

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Кваліфікаційна наукова
праця на правах рукопису

РОМАНЮК ОЛЕКСАНДР СЕРГІЙОВИЧ

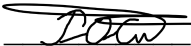
УДК 004.93.1: 004.032.26

ДИСЕРТАЦІЯ
НЕЙРОМЕРЕЖЕВЕ ПРОГНОЗУВАННЯ НЕСТАЦІОНАРНИХ
ПОСЛІДОВНОСТЕЙ

Спеціальність: 122 – Комп'ютерні науки
Галузь знань: 12 – Інформаційні технології

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

 О.С. Романюк

Науковий керівник
Руденко Олег Григорійович,
доктор технічних наук, професор

Харків – 2021

АНОТАЦІЯ

Романюк О.С. Нейромережеве прогнозування нестационарних послідовностей. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 122 «Комп'ютерні науки» (12 – Інформаційні технології). – Харківський національний університет радіоелектроніки, Міністерство освіти і науки України, Харків, 2021.

Дисертаційну роботу присвячено актуальному завданню створення математичних моделей, методів, алгоритмів та програм, орієнтованих на розв'язання задач прогнозування нестационарних послідовностей, які є складовими сучасних комп'ютерних технологій. У дисертаційній роботі на основі отриманих теоретичних і експериментальних досліджень вирішена задача побудови нейромережевих методів прогнозування нестационарних послідовностей, які на відміну від існуючих, прискорюють процес прогнозування та збільшують його точність в умовах апріорної і поточної невизначеності і наявності завад.

Метою дисертаційної роботи є розробка нейромережевих методів і моделей прогнозування нестационарних послідовностей з підвищеною точністю за наявності апріорної і поточної невизначеності.

Об'єкт дослідження – процеси прогнозування нестационарних часових рядів.

Предмет дослідження – методи та моделі прогнозування нестационарних часових рядів на основі штучних нейронних мереж.

Методи дослідження. В роботі використані методи теорії обчислювального інтелекту, а саме, методи теорії штучних нейронних мереж, що дозволило синтезувати нейромережеві моделі та отримати процедури їх навчання; методи теорії оптимізації, за допомогою яких були синтезовані швидкодіючі процедури навчання; методи робастного оцінювання, на основі яких були синтезовані робастні процедури навчання нейромережевих

моделей; методи імітаційного моделювання, що дозволили підтвердити ефективність отриманих результатів та розробити рекомендації щодо їх практичного використання. Експериментальні дослідження проводилися в лабораторних умовах і на реальних об'єктах.

Наукова новизна дисертаційної роботи полягає в такому:

1. Вперше запропоновано використання для навчання штучної нейронної мережі АДАЛТНИ в задачі оцінювання параметрів, які описуються марківською моделлю першого порядку, регуляризованих алгоритмів Качмажа та Нагумо-Ноди. Визначено умови збіжності та отримано неасимптотичні оцінки точності цих алгоритмів, що дозволяє встановити потенційно можливий виграш від їх використання.

2. Вперше запропоновано використання в задачах прогнозування комбінованого функціоналу, який складається з критерію четвертого ступеня і модульного, та градієнтну процедуру його мінімізації. Досліджено умови збіжності процедури в середньому і середньоквадратичному. Отримано аналітичні оцінки неасимптотичних та асимптотичних значень помилки оцінювання параметрів моделі та точності її ідентифікації, що забезпечило робастність оцінок шуканих параметрів мережі.

3. Отримав подальший розвиток нейромережевий метод прогнозування який, на відміну від існуючих використовує рекурентну форму l -крокової проєкційної процедури навчання, що дозволило підвищити швидкість навчання штучної нейронної мережі.

4. Удосконалено метод вибору оптимальних значень параметрів процедур навчання, що дозволяє забезпечити максимальну швидкість збіжності процедур і їхню стійкість та попередньо оцінити досліднику можливості та ефективність процедур, які використовуються, під час вирішення практичних завдань.

5. Отримав подальший розвиток метод факторизації l -крокових процедур навчання штучних нейронних мереж, що дозволило поліпшити їх обчислювальні властивості та підвищити стійкість.

У дисертаційній роботі проведено аналіз стану проблеми прогнозування. На основі системного підходу проаналізовано існуючі підходи до застосування нейронних мереж для вирішення задачі прогнозування і описані основні типи, які при цьому використовуються, сформульовано мету та задачі дослідження.

Пропонується вирішення комплексу взаємопов'язаних завдань, яке полягає у розробці методів та моделей прогнозування на основі штучних нейронних мереж. Розглянуто питання застосування нейронних мереж для вирішення завдань прогнозування нестационарних часових рядів і описані основні типи штучних нейронних мереж.

Розглянуто багат шаровий перцептрон в якості нейромережевої моделі для прогнозування. Описано процедуру вибору його структури та запропоновано процедуру навчання. Вивчено умови її збіжності та отримано вираз для оптимального значення параметра збіжності, що дозволяє зменшити час навчання мережі.

Запропоновано та досліджено використання для навчання штучної нейронної мережі АДАЛПНИ в заданні оцінювання параметрів, які описуються марківською моделлю першого порядку, регуляризованих алгоритмів Качмажа та Нагумо-Ноди.

З метою отримання робастних оцінок шуканих параметрів штучних нейронних мереж використано комбінований функціонал, що складається з критерію четвертого ступеня та модульного критерію. Досліджено градієнтну процедуру мінімізації функціоналу. Для підвищення швидкості навчання штучних нейронних мереж застосовано рекурентну форму l -крокової проекційної процедури навчання та здійснено її факторизацію.

Для забезпечення максимальної швидкості збіжності процедур та їх стійкості досліджено методи вибору оптимальних значень параметрів функціоналів та процедур навчання.

З метою перевірки отриманих теоретичних результатів та розробки рекомендацій щодо практичної реалізації здійснено імітаційне моделювання нейромережевих методів прогнозування нестационарних часових рядів.

На основі запропонованих моделей та методів програмно реалізована нейромережева система прогнозування параметрів технологічних процесів виробництва кальцинованої соди.

Практичне значення отриманих результатів полягає в такому:

- розроблено програмні засоби, які реалізують запропоновані методи побудови і навчання штучних нейронних мереж прямого поширення, що дозволяють автоматизувати процес побудови нейромережевих моделей досліджуваних об'єктів, здійснити структурно-параметричний синтез в умовах апіорної і поточної невизначеності;

- проведено експериментальні дослідження властивостей і характеристик розроблених методів, які показали, що запропоновані методи за рахунок використання додаткової інформації про властивості об'єкта і діючих перешкод дозволяють істотно скоротити час побудови моделей, а також забезпечити їх стійкість і робастність;

- синтезовані структури штучних нейронних мереж, їх моделі та процедури навчання можуть бути використані при розробці систем інтелектуального аналізу даних, систем інтелектуального управління об'єктами з безперервними технологічними процесами та прогнозування результатів їх роботи.

Отримані в роботі математичні моделі та методи прогнозування дозволили поліпшити стабільність вихідних параметрів технологічних процесів виробництва кальцинованої соди у Державному науково-дослідному і проектному інституті основної хімії (ДУ «НІОХІМ», м. Харків), про що свідчить акт від 19.11.2020 р. Також результати дисертаційного дослідження використані в освітньому процесі Харківського національного університету радіоелектроніки, зокрема, у навчальних дисциплінах «Нейронні обчислювальні структури», «Інтелектуальний аналіз даних», «Глибоке навчання в технологіях машинного зору» для здобувачів другого (магістерського) рівня вищої освіти за освітньою програмою «Комп'ютерні інтелектуальні технології» (акт від 10.01.2021 р.)

Матеріали дисертації повною мірою викладені у 18 публікаціях, з них – 6 статей у фахових періодичних виданнях України з технічних наук, з яких 1 – категорії А (індексується в Scopus), 2 – категорії Б; 2 розділи у колективних монографіях, 1 стаття у інших виданнях, 9 тез доповідей у матеріалах міжнародних наукових конференцій.

Ключові слова: прогнозування, математична модель, критерій втрат, процедура навчання, робастність, нейронна мережа.

Список публікацій здобувача

1. Rudenko O. Analysis of convergence of adaptive single-step algorithms for the identification of non-stationary objects / O. Rudenko, O. Bezsonov, V. Lebediev, A. Romanyk // Eastern-European Journal of Enterprise Technologies. 2019. – № 1/4 (97). – Р. 6–14. (Належить до категорії А, входить до міжнародної наукометричної бази Scopus).

2. Руденко О.Г. Робастна ідентифікація об'єктів на основі мінімізації комбінованого функціоналу / О.Г. Руденко О.Г., О.О. Безсонов, Н.М. Сердюк, К.О. Олійник, О.С. Романюк // Системи обробки інформації. – 2020. – №1 (160). – С.80–88 (Належить до категорії Б).

3. Бессонов А.А. Факторизованные алгоритмы обучения персептрона в задаче построения нелинейной модели / А.А. Бессонов, К.О. Олейник, С.А. Романюк, О.Г. Руденко, Н.Н. Сердюк / Бионика интеллекта.-2020. – №1(94). – С. 23-29 (Належить до категорії Б).

4. Руденко О.Г. Критерії вибору персептронної моделі для прогнозування: аналіз та практичні рекомендації щодо їх використання / О.Г. Руденко, О.О. Безсонов, О.Г. Лебедев, О.С. Романюк // Бионика интеллекта. –2018. –№ 2 (91). – С.31– 40.

5. Руденко О.Г. Про один алгоритм навчання нейронної мережі в задачі прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.Г. Лебедев, О.С. Романюк // Бионика интеллекта. – 2018. – №1(90). –С.79–83.

6. Руденко О.Г. Робастна ідентифікація об'єктів за наявністю негаусівських завад / О.Г. Руденко О.Г., О.О. Безсонов, Н.М. Сердюк, К.О. Олійник, О.С.Романюк // Бионика интеллекта. –2019. – №2(93). – С.7–12.

7. Руденко О.Г. Нейромережеве прогнозування часових рядів на основі багатошарового перцептрона / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Development Management. – 2019. –Vol. 5 – Issue 1. – С.23–34.

8. Олійник К.О. Градієнтні алгоритми навчання згортальних нейронних мереж / К.О. Олійник, О.С. Романюк // Інформаційні технології та системи: монографія / за заг. ред. В. С. Пономаренка. – Харків. : Видавництво «Стиль-іздат», 2020. – С. 37–51.

9. Rudenko O. Analysis of the influence of interference on the statistical properties of estimates of non-stationary parameters / O. Rudenko, A. Romanyuk // Інформаційні системи і технології: монографія / за заг. ред. В. С. Пономаренка. – Харків: ФОП Бровін О.В., 2019. – С.97–107.

10. Романюк О.С. Нейроеволюційний підхід до прогнозування нестаціонарних послідовностей / О.С. Романюк // Збірник матеріалів І Міжнар. наук.-практ. конф. молодих вчених, аспірантів і студентів; «Сучасні інформаційні технології та системи в управлінні». [Електронний ресурс] 19–20 квітня 2018 р. – Київ : КНЕУ, 2018. – С.204 – 205.

11. Руденко О.Г. Прогнозування нестаціонарних послідовностей за допомогою коеволюціонуючих штучних нейромереж/ О.Г. Руденко, О.С. Романюк // Збірник наукових праць за матеріалами II міжнародної науково-практичної конференції. «Синергетика, мехатроніка, телематика дорожніх машин і систем у навчальному процесі та науці». 29 травня 2018 р. – Харків, ХНАДУ, 2018. – С.159–161.

12. Руденко О.Г. Дослідження особливостей навчання штучних нейронних мереж в задачі прогнозування / О.Г. Руденко, О.С. Романюк // Матеріали 8-ої міжнародної науково-технічної конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління». 26–27 квітня 2018 р., Полтава, Баку, Харків, Жиліна. – 2018. – С. 32.

13. Rudenko O. Time series prediction based on evolving neural network CMAC / O. Rudenko, O. Bezsonov, A. Romanyk // 3-d International Conference on Computer Algebra and Information Technologies. – Odessa, August 20-25, 2018. – P. 49 –51.

14. Руденко О.Г. Про оцінювання нестационарних параметрів, що описуються марківською моделлю / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Тези доповідей шостої міжнародної науково-практичної конференції «Проблеми інформатизації». 14 – 16 листопада 2018 р., Черкаси, Баку, Бельско-Бяла, Харків. – 2018. – С.48.

15. Руденко О.Г. Прогнозування часових рядів на основі коеволюційних нейронних мереж / О.Г. Руденко, О.С. Романюк // Тези доповідей міжнародної науково–практичної конференції «Інформаційні технології та системи». 10-11 квітня 2019, Харків. – 2019. – С.15-16.

16. Руденко О.Г. Критерії вибору перцептронної моделі для прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Тези доповідей дев'ятої міжнародної науково-практичної конференції «Сучасні напрями розвитку інформаційно-комунікаційних систем та засобів управління», 11-12 квітня 2019 р., Баку, Харків, Жиліна. – 2019. – С.90 – 91.

17. Руденко О.Г. Про вибір математичної моделі для прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Матеріали XVII міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем». (MSSIS-2019) 20-22 листопада 2019 р., Дніпро. – 2019. – С. 221–222.

18. Олейник К.О. Дослідження спрощених алгоритмів навчання штучних нейронних мереж прямого поширення / К.О. Олейник, О.С. Романюк // Матеріали міжнародної науково-практичної конференції «Інформаційні технології та системи». 9-10 квітня 2020, Харків :ХНЕУ імені Семена Кузнеця. – 2020. – С. 19.

ABSTRACT

Romaniuk A.S. Neural Network Prediction of Non-Stationary Sequences. - Qualification scientific work as a manuscript copyright.

Thesis submitted for the degree of Doctor of Philosophy in the specialty 122 "Computer Science" (12 - Information Technology). - Kharkiv National University of Radio Electronics, Ministry of Education and Science of Ukraine, Kharkiv, 2021.

The thesis is devoted to the current problem of creating mathematical models, methods, algorithms and programs focused on solving the tasks of predicting non-stationary sequences that are the part of modern computer technology. Based on the theoretical and experimental studies obtained, the thesis solves the problem of constructing neural network methods of non-stationary sequences prediction, which, unlike existing ones, accelerate the prediction process and increase its accuracy under the conditions of a priori and current uncertainty and the presence of interference.

The aim of the thesis is to develop neural network methods and models for predicting non-stationary sequences with improved accuracy under a priori and current uncertainty.

The object of the research is the prediction processes of non-stationary time series.

The subject of the research is the methods and models of non-stationary time series prediction based on artificial neural networks.

Research methods. The paper uses the methods of the computational intelligence theory, namely the methods of the artificial neural networks theory, which made it possible to synthesize neural network models and obtain procedures for their training; methods of optimization theory, by means of which the high-speed training procedures have been synthesized; robust estimation methods, on the basis of which the robust training procedures for neural network models have been synthesized; simulation modeling methods, which made it possible to confirm the effectiveness of the obtained results and develop recommendations for their practical

use. The experimental studies have been carried out under laboratory conditions and on real objects.

The scientific novelty of the thesis is as follows:

1. It has first been proposed to use the regularized algorithms of Kaczmarz and Nagumo-Noda for training the artificial neural network ADALINE for the task of estimating the parameters described by the Markov model of the first order. The convergence conditions have been determined and non-asymptotic estimates of the accuracy of these algorithms have been obtained, which makes it possible to establish the potential gain from their use.

2. The use of a combined functional consisting of a fourth-degree criterion and a modular degree criterion in prediction problems and a gradient procedure for its minimization have been proposed for the first time. The convergence conditions of the procedure in the mean and the RMS have been investigated. The analytical estimates of non-asymptotic and asymptotic values of the model parameter estimation error and accuracy of its identification have been obtained, which ensured the productivity of the estimates of the required network parameters.

3. The neural network prediction method has been further developed, which, unlike the existing ones, uses a recurrent form of l -step projection training procedure, which enabled to increase the training rate of the artificial neural network.

4. The method of choosing the optimal values of the parameters of training procedures has been improved, enabling to provide the maximum rate of convergence of the procedures and their stability and to preliminarily estimate by the researcher the possibility and efficiency of the used procedures when solving practical tasks.

5. The method of factorization of l -step procedures for training artificial neural networks has been further developed, which has improved their computational properties and stability.

The thesis analyzes the state of the prediction problem. Based on a systematic approach, the existing approaches to the application of neural networks for solving

the prediction problem have been analyzed and the main types used have been described, the goal and the objectives of the research have been formulated.

The solution of a complex of interrelated problems has been proposed, which consists in the development of methods and prediction models based on artificial neural networks. The issues of the neural networks application for solving the problems of non-stationary time series prediction have been considered and the main types of artificial neural networks have been described.

A multilayer perceptron has been considered as a neural network model for the prediction. The procedure for selecting its structure has been described and the training procedure has been proposed. The conditions of its convergence have been researched and the expression for the optimal value of the convergence parameter has been obtained, enabling to reduce the network training time.

The use of regularized Kaczmarz and Nagumo-Noda algorithms for training the artificial neural network ADALINE in the task of estimating the parameters that are described by a first-order Markov model, has been proposed and investigated.

In order to obtain robust estimates of the required parameters of artificial neural networks, a combined functional consisting of the fourth-degree criterion and the modular criterion has been used. The gradient procedure of the functional minimization has been studied. The recurrent form of l -step projective training procedure has been applied to increase the training rate of artificial neural networks and its factorization has been carried out.

To ensure the maximum rate of convergence of the procedures and their stability, the methods for selecting the optimal values of the parameters of the functionals and the training procedures have been investigated.

In order to verify the obtained theoretical results and to develop the recommendations for their practical implementation, the simulation of neural network methods of non-stationary time series prediction has been carried out.

Based on the proposed models and methods, the neural network system of the prediction parameters of the technological processes of the soda ash production has been implemented programmatically.

The practical value of the obtained results is as follows:

- the software tools have been developed, which implement the proposed methods of construction and training of the artificial neural networks of direct propagation, enabling to automate the process of designing neural network models of objects under research and to carry out the structural and parametric synthesis in terms of a priori and current uncertainties;

- the experimental studies of the properties and characteristics of the developed methods have been conducted, which have shown that the proposed methods by using additional information concerning the properties of the object and the existing disturbances can significantly reduce the time to build models, as well as ensure their stability and robustness;

- the synthesized structures of the artificial neural networks, their models and training procedures can be used to develop systems of data mining, intelligent control systems of objects with continuous technological processes and predict the results of their work.

The mathematical models and prediction methods obtained in the work have improved the stability of the initial parameters for the technological processes of the soda ash production in the State Research and Design Institute of Basic Chemistry (SI "NIOCHIM", Kharkiv), as evidenced by the act of 19.11.2020, the results of the dissertation research have been used in the educational process of Kharkiv National University of Radio Electronics, particularly in the academic disciplines "Neural Computer Structures", "Intelligent Data Analysis", "Deep Training in Machine Vision Technologies" for the students of the second (master's) level of higher education on the educational program "Computer Intelligent Technologies" (act of 10.01.2021).

The materials of the thesis are fully described in 18 publications, including six articles in the professional periodicals of Ukraine in the technical sciences, including one of A category (indexed in Scopus), two of B category; two sections in collective monographs, one article in other publications, and nine abstracts of reports in the proceedings of international scientific conferences.

Keywords: prediction, mathematical model, loss criterion, training procedure, performance, neural network.

List of the applicant's publications

1. Rudenko O. Analysis of convergence of adaptive single-step algorithms for the identification of non-stationary objects / O. Rudenko, O. Bezsonov, V. Lebediev, A. Romanyk // Eastern-European Journal of Enterprise Technologies. 2019. – № 1/4 (97). – P. 6–14. (Належить до категорії А, входить до міжнародної наукометричної бази Scopus).

2. Руденко О.Г. Робастна ідентифікація об'єктів на основі мінімізації комбінованого функціоналу / О.Г. Руденко О.Г., О.О. Безсонов, Н.М. Сердюк, К.О. Олійник, О.С. Романюк // Системи обробки інформації. – 2020. – №1 (160). – С.80–88 (Належить до категорії Б).

3. Бессонов А.А. Факторизованные алгоритмы обучения персептрона в задаче построения нелинейной модели / А.А. Бессонов, К.О. Олейник, С.А. Романюк, О.Г. Руденко, Н.Н. Сердюк / Бионика интеллекта.-2020. –№1(94). – С. 23-29 (Належить до категорії Б).

4. Руденко О.Г. Критерії вибору персептронної моделі для прогнозування: аналіз та практичні рекомендації щодо їх використання / О.Г. Руденко, О.О. Безсонов, О.Г. Лебедев, О.С. Романюк // Бионика интеллекта. –2018. –№ 2 (91). – С.31– 40.

5. Руденко О.Г. Про один алгоритм навчання нейронної мережі в задачі прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.Г. Лебедев, О.С. Романюк // Бионика интеллекта. – 2018. – №1(90). – С.79–83.

6. Руденко О.Г. Робастна ідентифікація об'єктів за наявністю негаусівських завад / О.Г. Руденко О.Г., О.О. Безсонов, Н.М. Сердюк, К.О. Олійник, О.С.Романюк // Бионика интеллекта. –2019. – №2(93). – С.7–12.

7. Руденко О.Г. Нейромережеве прогнозування часових рядів на основі багатосарового персептрона / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Development Management. – 2019. –Vol. 5 – Issue 1. – С.23–34.

8. Олійник К.О. Градієнтні алгоритми навчання згортальних нейронних мереж / К.О. Олійник, О.С. Романюк // Інформаційні технології та системи: монографія / за заг. ред. В. С. Пономаренка. – Харків. : Видавництво «Стиль-іздат», 2020. – С. 37–51.

9. Rudenko O. Analysis of the influence of interference on the statistical properties of estimates of non-stationary parameters / O. Rudenko, A. Romanyk // Інформаційні системи і технології: монографія / за заг. ред. В. С. Пономаренка. – Харків: ФОП Бровін О.В., 2019. – С.97–107.

10. Романюк О.С. Нейроеволюційний підхід до прогнозування нестационарних послідовностей / О.С. Романюк // Збірник матеріалів І Міжнар. наук.-практ. конф. молодих вчених, аспірантів і студентів; «Сучасні інформаційні технології та системи в управлінні». [Електронний ресурс] 19–20 квітня 2018 р. – Київ : КНЕУ, 2018. – С.204 – 205.

11. Руденко О.Г. Прогнозування нестационарних послідовностей за допомогою коеволюціонуючих штучних нейромереж/ О.Г. Руденко, О.С. Романюк // Збірник наукових праць за матеріалами ІІ міжнародної науково-практичної конференції. «Синергетика, мехатроніка, телематика дорожніх машин і систем у навчальному процесі та науці». 29 травня 2018 р. – Харків, ХНАДУ, 2018. – С.159–161.

12. Руденко О.Г. Дослідження особливостей навчання штучних нейронних мереж в задачі прогнозування / О.Г. Руденко, О.С. Романюк // Матеріали 8-ої міжнародної науково-технічної конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління». 26-27 квітня 2018 р., Полтава, Баку, Харків, Жиліна. – 2018. – С. 32.

13. Rudenko O. Time series prediction based on evolving neural network CMAC / O. Rudenko, O. Bezsonov, A. Romanyk // 3-d International Conference on Computer Algebra and Information Technologies. – Odessa, August 20-25, 2018. – P. 49 –51.

14. Руденко О.Г. Про оцінювання нестационарних параметрів, що описуються марківською моделлю / О.Г. Руденко, О.О. Безсонов, О.С.

Романюк // Тези доповідей шостої міжнародної науково-практичної конференції «Проблеми інформатизації». 14 – 16 листопада 2018 р., Черкаси, Баку, Бельско-Бяла, Харків. – 2018. – С.48.

15. Руденко О.Г. Прогнозування часових рядів на основі коеволюційних нейронних мереж / О.Г. Руденко, О.С. Романюк // Тези доповідей міжнародної науково–практичної конференції «Інформаційні технології та системи». 10-11 квітня 2019, Харків. – 2019. – С.15-16.

16. Руденко О.Г. Критерії вибору перцептронної моделі для прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Тези доповідей дев'ятої міжнародної науково-практичної конференції «Сучасні напрями розвитку інформаційно-комунікаційних систем та засобів управління», 11-12 квітня 2019 р., Баку, Харків, Жиліна. – 2019. – С.90 – 91.

17. Руденко О.Г. Про вибір математичної моделі для прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Матеріали XVII міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем». (MSSIS-2019) 20-22 листопада 2019 р., Дніпро. – 2019. – С. 221–222.

18. Олейник К.О. Дослідження спрощених алгоритмів навчання штучних нейронних мереж прямого поширення / К.О. Олейник, О.С. Романюк // Матеріали міжнародної науково-практичної конференції «Інформаційні технології та системи». 9-10 квітня 2020, Харків :ХНЕУ імені Семена Кузнеця. – 2020. – С. 19.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	19
ВСТУП.....	20
1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ.....	28
1.1 Прогнозування як синтез моделі.....	28
1.2 Моделі прогнозування.....	33
1.2.1 Лінійні регресійні моделі.....	33
1.2.2 Нелінійні регресійні моделі.....	37
1.2.3 Нейромережеві нелінійні моделі.....	39
1.3 Порівняльний аналіз моделей прогнозування. Переваги і недоліки моделей.....	41
1.4 Аналіз рекомендацій щодо вибору критерію прогнозування...	43
1.5 Постановка задач дослідження.....	45
1.6 Висновки за першим розділом.....	46
2 ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ НА ОСНОВІ ШНМ.....	48
2.1 Штучні нейронні мережі, що застосовуються для вирішення завдань ідентифікації, управління і прогнозування.....	48
2.1.1 Багатошаровий персептрон.....	48
2.1.2 Радіально - базисна мережа (РБМ)	49
2.1.3 Узагальнено - регресійна мережа (УРМ)	50
2.1.4 Штучна нейронна мережа АДАЛІНА.....	50
2.1.5 Нейронна мережа СМАС.....	50
2.1.6 Мережа локально - лінійних моделей (МЛЛМ).	51
2.1.7 Динамічні ШНМ.....	51
2.2. Аналіз процедур навчання ШНМ.....	52
2.3 Багатошаровий персептрон як нейромережева модель.....	56
2.4 Прогнозування часових рядів на основі ШНМ АДАЛІНА.....	60
2.4.1 Прогнозування з використанням процедури Качмажа.....	61

2.4.2 Прогнозування з використанням процедури Нагумо-Ноди...	64
2.5 Прогнозування часових рядів на основі коеволюційних нейронних мереж.....	67
2.6 Висновки за другим розділом.....	72
3 РОБАСТНЕ НАВЧАННЯ ШНМ ПРЯМОГО ПОШИРЕННЯ.....	74
3.1 Деякі підходи, що використовуються при робастному навчанні.....	74
3.2 Вибір критерію робастного навчання.....	75
3.3 Процедури робастного навчання.....	78
3.3.1 Оптимальні процедури навчання.....	78
3.3.2 Однокрокові робастні процедури навчання.....	80
3.3.3 Робастна процедура навчання, яка мінімізує комбінований функціонал	81
3.4 Робастне навчання на основі мінімізації модифікованого комбінованого функціоналу.....	83
3.5 Багатокрокові робастні процедури навчання.....	87
3.6 Висновки за третім розділом.....	89
4 ОПТИМІЗАЦІЯ ПРОЦЕДУР НАВЧАННЯ.....	92
4.1 Оцінювання параметрів функціоналів.....	92
4.1.1 Оцінювання параметра масштабу в режимі off-line.....	92
4.1.2 Оцінювання параметрів завад в режимі on-line	94
4.2 Вибір оптимальних параметрів процедур навчання.....	94
4.2.1 Вибір параметра регуляризації δ	94
4.2.2 Визначення параметра швидкості навчання γ	100
4.3 Факторизація процедур навчання.....	102
4.3.1 Факторизація моделі методом квадратного кореня.....	103
4.3.2 Факторизація за допомогою алгоритма Хаусхолдера.....	108
4.3.3 Ортогоналізація Грама-Шмідта.....	112
4.4 Висновки за четвертим розділом.....	116

5. ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ВИРІШЕННЯ ПРАКТИЧНИХ ЗАДАЧ.....	118
5.1 Рішення задачі ідентифікації.....	118
5.2 Рішення задачі прогнозування.....	127
5.3 Побудова нейромережевої системи ідентифікації процесу виробництва кальцинованої соди.....	132
5.4 Висновки за п'ятим розділом.....	138
ВИСНОВКИ.....	140
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	142
Додаток А.....	156
Додаток Б.....	171
Додаток В.....	181
Додаток Г.....	185
Додаток Д.....	189
Додаток Е.....	193
Додаток Ж.....	196

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ШНМ – штучна нейронна мережа

ЧР – часовий ряд

МГУА – метод групового урахування аргументів

AR – модель авторегресії

MA – модель змінного середнього

ARMA – модель авторегресії-ковзного середнього

ARIMA – інтегрована модель авторегресії-ковзного середнього

NARMAX – нелінійна ARMAX модель

NARX – нелінійна ARX модель

БШП – багатошаровий персептрон

РБМ – радіально - базисна мережа

ADALINE – ADaptive LInear NEuron

ЗП – алгоритм зворотного поширення помилки

АФ – активаційна функція

МНК – метод найменших квадратів

МНМ – метод найменших модулів

MAD – Median of Absolute Deviations

ENR – echo-to-noise ratio

ВСТУП

Актуальність теми дослідження.

Прогнозування часових рядів (ЧР) є актуальною науковою проблемою, що має безліч застосувань у різних галузях. В даний час методи прогнозування представляють великий практичний інтерес і дозволяють вирішувати широкий спектр завдань в науці, техніці і економіці. До їх числа можна віднести аналіз соціальних, економічних, геофізичних процесів, передбачення природних явищ, економічних подій тощо. Методи прогнозування використовують для дослідження системних зв'язків і закономірностей функціонування і розвитку об'єктів і процесів із застосуванням сучасних методів обробки інформації. Вони є важливим засобом в аналізі складних прикладних систем, в роботі з інформацією, цілеспрямованого впливу людини на об'єкти дослідження з метою підвищення ефективності їх функціонування. Найбільш поширеною постановкою завдання прогнозування є завдання прогнозування часових рядів, тобто функції, визначеної на осі часу.

Розвиток прогнозування, як сфери науки в останні десятиліття призвів до створення безлічі моделей і методів, процедур, прийомів прогнозування, нерівноцінних за своїм значенням. За оцінками зарубіжних і вітчизняних фахівців з прогнозування вже налічується понад ста методів, в зв'язку з чим постає завдання вибору методів, які давали б адекватні прогнози для досліджуваних процесів або систем.

До останнього часу основним при вирішенні задачі прогнозування був статистичний підхід. В рамках статистичних моделей вирішуються задачі прогнозування, знаходження прихованої періодичності даних, аналізу залежностей, оцінки ризиків під час прийняття рішень та інші. Загальним недоліком статистичних моделей є складність вибору типу моделі і підбору її параметрів. Крім того, при використанні статистичного підходу однією з головних вимог до часового ряду є його стаціонарність, яка полягає в тому, що

розподіл його значень є інваріантним щодо моменту часу, для якого його побудовано.

Традиційні статистичні методи аналізу ЧР часто виявляються неефективними з огляду на те, що ці методи потребують апріорної наявності досить великої і репрезентативної вибірки.

Слід, однак, відзначити, що більшість практичних задач прогнозування ЧР характеризуються високим рівнем нелінійності і нестационарності, зашумленістю, наявністю нерегулярних трендів, стрибків, аномальних викидів. У цих умовах жорсткі статистичні припущення про властивості ЧР часто обмежують можливості класичних методів прогнозування.

Альтернативою статистичним методам можуть служити методи обчислювального інтелекту, до числа яких, в першу чергу, слід віднести штучні нейронні мережі (ШНМ). Виходячи з того, що вони є універсальними апроксиматорами, деякі типи ШНМ дозволяють відновити з заданою точністю будь-яку складну неперервну нелінійну функцію, використовуючи представлення функції, що апроксимується, у вигляді нейронної мережі, утвореної нейронами, параметри яких визначаються шляхом її навчання. Здатність нейронної мережі до різнобічної обробки інформації впливає з її здатності до узагальнення і виділення прихованих залежностей між вхідними та вихідними даними. Істотною перевагою нейронних мереж є те, що вони здатні до навчання й узагальнення накопичених знань. Особливо привабливим є використання ШНМ для прогнозування нестационарних послідовностей. Підвищення точності та швидкості процесу прогнозування нестационарних послідовностей методами ШНМ є актуальною науковою задачею.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана відповідно до плану науково-дослідних робіт Харківського національного університету радіоелектроніки в рамках держбюджетної НДР «Нейро-фаззі системи для поточної кластеризації і класифікації послідовностей даних за умов їх викривленості відсутніми та аномальними спостереженнями», розділ «Адаптивні методи та моделі

класифікації даних і прогнозування часових рядів за умов їх викривленості відсутніми та аномальними спостереженнями на основі штучних імунних систем» (№ ДР 0113U000361), затвердженими Міністерством освіти і науки України. Автор був одним з виконавців робіт за даними темами.

Мета та задачі дослідження. Метою дисертаційної роботи є розробка нейромережових методів і моделей прогнозування нестационарних послідовностей з підвищеною точністю за наявності апріорної і поточної невизначеності.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати стан проблеми і особливості задач прогнозування;
- розглянути методи вибору критерія якості прогнозування і провести їх порівняльний аналіз;
- розвинути нейромережовий метод прогнозування нестационарного процесу, що описується марківською моделлю першого порядку;
- дослідити властивості адаптивних процедур прогнозування за умов наявності завад;
- розробити просту в обчислювальному відношенні робастну процедуру прискореного навчання нейронних мереж на основі мінімізації комбінованого функціоналу;
- вирішити задачу оптимізації процедур навчання з метою скорочення процесу навчання та підвищення їх обчислювальної стійкості;
- застосувати розроблені процедури та моделі для побудови моделей та прогнозування параметрів технологічних процесів виробництва кальцинованої соди.

Об'єкт дослідження – процеси прогнозування нестационарних часових рядів.

Предмет дослідження – методи та моделі прогнозування нестационарних часових рядів на основі штучних нейронних мереж.

Методи дослідження. В роботі використані методи теорії обчислювального інтелекту, а саме, методи теорії штучних нейронних мереж,

що дозволило синтезувати нейромережеві моделі та отримати процедури їх навчання; методи теорії оптимізації, за допомогою яких були синтезовані швидкодіючі процедури навчання; методи робастного оцінювання, на основі яких були синтезовані робастні процедури навчання нейромережевих моделей; методи імітаційного моделювання, що дозволили підтвердити ефективність отриманих результатів та розробити рекомендації щодо їх практичного використання. Експериментальні дослідження проводилися в лабораторних умовах і на реальних об'єктах.

Наукова новизна результатів дисертаційної роботи:

1. Вперше запропоновано використання для навчання штучної нейронної мережи АДАЛЛНИ в задачі оцінювання параметрів, які описуються марківською моделлю першого порядку, регуляризованих алгоритмів Качмажа та Нагумо-Ноди. Визначено умови збіжності та отримано неасимптотичні оцінки точності цих алгоритмів, що дозволяє встановити потенційно можливий виграш від їх використання.

2. Вперше запропоновано використання в задачах прогнозування комбінованого функціоналу, який складається з критерію четвертого ступеня і модульного, та градієнтну процедуру його мінімізації. Досліджено умови збіжності процедури в середньому і середньоквадратичному. Отримано аналітичні оцінки неасимптотичних та асимптотичних значень помилки оцінювання параметрів моделі та точності її ідентифікації, що забезпечило робастність оцінок шуканих параметрів мережі.

3. Отримав подальший розвиток нейромережевий метод прогнозування який, на відміну від існуючих використовує рекурентну форму l-крокової проекційної процедури навчання, що дозволило підвищити швидкість навчання штучної нейронної мережі.

4. Удосконалено метод вибору оптимальних значень параметрів процедур навчання, що дозволяє забезпечити максимальну швидкість збіжності процедур і їхню стійкість та попередньо оцінити досліднику

можливості та ефективність процедур, які використовуються, під час вирішення практичних завдань.

5. Отримав подальший розвиток метод факторизації 1-крокових процедур навчання штучних нейронних мереж, що дозволило поліпшити їх обчислювальні властивості та підвищити стійкість.

Практичне значення одержаних результатів дисертаційної роботи полягає в тому, що:

- розроблено програмні засоби, які реалізують запропоновані методи побудови і навчання штучних нейронних мереж прямого поширення, що дозволяють автоматизувати процес побудови нейромережових моделей досліджуваних об'єктів, здійснити структурно-параметричний синтез в умовах апріорної і поточної невизначеності;

- проведено експериментальні дослідження властивостей і характеристик розроблених методів, які показали, що запропоновані методи за рахунок використання додаткової інформації про властивості об'єкта і діючих перешкод дозволяють істотно скоротити час побудови моделей, а також забезпечити їх стійкість і робастність;

- синтезовані структури штучних нейронних мереж, їх моделі та процедури навчання можуть бути використані при розробці систем інтелектуального аналізу даних, систем інтелектуального управління об'єктами з безперервними технологічними процесами та прогнозування результатів їх роботи.

Отримані в роботі математичні моделі та методи прогнозування дозволили поліпшити стабільність вихідних параметрів технологічних процесів виробництва кальцинованої соди у Державному науково-дослідному і проектному інституті основної хімії (ДУ «НІОХІМ», м. Харків), про що свідчить акт від 19.11.2020 р. Також результати дисертаційного дослідження використані в освітньому процесі Харківського національного університету

радіоелектроніки, зокрема, у навчальних дисциплінах «Нейронні обчислювальні структури», «Інтелектуальний аналіз даних», «Глибоке навчання в технологіях машинного зору» для здобувачів другого (магістерського) рівня вищої освіти за освітньою програмою «Комп'ютерні інтелектуальні технології» (акт від 10.01.2021 р.)

В [1] отримано оцінки точності визначення нестационарних параметрів за наявністю завад; в [2] запропоновано використання комбінованого функціоналу для оцінювання параметрів за наявністю негаусівських завад; в [3] розроблено факторизовані багатокрокові процедури навчання; в [4, 16] сформульовані критерії вибору перцептронної моделі для задач прогнозування; в [5] отримано оцінки швидкості збіжності однокрокових адаптивних процедур при побудові моделей нестационарних об'єктів; в [6] розроблено процедуру оцінювання нестационарних параметрів та досліджено умови її сталого режиму; в [7] визначено особливості прогнозування нестационарних послідовностей за допомогою багат шарового перцептрона; в [8] проведено аналіз однокрокових градієнтних алгоритмів навчання нейронних мереж; в [9] отримано оцінки впливу завад на властивості алгоритмів оцінювання нестационарних послідовностей; в [10, 11] запропоновано використання коєволюціонуючих штучних нейромереж для прогнозування нестационарних послідовностей; в [12] отримано оцінки точності і часу навчання штучних нейронних мереж в задачі прогнозування часових рядів; в [13] запропоновано використання штучних нейронних мереж СМАС для прогнозування часових рядів; в [14] отримано оцінки визначення нестационарних параметрів, що описуються марківською моделлю; в [15] запропоновано використання коєволюційних штучних нейронних мереж для вирішення задач прогнозування; в [17] проведено аналіз структур математичних моделей в задачах прогнозування; в [18] запропоновано різні варіанти спрощення процедур навчання штучних нейронних мереж.

Апробація результатів дисертації.

Основні положення дисертаційної роботи представлено на таких міжнародних конференціях і форумах:

– на I Міжнародній науково-практичній конференції молодих вчених, аспірантів і студентів «Сучасні інформаційні технології та системи в управлінні» (Київ 2018);

– на II Міжнародній науково-практичній конференції «Синергетика, мехатроніка, телематика дорожніх машин і систем у навчальному процесі та науці» (Харків, 2018);

– на 8-й Міжнародній науково-технічній конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління» (Полтава, Баку, Харків, Жиліна, 2018);

– на 3-d International Conference on Computer Algebra and Information Technologies (Odessa, 2018);

– на 6-й Міжнародній науково-практичній конференції «Проблеми інформатизації» (Черкаси, Баку, Бельско-Бяла, Харків, 2018);

– на Міжнародній науково-практичній конференції «Інформаційні технології та системи» (Харків, 2019);

– на 9-й Міжнародній науково-практичній конференції «Сучасні напрями розвитку інформаційно-комунікаційних систем та засобів управління» (Баку, Харків, Жиліна, 2019);

– на XVII -й Міжнародній науково-практичній конференції «Математичне та програмне забезпечення інтелектуальних систем». (MSSIS-2019, Дніпро);

– на Міжнародній науково-практичній конференції «Інформаційні технології та системи» (Харків, 2020).

Публікації.

у 18 публікаціях, з них – 6 статей у фахових періодичних виданнях України з технічних наук, з яких 1 – категорії А (індексується в Scopus), 2 – категорії Б; 2 розділи у колективних монографіях, 1 стаття у інших виданнях, 9 тез доповідей у матеріалах міжнародних наукових конференцій.

Структура та обсяг роботи. Дисертація складається зі вступу, 5 розділів, висновків, списку використаних джерел, додатків. Загальний обсяг роботи складає 198 сторінок тексту, що містять 155 сторінок основного тексту, анотації на 15 сторінках, 25 рисунків, 1 таблиця, список використаних джерел з 140 найменування на 14 сторінках, 7 додатків на 43 сторінках.

1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Прогнозування як синтез моделі

Будь-яке прогнозування (екстраполяція) спирається на деяке формалізоване уявлення про існуючий зв'язок F^0 між причинами X і наслідками Y (або вхідними та вихідними змінними відповідно).

Формалізація зв'язку F^0 необхідна перш за все для однозначної оцінки слідства Y при заданій причині X і для її оптимізації за заданими критеріями. Нехай ϕ – алгоритм такої формалізації, що включає і оптимізацію. Тоді формалізований зв'язок можна уявити як результат переробки вихідної інформації алгоритмом ϕ

$$F = \phi(I, J), \quad (1.1)$$

де I – протокол спостережень різних реалізацій формалізованого причинно-наслідкового зв'язку F ;

J – апріорні уявлення про вигляд синтезованого зв'язку F (наприклад, її структура, порядок полінома і т.д.).

Зв'язок F^0 тут усюди передбачається статичним, що дозволяє вважати його модель F функцією (точніше, векторної функцією векторного аргументу)

$$F = (f_1, \dots, f_m). \quad (1.2)$$

Це означає, що

$$y_i = f_i(x), \quad i = 1, \dots, m$$

тобто задачу синтезу вектор-функції F завжди можна визначити як m задач синтезу скалярних функцій f_i . Як видно, визначення F по I і J

визначає собою завдання виявлення закономірності або завдання синтезу моделі. Для вирішення цих завдань розроблено численні методи, які можуть бути використані для екстраполяції. Розглянемо найбільш поширені методи і проаналізуємо їх специфіку з позицій екстраполяції або прогнозування.

Різні методи синтезу F відрізняються різними способами завдання апріорної інформації.

Розглянемо випадок, коли J задає структуру F і критерій ефективності моделі Q . Нехай F_{st} - задана структура формалізованого зв'язку, тобто

$$Y = F_{st}(X, C),$$

де $C = (c_1, \dots, c_k)$ – невідомі параметри, які визначаються мінімізацією заданого критерію ефективності

$$Q(F_{st}(X, C)I) \rightarrow \min_{C \in S} \Rightarrow C^*.$$

Тут S – область визначення шуканих параметрів C (наприклад, $R^k - k$ -вимірний декартів простір), а C^* – рішення цієї задачі. Отримана таким чином модель $Y = F_{st}(X, C^*)$ і є реалізацією шуканого зв'язку F^0 , яка використовується при екстраполяції. Як критерій ефективності Q зазвичай використовують величину типу залишкової дисперсії, наприклад,

$$Q = \sum_{i=1}^N |F_{st}(X_i, C) - Y_i|^2. \quad (1.4)$$

Очевидно, найскладнішим тут є завдання структури зв'язку F_{st} . Для того, як правило, необхідно розташовувати теорією цієї причинно-наслідкового зв'язку F^0 , що в значній мірі знецінює вказаний підхід. Дійсно, в переважній більшості практичних завдань щодо структури оператора F сказати нічого не можна, а тому і застосовувати цей підхід не можна.

Справа дещо полегшується у разі, коли оператор моделі представляється у вигляді розкладання по заданій системі функцій.

Наприклад, при розгляді скалярного випадку ($m = 1$), який тривіально узагальнюється на загальний, модель можна представити у вигляді

$$y = \sum_{j=1}^k c_j \varphi_j(X) \quad ,$$

де $\Phi(X) = \{\varphi_1(X), \dots, \varphi_k(X)\}$ – задана система лінійно-незалежних функцій, що визначає структуру моделі.

Параметри C моделі можна обчислити методом найменших квадратів

$$Q(C) = \sum_{i=1}^N \left[\sum_{j=1}^k c_j \varphi_j(X_i) - y_i \right]^2 \rightarrow \min_C, \quad (1.5)$$

де використовується протокол

$$I = \langle X_i, y_i (i = 1, \dots, N) \rangle.$$

Рішення задачі (1.5), як відомо, зводиться до вирішення системи лінійних алгебраїчних рівнянь

$$\partial Q / \partial c_l = 0 (l = 1, \dots, k), \quad (1.6)$$

коефіцієнти якої визначають інформаційну матрицю

$$A = \left\| \sum_{i=1}^N \varphi_j(X_i) \varphi_l(X_i) \right\|_{k \times k}.$$

Очевидно, що для існування рішення системи (1.6) необхідно, щоб $N \geq k$. Оцінимо величину k для нелінійної системи функцій (1.4), яка найчастіше використовується в цьому випадку:

$$\phi_l(X) = x_1^{\gamma_{1l}} x_2^{\gamma_{2l}} \dots x_n^{\gamma_{nl}}, \quad l = 1, \dots, k,$$

де γ_{il} , $i = 1, \dots, n$ – показники ступеня – цілі невід'ємні числа (включно з нулем). Для лінійної моделі $\phi_l(X) = x_{l-1}$ число невідомих параметрів S дорівнює $k = n + 1$ ($l = 1, \dots, n + 1, x_0 = 1$). Для повної квадратичної моделі $k = 1 + n + n(n + 1) / 2$.

Як видно, з ростом розмірності моделі і її порядку число k дуже швидко зростає. Так само швидко зростає об'єм N необхідного протоколу. Ця обставина сильно звужує можливість застосування даного методу. Дійсно, невдалий вибір системи функцій (1.5) вимагає для оптимальної моделі значно збільшити число членів k , що в свою чергу змушує збільшити об'єм протоколу I .

Для складних об'єктів визначити систему функцій розкладання апріорі зазвичай не вдається. Однак існують методи апостеріорного визначення поліноміальної структури, до числа найбільш ефективних з яких відноситься метод групового обліку аргументів (МГОА).

Синтез структури моделі F по МГОА відбувається по етапах. На першому етапі утворюються приватні опису виду квадратичної форми

$$y^{(li)} = \psi(x_l, x_j, C^{(li)}), \quad i = 1, \dots, q > n, \quad l \neq j = 1, \dots, n, \quad (1.7)$$

де номери змінних l і j визначаються або повним перебором для малих n (і тоді $q = (n - 1)!$), або для великих n випадковим чином;

$C^{(li)}$ – параметри i -ї часткової моделі на першому етапі,

ψ – найпростіша квадратична форма, наприклад, виду

$$y^{(li)} = c_1^{(li)} + c_2^{(li)} x_l + c_3^{(li)} x_j + c_4^{(li)} x_l x_j,$$

$C^{(li)} = (c_1^{(li)}, \dots, c_4^{(li)})$ – параметри i -го приватного опису визначаються методом найменших квадратів (1.5) по протоколу I . Далі слід селекція приватних моделей.

Потім відбираються n (або більше) часткових моделей (1.7), що мають мінімальну залишкову дисперсію, які використовуються на другому етапі для аналогічного побудови q часткових описів виду

$$y^{(2i)} = \psi(y^{(1l)}, y^{(1j)}, C^{(2i)}), \quad (1.8)$$

де роль аргументу виконують часткові описи, що пройшли селекцію першого етапу. Як легко помітити, описи виду (1.8) є поліномами четвертого порядку. Параметри $C^{(2i)}$ визначаються методом найменших квадратів за тим же протоколом I . Далі здійснюється селекція отриманих часткових описів за критерієм мінімуму дисперсії, і моделі, що залишилися, надходять на третій етап і т.д. Легко помітити, що порядок моделей подвоюється на кожному етапі. Експерименти показують, що МГОА дає можливість отримати хороші структури моделей при малому об'ємі протоколу N .

Однак МГОА є евристичним методом, і як у всякого евристичного метода, його ефективність суттєво залежить від вдалого вибору ряду факторів (числа етапів селекції q , додаткових критеріїв, числа часткових моделей і т.д.).

Розглянуті вище методи є методами "наскрізного" опису. Це означає, що будується одна формула моделі, що претендує на опис поведінки об'єкта у всій області S , охопленій протоколом I . Очевидно, що такий підхід гарний лише в тому випадку, коли об'єкт поводить досить однаково в усій області S . При наявності якихось локальних відхилень наскрізне опис погано "справляється" зі своїм завданням. Потрібно вводити локальний опис моделі в районі тієї точки (X_{N+1}) , де необхідна екстраполяція.

Локальний метод найменших квадратів (ЛМНК) спирається на локальну модель в районі точки $Z = X_{N+1}$:

$$Y = F_Z(X, C_Z), \quad (1.9)$$

яка відрізняється від (1.3) більш простою структурою (знак структури *st* тут опущений). Очевидно, що параметри C_Z такої моделі залежать від точки Z . Для виявлення цієї залежності введемо скалярну вагову функцію $\rho(W) \geq 0$ векторного аргументу W , максимум якої відповідає $W = 0$ і яка монотонно убиває з ростом $|W|$. Вважаючи $W = X - Z$, можна величиною $\rho(X - Z)$ визначити міру важливості інформації протоколу I в районі точки екстраполяції Z . Дійсно, чим більше відмінність X_i і Z , тим менше інформації містить X_i для екстраполяції в Z .

Тепер параметри C_Z можна визначити, мінімізуючи функцію нев'язки:

$$\sum_{i=1}^N \rho(X_i - Z) [F(X_i, C_Z) - Y_i]^2 \rightarrow \min_{C_Z} \Rightarrow C_Z^*. \quad (1.10)$$

Вирішення цієї задачі C_Z^* і дає рішення задачі екстраполяції

$$Y_{N+1} = f(X_{N+1}, C_{X_{N+1}}).$$

З огляду на локальність моделі (1.9) її структура може бути дуже простою.

1.2 Моделі прогнозування

1.2.1 Лінійні регресійні моделі

1) Модель авторегресії (AR) Загальний вид моделі авторегресії p -го порядку - AR (p) може бути представлений наступним рівнянням:

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_p y_{t-p} + \varepsilon_t \quad (t=1,2,\dots,n), \quad (1.11)$$

де $\beta_1, \beta_2, \dots, \beta_p$ – деякі константи;

ε_t – випадкові помилки, що утворюють «білий шум»:

$$\begin{aligned} M(\varepsilon_t) &= 0, \\ D(\varepsilon_t, \varepsilon_{t \pm \tau}) &= \begin{cases} \sigma_0^2 & \text{при } \tau = 0 \\ 0 & \text{при } \tau \neq 0 \end{cases}, \end{aligned} \quad (1.12)$$

AR (p) - модель описує досліджуваний процес в момент t в залежності від його значень у попередні моменти $t-1, t-2, \dots, t-p$.

2) Модель змінного середнього (МА)

У моделях змінного середнього поточне значення стаціонарного випадкового процесу другого порядку y_t , представляється у вигляді лінійної комбінації поточного і минулих значень помилки $\varepsilon_t, \varepsilon_{t-1}, \dots, \varepsilon_{t-q}$, за своїми властивостями відповідної «білого шуму». Таке подання може бути виражено наступним рівнянням (модель змінного середнього порядку q – МА (q)):

$$y_t = \varepsilon_t - \gamma_1 \varepsilon_{t-1} - \gamma_2 \varepsilon_{t-2} - \dots - \gamma_q \varepsilon_{t-q}, \quad (1.13)$$

де $\gamma_1, \gamma_2, \dots, \gamma_q$ – параметри моделі.

Відповідно до визначення «білого шуму» помилка ε_t характеризується наступними властивостями:

$$\begin{aligned} M(\varepsilon_t) &= 0, \\ D(\varepsilon_t, \varepsilon_{t-\tau}) &= \begin{cases} \sigma_0^2 & \text{при } \tau = 0 \\ 0 & \text{при } \tau \neq 0 \end{cases}. \end{aligned} \quad (1.14)$$

3) Модель авторегресії-ковзного середнього (ARMA)

Для досягнення більшої гнучкості при побудові моделі досліджуваних процесів корисно включати в неї і члени змінного середнього, і авторегресійні члени. Це призводить до змішаної моделі ARMA (p, q) [19]:

Загальний вигляд моделі авторегресії-ковзного середнього - ARMA (p, q) визначається наступним рівнянням:

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_p y_{t-p} + \varepsilon_t - \gamma_1 \varepsilon_{t-1} - \gamma_2 \varepsilon_{t-2} - \dots - \gamma_q \varepsilon_{t-q}, \quad (1.15)$$

де $\beta_1, \beta_2, \dots, \beta_p, \gamma_1, \gamma_2, \dots, \gamma_q$ – коефіцієнти моделі;

p – порядок авторегресії;

q – порядок змінного середнього.

Зауважимо, що модель (1.15) може бути перетворена або в модель авторегресії AR (p)

$$y_t = \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_p y_{t-p} + \xi_t, \quad (1.16)$$

де помилка ξ_t , задовольняє властивостям процесу змінного середнього порядку q , або в модель змінного середнього – MA (q), шляхом вираження змінних $y_{t-\tau}$, через лінійні комбінації помилок:

$$y_t = \beta_1 (\varepsilon_{t-1} - \gamma_1 \varepsilon_{t-2} - \gamma_2 \varepsilon_{t-3} - \dots - \gamma_q \varepsilon_{t-q-1}) + \beta_2 (\varepsilon_{t-2} - \gamma_1 \varepsilon_{t-3} - \gamma_2 \varepsilon_{t-4} - \dots - \gamma_q \varepsilon_{t-q-2}) + \dots + \beta_p (\varepsilon_{t-p} - \gamma_1 \varepsilon_{t-p-1} - \gamma_2 \varepsilon_{t-p-2} - \dots - \gamma_q \varepsilon_{t-p-q}) + \varepsilon_t - \gamma_1 \varepsilon_{t-1} - \gamma_2 \varepsilon_{t-2} - \dots - \gamma_q \varepsilon_{t-q} \quad (1.17)$$

4) Інтегрована модель авторегресії-ковзного середнього.

Зазначені вище моделі припускають, що часовий ряд є стаціонарним, тобто його властивості не змінюються в часі. На практиці це часто буває не так, і при наявності лінійного росту моделі виявляються неефективними. Наступна модифікація моделі, ARIMA (p, d, q) (Autoregressive Integrated Moving Average, модель Бокса-Дженкінса) дозволяє за допомогою лагового оператора привести ряд до стаціонарного виду. Для цього проводиться взяття різниць порядку d від вихідного часового ряду. Модель ARIMA (p, d, q)

означає, що різниці часового ряду порядку d підкоряються моделі ARMA (p, q) .

Для роботи з іншими формами нестационарності, крім наявності тренда, можуть застосовуватися інші моделі, такі як сезонна модель ARIMA (SARIMA) або моделі авторегресійної умовної гетероскедастичності (ARCH / GARCH).

Відзначимо, що багато моделей є спеціальними випадками моделі ARIMA [20,21].

Підхід Бокса-Дженкінса до аналізу часових рядів є дуже потужним інструментом для побудови точних прогнозів з малою дальністю прогнозування. Моделі ARIMA досить гнучкі і можуть описувати широкий спектр характеристик часових рядів, що зустрічаються на практиці. Формальна процедура перевірки моделі на адекватність є простою і доступною. Крім того, прогнози і інтервали прогнозування витікають безпосередньо з підбраної моделі. Однак використання моделей ARIMA має і кілька недоліків.

1. Необхідна відносно велика кількість вихідних даних.
2. Не існує простого способу коригування параметрів моделей ARIMA, такого як в деяких методах згладжування, коли використовуються нові дані. Модель необхідно періодично повністю перебудовувати, а іноді потрібно вибрати абсолютно нову модель.
3. Побудова задовільної моделі ARIMA часто вимагає великих витрат часу і ресурсів. Для моделей ARIMA витрати на побудову моделі, час виконання обчислень і обсяги необхідних баз даних можуть виявитися істотно вище, ніж для більш традиційних методів прогнозування, таких як згладжування.

Модель експоненціального згладжування (exponential smoothing, ES) застосовується для моделювання фінансових і економічних процесів [20]. В основу експоненціального згладжування закладена ідея постійного перегляду прогнозних значень в міру надходження фактичних. Модель ES привласнює

експоненційно спадні ваги спостереженням в міру їх старіння. Таким чином, останні доступні спостереження мають більший вплив на прогнозне значення, ніж старші спостереження.

Моделі прогнозування на основі ланцюгів Маркова (Markov chain model) припускають, що майбутній стан процесу залежить тільки від його поточного стану і не залежить від попередніх [22]. У зв'язку з цим процеси, що моделюються ланцюгами Маркова, повинні відноситися до процесів з короткою пам'яттю.

Класифікаційно–регресійні дерева (classification and regression trees, CART) є ще однією популярною структурною моделлю прогнозування часових рядів [23]. Структурні моделі CART розроблені для моделювання процесів, на які впливають як неперервні зовнішні фактори, так і категоріальні. Якщо зовнішні фактори, що впливають на процес $Z(t)$, неперервні, то використовуються регресійні дерева; якщо чинники категоріальні, то – класифікаційні дерева. У разі, якщо необхідно враховувати фактори обох типів, використовуються змішані класифікаційно-регресійні дерева [24].

1.2.2 Нелінійні регресійні моделі

Досить зручним і ефективним виявилось застосування нелінійних моделей, запропонованих і розвинених в роботах [25-27]. Так, на основі представлення лінійних систем ARMAX моделлю, в [27] для опису нелінійних систем використовується нелінійна модель авторегресії змінного середнього з додатковими вхідними сигналами (NARMAX модель)

$$y(k) = f[y(k-1), \dots, y(k-k_y), u(k-1), \dots, u(k-k_u), e(k-1), \dots, e(k-k_e)] + e(k), \quad (1.18)$$

де $y(k)$ – вихідний сигнал об'єкта;

$u(k)$ – вхідний сигнал об'єкта;

$e(k)$ – помилка вимірювань;

$f[\cdot]$ – нелінійна функція перетворення;

k_y, k_u, k_e – порядки запізнювання по вихідному, вхідному сигналам об'єкта і помилці вимірювань відповідно.

За аналогією з ARX-моделями лінійних систем нелінійні системи можуть бути представлені нелінійної NARX-моделлю

$$y(k) = f[y(k-1), \dots, y(k-k_y), u(k-1), \dots, u(k-k_u)] + e(k). \quad (1.19)$$

Моделі (1.18), (1.19) є непараметричних, проте використання теореми Стоуна - Вейерштраса дає можливість їх параметризувати [28].

Поліноміальне уявлення, яке при цьому отримується, дозволяє перейти від задачі визначення нелінійної функції, що зв'язує вхідні і вихідні сигнали моделі, до задачі оцінювання параметрів, які входять лінійно в шукану функцію.

Використання параметризованих моделей також значно спрощує обчислення, так як призводить до лінійного щодо невідомих параметрів рівняння, яке може бути вирішене добре розробленими і досить повно дослідженими методами. Слід зазначити, що представлення об'єктів у вигляді NARMAX або NARX моделей (1.20), (1.21) відіграють фундаментальну роль при дослідженні процесів за допомогою ШНМ.

Узагальненням NARMAX і NARX моделей є моделі, що враховують різні види нелінійностей по різних змінних. Серед таких моделей виділяють наступні [29].

Модель 1. Прогнозований сигнал лінійно залежить від останніх його значень і нелінійно від останніх значень сигналу

$$y(k+1) = \sum_{i=0}^{n-1} \alpha_i y(k-i) + f[u(k), u(k-1), \dots, u(k-m+1)]. \quad (1.20)$$

Модель 2. Прогнозований сигнал лінійно залежить від сигналу і його останніх значень і нелінійно від своїх власних останніх значень (рис.1.4)

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1)] + \sum_{i=0}^{m-1} \beta_i u(k-i). \quad (1.21)$$

Модель 3. Прогнозований сигнал залежить нелінійно як від останніх значень сигналу $u(k)$, так і від останніх значень сигналу, а результуючі значення є адитивними функціями від i

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1)] + g[u(k), u(k-1), \dots, u(k-m+1)]. \quad (1.22)$$

Модель 4. Прогнозований сигнал в будь-який момент часу є нелінійною функцією останніх значень як сигналу, так і сигналу

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n+1); u(k), u(k-1), \dots, u(k-m+1)]. \quad (1.23)$$

Дана модель є найбільш загальною з усіх моделей, описаних вище. Вона являє собою суму моделей (1.20) – (1.22), є аналітично розв'язуємою, а, отже, для практичних додатків в деяких випадках може бути навіть більш привабливою.

1.2.3 Нейромережеві нелінійні моделі

При побудові нелінійних моделей використовують два принципово різні підходи, засновані на різних способах формування за допомогою моделі прогнозованого сигналу об'єкта [28,29]. Перший підхід використовує при формуванні сигналу моделі вимірних сигналів $u(k), u(k-1), \dots$, і

$y(k), y(k-1), \dots$, а другий - сигнали $u(k), u(k-1), \dots$ і оцінки вихідних сигналів моделі $\hat{y}(k), \hat{y}(k-1), \dots$.

Відповідно до цього розрізняють паралельну модель, що описується рівнянням

$$\hat{y}(k+1) = f(\hat{y}(k), \hat{y}(k-1), \dots, u(k), u(k-1), \dots) \quad (1.24)$$

і послідовно-паралельну виду

$$\hat{y}(k+1) = f(y(k), y(k-1), \dots, u(k), u(k-1), \dots). \quad (1.25)$$

Незважаючи на те, що паралельна модель дозволяє здійснювати прогнозування вихідного сигналу на будь-яке число тактів вперед, а послідовно-паралельна тільки на один такт, найбільшого поширення при вирішенні задач ідентифікації набуло використання послідовно-паралельної моделі.

При нейромережевому підході досліджуваний об'єкт представляється у вигляді нейронної мережі, що містить крім вхідного і вихідного один або кілька прихованих шарів, кожен з яких складається з певної кількості нейронів, які реалізують задану базисну функцію (функцію активації).

При побудові моделі використовується апріорна інформація щодо класу, до якого вона належить. Якщо відомо, що модель має структуру, яка описується співвідношенням (1.24), при використанні паралельної моделі структура нейромережевої моделі буде аналогічна структурі моделі з та заміною f і g на N_2 і N_1 відповідно. Задача побудови такої моделі полягає в обчисленні ваг двох нейронних мереж N_1 і N_2 , так щоб представлення N_1 було еквівалентним $g[\cdot]$, а представлення $N_2 - f$.

При використанні послідовно-паралельної моделі необхідно враховувати той факт, що для формування виходу моделі використовується $y(k+1)$.

Таким чином, рівняння моделі має вигляд

$$\hat{y}(k+1) = N_2 [y(k), \dots, y(k-n+1)] + N_1 [u(k), \dots, u(k-m+1)].$$

При нейромережевому підході побудова математичної моделі ґрунтується на концепції «чорного ящика», коли дослідник фіксує реакцію мережі та вхідні фактори, що її викликають а функція, що зв'язує чинники і реакцію мережі, є невідомою. У зв'язку з цим вибір структури моделі, під яким, як і в розглянутому вище випадку традиційних моделей, розуміється визначення необхідної кількості регресорів, є одним з найважливіших етапів побудови моделі. У зв'язку з тим, що при нейромережевому підході цей етап розроблений і формалізований недостатньо, при виборі структури нейромережевої моделі використовуються ті ж критерії, що й при виборі традиційної моделі.

1.3 Порівняльний аналіз моделей прогнозування. Переваги і недоліки моделей

Регресійні моделі і методи. До переваг даних моделей прогнозування відносять простоту, гнучкість, а також однаковість їх аналізу та проектування. При використанні лінійних регресійних моделей результат прогнозування може бути отриманий швидше, ніж при використанні інших моделей. Крім того, перевагою є прозорість моделювання, тобто доступність для аналізу всіх проміжних обчислень.

Основним недоліком нелінійних регресійних моделей є складність визначення виду функціональної залежності, а також трудомісткість визначення параметрів моделі. Недоліками лінійних регресійних моделей є низька адаптивність і відсутність здатності моделювання нелінійних процесів [30].

1) Авторегресійні моделі і методи. Важливими перевагами даного класу моделей є їх простота і прозорість моделювання. Ще однією перевагою

є однаковість аналізу і проектування. На сьогоднішній день даний клас моделей є одним з найбільш популярних, а тому у відкритому доступі легко знайти приклади застосування авторегресійних моделей для вирішення задач прогнозування часових рядів різних предметних областей [31].

Недоліками даного класу моделей є: велике число параметрів моделі, ідентифікація яких неоднозначна і ресурсомісткість; низька адаптивність моделей, а також лінійність і, як наслідок, відсутність здатності моделювання нелінійних процесів, часто зустрічаються на практиці [32].

2) Моделі і методи експоненціального згладжування. Перевагами даного класу моделей є простота і однаковість їх аналізу та проектування. Даний клас моделей частіше інших використовується для довгострокового прогнозування [33].

Недоліком даного класу моделей прогнозування є відсутність гнучкості.

3) Нейромережеві моделі і методи. Основною перевагою нейромережевих моделей є нелінійність, тобто здатність встановлювати нелінійні залежності між майбутніми та фактичними значеннями процесів. Іншими важливими перевагами є: адаптивність, масштабованість (паралельна структура ШНМ прискорює обчислення) і однаковість їх аналізу та проектування [34].

При цьому недоліками ШНМ є відсутність прозорості моделювання; складність вибору архітектури, високі вимоги до несуперечності навчальної вибірки; складність вибору алгоритму навчання і ресурсомісткість процесу їх навчання [35].

4) Моделі і методи на базі ланцюгів Маркова. Простота і одноманітність аналізу і проектування є перевагами моделей на базі ланцюгів Маркова.

Недоліком даних моделей є відсутність можливості моделювання процесів з довгою пам'яттю [36].

5) Моделі на базі класифікаційно-регресійних дерев. Перевагами даного класу моделей є: масштабованість, за рахунок якої можлива швидка

обробка надвеликих обсягів даних; швидкість і однозначність процесу навчання дерева (на відміну від ШНМ) [22], а також можливість використовувати категоріальні зовнішні чинники.

Недоліками даних моделей є неоднозначність алгоритму побудови структури дерева; складність питання зупинки, тобто питання про те, коли варто припинити подальші розгалуження; відсутність однаковості їх аналізу та проектування [33].

Необхідно додатково відзначити, що ні для однієї з розглянутих груп моделей (і методів) в перевагах не зазначена точність прогнозування. Це зроблено у зв'язку з тим, що точність прогнозування того чи іншого процесу залежить не тільки від моделі, але і від досвіду дослідника, від доступності даних, від можливої апаратної потужності і багатьох інших факторів.

На сьогоднішній день найбільш поширеними моделями прогнозування є авторегресійні моделі (ARIMAX), а також нейромережеві моделі (ШНМ).

1.4 Аналіз рекомендацій щодо вибору критерію прогнозування

Однією з найбільш складних проблем при прогнозуванні є питання вибору найбільш підходящої міри (критерію) оцінювання якості прогнозу.

На цей час існує досить багато досліджень, присвячених аналізу критеріїв якості прогнозування та розробці рекомендацій їх ефективного вибору. Найбільш поширені критерії прогнозування наведено у додатку А.

Розглянемо деякі з них.

Якщо оцінка прогнозу здійснюється для часових рядів з одним масштабом і були виконані процедури попередньої обробки даних (очищення даних, виявлення аномалій), розумно вибрати MAE, MdAE, RMSE. У разі використання різних масштабів ці міри критерії не застосовуються [37].

Щодо порівняння критеріїв MSE та MAE слід зауважити, що є тільки одна ситуація, в якій помилкам на основі MSE можна віддати перевагу щодо помилок на основі MAE – це прогнозування цілочисельного попиту. Обидві ці

оцінки MAE і MSE добре підходять, наприклад, для визначення точності прогнозу однієї і тієї ж послідовності при різних параметрах моделі або при використанні різних моделей, але вони виявляють помилок прогнозу менше, ніж при наївному методі. Ці оцінки так само придатні для визначення точності прогнозу на два або більше кроків вперед. Для цього потрібно лише замість помилки прогнозу на один крок при їх розрахунку використовувати значення помилок прогнозу на відповідну кількість кроків вперед.

Якщо дані набору не містять викидів, і часовий проміжок, на якому проводилися розрахунки, не є настільки великим, щоб не мати близьких за значенням даних, доцільним є використання масштабованих оцінок.

Якщо дані мають сезонні або циклічні закономірності, рекомендується використовувати нормовані помилки, коли можуть бути розраховані коефіцієнти нормування в межах інтервалу, рівного циклу або сезону.

При оцінці точності декількох прогнозів потрібен спосіб агрегування. Якщо прийняти середнє арифметичне, тоді слід використовувати середню абсолютну процентну похибку(MAPE).

Для різномасштабних випадків можна надати наступні рекомендації.

Незважаючи на те, що процентні помилки зазвичай використовуються при прогнозуванні в реальних задачах, внаслідок несиметрії застосовувати їх не рекомендують. Якщо діапазон значень лежить в позитивній півплощині, і в даних немає викидів, доцільно використовувати симетричні міри помилки.

Якщо дані «забруднені», тобто містять викиди, доцільно застосовувати масштабні міри, такі як MASE, inRSE. У цьому випадку необхідно [37], щоб:

- 1) горизонт прогнозування був достатнім;
- 2) не було однакових значень прогнозуємого ряду;
- 3) значення нормалізованого коефіцієнта не повинно дорівнювати нулю.

У зв'язку з тим, що на різних наборах даних виходять різні результати оцінок помилок прогнозування, при відсутності результатів попереднього

аналізу і необхідності узагальнення методів замість однієї метрики прогнозування розумно використовувати певний набір метрик помилки.

Слід зазначити, що іноді в літературі зустрічається порівняння моделей на основі RMSE. Таке порівняння в загальному випадку некоректно [38] і зазвичай призводить до вибору неправильної, тобто не самої точної, моделі. Це обумовлено тим, що RMSE сильно схильна до впливу викидів (через квадрат у формулі). Тому в результаті використання цього критерію перевага віддається менш точним моделям, які посередньо прогнозують дані, але при цьому не допускають жодної великої помилки. Очевидно, що будь-які коефіцієнти, засновані на RMSE, також будуть приводити до спотворених результатів.

У зв'язку з тим, що на різних наборах даних виходять різні результати оцінок помилок прогнозування, а кожен критерій має недоліки, які можуть призводити до неточного визначення якості прогнозування, неможливо виділити один універсальний критерій. Тому розумним є використання певного набору метрик помилки.

1.5 Постановка задач дослідження

На основі системного аналізу проблеми прогнозування в різних сферах застосування можна зробити висновок щодо актуальності задач прогнозування та необхідності їх подальшого дослідження з метою підвищення ефективності розв'язання задач зазначеного класу.

Як показує аналіз існуючих методів прогнозування, одними з найбільш ефективних є нейромережеві моделі і методи, основною перевагою яких є нелінійність, тобто здатність встановлювати нелінійні залежності між майбутніми та фактичними значеннями процесів.

Однак необхідно зауважити, що задача прогнозування значно ускладнюється, якщо досліджуваний процес нестационарний.

Нестационарність процесу потребує розробки швидкодіючих процедур навчання ШНМ та дослідження їх властивостей.

У зв'язку з тим, що метою дисертаційної роботи є розробка нейромережових методів і засобів прогнозування нестационарних послідовностей з підвищеною точністю при наявності апріорної і поточної невизначеності, для досягнення поставленої мети необхідно вирішити такі завдання:

- розглянути методи вибору критерія якості прогнозування і провести їх порівняльний аналіз;
- розвинути нейромережовий метод прогнозування, нестационарного процесу, що описується марківською моделлю першого порядку;
- дослідити властивості адаптивних процедур прогнозування за умов наявності завад;
- розробити просту в обчислювальному відношенні робастну процедуру прискореного навчання нейронних мереж на основі мінімізації комбінованого функціоналу;
- вирішити задачу оптимізації процедур навчання з метою скорочення процесу навчання та підвищення їх обчислювальної стійкості;
- застосувати розроблені процедури та моделі для отримання моделей та прогнозування параметрів технологічного процесу виробництва кальцинованої соди.

Вирішенню поставлених завдань, що виникають в процесі прогнозування нестационарних послідовностей на основі нейромережевого підходу, присвячені наступні розділи.

1.6 Висновки за першим розділом

1. Проведено системний аналіз проблеми прогнозування в різних сферах застосування. В результаті зроблено висновки:

– існуючі методи прогнозування в основному базуються на використанні лінійних та нелінійних (NARMAX, NARX тощо) регресійних моделей. Проведений порівняльний аналіз моделей прогнозування дозволив виявити їх переваги та недоліки;

– одними з найбільш ефективних є нейромережеві моделі і методи, основною перевагою яких є нелінійність, тобто здатність встановлювати нелінійні залежності між майбутніми та фактичними значеннями процесів. Іншими важливими перевагами є: адаптивність, масштабованість (паралельна структура ШНМ прискорює обчислення) і однаковість їх аналізу та проектування.

2. Здійснено огляд та аналіз нейромережевих структур, що використовуються для вирішення задач прогнозування.

3. Проаналізовано критерії оцінки роботи нейронної мережі в разі прогнозування. У зв'язку з тим, що на різних наборах даних виходять різні результати оцінок помилок прогнозування, а кожен критерій має недоліки, які можуть призводити до неточного визначення якості прогнозування, неможливо виділити один універсальний критерій. Тому розумним є використання певного набору метрик помилки.

4. Проаналізовано методи побудови ШНМ для вирішення задачі прогнозування та розглянуто задачі їх структурної та параметричної оптимізації.

5. Сформульовано мету та задачі дослідження дисертаційної роботи

Список використаних джерел у даному розділі наведено у повному списку використаних джерел під номерами: 19-37.

2 ПРОГНОЗУВАННЯ ЧАСОВИХ РЯДІВ НА ОСНОВІ ШНМ

2.1 Штучні нейронні мережі, що застосовуються для вирішення завдань ідентифікації, управління і прогнозування

Деякі типи ШНМ внаслідок хороших апроксимуючих властивостей з'явилися зручним інструментом для розв'язання задачі ідентифікації нелінійних об'єктів та прогнозування. До числа таких найбільш використовуваних ШНМ відносяться статичні мережі, такі як багатошаровий перцептрон, радіально-базисна, узагальнено-регресійна мережа, мережа АДАЛІНА, нейронна мережа СМАС та динамічні ШНМ.

При використанні ШНМ, як і при традиційному підході, також виникають завдання структурної та параметричної оптимізації, відповідні вибору оптимальної топології мережі і її навчання (налаштування параметрів). Якщо завдання визначення структури є дискретною оптимізаційною (комбінаторною), то пошук оптимальних параметрів здійснюється в безперервному просторі за допомогою класичних методів оптимізації.

2.1.1 Багатошаровий перцептрон (БШП)

БШП є нейронну мережу з декількома шарами, кожен з яких складається з обчислювальних вузлів (нейронів) [22,38,38-39]. Архітектура багатошарового перцептрона може бути зручно записана як $n_0 - n_1 - \dots - n_l$, де n_0 є розмірністю вхідного вектора мережі, а n_i , $1 \leq i \leq l$ позначає номери вузлів у відповідних шарах. Загальна залежність входу-виходу i -го нейрона в k -му шарі записується як

$$\begin{aligned} z_i^{(k)} &= \sum_{j=1}^{n_{k-1}} w_{ij}^{(k)} u_j^{(k-1)} + \mu_i^{(k)}; \\ u_i^{(k)} &= f\left(z_i^{(k)}\right), \end{aligned} \quad (2.1)$$

де $w_{ij}^{(k)}$, $\mu_i^{(k)}$ – відповідно ваги на з'єднаннях нейронів і зсуву (порогові значення), що визначаються в процесі навчання мережі;

$f(\cdot)$ – функція активації нейрона (зазвичай логістична або функція гіперболічного тангенса).

Апроксимуючі можливості БШП були досліджені і описані багатьма авторами [22,28,38-41].

2.1.2 Радіально - базисна мережа (РБМ)

Представлення входу-виходу РБМ можна виразити як

$$\hat{f}_i(u) = \sum_{j=1}^{n_h} w_{ji} \phi_j(u), \quad 1 \leq i \leq m, \quad (2.5)$$

де w_{ji} – ваги лінійних суматорів;

n_h – число вузлів в прихованому шарі.

У даних мережах в якості базисних вибираються деякі функції відстані між векторами $\phi_i(u) = f(\|u - c_i\|)$, де c_i – центри базисних функцій (БФ).

Вихідний нейрон, як правило, лінійний, а його роль зводиться до зваженого підсумовування сигналів, що надходять від нейронів прихованого шару.

Слід також зазначити, що при наявності великої кількості невідомих параметрів (ваг, центрів, дисперсій) використання великої навчальної вибірки призводить до того, що при ідентифікації складних нелінійних об'єктів різко

зростає складність обчислень, і застосування даної мережі для вирішення задачі ідентифікації в режимі on-line стає проблематичним.

2.1.3 Узагальнено – регресійна мережа (УРМ)

Дана мережа, запропонована Шпехтом Д. [42] для побудови узагальнених (лінійних і нелінійних) регресій. Як і РБС, дана мережа використовує БФ $\phi_i(n)$, які, однак, є нормованими [43-46].

При цьому

$$\sum_{i=1}^{n_h} \phi_i(n) = 1. \quad (2.9)$$

2.1.4 Штучна нейронна мережа АДАЛІНА

Дана ШНМ, що є першою лінійною нейронною мережею, була запропонована Б.Уїдроу і М.Е.Хоффом в роботі [47] в 1960 р. Розроблена ними модель була названа адаптивним нейроном, а потім ADALINE – ADaptive LInear NEuron. З плином часу свою популярність в якості ШНМ вона втратила і тепер ADALINE це ADaptive LInear Element.

Основна відмінність АДАЛІНИ від персептрону полягає в використовуваному алгоритмі навчання.

2.1.5 Нейронна мережа СМАС

У 1975 р. Дж. Альбусом була запропонована модель, яка описує процеси управління рухом, що відбуваються в мозочку, яка згодом була реалізована в нейромережевому регуляторі для управління роботом-маніпулятором, названому ім СМАС – Cerebellar Model Articulation Controller (церебральна модель артикуляційного контролера) [48]. Згодом модель стала основою для

розробки нейромережевої асоціативної пам'яті, званої також СМАС, проте в цьому випадку дана аббревіатура часто розшифровується як Cerebellar Model Arithmetic Computer (церебральна модель арифметичного комп'ютера). Простота реалізації і хороші апроксимуючі властивості мережі забезпечили їй досить широке застосування не тільки при управлінні у реальному часі роботом-маніпулятором, але і при вирішенні задач розпізнавання образів, кодування зображень, цифрової обробки і фільтрації сигналів, апроксимації, відновлення і інтерполяції одно - і багатовимірних функцій.

2.1.6 Мережа локально - лінійних моделей (МЛЛМ)

Ідея побудови такої мережі полягає в можливості апроксимації нелінійної функції кусково-лінійними функціями (моделями) [49-50]. Завдання визначення оптимального числа таких функцій аналогічна класичній задачі структурної ідентифікації і вирішується в кожному конкретному випадку окремо. Кожна модель грає роль вагового коефіцієнта за аналогією з РБМ. Як активаційні функції в МЛЛМ вибирають зазвичай нормовані гаусові функції.

Вихідний сигнал мережі обчислюється таким чином:

$$\hat{y} = \sum_{i=1}^N (w_{0i} + w_{1i}u_1 + w_{2i}u_2 + \dots + w_{Ni}u_N) \cdot \phi_i(u, c_i, \sigma_i), \quad (2.10)$$

де $\phi_i(u, c_i, \sigma_i)$ – БФ мережі.

Як і в разі РБМ, в даній мережі шляхом навчання повинні бути визначені її параметри w, c і σ .

2.1.7 Динамічні ШНМ

В теорії ШНМ розрізняють повнозв'язні і частково-рекурсивні динамічні мережі, причому перші зважаючи на їх складної архітектури, як правило, не

знаходять практичного застосування. До більш ефективним частково-рекурсивним ІНС відносять мережу з елементами затримки сигналу, мережі Елмана і Джордана [43,51].

2.2 Аналіз процедур навчання ШНМ

Навчання або адаптація ШНМ відбувається зазвичай шляхом налаштування ваг і інших параметрів мережі так, щоб відобразити можливі вхідні значення в значення на виході мережі.

Базовими стратегіями навчання є: навчання з учителем і навчанням без учителя. У разі навчання без учителя інформація про бажане значення на виході мережі, яка відповідає кожній вхідного образу, відсутня.

У зв'язку з тим, що в задачах нейромережевої ідентифікації та прогнозування використовується навчання з учителем, зупинимося на ньому докладніше. Навчання ШНМ – ітеративний процес. На кожній ітерації обчислюються мережеві виходи для одного (або більше) зразка в наборі навчання, і коригуються мережеві ваги з метою зменшення помилки між фактичним мережевим виходом $(y_{i,p}^L)$ ($i=1, \dots, N^L$) і цільовим виходом $(y_{i,p}^*)$ для даного зразка. При використанні підходящої мережної архітектури і алгоритму навчання мережеві ваги прагнуть до значень, при яких мережевий вихід стає прийнятно близьким до цільового виходу для кожного зразка в наборі навчання.

Задача навчання мережі полягає в мінімізації опуклої функції втрат $F[e]$:

$$F[e_i] = \sum_{i=1}^k \rho(e_i), \quad (2.12)$$

де $\rho(e(i))$ – деяка функція втрат, $e_i = y_i - \hat{y}_i$, і зводиться до пошуку оцінки $\hat{\theta}$

$$\hat{\theta}_k = \arg \min_{\theta} F_k(\theta), \quad (2.13)$$

яка визначається як рішення системи рівнянь

$$\nabla F(\theta_j) = \frac{\partial F(\theta)}{\partial \theta_j} = \sum_{i=1}^K \psi(e(i, \theta)) \frac{\partial e(i, \theta)}{\partial \theta_j} = 0, \quad (2.14)$$

де $\psi(e(i, \theta)) = \frac{\partial \rho(e(i, \theta))}{\partial e(i, \theta)}$ – функція впливу.

Введення вагової функції дозволяє представити систему рівнянь (2.14) наступним чином:

$$\sum_{i=1}^k \omega(e_i) e_i \frac{\partial e_i}{\partial \theta_j} = 0. \quad (2.15)$$

В цьому випадку мінімізація функціоналу (2.12) буде еквівалентна мінімізації зваженого квадратичного функціоналу.

Зазвичай в якості $\rho(e(i))$ вибирається квадратична функція, що призводить до оцінки найменших квадратів. При цьому функція впливу $\psi(e(i, \theta))$, росте лінійно зі збільшенням i , що і пояснює нестійкість оцінки МНК до викидів та перешкод, розподіли яких мають великі хвости.

Перший теоретично доведений алгоритм для налаштування ваг при навчанні багат шарових ШНМ відомий як алгоритм зворотного поширення помилки (ЗП) [22,28,38-40,43,51]. У стандартній реалізації алгоритму ЗП для мінімізації цільової функції (функції помилки) використовується градієнтний метод пошуку (заснований на локальній лінійній апроксимації цільової функції), згідно з яким на кожній ітерації обчислюється градієнт мінімізуемого функціоналу і коригуються мережеві ваги за правилом

$$\theta_{n+1} = \theta_n - \gamma \nabla f_n, \quad (2.16)$$

де γ – ітераційний параметр, який впливає на швидкість навчання (зазвичай $\gamma \in (0,1)$);

$$\nabla f_n = \left(\frac{\partial F(e)}{\partial \theta_{1n}} \quad \frac{\partial F(e)}{\partial \theta_{2n}} \quad \dots \quad \frac{\partial F(e)}{\partial \theta_{Nn}} \right)^T \quad (2.17)$$

Правило (2.16) гарантує зменшення загальної мережевої помилки в кожному періоді. Градієнт в (2.16) обчислюється в двох різних періодах - при прямому і зворотному проході.

Головний недолік градієнтного методу – його повільна збіжність. Тому замість цього алгоритму використовують його якусь модифікацію або ж, якщо ж цільова функція $f(w_n)$ двічі диференційована, метод Ньютона, або його модифікацію – метод Левенберга-Марквардта [44-46,52-53].

У ряді випадків більш ефективними виявляються квазіньютонівські методи (метод Давідона-Флетчера-Пауелла, метод Бройде, метод Бройде-Флетчера-Шенно), в основі яких лежить ідея відновлення квадратичної апроксимації цільової функції за значеннями її градієнтів у ряді точок.

Головного недоліку квазіньютонівських методів – необхідності зберігати і перераховувати матрицю Гессе, що для великих розмірностей вимагає великого обсягу пам'яті і виконання великого числа арифметичних операцій, позбавлені методи сполучених градієнтів (метод Флетчера-Ривса, метод Полака-Ріб'єра тощо) [45].

Однак найбільш широке поширення при навчанні ШНМ отримав рекурентний метод найменших квадратів (РМНК), що дозволяє налаштовувати параметри мережі в режимі on-line. Для оцінювання нестационарних параметрів або для прогнозування нестационарних процесів досить ефективним виявився РМНК з експоненціальним зважуванням інформації, що виходить в результаті мінімізації критерію якості навчання виду [22,28,38-39]:

$$E = \sum_{i=1}^n \lambda^{n-i} e_i^2, \quad (2.21)$$

і має вигляд:

$$w_{n+1} = [\nabla F_{n+1}^T A \nabla F_{n+1}]^{-1} \nabla F_{n+1}^T A D_{n+1}, \quad (2.22)$$

де

$$\nabla F_{n+1} = [\nabla f_1^T \nabla f_2^T \dots \nabla f_{n+1}^T] - \text{матриця } (n+1) \times N; \quad (2.23)$$

$$D(k+1) = [d(1)d(2)\dots d(k+1)] - \text{вектор } (n+1) \times 1; \quad (2.24)$$

$$A = \begin{bmatrix} \lambda^{n+1} & 0 & \dots & 0 \\ 0 & \lambda^n & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} - \text{матриця } (n+1) \times (n+1); \lambda \in (0,1]; \quad (2.25)$$

$e_i = d_i - w_{i-1}^T \nabla f_i$ – помилка навчання.

Рекурентна оцінка (2.22) записується таким чином:

$$w_{n+1} = w_n + \frac{P_n \nabla f_{n+1}}{\lambda + \nabla f_{n+1}^T P_n \nabla f_{n+1}} e_{n+1}, \quad (2.26)$$

$$P_{n+1} = \frac{1}{\lambda} \left(P_n - \frac{P_n \nabla f_{n+1} \nabla f_{n+1}^T P_n}{\lambda + \nabla f_{n+1}^T P_n \nabla f_{n+1}} \right). \quad (2.27)$$

Використання такого механізму зважування інформації дозволяє навчати мережу, що описує нестационарні динамічні об'єкти. Зазвичай в практичних додатках приймають $\lambda = 0.995 \div 0.999$.

Більш простими в обчислювальному відношенні є градієнтні однокрокові алгоритми, серед яких найбільш швидкодіючим є алгоритм Качмажа [54], відомий в теорії ШНМ як алгоритм Уїдроу-Хоффа [47,55]

$$w_{n+1} = w_n - \gamma \frac{e_{n+1}}{\|\nabla f_{n+1}\|^2} \nabla f_{n+1}, \quad (2.28)$$

де $\gamma \in (0,2)$ – коефіцієнт навчання.

Останнім часом все більш популярними, зокрема в загорткових ШНМ, стають процедури навчання на основі стохастичного градієнту. Декілька найбільш поширених процедур наведено в додатку Б.

2.3 Багатошаровий перцептрон як нейромережева модель

Метою будь-якого прогнозування є створення моделі, яка дозволяє прогнозувати майбутнє і оцінити тенденції в змінах того чи іншого фактора.

Побудова математичної моделі, використовуваної при прогнозуванні, пов'язана з апроксимацією деяких, в загальному випадку нелінійних, функцій

$$y(x) = f(x) + \xi, \quad (2.29)$$

де $x = [x_1, x_2, \dots, x_N]^T$ – вектор узагальненого вхідного сигналу $N \times 1$;

$f(\bullet)$ – невідома нелінійна функція;

ξ – завада;

T – символ транспонування, тобто полягає в отриманні оцінки функції $f(\bullet)$ за вимірюваннями вхідних і вихідних змінних.

Серед найбільш широко застосовуваних для вирішення цієї задачі ШНМ досить ефективним є БП, який використовує апроксимації нелінійного оператора $f(\bullet)$ виду

$$\begin{aligned} \hat{y}(x) &= \hat{f}(x) = \\ &= f^q \left[(W^q)^T f^{q-1} \left[(W^{q-1})^T f^{q-2} \left[\dots f^1 \left[(W^1 x + b_1)^T \right] \dots \right] \right] \right] + b_q, \end{aligned} \quad (2.30)$$

де W^i – вектор вагових параметрів нейронів i -го шару мережі;

$f^i [\bullet]$ – активаційна функція (АФ) i -го шару;

b_i – зміщення i -го нейрону.

Існують два основних підходи для надання нейронним мережам на базі БП властивостей, необхідних для обробки динамічних даних: додавання елементів затримок сигналу на вході мережі та додавання рекурентних зв'язків до внутрішньої структури мережі. В першому випадку, при використанні «методу часового вікна» [46], нейронна мережа отримує на вхід разом з поточним значенням затримані в часі минулі вхідні сигнали. Навчання нейромережі здійснюється методом зворотного поширення і з застосуванням градієнтних методів оптимізації.

Іншим підходом введення динаміки в нейромережі прямого поширення є додавання внутрішніх рекурентних зв'язків у шари нейромережі (мережі Джордана, Елмана тощо) [22,28,38-40,51].

Кількість нейронів прихованого шару БП попередньо можна оцінити таким чином [4,7,16-17, 56]:

$$\frac{N}{10} - n - m \leq n_k \leq \frac{N}{2} - n - m, \quad (2.33)$$

де n_k – кількість нейронів в прихованому шарі;

n – кількість вхідних сигналів;

m – кількість вихідних сигналів;

N – кількість елементів статичної вибірки, необхідної для навчання.

Кількість синаптичних ваг для персептрона с одним прихованим шаром обчислюється так [57]

$$K_w = n \cdot n_k \cdot m. \quad (2.37)$$

Розглянемо задачу прогнозування часового ряду, що описується рівнянням [7,17]

$$y_{n+1} = f(z_{n+1}) = f(w^{*T} x_{n+1}), \quad (2.38)$$

де $z_{n+1} = w^{*T} x_{n+1}$, $x_{n+1} = (x_{1,n+1}, x_{2,n+1}, \dots, x_{N,n+1})^T$ – вектор вхідних сигналів $N \times 1$;

$x_{ik} = x_{k-i-1}$, $i = 1, 2, \dots, N$; $f(\bullet)$ – невідома нелінійна функція;

$w^* = (w_1^*, w_2^*, \dots, w_N^*)^T$ – вектор вагових параметрів $N \times 1$.

Помилка прогнозування визначається таким чином

$$e_{n+1} = y_{n+1}^* - y_{n+1},$$

де y_{n+1}^* – сигнал, що прогнозується.

Точність прогнозування будемо оцінювати за критерієм

$$E_{n+1} = 0,5e_{n+1}^2. \quad (2.39)$$

Як вже зазначалося вище, задача навчання ШНМ полягає в налаштуванні вектора її параметрів w_{n+1} за формулою

$$w_{n+1} = w_n + \Delta w_n, \quad (2.40)$$

де Δw_n – приріст значень вектора параметрів;

$$\Delta w_n = -\gamma \nabla_w [E_n]_{w=w_n}, \quad (2.41)$$

γ – коефіцієнт, що впливає на час навчання.

Для отримання правила налаштування розглянемо розклад помилки прогнозування на $k+1$ -у кроці в ряд Тейлора і обмежимося лінійними членами

$$e_{n+1} = e_n + \left(\frac{\partial e_n}{\partial w_n} \right)^T \Delta w_n + \left(\frac{\partial e_n}{\partial x_n} \right)^T \Delta x_n + \frac{\partial e_n}{\partial y_n^*} \Delta y_n^* + \dots \quad (2.42)$$

У зв'язку з цим квадрат помилки можна представити таким чином

$$e_{n+1}^2 = e_n^2 + 2e_n \left(\frac{\partial e_n}{\partial w_n} \right)^T \Delta w_n + \left\| \left(\frac{\partial e_n}{\partial w_n} \right)^T \Delta w_n \right\|^2 \quad (2.43)$$

Враховуючи (2.38), (2.39), можна отримати наступні вирази для похідних в рівнянні (2.43)

$$\frac{\partial e_n}{\partial w_n} = -f'(z_n x_n^T),$$

а

$$\left\| \left(\frac{\partial e_n}{\partial w_n} \right)^T \Delta w_n \right\|^2 = \Delta w_n^T \frac{\partial e_n}{\partial w_n} \left(\frac{\partial e_n}{\partial w_n} \right)^T \Delta w_n.$$

Це дозволяє записати (2.43) наступним чином

$$\Delta w_n = -\gamma e_n f'(z_n) x_n. \quad (2.44)$$

Враховавши, що

$$\Delta w_n^T \frac{\partial e_n}{\partial w_n} \left(\frac{\partial e_n}{\partial w_n} \right)^T \Delta w_n = \Delta w_n^T (-f'(z_n) x_n) (x_n^T (-f'(z_n))) \Delta w_n,$$

можна записати наступний вираз для e_{n+1} :

$$e_{n+1} \approx e_n [1 - \gamma |f'(z_n)|^2 \|x_n\|^2],$$

що дозволяє представити e_{n+1}^2 так

$$e^2_{n+1} = e^2_n \left[1 - 2\gamma(f'(z_{n+1}))^2 \|x_{n+1}\|^2 + \gamma^2(f'(z_{n+1}))^4 \|x_{n+1}\|^4 \right].$$

Для збіжності алгоритму (2.40), (2.41) необхідне виконання нерівності

$$\left| 1 - 2\gamma(f'(z_{n+1}))^2 \|x_{n+1}\|^2 + \gamma^2(f'(z_{n+1}))^4 \|x_{n+1}\|^4 \right| < 1,$$

звідки маємо

$$0 < \gamma < \frac{2}{(f'(z_{n+1}))^2 \|x_{n+1}\|^2}. \quad (2.45)$$

З умови $\partial e^2_{n+1} / \partial \gamma = 0$ можна отримати вираз для оптимального значення параметра γ , при якому швидкість збіжності процедури навчання буде максимальною

$$\gamma_{n+1}^{opt} = \frac{1}{(f'(z_{n+1}))^2 \|x_{n+1}\|^2}. \quad (2.46)$$

Тобто отримуємо алгоритм Качмажа (Уідроу-Хоффа) [22,38].

Для підвищення обчислювальної стійкості процедури навчання слід скористатися регуляризацією процедури навчання, тобто замість (2.46) взяти

$$\gamma'_{n+1} = \frac{1}{(f'(z_{n+1}))^2 \|x_{n+1}\|^2 + \delta_{n+1}}, \quad (2.47)$$

де $\delta_{n+1} > 0$ – параметр регуляризації.

2.4 Прогнозування часових рядів на основі ШНМ АДАЛІНА

Як було відмічено вище, основна відмінність АДАЛІНИ від персептрону полягає в використовуваній процедурі навчання.

2.4.1 Прогнозування з використанням процедури Качмажа

Хоча навчання АДАЛПНИ також складалося в корекції її вагових коефіцієнтів на підставі пред'явлених навчальних пар і порівняння реальних вихідних сигналів моделі \hat{y}_{n+1} бажаними (необхідними) y_{n+1} , Уїдроу і Хофф, ґрунтуючись на роботах Н. Вінера, пов'язаних з дослідженням фільтрів, використовували для цього градієнтна процедура Качмажа

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \frac{e_{n+1}x_{n+1}}{\|x_{n+1}\|^2}, \quad (2.51)$$

де $e_{n+1} = y_{n+1} - \hat{y}_{n+1} = y_{n+1} - \hat{\theta}_n^T x_{n+1}$, \hat{y}_{n+1} – вихідний сигнал моделі;

$\hat{\theta}_n = (\hat{\theta}_{1,n}, \hat{\theta}_{2,n}, \dots, \hat{\theta}_{N,n})^T$ – вектор оцінок параметрів $N \times 1$;

γ – деякий параметр, що впливає на швидкість збіжності процедури;

$\|\bullet\|$ – евклідова норма

Ця процедура була запропонована в роботі [54] для вирішення систем лінійних алгебраїчних рівнянь, згодом, починаючи з [58-60], з успіхом її почали застосовувати для розв'язання задачі ідентифікації при побудові моделі. Оцінки швидкості збіжності даної процедури при ідентифікації стаціонарних об'єктів вперше були отримані в [61,62]. Слід, однак, відзначити, що даний алгоритм застосовується не тільки в системах ідентифікації, але і набув широкого поширення при вирішенні задач фільтрації [63-66].

Для підвищення обчислювальної стійкості (2.51) В.М. Чадеєвим [58] була запропонована його модифікація - регуляризована процедура

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \frac{e_{n+1}x_{n+1}}{\|x_{n+1}\|^2 + \delta}, \quad (2.52)$$

де $\delta > 0$ – параметр регуляризації.

Передбачається, що оцінювані нестационарні параметри, які є стаціонарними випадковими процесами, можна описати марківською моделлю першого порядку [67,1,14]

$$\theta_{n+1} = \theta_n + S_{n+1}, \quad (2.53)$$

де $S_{n+1} = (S_{n+1,1}, S_{n+1,2}, \dots, S_{n+1,N})^T$ – вектор випадкової послідовності $N \times 1$; $S_i \sim N(0, \sigma_s^2)$.

Враховуючи наявність завад ξ_{n+1} і ς_{n+1} , вихідний сигнал об'єкта може бути представлений наступним чином:

$$y_{n+1} = \theta_n^T x_{n+1} + s_{n+1} x_{n+1} + \theta_n^T \varsigma_{n+1} - \theta_n^T \varsigma_{n+1} + \xi_{n+1}.$$

Введемо вектор помилок оцінювання

$$\tilde{\theta}_{n+1} = \theta_{n+1} - \hat{\theta}_{n+1}. \quad (2.54)$$

Враховуючи (2.53), вираз для можна записати так:

$$e_{n+1} = \tilde{\theta}_n^T x_{n+1} - \theta_n^T \varsigma_{n+1} + S_{n+1}^T x_{n+1} + \xi_{n+1}. \quad (2.55)$$

Передбачається, що завади мають нормальний закон з

$$\xi_{n+1} \sim N(0, \sigma_\xi^2), \quad \varsigma_{i,n+1} \sim N(0, \sigma_\varsigma^2).$$

Враховуючи прийнятий тип нестационарності (2.53), процедуру Качмажа відносно помилок ідентифікації можна записати як

$$\begin{aligned}
\tilde{\theta}_{n+1} &= \tilde{\theta}_n + \\
&+ S_{n+1} - \gamma \frac{\tilde{x}_{n+1} \tilde{x}_{n+1}^T}{\|\tilde{x}_{n+1}\| + \delta} \tilde{\theta}_n + \gamma \frac{\tilde{x}_{n+1} \zeta_{n+1}^T}{\|\tilde{x}_{n+1}\| + \delta} \theta_n - \gamma \frac{\tilde{x}_{n+1} x_{n+1}^T}{\|\tilde{x}_{n+1}\| + \delta} S_{n+1} - \frac{\tilde{x}_{n+1}}{\|\tilde{x}_{n+1}\|^2 + \delta} \xi_{n+1} = \\
&= \left(1 - \gamma \frac{\tilde{x}_{n+1} \tilde{x}_{n+1}^T}{\|\tilde{x}_{n+1}\|^2 + \delta} \right) (\tilde{\theta}_n + S_{n+1}) + \gamma \frac{\tilde{x}_{n+1} \zeta_{n+1}^T}{\|\tilde{x}_{n+1}\| + \delta} \theta_n - \gamma \frac{\tilde{x}_{n+1}}{\|\tilde{x}_{n+1}\|^2 + \delta} \xi_{n+1}.
\end{aligned} \tag{2.56}$$

Передбачається, що компоненти вектора похибки оцінки $\tilde{\theta}_n$ підпорядковуються нормальному закону розподілу із $\tilde{\theta}_{i,n} \triangleq M\{\tilde{\theta}_{i,n}\}$ та дисперсією $\sigma_{i,n}^2$.

Аналіз впливу завади ζ_{n+1} на статистичні властивості оцінки наведено у додатку В [9]. З наведених там результатів видно, що вектор оцінки буде пов'язаний з вектором шуканих параметрів співвідношенням:

$$M\{\hat{\theta}_{n+1}\} = \frac{(\sigma_{\tilde{x}}^2 - N\sigma_{\zeta}^2 + \delta) \left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^3}{\left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^3 - \delta N \left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^2 + 2N\delta\sigma_{\tilde{x}}^2} \theta_{n+1}, \tag{2.57}$$

звідки випливає, що

$$\lim_{n \rightarrow \infty} M\{\hat{\theta}_{n+1}\} \neq 0, \tag{2.58}$$

тобто оцінка буде зміщеною.

Зауважимо, що формула (2.57) не містить характеристик нестаціонарності та статистичних характеристик завади ξ_{n+1} . Це пов'язано з типом прийнятої моделі нестаціонарності та властивостями ξ_{n+1} .

Використовуючи вищезазначені формули, легко отримати, що для регуляризованої процедури Качмажа

$$M\{\hat{\theta}_{n+1}\} = \frac{\sigma_x^2 + \delta}{\sigma_x^2 + \sigma_\zeta^2 + \delta} \theta_{n+1}, \quad (2.59)$$

а для класичної процедури Качмажа з $\delta = 0$

$$M\{\hat{\theta}_{n+1}\} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_\zeta^2} \theta_{n+1}. \quad (2.60)$$

Якщо вхідний сигнал не має завад, оцінка буде незміщеною.

Як видно з вищезазначених результатів, наявність завади ξ_{n+1} не призводить до зміщення в оцінці.

Вплив завади ξ_{n+1} на дисперсію оцінки також вивчено в Додатку В. З (В.10) випливає, що для збіжності процедури (2.52) у середньоквадратичному необхідно виконання умови

$$\left| 1 - \frac{\gamma}{N} \left[(2 - \gamma) - \frac{2(\gamma - 1)(\delta(N - 2)\sigma_x^2 + \delta)^2 + 2\sigma_x^2\delta}{((N - 2)\sigma_x^2 + \delta)^3} - \frac{\gamma^2\delta^2}{((N - 2)\sigma_x^2 + \delta)^2} \right] \right| < 1 \quad (2.61)$$

2.4.2 Прогнозування з використанням процедури Нагумо-Ноди

Регуляризована процедура Нагумо-Ноди [68], відома в задачах фільтрації як signed-regressor NLMS (SR-NMLS) [69], має вигляд

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \gamma \frac{e_{n+1} \text{sign} x_{n+1}}{|x_{n+1}| + \delta}, \quad (2.62)$$

де γ – деякий параметр (параметр релаксації), що впливає на швидкість збіжності процедури.

$$|x_{n+1}| = x_{n+1}^T \text{sign} x_{n+1};$$

$$\text{sign} x_{n+1} = \begin{cases} 1, & \text{при } x_{n+1} > 0, \\ 0 & \text{при } x_{n+1} = 0, \\ -1 & \text{при } x_{n+1} < 0. \end{cases}$$

При оцінюванні нестационарних параметрів, описуваних моделлю (2.53), і наявності завад процедура може бути записана щодо помилок ідентифікації таким чином:

$$\tilde{\theta}_{n+1} = \left(I - \gamma \frac{\text{sign} x_{n+1} x_{n+1}^T}{|x_{n+1}| + \delta} \right) (\tilde{\theta}_n + S_{n+1}) - \frac{\text{sign} x_{n+1} x_{n+1}^T}{|x_{n+1}| + \delta} \xi_{n+1}. \quad (2.63)$$

Для аналізу збіжності в середньому обчислимо $M \{ \tilde{\theta}_{n+1} \}$.

Як було показано вище, наявність завади ζ_{n+1} у вхідному сигналі призводить до зміщення оцінки. Тому при дослідженні процедури Нагумо-Ноди будемо розглядати тільки заваду на виході ξ_{n+1} .

З огляду на статистичну незалежність сигналів і завад, а також враховуючи формули [61]

$$M \left\{ \frac{1}{|x_{n+1}| + \delta} \right\} \approx \frac{\sqrt{\pi\omega}}{N + \delta\sqrt{\pi\omega}}; \quad (2.64)$$

$$M \left\{ \frac{\text{sign} x_{n+1} x_{n+1}^T}{|x_{n+1}| + \delta} \right\} = \left(\frac{1}{N} - \delta M \left\{ \frac{1}{|x_{n+1}| + \delta} \right\} \right) I \approx \frac{1}{N} - \delta \frac{\sqrt{\pi\omega}}{N + \delta\sqrt{\pi\omega}}, \quad (2.65)$$

де $\omega = (2\sigma_x^2)^{-1}$, отримаємо співвідношення, з яких випливає, що процедура

(2.62) буде збігатися в середньому при виконанні умови

$$\left| 1 - \gamma \frac{N + \sqrt{\pi\omega} - N\delta\sqrt{\pi\omega}}{N(N + \sqrt{\pi\omega})} \right| < 1, \quad (2.66)$$

або

$$0 < \gamma < \frac{2N(N + \sqrt{\pi\omega})}{N + \sqrt{\pi\omega} - N\delta\sqrt{\pi\omega}}, \quad (2.67)$$

тобто при виконанні цієї умови, оцінка, що отримується за допомогою даної процедури, буде незміщеною.

Для встановлення збіжності в середньоквадратичному вчинимо по аналогії з вищевикладеним. Після множення обох частин (2.63) зліва на $\tilde{\theta}_{n+1}^T$ і обчислення математичного сподівання з урахуванням того, що [61]

$$M \left\{ \frac{\|x_{n+1}\|^2}{(|x_{n+1}| + \delta)^2} \right\} = \frac{\pi N}{2(N + \delta\sqrt{\pi\omega})^2} + O\left(\frac{1}{N^2}\right), \quad (2.68)$$

отримаємо [1,14]

$$\begin{aligned} & M\{\|\tilde{\theta}_{n+1}\|^2\} \\ &= \left(1 - \frac{\gamma}{N + \delta\sqrt{\pi\omega}} \left(2 - \gamma \frac{\pi N}{2(N + \delta\sqrt{\pi\omega})} \right) \right) M\{\|\tilde{\theta}_n\|^2\} + \sigma_s^2 + \\ &+ \gamma^2 \frac{\pi N \omega \sigma_\xi^2}{(N + \delta\sqrt{\pi\omega})^2}. \end{aligned}$$

З цього виразу випливає, що процедура (2.62) буде збігатися в середньоквадратичному при виконанні умови:

$$\left| 1 - \frac{\gamma}{N + \delta\sqrt{\pi\omega}} \left(2 - \gamma \frac{\pi N}{2(N + \delta\sqrt{\pi\omega})} \right) \right| < 1. \quad (2.69)$$

При виконанні (2.69) область збіжності процедури (2.62) (з урахуванням того, що $\omega = \left(2\sigma_x^2\right)^{-1}$) визначається наступним виразом:

$$\lim_{n \rightarrow \infty} M \left\{ \left[\tilde{\theta}_{n+1} \right]^2 \right\} = \frac{\gamma \pi N \sigma_{\xi}^2}{\left(4(N + \delta \sqrt{\pi \omega})^2 - \gamma \pi N \right) \sigma_x^2}. \quad (2.70)$$

З отриманих виразів при $\delta = 0$ слідують відомі співвідношення для алгоритму Нагумо-Ноди.

2.5 Прогнозування часових рядів на основі коеволюційних нейронних мереж

Слід зазначити, що останнім часом зазначені завдання все частіше прагнуть вирішувати одночасно за допомогою коеволюційних адаптивних систем, тобто систем, що складаються з різноманітних груп особин, які еволюціонують та діють спільно для виконання складних обчислень або вироблення спільного ефективного поведінки [12,70-73].

Використання коеволюційних ШНМ в задачах прогнозування вивчалось в багатьох роботах, зокрема в [10,13,15,73].

Розглянемо використання мережі СМАС для вирішення цієї задачі.

В запропонованій Альбусом мережі використовуються два загальних перетворення:

$$S: X \Rightarrow A, \quad (2.71)$$

$$P: A \Rightarrow y, \quad (2.72)$$

де $X - N$ – вимірний простір неперервних вхідних сигналів;

$A - n$ – вимірний простір асоціацій;

y – одновимірний вихідний сигнал.

Перетворення $S \Rightarrow A$ в свою чергу складається з двох перетворень

$$S: X \Rightarrow A, \quad (2.73)$$

$$P: A' \Rightarrow y, \quad (2.74)$$

де A' – перетворений алгоритмом хешування простір асоціацій;

y – вектор вихідних сигналів.

Перетворення (2.73) відповідає кодуванню інформації, а (2.74) – обчисленню вихідного сигналу

$$\hat{y} = P(a') = (a')^T w = (H(a'))^T w, \quad (2.75)$$

Вираз (2.75) описує перетворення, яке здійснюється в традиційній СМАС з використанням хешування інформації. Якщо ж в мережі використовується апарат нечіткої логіки, то перетворення (2.75) приймає вид

$$\hat{y} = H(a^T \Phi(x))w, \quad (2.76)$$

$$\text{де } \Phi(x) = \begin{bmatrix} \Phi_1(x) & 0 & \dots & 0 \\ 0 & \Phi_2(x) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \Phi_n(x) \end{bmatrix}.$$

Для традиційної СМАС $\Phi(x) = I$.

Асоціативні властивості СМАС проявляються в використуваному вигляді адресації, заснованому на спеціальному кодуванні вхідної інформації [37,35,20]. Структура СМАС наведена на рисунку 2.1.

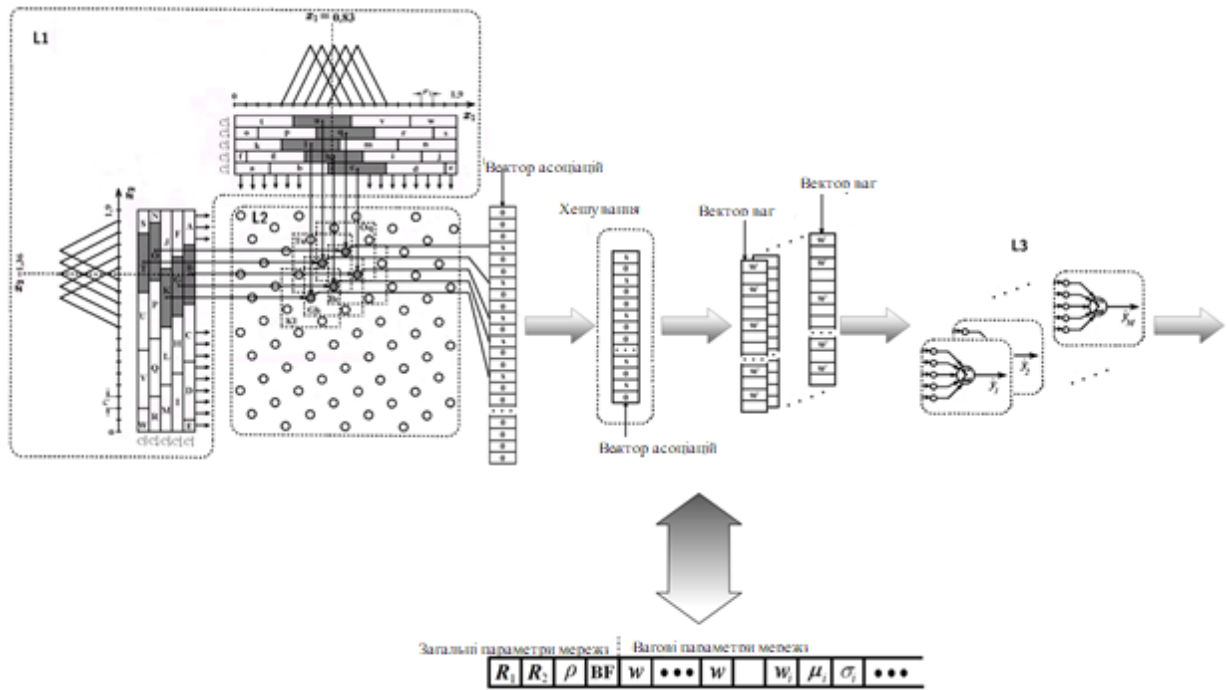


Рисунок 2.1 – Мережа СМАС

Кодування інформації в даній мережі полягає в тому, що кожному N -вимірному вхідному вектору $x(i)$ ставиться у відповідність n -вимірний вектор асоціацій $a(i)$, що зберігається в віртуальній пам'яті, елементи якого можуть набувати значень з інтервалу $[0,1]$ (в цитованих вище роботах прийнято, що ці елементи приймають значення 0 або 1). При цьому тільки $\rho \ll n$ елементів даного вектора мають відмінні від нуля значення, тобто тільки ρ елементів пам'яті є активними. Розмірність цих векторів, n , що дорівнює сумі елементів всіх матриць A_i ($i = \overline{1, \rho}$), може бути обчислена за формулою

$$n = \left\lceil \rho \left(\frac{R-1}{\rho} + 1 \right)^N \right\rceil, \quad (2.77)$$

де R – число рівнів квантування вхідних сигналів, яке використовується;

N – розмірність вхідного вектора;

$\lceil \bullet \rceil$ – означає округлення в бік найближчого більшого цілого числа.

Вибір БФ нейронів шару L1 істотно впливає на апроксимуючі властивості мережі СМАС. Як уже зазначалося, традиційна СМАС здійснює

кусково-постійну апроксимацію, що є наслідком використання в ній нейронів з прямокутною активаційною функцією. При виборі БФ прямокутної форми обчислювальні витрати будуть мінімальними. У мережі СМАС в якості БФ також широко використовуються гауссівські БФ, а також В-сплайни, безсумнівною перевагою яких є можливість рекурентного обчислення як самих В-сплайнів, так і їх похідних.

Визначення параметрів мережі (її ваг, параметрів базисних функцій і т.д.) здійснюється шляхом її навчання з учителем.

При використанні функцій приналежності з формою, відмінною від прямокутної, даний алгоритм може бути записаний у такий спосіб:

$$w(k+1) = w(k) + \gamma(k) \left(\frac{y(k) - a^T(k)\Phi(x)w(k)}{\|\Phi(x)a(k)\|^2} \Phi(x)a(k) \right), \quad (2.78)$$

де $\gamma(k)$ – деякий в загальному випадку змінний параметр.

При переході від ШНМ до ЕШНМ для всіх типів мереж використовуються загальні еволюційні процедури (ініціалізація популяції, оцінка популяції, селекція, схрещування, мутації), а відмінності полягають лише в способі кодування структури і параметрів тієї чи іншої ШНМ у вигляді хромосоми. На початку роботи ЕА випадковим чином ініціюється популяція, що складається з N особин (ШНМ).

На рисунку 2.1 показана ЕСМАС, формат її хромосоми і відповідність між параметрами мережі і генами, що зберігаються в хромосомі.

Слід зазначити, що довжина хромосоми залежить від розмірності задачі і максимально допустимої кількості пам'яті. Як видно з рисунка, хромосома складається з генів, в яких зберігається інформація про відповідні параметри мережі. На початку хромосоми йдуть гени, які містять інформацію про параметри завади і є активними лише в разі наявності завад вимірів. Наступний блок генів кодує кількість рівнів і шаблів квантування, форму рецептивного поля нейронів і тип застосовуваного гешування інформації. У

зв'язку з великою кількістю БФ, які можуть застосовуватися в ЄМАС, в її хромосому вводиться спеціальний ген ВФ, що відповідає за кодування виду використовуваної функції. Потім в хромосомі йде група генів, що кодує безпосередньо вагові параметри відповідних асоціативних нейронів. На етапі ініціалізації всім цим параметрам за допомогою датчика випадкових чисел привласнюють початкові значення, що знаходяться в деякому допустимому діапазоні. У зв'язку з тим, що в процесі еволюції може відбуватися мутація в параметрах, що впливають на кількість використовуваних асоціативних нейронів, довжина хромосоми може змінюватися [27,29,74].

Після того як початкова популяція сформована, проводиться оцінка пристосованості кожної що входить в неї особини, яка визначається деякою функцією пристосованості (фітнес-функцією). Зазвичай в якості такої функції використовується квадратична:

$$F(x) = \frac{1}{M} \sum_{i=1}^M (y^*(x_i) - \hat{y}(x_i))^2, \quad (2.79)$$

де y_n^* – бажана реакція мережі;

\hat{y}_n – реальний вихідний сигнал;

M – розмір вибірки.

Наступним кроком є відбір особин, хромосоми яких братимуть участь у формуванні нового покоління, і їх подальше схрещування, з метою передачі генетичної інформації від батьківських особин до їх нащадків. Після завершення роботи даного оператора ген будь-якої особини в новій популяції може мутувати, тобто змінити своє значення. Оскільки в хромосомі використовується гібридне кодування, при мутації необхідно здійснювати різні операції для різних методів кодування.

Таким чином, алгоритм роботи ЄМАС може бути представлений таким чином.

1. Створення початкової популяції.

- 1.1. Ініціювання хромосоми кожної особини.

- 1.2. Оцінювання початкової популяції.
2. Етап еволюції – побудова нового покоління.
 - 2.1. Відбір кандидатів на схрещування (селекція).
 - 2.2 Схрещування, тобто породження кожною парою відібраних кандидатів нових індивідів.
 - 2.3. Мутація.
 - 2.4. Оцінювання нової популяції.
3. Перевірка критерію завершення, якщо не виконано - перехід на 2.
- 2.6 Висновки за другим розділом

1. Деякі типи ШНМ внаслідок хороших апроксимуючих властивостей з'явилися зручним інструментом для розв'язання задачі ідентифікації нелінійних об'єктів та прогнозування. До числа таких найбільш використовуваних ШНМ відносяться статичні мережі, такі як багатошаровий перцептрон, радіально-базисна, узагальнено-регресійна мережі, мережа АДАЛІНА, нейронна мережа СМАС та динамічні ШНМ.

2. Розглянуто задачу навчання ШНМ, яка полягає в мінімізації деякої опуклої функції втрат, вибір якої залежить від наявності інформації щодо статистичних властивостей сигналів і завад. В основі процедур навчання зазвичай лежать градієнтні алгоритми мінімізації. Розглянуто найбільш поширені на цей час алгоритми, зокрема РМНК та найбільш швидкодіючий серед однокрокових алгоритм Качмажа (Уїдроу–Хоффа).

3. Розглянуто БШП в якості нейромережевої моделі для прогнозування. Описано процедуру вибору його структури та запропоновано процедуру навчання. Вивчено умови її збіжності та отримано вираз для оптимального значення параметра γ , при якому швидкість збіжності процедури навчання буде максимальною.

4. Досліджено особливості прогнозування нестационарних часових рядів на основі ШНМ АДАЛІНА. У зв'язку з тим, що для навчання АДАЛІНИ

використовується алгоритм Качмажа, проведено дослідження його властивостей для оцінювання параметрів, які описуються марківською моделлю першого порядку. Отримано неасимптотичні оцінки точності регуляризованого алгоритму Качмажа.

5. Запропоновано використання для навчання АДАЛІНИ в нестационарному випадку більш простого, ніж алгоритм Качмажа, регуляризованого алгоритму Нагумо-Ноди. Визначено умови його збіжності та отримано неасимптотичні оцінки точності цього алгоритму, які дозволяють встановити потенційно можливий виграш від його використання.

6. Запропоновано використання для прогнозування нестационарних часових рядів еволюційну ШНМ СМАС, розглянуто формат її хромосоми і відповідність параметрів мережі генам, що зберігаються в хромосомі. Наведено її алгоритм роботи.

Список використаних джерел у даному розділі наведено у повному списку використаних джерел під номерами: 1, 7, 9, 14, 17, 20, 22, 27-29, 35, 37-74.

3 РОБАСТНЕ НАВЧАННЯ ШНМ ПРЯМОГО ПОШИРЕННЯ

3.1 Деякі підходи, що використовуються при робастному навчанні

Метод найменших квадратів не є стійким, оскільки цільова функція може рости до безкінечності і викиди можуть стати домінуючими вимірами, які фактично перевіряють реальну модель. В якості альтернативи, для забезпечення робастності цільову функцію модифікують таким чином, щоб обмежити вплив найбільших вимірів. Основним наслідком цього є, як правило, більш низька швидкість збіжності алгоритмів оптимізації. Це пояснюється тим, що дуже складно розрізнати в перший раз викиди і корисні вимірювання. У зв'язку з цим, деякі викиди можуть бути відфільтровані, що призводить до зменшення швидкості збіжності. У найбільш випадку невеликі, але зміщені вимірювання переміщують мінімум цільової функції.

Якщо інформація про приналежність завади деякому певному класу розподілів відома, то шляхом мінімізації оптимального критерію, що представляє собою узятий з оберненим знаком логарифм функції розподілу перешкоди, може бути отримана оцінка максимальної правдоподібності (М-оцінка). Якщо ж такої інформації немає, то для оцінювання шуканого вектора параметрів слід застосувати будь-який неквадратичний критерій, що забезпечує робастність одержуваної оцінки [75-85].

В [78,79] розглянуті деякі типи класів розподілів, що зустрічаються при вирішенні практичних завдань: P_1 – клас не вироджених розподілів, P_2 – клас розподілів з обмеженою дисперсією, P_5 – клас фінітних розподілів (завада обмежена по абсолютній величині, а будь-які відомості про щільність її розподілу відсутні), P_3 , P_4 і P_6 – класи наближено нормальних, наближено рівномірних і наближено фінітних розподілів відповідно, описуваних моделлю Тьюки-Хьюбера [75]

$$p(x) = (1 - \varepsilon)p_0(x) + \varepsilon q(x), \quad (3.1)$$

де $p_0(x)$ - щільність відповідного основного розподілу;

$q(x)$ - щільність розподілу (довільного), що засмічує;

$\varepsilon \in [0,1]$ – параметр, що характеризує ступінь засмічення основного розподілу.

Навіть якщо основний $p_0(x)$ розподіл і розподіл, що засмічує, $q(x)$ є гауссівськими з нульовими математичними сподіваннями і дисперсіями σ_1^2 і σ_2^2 , $\sigma_1^2 \ll \sigma_2^2$ відповідно, оцінки, одержувані при виборі $\rho(e) = 0,5e^2$, і є оптимальними для гауссівського розподілу, будуть нестійкими, так як $\rho'(e(i, \theta)) = e(i, \theta)$ зростає лінійно зі збільшенням $e(i, \theta)$.

Для всіх цих класів були знайдені найменш сприятливі розподіли, тобто розподіли, що мінімізують фішерівську інформацію. Так, мінімум фішерівській інформації для класу P_1 дає розподіл Лапласа $p^*(\xi) = L(0, s_\xi)$, для класу P_2 – Гаусса $p^*(\xi) = N(0, \sigma^2)$, для класів фінітних розподілів найменш сприятливою щільністю є

$$p^*(\xi) = \begin{cases} \frac{1}{l} \cos^2 \frac{\pi \xi}{2l} & \text{при } |\xi| \leq l; \\ 0 & \text{при } |\xi| > l. \end{cases} \quad (3.2)$$

З іншого боку, вибір функції втрат, відмінної від квадратичної, дозволяє забезпечити робастність оцінок, тобто їх працездатність практично для всіх розподілів завад.

3.2 Традиційні критерії М-навчання

Робастні оцінки визначаються з умови

$$w(k) = \arg \min I_k(w),$$

$$I_k(w) = \sum_{i=1}^k F[e(w, x(k))], F(e) = -\ln p(\xi)|_{\xi=e} \quad (3.3)$$

і є оптимальними в мінімаксному сенсі на відповідних класах розподілів.

Оптимальна функція втрат, тобто функція, яка мінімізує асимптотичну матрицю коваріації помилок оцінювання, дорівнює логарифму щільності розподілу завади при $\xi = e(\mathbf{x}, \mathbf{w})$, взятої з протилежним знаком

$$F_0[e(\mathbf{x}, \mathbf{w})] = -\ln p_0(\xi) \Big|_{\xi=e(\mathbf{x}, \mathbf{w})}, \quad (3.4)$$

тобто логарифмічній функції неправдоподібності.

Таким чином, оптимальною функцією втрат для класу P_1 буде модульна, для класу P_2 - квадратична, для класу P_5

$$F_0[e(\mathbf{x}, \mathbf{w})] = -\ln \cos\left(\frac{\pi e}{c}\right). \quad (3.5)$$

Для ε -засмічених імовірнісних розподілів F_0 є нелінійною на деякому інтервалі, який визначається параметрами завад, і лінійною поза цим інтервалом.

Існує досить велика кількість функціоналів, що забезпечують отримання робастних М-оцінок [75,84-86], проте найбільш поширеними є комбіновані функціонали, запропоновані Хьюбером [75] і Хемпелем [80,81] і що складаються з квадратичного, який забезпечує оптимальність оцінок для гауссівського розподілу, і модульного, що дозволяє отримати більш робастну до розподілів з важкими «хвостами» (викидами) оцінку. Дані функціонали і їх функції впливу мають відповідно вид

$$\rho_1(e) = \begin{cases} \frac{e^2}{2}, & |e| \leq c; \\ c|e| - \frac{c^2}{2}, & c < |e|; \end{cases} \quad \psi_1(e) = \begin{cases} e, & |e| \leq c; \\ c \operatorname{sign}(e), & c < |e|; \end{cases} \quad (3.6)$$

$$\psi'_1(e) = \begin{cases} 1, & |e| \leq c; \\ 0, & c < |e|; \end{cases} \quad \omega_1(e) = \begin{cases} 1, & |e| \leq c; \\ \frac{c}{e}, & c < |e|; \end{cases} \quad (3.7)$$

$$\rho_2(e) = \begin{cases} \frac{e^2}{2}, & 0 \leq |e| < b; \\ b|e| - \frac{b^2}{2}, & b \leq |e| < c; \\ \frac{b}{2}(c+d) - \frac{b}{2} \left(b - \frac{(|e|-d)^2}{c-d} \right), & c \leq |e| < d; \\ \frac{b}{2}(c+d) - \frac{b^2}{2}, & d \leq |e|; \end{cases} \quad (3.8)$$

$$\psi_2(e) = \begin{cases} e, & 0 \leq |e| < b; \\ b \operatorname{sign}(e), & b \leq |e| < c; \\ \frac{b(|e|-d)}{c-d} \operatorname{sign}(e), & c \leq |e| < d; \\ 0, & d \leq |e|; \end{cases} \quad (3.9)$$

$$\psi_2'(e) = \begin{cases} 1, & 0 \leq |e| < b; \\ 0, & b \leq |e| < c; \\ \frac{b}{c-d}, & c \leq |e| < d; \\ 0, & d \leq |e|; \end{cases} \quad \omega_2(e) = \begin{cases} 1, & 0 \leq |e| < b; \\ \frac{b}{|e|}, & b \leq |e| < c; \\ \frac{b(|e|-d)}{(c-d)|e|}, & c \leq |e| < d; \\ 0, & d \leq |e|. \end{cases}$$

Зазвичай М-оцінки описуються шляхом завдання функції впливу ψ , а не функції ρ .

Функція, запропонована Хьюбером, є монотонною, а ψ -функція Хемпела – немонотонною. Як зазначається в [87], при наявності в розподілі важких хвостів, застосування немонотонних ψ -функцій дозволяє поліпшити результати оцінювання.

Ефективність застосування цих функціоналів залежить від того, наскільки вдало вибрані константи a , b , c і d , які входять до них і визначають ступінь завадостійкості. У зазначених вище роботах рекомендується вибирати значення a , з інтервалу $[\sigma, 2\sigma]$, де σ – стандартне відхилення спостереження x , а значення b , c і d задавати рівними 1.5, 3.5 і 8 відповідно.

Класичні робастні методи, які мінімізують (3.6-3.9), так і інші неквадратичні функціонали, орієнтовані на симетричність засмічення, коли

викиди однаково часто з'являються як в області негативних, так і в області позитивних значень.

Зазначені методи дозволяють ефективно боротися з завадами, що описуються моделлю (3.1), коли основний і засмічуючий розподіли є гауссівськими з нульовими математичними сподіваннями і дисперсіями σ_1^2 і σ_2^2 , $\sigma_1^2 \ll \sigma_2^2$.

У більш загальній ситуації довільного виду засмічення, наприклад, коли гауссівський засмічуючий розподіл має ненульове математичне сподівання або коли засмічуючий розподіл є несиметричним, оцінки, що даються цими методами, будуть зміщеними [88-90].

Відзначимо, що для забезпечення робастних властивостей одержуваних оцінок досить ефективним є застосування функціоналу четвертого ступеня [91-96]

$$F[e_{n+1}] = M \left\{ e_{n+1}^4 \right\} \quad (3.10)$$

та комбінованого функціоналу навчання [97-100]

$$F[e(k)] = \lambda e^2(k) + (1 - \lambda) |e(k)|, \quad (3.11)$$

де $\lambda \in [0,1]$; $M \{ \bullet \}$ - символ математичного сподівання.

3.3 Процедури робастного навчання

3.3.1 Оптимальні процедури навчання

Якщо щільність розподілу завад $p_0(\xi)$ повністю не відома, а відомо лише, що вона належить деякому класу $p_0(\xi) \in p_k$, то для найменш сприятливого на цьому класі щільності розподілу $p^*(\xi)$ можна отримати

відповідний оптимальний алгоритм. Якщо $p_0(\xi) = p^*(\xi)$, цей алгоритм буде абсолютно оптимальним для класу p_k .

Оптимальні алгоритми навчання мережі є градієнтні процедури виду

$$\theta_{n+1} = \theta_n - H_{n+1} \nabla F_0(e_{n+1}), \quad (3.12)$$

де H_{n+1} – матриця посилення.

Враховуючи те, що $\nabla F_0(e_{n+1}) = F'(e_{n+1}) \nabla f_{n+1}$, де $\nabla f_{n+1} = \frac{\partial F'_{n+1}}{\partial \theta}$,

можна записати оптимальні алгоритми навчання наступним чином:

$$\theta_{n+1} = \theta_n + \frac{P_n \nabla f_{n+1}}{\gamma + \nabla f_{n+1} P_n \nabla f_{n+1}} F'(e_{n+1}), \quad (3.13)$$

$$P_{n+1} = P_n - \frac{P_n \nabla f_{n+1} \nabla f_{n+1}^T P_n}{\gamma + \nabla f_{n+1}^T P_n \nabla f_{n+1}}, \quad (3.14)$$

де

$$F'(e_{n+1}) = \begin{cases} e_{n+1} \text{ для } p_0(\xi) = N(0, \sigma_\xi)^2; & (3.15) \\ \text{sign}(e_{n+1}) \text{ для } p_0(\xi) = L(0, s_\xi); & (3.16) \\ \frac{2e_{n+1}}{s_\xi^2 + e_{n+1}^T} \text{ для } p_0(\xi) = C(0, s_\xi); & (3.17) \end{cases}$$

$$\gamma = \begin{cases} 1 \text{ для } p_0(\xi) = N(0, \sigma_\xi)^2; & (3.18) \\ s_\xi \text{ для } p_0(\xi) = L(0, s_\xi); & (3.19) \\ s_\xi^2 \text{ для } p_0(\xi) = C(0, s_\xi). & (3.20) \end{cases}$$

Як випливає з наведених співвідношень, якщо алгоритм МНК (3.13) – (3.15), (3.18) є інваріантним до параметру масштабу нормального розподілу σ_ξ^2 , то оцінки МНМ (3.13), (3.14), (3.16), (3.19) і оцінка, відповідна розподілу Коші залежать від параметра S_ξ . При невідомій величині S_ξ в цих алгоритмах слід використовувати оцінку \hat{S} , отриману, наприклад, за допомогою алгоритму стохастичної апроксимації.

Для ε -засмічених імовірнісних розподілів оптимальні алгоритми навчання представляють собою комбіновані процедури виду

$$\theta_{n+1} - \theta_n = \begin{cases} -H_{1,n+1} F' [e_{n+1}] \nabla f_{n+1} & \text{при } |e_{n+1}| \leq \Delta; \\ -H_{2,n+1} \nabla f_{n+1} \operatorname{sign} e_{n+1} & \text{при } |e_{n+1}| > \Delta, \end{cases} \quad (3.21)$$

де $H_{1,n+1}$, $H_{2,n+1}$ – матриці посилення.

3.3.2 Однокрокові робастні процедури навчання

У робастній ідентифікації та фільтрації досить широкого поширення набули релейні алгоритми [100] або алгоритми, що використовують нелінійне перетворення помилки ідентифікації [101]. Основу даних процедур, які можуть бути використані для навчання лінійної АДАЛПНИ, становить наступна процедура

$$\theta_{n+1} = \theta_n + \gamma_{n+1} x_{n+1} \operatorname{sign} e_{n+1}. \quad (3.22)$$

Очевидно, що з метою прискорення швидкості оцінювання можуть бути запропоновані наступні модифікації (3.22):

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \frac{\text{sign } e_{n+1}}{\|x_{n+1}\|^2} x_{n+1}, \quad (3.23)$$

та алгоритм Гудвіна-Ремеджа-Кейнеса [102]

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \frac{\text{sign } e_{n+1}}{r_{n+1}} x_{n+1}. \quad (3.24)$$

$$\text{де } r_{n+1} = r_n + \|x_{n+1}\|^2;$$

$r_0 = 1$, який поєднує властивості нормалізованого алгоритму (3.23) и алгоритму стохастичної апроксимації.

Дані процедури мінімізують модульний критерій. При використанні комбінованого критерію (3.11) градієнтна процедура мінімізації має вигляд

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \left[\lambda 2e_{n+1} + (1-\lambda) \text{sign } e_{n+1} \right] x_{n+1}. \quad (3.25)$$

Дана процедура поєднує властивості МНК з властивостями МНМ, тому що при $\lambda = 1$ з (3.25) маємо алгоритм МНК, а при $\lambda = 0$ – алгоритм МНМ, і дозволяє боротися з імпульсними перешкодами. Варіюючи параметр λ , можна змінювати властивості алгоритму.

3.3.3 Робастна процедура навчання, яка мінімізує комбінований функціонал

Запишемо (3.25) щодо помилок навчання $\tilde{\theta}(i)$

$$\tilde{\theta}_{n+1} = \tilde{\theta}_n + \gamma_{n+1} \left[\lambda 2e_{n+1} + (1-\lambda) \text{sign } e_{n+1} \right] x_{n+1}. \quad (3.26)$$

Дослідження питань збіжності процедури (3.25) наведено в додатку В.

Як витікає з результатів аналізу збіжності за відсутністю завад, умовою збіжності є те, що параметр γ повинен задовольняти нерівності (В.4)

Оптимальне ж значення параметра γ визначаємо з рівняння, одержуваного шляхом диференціювання (В.1) по γ і прирівнювання похідної нулю. Таким чином

$$\gamma^{opt} = \frac{e_{n+1}}{\left(\lambda e_{n+1} + (1-\lambda) \text{sign } e_{n+1} \right) \|x_{n+1}\|^2}. \quad (3.27)$$

При дослідженні статистичних властивостей процедури навчання (3.25) при наявності завад виміру, тобто $y_{n+1}^* = \theta^{*T} x_{n+1} + \xi_{n+1}$, $\xi_{n+1} \square N\left(0, \sigma_\xi^2\right)$, з отриманого виразу (В.8) випливає, що процедура (3.25) буде збігатися в середньому, якщо параметр γ задовольняє нерівності (В.9)

$$0 < \gamma < \frac{2}{(2\lambda + (1-\lambda)\beta) \text{tr} R_{xx}}$$

або (В.10)

$$0 < \gamma < \frac{2}{(2\lambda + (1-\lambda)\beta) N \sigma_x^2}.$$

Тут $\beta = \sqrt{\frac{1}{\pi \sigma_e^2}}$, а $\text{tr} R_{xx}$ - слід матриці R_{xx} .

Дослідження питань збіжності в середньоквадратичному показують, що у відповідності з (В.17), що процедура (3.25) буде збігатися в середньоквадратичному при виконанні умови

$$\left| 1 - 4\gamma\lambda\sigma_x^2 + 2\gamma(1-\lambda)\beta\sigma_x^2 + 12\gamma^2\lambda^2\sigma_x^4 + 12\gamma^2(1-\lambda)\sigma_x^4 \right| < 1,$$

звідки випливають обмеження для параметра γ

$$0 < \gamma < \frac{2\lambda + (1-\lambda)\beta}{6\lambda(\lambda + (1-\lambda)\beta)\sigma_x^2}. \quad (3.28)$$

Наявність завади ξ_{n+1} призводить до того, що процедура (3.25) буде збігатися до області, яка визначається цією перешкодою, і рівної

$$\lim_{n \rightarrow \infty} M \left\{ \|\theta_n\|^2 \right\} = \gamma^2 (1-\lambda)^2 N \sigma_\xi^2 \sigma_x^2 \quad (3.29)$$

З отриманого виразу видно, що для забезпечення $\lim_{n \rightarrow \infty} M \left\{ \|\tilde{\theta}_n\|^2 \right\} = 0$ параметр γ повинен обиратися змінним і з ростом n прагнути до нуля, тобто задовольняти умовам Дворецького [103].

3.4 Робастне навчання на основі мінімізації модифікованого комбінованого функціоналу

Для забезпечення робастних властивостей одержуваних оцінок досить ефективним є застосування модифікованого комбінованого функціоналу навчання

$$F[e(k)] = \frac{1}{4} \lambda e^4(k) + (1-\lambda)|e(k)|, \quad (3.30)$$

$$\text{де } e_{n+1} = y_{n+1} - \hat{y}_{n+1} + \xi_{n+1} = y_n - \hat{\theta}_n^T x_{n+1} + \xi_{n+1},$$

\hat{y}_{n+1} – вихідний сигнал моделі;

$\hat{\theta}_n = (\hat{\theta}_{1,n}, \hat{\theta}_{2,n}, \dots, \hat{\theta}_{N,n})^T$ – вектор оцінок параметрів $N \times 1$; $\lambda \in [0, 1]$ –

параметр змішування.

При використанні комбінованого критерію (3.30) градієнтна процедура мінімізації має вигляд

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \left[\lambda e_{n+1}^3 + (1-\lambda) \text{sign} e_{n+1} \right] x_{n+1}, \quad (3.31)$$

де γ – деякий параметр, що впливає на швидкість збіжності алгоритму.

Дана процедура поєднує властивості МНК з властивостями МНМ, тому що при $\lambda=1$ з (3.31) маємо алгоритм МНК, а при $\lambda=0$ – алгоритм МНМ, і дозволяє боротися з негаусівськими завадами. Варіюючи параметр λ , можна змінювати властивості алгоритму.

Введемо в розгляд помилку оцінювання:

$$\tilde{\theta}_{n+1} = \theta_{n+1} - \hat{\theta}_{n+1}, \quad (3.32)$$

що дозволяє записати вираз для e_{n+1} в такий спосіб:

$$e_{n+1} = \tilde{\theta}_n^T x_{n+1} + \xi_{n+1} = e_{an+1} + \xi_{n+1}, \quad (3.33)$$

де $e_{an+1} = \tilde{\theta}_n^T x_{n+1}$ – апіорна помилка ідентифікації.

У зв'язку з тим, що передбачається $\xi_{n+1} \sim N(0, \sigma_\xi^2)$, маємо

$$M \left\{ e_{n+1}^2 \right\} = \sigma_\xi^2 + \sigma_x^2 M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}, \quad (3.34)$$

де $M \{ \bullet \}$ – символ математичного сподівання; $\| \bullet \|$ – евклідова норма.

Розглянемо питання збіжності процедури (3.31) за відсутністю завад. При дослідженні питань збіжності скористаємося підходом, який застосовувався в [1-2,6,18]. З цією метою введемо функцію Ляпунова $\left\| \tilde{\theta}_{n+1} \right\|^2$.

Запишемо алгоритм (3.31) щодо помилок ідентифікації $\tilde{\theta}_i$:

$$\tilde{\theta}_{n+1} = \tilde{\theta}_n - \gamma \left[\lambda e_{n+1}^3 + (1-\lambda) \text{sign} e_{n+1} \right] x_{n+1}. \quad (3.35)$$

Помноживши обидві частини (3.35) зліва на $\tilde{\theta}_{n+1}^T$, з урахуванням того, що $e_{n+1} = \tilde{\theta}_n^T x_{n+1}$, маємо

$$\begin{aligned} \|\tilde{\theta}_{n+1}\|^2 &= \|\tilde{\theta}_n\|^2 - 2\gamma\lambda e_{n+1}^4 - 2\gamma(1-\lambda)e_{n+1} \operatorname{sign} e_{n+1} - \\ &- \gamma^2\lambda^2 e_{n+1}^6 \|x_{n+1}\|^2 + 2\gamma^2\lambda(1-\lambda)e_{n+1}^3 \operatorname{sign} e_{n+1} \|x_{n+1}\|^2 + \gamma^2(1-\lambda)^2 \|x_{n+1}\|^2. \end{aligned} \quad (3.36)$$

З (3.36) видно, що при $\gamma > 0$ приріст функції Ляпунова

$$\Delta V_{n+1} = \|\tilde{\theta}_{n+1}\|^2 - \|\tilde{\theta}_n\|^2,$$

буде негативним, якщо

$$\left(\lambda e_{n+1}^4 + 2(1-\lambda)|e_{n+1}|\right) > \gamma \left(\lambda e_{n+1}^3 + (1-\lambda)|e_{n+1}|\right)^2 \|x_{n+1}\|^2. \quad (3.37)$$

Таким чином, умова збіжності процедури (3.31) буде виконуватися, якщо параметр γ задовольняє нерівності

$$0 < \gamma < \frac{2e_{n+1}}{\left(\lambda e_{n+1}^3 + (1-\lambda)\operatorname{sign} e_{n+1}\right) \|x_{n+1}\|^2}. \quad (3.38)$$

Оптимальне значення параметра γ визначаємо з рівняння, одержуваного шляхом диференціювання (3.38) по γ і прирівнювання похідної нулю. Таким чином,

$$\gamma^{opt} = \frac{e_{n+1}}{\left(\lambda e_{n+1}^3 + (1-\lambda)\operatorname{sign} e_{n+1}\right) \|x_{n+1}\|^2}. \quad (3.39)$$

Дослідження статистичних властивостей процедури навчання (3.31) при наявності завад вимірів, тобто $y_{n+1}^* = \theta^{*T} x_{n+1} + \xi_{n+1}$, $\xi_{n+1} \sim N(0, \sigma_\xi^2)$, проведено у Додатку Г.

Отримані там співвідношення дозволяють визначити оптимальне значення параметра γ , що забезпечує максимальну швидкість збіжності алгоритму. Це значення отримується шляхом вирішення рівняння

$$\partial M \left\{ \|\tilde{\theta}_{n+1}\|^2 \right\} / \partial \gamma = 0, \quad (3.40)$$

матиме такий вигляд:

$$\gamma^{opt} = \frac{3\lambda\sigma_\xi^2 + (1-\lambda)\beta}{15(N+2)\lambda^2 M\{\xi_{n+1}^4\}}. \quad (3.41)$$

Зі співвідношення (Г.10) можна отримати вираз для асимптотичної помилки оцінювання

$$M \left\{ \|\tilde{\theta}(\infty)\|^2 \right\} = \frac{\gamma N \left[\lambda^2 M\{\xi^6\} + (1-\lambda)^2 \right]}{15\gamma(N+2)M\{\xi^4\} - 6\lambda\sigma_\xi^2 - 2(1-\lambda)\beta}, \quad (3.42)$$

тобто для забезпечення $\lim_{n \rightarrow \infty} M \left\{ \|\tilde{\theta}_n\|^2 \right\} = 0$ параметр γ повинен обиратися змінним і з ростом n прагнути до нуля, тобто задовольняти умовам Дворецького [103].

Підстановка (3.42) в (3.34) дає вираз для асимптотичної помилки ідентифікації

$$M \left\{ e^2(\infty) \right\} = \sigma_\xi^2 + \frac{\sigma_x^2 \gamma N \left[\lambda^2 M\{\xi^6\} + (1-\lambda)^2 \right]}{15\gamma(N+2)M\{\xi^4\} - 6\lambda\sigma_\xi^2 - 2(1-\lambda)\beta}. \quad (3.43)$$

3.5 Багатокрокові робастні процедури навчання

Розглянуті вище алгоритми навчання, які базуються на МНК, будуть працездатними після $n \geq N + 1$ тактів. Якщо ж $n < N + 1$, навчання мережі може здійснюватися за допомогою проєкційних алгоритмів[104]. Слід також зазначити, що застосування цих алгоритмів доцільно і в разі, коли досліджуваний об'єкт є нестационарним, і параметри його змінюються з плином часу.

l -кроковий проєкційний алгоритм навчання має вигляд

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \nabla f_{l,n+1} [\nabla f_{l,n+1}^T \nabla f_{l,n+1}]^{-1} F'_{l,n+1}(e_{n+1}), \quad (3.44)$$

де $F'_{l,n+1}(e_{n+1}) = (F'_{l,n}(e_n), 0, 0, \dots, 0)^T$;

$\nabla f_{l,n+1} = (\nabla f_{n+1}, \nabla f_n, \dots, \nabla f_{n-l+1})$ – матриці $(N+1) \times l$;

γ_{n+1} – параметр, який задовольняє умовам Дворецького.

Більш зручна в обчислювальному відношенні процедура, що використовує рекурентне обчислення матриці $[\nabla f_{l,n+1}^T \nabla f_{l,n+1}]^{-1}$ замість її безпосереднього обертання може бути отримана з урахуванням правил формування матриць $\nabla f_{l,n} = (\nabla f_{l-1,n} \vdots \nabla f_{n-l}(k-l))$, $\nabla f_{l,n+1} = (\nabla f_{n+1} \vdots \nabla f_{l-1,n})$.

В цьому випадку можна записати

$$\theta_{n+1} = \theta_n + \gamma_{n+1} \frac{R_{l-1,n}}{\nabla f_{n+1}^T R_{l-1,n} \nabla f_{n+1}} F'_{l,n+1}(e_{n+1}), \quad (3.45)$$

де $R_{l-1,n} = I - \nabla f_{l-1,n} [f_{l-1,n}^T \nabla f_{l-1,n}]^{-1} \nabla f_{l-1,n}^T = I - \nabla f_{l-1,n}^+ \nabla f_{l-1,n}^T$. Тут $\nabla f_{l-1,n}^+$ – матриця, псевдообернена до $\nabla f_{l-1,n}$.

Матриця $R_{l,n+1}$ також обчислюється рекурентно. Якщо отримання співвідношень для обчислення цієї матриці при надходженні нової інформації є досить тривіальним, то для реалізації процедури скидання слід скористатися теоремою Гревилля і наступними співвідношеннями між матрицями $\nabla f_{l+1,n+1}^+$ та $\nabla f_{l,n+1}^+$:

$$(\nabla f_{l,n+1}^+) \div 0 = \nabla f_{l+1,n+1}^+ (I - \mathbf{g} \mathbf{K}_{n+1}^T),$$

де для випадку $l < N$

$$\mathbf{K}_{n+1} = \frac{(I - \nabla f_{l+1,n+1} \nabla f_{l+1,n+1}^+) \mathbf{g}}{\mathbf{g}^T (I - \nabla f_{l+1,n+1} \nabla f_{l+1,n+1}^+) \mathbf{g}},$$

де $\mathbf{g} = (0, 0, \dots, 1)^T$ – вектор $l \times 1$.

Проводячи нескладні обчислення, отримаємо, що при надходженні нової інформації на n -му такті матриця $R_{l,n+1}$ розраховується за формулою

$$R_{l,n+1} = R_{l-1,n} + \frac{R_{l-1,n} \nabla f_{n+1} \nabla f_{n+1}^T R_{l-1,n}}{\nabla f_{n+1}^T R_{l-1,n} \nabla f_{n+1}}, \quad (3.46)$$

а при скиданні застарілої інформації про $(n-l-1)$ -й такт – за формулою

$$R_{l,n+1} = R_{l+1,n+1} + \frac{\nabla \hat{f}_{n-l+1} \nabla \hat{f}_{n-l+1}^T}{\|\nabla \hat{f}_{n-l+1}\|^2}, \quad (3.47)$$

де $\nabla \hat{f}_{n-l+1} = U_{l+1,n+1} \nabla f_{n-l+1}$; $R_0 = I$;

$$U_{l+1,n+1} = (\nabla f_{l+1,n+1}^T \nabla f_{l+1,n+1})^+.$$

Матриця $U_{l,n}$, яка входить в (3.46), (3.47), в свою чергу при надходженні нової інформації та сбросі застарілої обчислюється відповідно за формулами:

$$U_{l+1,n+1} = U_{l,n} - \frac{(U_{l,n} + R_{l,n}) \nabla f_{n+1}(k) \nabla f_{n+1}^T U_{l,n}}{\nabla f_{n+1}^T U_{l,n} \nabla f_{n+1}(k)} - \quad (3.48)$$

$$+ \frac{R_{l,n}(k) \nabla f_{n+1} \nabla f_{n+1}^T R_{l,n} R_{l,n} \nabla f_{n+1} \nabla f_{n+1}^T U_{l,n} \nabla f_{n+1} \nabla f_{n+1}^T R_{l,n}}{\nabla f_{n+1}^T U_{l,n} \nabla f_{n+1} \left\| R_{l,n} \nabla f_{n+1}^T \right\|^2}$$

$$U_{l,n+1} = U_{l+1,n+1} - \frac{U_{l+1,n+1} \nabla \hat{f}_{n-l+1} \nabla \hat{f}_{n-l+1}^T + \nabla \hat{f}_{n-l+1} \nabla \hat{f}_{n-l+1}^T U_{l+1,n+1}}{\nabla \hat{f}_{n-l+1}^2} + \quad (3.49)$$

$$+ \frac{\nabla \hat{f}_{n-l+1}^T U_{l+1,n+1} \nabla \hat{f}_{n-l+1}}{\left\| \nabla \hat{f}_{n-l+1} \right\|^4} \nabla \hat{f}_{n-l+1} \nabla \hat{f}_{n-l+1}^T$$

Таким чином, процедури (3.45) - (3.47) відповідають накопиченню, а (3.45), (3.48), (3.49) – скиданню застарілої інформації.

3.6 Висновки за третім розділом

1. Найбільш широко використовуваний на практиці квадратичний функціонал призводить до різних алгоритмів навчання, базованих на МНК-рішенні, яке за цих умов є асимптотично оптимальним з мінімальною дисперсією в класі незміщених оцінок. Однак припущення гаусовості, як правило, не є вірним в реальних умовах, тому що майже завжди апріорна інформація про розподіл зазвичай недоступна або завада є засміченою

негаусівським шумом, через що деякі вимірювання видалені на відносно велику відстань від основного обсягу даних і утворюють так звані “хвости”. В цих умовах оцінка МНК є нестійкою.

2. Якщо інформація про приналежність завади ξ деякому певному класу розподілів є відомою, то шляхом мінімізації оптимального критерію, що представляє собою узятий з оберненим знаком логарифм функції розподілу завади, може бути отримана оцінка максимальної правдоподібності (М-оцінка). Якщо ж такої інформації немає, то для оцінювання шуканого вектору параметрів θ слід застосувати будь-який неквадратичний критерій, що забезпечує робастність одержуваної оцінки. Одним з таких критеріїв є модульний критерій, що приводить до знакового алгоритму. Позитивні властивості модульного критерію використовуються в так званих комбінованих критеріях, найбільш поширеними серед яких є комбіновані функціонали, запропоновані Хьюбером і Хемпелем. Вони складаються з квадратичного функціоналу, що забезпечує оптимальність оцінок для гаусівського розподілу, і модульного, що дозволяє отримати більш робастну до розподілів з важкими «хвостами» (викидами) оцінку. Однак ефективність одержуваних робастних оцінок істотно залежить від численних параметрів, що використовуються в даних критеріях, і які обираються на основі досвіду дослідника.

Іншим підходом до отримання робастних оцінок, позбавлених зазначеного недоліку, є використання комбінованого критерію, заснованого на поєднанні квадратичного і модульного критеріїв більш простим способом, ніж запропонованих Хьюбером і Хемпелем.

3. Розглянуто особливості робастного навчання на основі мінімізації комбінованих функціоналів, які поєднують квадратичний і модульний критерій, а також запропоновано використання комбінованого функціоналу, що складається з критерію четвертого ступеня та модульного. Визначено умови збіжності градієнтного алгоритму навчання в середньому і середньоквадратичному. Отримано аналітичні оцінки неасимптотичних та

асимптотичних значень помилки оцінювання параметрів і точності ідентифікації моделі. Показано, що ці значення помилки оцінювання і точності ідентифікації залежать від вибору параметра змішування. Отримані оцінки є досить загальними і залежать від статистичних характеристик корисних сигналів і завад. Тому для їх практичного застосування слід скористатися оцінками цих параметрів.

Запропонована l -крокова проєкційна процедура навчання, яка має більшу швидкість збіжності, та розроблена її рекурентна форма.

Список використаних джерел у даному розділі наведено у повному списку використаних джерел під номерами: 1-2, 6, 18, 75-104.

4 ОПТИМІЗАЦІЯ ПРОЦЕДУР НАВЧАННЯ

В цьому розділі розглядаються питання оптимізації процедур навчання нейронних мереж, що повинно забезпечити суттєве скорочення часу налаштування ШНМ та поліпшити якість вирішення задачі прогнозування [104].

4.1 Оцінювання параметрів функціоналів

Як випливає з співвідношень (3.8), (3.9,) (3.14)–(3.15), параметри, що входять у вираз для функціоналів, впливають на їх вигляд і залежать від статистичних властивостей завади ξ_n . При цьому необхідно мати на увазі наступне. У разі нормального розподілу, оптимальні оцінки, одержувані за допомогою методу найменших квадратів, інваріантні до параметру масштабу σ_ξ^2 . При М-оцінюванні, внаслідок неоднорідності функції ρ , одержувані оцінки не зберігають властивості інваріантності. Для того щоб властивість інваріантності масштабу виконувалося, в функціоналі (3.3) замість помилки $e(i, \theta)$ слід брати $\tilde{e}(i, \theta) = (e(i, \theta) - m)/S$, де S – завадостійка оцінка параметра масштабу або міра розсіювання залишкових різниць (в разі нормального розподілу S є оцінкою σ_ξ); m - математичне сподівання засмічуючої завади $q(x)$ в моделі (3.1), яке в загальному випадку відмінно від нуля.

При нульовому математичному сподіванні перешкоди оцінка параметра масштабу використовуються в якості константи, що входить в розглянуті вище функціонали і функції впливу.

4.1.1 Оцінювання параметра масштабу в режимі off-line

Якщо навчання мережі відбувається в режимі off-line, то в якості оцінки слід взяти будь-яку MAD-оцінку (Median of Absolute Deviations, медіана

абсолютних відхилень) [105,106], яка, як показано в [106], є найбільш завадостійкою.

У ранніх роботах по робастному оцінюванню в якості стійких оцінок параметра масштабу розглядалися медіанний.

Так в [106] відзначається, що в якості оцінки S можна використовувати медіану модуля

$$S = med \{|e|\} \quad (4.1)$$

Більш точна, MAD-оцінка

$$S = \frac{med \{|e|\}}{0.6745} \quad (4.2)$$

розглядалася в роботах [107,108]. Крім того, в [108] доведено, що MAD-оцінка є найбільш перешкодостійкою оцінкою дисперсії.

В [109] використовувалася медіанна оцінка виду

$$S = \frac{med \{|e - med(e)|\}}{0.6745} \quad (4.3)$$

Децю інший підхід оцінювання S вивчався в [110]. Тут розглядалося запропонований Хьюбером підхід отримання оцінки параметра S шляхом рішення щодо S рівняння

$$\frac{1}{n - 2p - 1} \sum_{i=p+1}^n \psi^2 \left[\frac{e_i}{S_i} \right] = \beta, \quad (4.4)$$

де константа β обирається так, що S є спроможною оцінкою σ_ε в разі $\xi \in N(0, \sigma_\varepsilon^2)$, тобто $\beta = E_\Phi \{\psi^2(e)\}$, а Φ – стандартний нормальний розподіл:

$$\beta = \frac{1}{\sqrt{2\pi}} \int [\Phi(x)]^2 e^{-\frac{x^2}{2}} dx \quad (4.5)$$

Значення константи β для різних функцій впливу $\psi'(e)$ наведені в [110].

4.1.2 Оцінювання параметрів завод в режимі on-line

При вирішенні завдань в режимі on-line природним є рекуррентне оцінювання параметрів перешкоди.

Завадостійке оцінювання параметра положення на основі алгоритму стохастичної апроксимації Роббінса-Монро для випадку, коли безліч розподілів перешкоди належить класу асиметричних ε -засмічених нормальних розподілів (3.1), розглядалося в [110], а для розподілів довільного виду - в роботі [109]. Такий підхід є досить привабливим в обчислювальному аспекті, тому що оцінки визначаються за допомогою простих рекуррентних формул і не використовують в явному вигляді даних минулих спостережень.

4.2 Вибір оптимальних параметрів процедур навчання

4.2.1 Вибір параметра регуляризації δ

Метою використання параметра регуляризації δ є підвищення обчислювальної стійкості алгоритму оцінки. Однак при цьому виникає проблема ефективного вибору цього параметра. Принципово в літературі це питання майже не досліджене. Ймовірно, це пов'язано з тим, що для її вирішення потрібен достатньо великий обсяг інформації про статистичні властивості корисних сигналів та завод, що впливають на властивості алгоритму. Деякі рекомендації даються в ряді робіт.

Розглянемо це питання більш детально.

Як впливає з розгляду однокрокових процедур, підвищення їх обчислювальної стійкості пов'язано з введенням параметра регуляризації δ .

Очевидно, що даний параметр впливає на властивості процедур навчання, так як при такому підході замість звичайного квадратичного функціоналу використовується регуляризоване, тобто вирішується задача

$$\min_{\theta_n} \left\{ \left(y_{n+1} - \theta_n^T x_{n+1} \right)^2 + \delta \|\theta_n\|^2 \right\} \quad (4.6)$$

Як і для будь-якого штучно вводиться в критерій або оцінку параметра, для параметра регуляризації δ не існує загальних рекомендацій по його вибору. Неважко бачити, що з метою збільшення швидкості збіжності даний параметр повинен прагнути до нуля, а з метою підвищення стійкості оцінки повинен бути відмінним від нуля.

Вибір оптимального значення δ залежить від наявності апріорної інформації щодо властивостей досліджуваного об'єкта, статистичних властивостей корисних сигналів і завад і т.д. Як правило, на практиці така інформація відсутня, тому при практичній реалізації регуляризоване алгоритмів застосовують різного роду припущення і апроксимації. Так, наприклад, в [110] пропонується апроксимувати регуляризоване квадратичний функціонал $E(\delta)$ параболою в околиці точки $\delta = 0$ і скористатися методом Ньютона–Рафсона. Отримане в цій роботі значення δ_{opt} визначається виразом

$$\delta^{opt} = \frac{\sigma_{\xi}^2 \sum_{i=1}^N \frac{1}{\mu_i^2}}{\sum_{i=1}^N \frac{3\sigma_{\xi}^2 + \gamma_i^2 \mu_i}{\mu_i^2}} \quad (4.7)$$

де μ_i і γ_i відповідно характеристичні числа матриць $X^T X$

P – ортогональної матриці, стовпці якої складені з характеристичних чисел матриці $X^T X$.

Однак, як зазначається в даній роботі, скористатися виразом для δ_{opt} на практиці не представляється можливим, тому що це вираз залежить від невідомих параметрів γ_i ($i=1 \div N$) і μ_i ($i=1 \div N$) та часто невідомою σ_ξ^2 .

У зв'язку з цим пропонується досить складна ітеративна процедура оцінювання δ , незастосовні при on-line навчанні.

Відзначимо, що досить зручною on-line процедурою налаштування параметра δ для нелінійної АДАЛІНИ є наступна, що отримується шляхом мінімізації функціонала і записується у вигляді

$$\delta_{n+1} = \delta_n + \alpha \nabla_{\beta} E(k) |_{\delta=\delta_n}, \quad (4.8)$$

де α – деякий параметр, що впливає на швидкість збіжності;

$$\nabla_{\delta} E_{n+1} = \frac{\partial E_{n+1}}{\partial \delta_{n+1}} = e_{n+1} \frac{\partial e_{n+1}}{\partial \delta_{n+1}}. \quad (4.9)$$

У зв'язку з тим, що

$$\frac{\partial e_{n+1}}{\partial \delta_{n+1}} = - \frac{\partial w_{n+1}}{\partial \delta_{n+1}} \nabla f_{n+1} x_{n+1},$$

а

$$\frac{\partial w_{n+1}}{\partial \delta_{n+1}} = - \frac{e_{n+1} \nabla f_{n+1} x_{n+1}}{\left[\left\| \nabla f_{n+1} x_{n+1} \right\|^2 + \delta_{n+1} \right]^2},$$

то

$$\frac{\partial E_{n+1}}{\partial \delta_{n+1}} = - \frac{e_{n+1} e_n \nabla f(x_{n+1}) \nabla f(x_n) x_{n+1}^T x_n}{\left[\left\| \nabla f(x_{n+1}) x_{n+1} \right\|^2 + \delta_{n+1} \right]^2}.$$

Таким чином, остаточно рекурентна процедура налаштування параметра набуває вигляду

$$\delta_{n+1} = \delta_n - \alpha \frac{e_{n+1} e_n \nabla f_{n+1} \nabla f_n x_{n+1}^T x_n}{\left[\|\nabla f_{n+1} x_n\|^2 + \delta_n \right]^2}. \quad (4.10)$$

З огляду на (3.32) і (3.38), можна записати

$$0 < \frac{1}{\|\nabla f_{n+1} x_{n+1}\|^2 + \delta_{n+1}} < \frac{2}{\|\nabla f_{n+1} x_{n+1}\|^2}, \quad (4.11)$$

звідки отримуємо обмеження для параметра регуляризації

$$-0.5 \|\nabla f_{n+1} x_{n+1}\|^2 < \delta_{n+1}.$$

Для підвищення стійкості оцінок можна застосувати більш грубі значення параметра регуляризації, одержувані, наприклад, шляхом побудови так званих гребньових слідів [111], що представляють собою графіки залежностей суми квадратів відхилень, середньої суми квадратів помилок, координат вектора гребньовій оцінки і т.д. від параметра регуляризації. Зіставлення графіків гребньових слідів для різних показників допомагає встановити в яких межах повинен лежати параметр регуляризації δ і до яких наслідків призводить той чи інший вибір δ .

При використанні регуляризоване оцінок слід очікувати наступне. При виборі досить малих значень δ значення суми квадратів відхилень збільшується незначно в порівнянні з сумою квадратів відхилень МНК – оцінок, в той час як самі регуляризоване оцінки будуть відрізнятися від МНК – оцінок більш істотно. Тому вибір параметра регуляризації може здійснюватися також, виходячи із задачі максимально допустимого значення суми квадратів відхилень. Якщо гребневий слід дає якісну інформацію про

поведінку найважливіших характеристик оцінок в залежності від зміни δ , то сума квадратів відхилень має гарну інтерпретацією.

Однак практично всім відомим класичним процедурам вибору параметра регуляризації притаманний спільний недолік - вони не враховують статистичних властивостей одержуваних оцінок. Тому на практиці слід, швидше за все, виходити з конкретної ситуації і задавати параметр δ досить малим, але в той же час дозволяє отримувати оцінки в умовах мультиколінеарності.

Спроба непрямого обліку цих властивостей була зроблена в роботі [110], в якій в якості критерію запропоновано розглядати відношення сигнал-шум (ENR – echo-to-noise ratio), тобто нормалізовану величину

$$ENR = \frac{\sigma_y^2}{\sigma_\xi^2} = \frac{\theta^T R_x \theta}{\sigma_\xi^2}, \quad (4.12)$$

де $\sigma_y^2 = M \{y_n^2\}$; $\sigma_\xi^2 = M \{\xi_n^2\}$; $R_x = M \{x_n x_n^T\}$.

Мінімізація $M \{\tilde{e}_{n+1}^2\} = \sigma_\xi^2$, де $\tilde{e}_{n+1} = y_{n+1}^* - w_n^T x_{n+1}$, в припущенні, що

$$\|x_n\|^2 \approx N \sigma_x^2 \text{ і}$$

$$M \{|x_n|\} = \sqrt{\frac{2}{\pi \sigma_x^2}} \sigma_x^2 = \alpha_x \sigma_x^2,$$

де $\alpha_x = \sqrt{\frac{2}{\pi \sigma_x^2}}$, дозволяє отримати наступні вирази для нормалізованого параметра регуляризації

– для процедури (3.10)

$$\delta = \frac{N(1 + \sqrt{1 + ENR})}{ENR} \sigma_x^2; \quad (4.13)$$

– для процедури (3.19)

$$\delta = \frac{N\alpha_x(1 + \sqrt{1 + ENR})}{ENR} \sigma_x^2; \quad (4.14)$$

– для процедури (3.16)

$$\delta = \frac{1 + \sqrt{1 + ENR}}{ENR} \sigma_x^2; \quad (4.15)$$

– для процедури (3.23)

$$\delta = \frac{1 + \sqrt{1 + ENR}}{ENR}. \quad (4.16)$$

Слід зазначити, що отримані вирази для параметра регуляризації також важко застосовувати, тому що вимагають значення σ_x^2 і σ_ξ^2 , що не представляється можливим. Тому для обчислення даних значень також слід використовувати оцінки σ_x^2 і σ_ξ^2 , отримані за допомогою деяких зворотних процедур (отримання таких оцінок буде розглянуто нижче).

Зазвичай в роботах, що мають практичну спрямованість, рекомендується вибирати δ досить малим позитивним числом і не розглядається питання корекції цього параметра в процесі навчання мережі.

Можна розглянути задачу визначення, що забезпечує мінімальне значення помилки апроксимації, яка в підсумку буде визначатися величиною σ_ξ^2 .

Таким чином, значення параметра залежить від розмірності моделі N , дисперсії вхідного сигналу σ_x^2 і ENR . Якщо значення N відоме, а σ_x^2 можна визначити або оцінити, значення ENR часто не відомо.

У ряді робіт було розглянуто можливість визначення оптимального значення параметра δ з умов максимізації швидкості збіжності алгоритму. Такий підхід дозволив отримати наступний вираз для δ_{n+1}^{opt}

$$\delta_{n+1}^{opt} = \frac{N\sigma_{\xi}^2}{M\left\{\|\tilde{\theta}_n\|^2\right\}}. \quad (4.17)$$

Вираз (4.19) містить, крім невідомої σ_{ξ}^2 , невідоме значення $M\left\{\|\tilde{\theta}_n\|^2\right\}$.

Тому ці рекомендації представляють більше теоретичний інтерес. Їх аналіз дозволяє зробити висновок, що значення δ_{n+1}^{opt} повинно вибиратися змінним. У той же час, коли $ENR \rightarrow 0$ і $M\left\{\|\tilde{\theta}_n\|^2\right\} \rightarrow 0$. При $\delta_{n+1}^{opt} \rightarrow \infty$, усувається вплив завад ξ_n , проте алгоритм не буде відслідковувати нестационарність параметрів.

4.2.2 Визначення параметра швидкості навчання γ

Для визначення параметру γ , який характеризує швидкість навчання, зазвичай обирають критерій у вигляді функції Ляпунова $V_{n+1} = M\{\|\theta_{n+1}\|^2\} - M\{\|\theta_n\|^2\}$.

Оптимальне значення параметра γ , що забезпечує максимальну швидкість збіжності алгоритму, можна отримати, розв'язавши рівняння

$$\frac{\partial V_{n+1}}{\partial \gamma} = 0, \quad (4.18)$$

або

$$\frac{\partial M\{\|\theta_{n+1}\|^2\}}{\partial \gamma} = 0, \quad (4.19)$$

Так для регяризованого алгоритму Качмажа (2.52) слід використовувати співвідношення (2.64), звідки після нескладних перетворень маємо

$$A = (N - 2)\sigma_x^2 + \delta, \quad (4.20)$$

де $A = (N - 2)\sigma_x^2 + \delta$.

Слід зазначити, що отримані співвідношення узагальнюють відомі результати. У нашому випадку для класичного алгоритму Качмажа ($\delta = 0$) з (2.61) впливає відома умова збіжності у середньому

$$0 < \gamma < 2N, \quad (4.21)$$

а з (2.63) умови збіжності в середньоквадратичному. При цьому

- для нестационарного випадку

$$\begin{aligned} \gamma^{opt} &= \frac{((N-2)^3 \sigma_x^6 (M\{\|\theta_n\|^2\} + \sigma_S^2))}{((N-2)\sigma_x^6 M\{\|\theta_n\|^2\} + \sigma_S^2) + N(N-2)^2 \sigma_x^4 \sigma_\xi^2} = \\ &= 1 - \frac{N\sigma_\xi^2}{(N-2)\sigma_x^2 M\{\|\theta_n\|^2\} + \sigma_S^2} \end{aligned} \quad (4.22)$$

та

$$\gamma^{opt} = 1 \quad (4.23)$$

для стаціонарного випадку та відсутності завад.

Підхід до визначення оптимального значення параметра для алгоритму Нагумо-Ноди залишається тим самим. Опускаючи нескладні перетворення, отримаємо такий вираз для γ :

$$\gamma^{opt} = \frac{2(N + \delta\sqrt{\pi\omega})(M\{\|\tilde{\theta}_n\|^2\} + \sigma_S^2)}{\pi N\{[M\{\|\tilde{\theta}_n\|^2\} + \sigma_S^2] + 2\omega\sigma_\xi^2\}}. \quad (4.24)$$

Аналіз цього виразу проводиться аналогічно вищенаведеним аналізу вибору параметра релаксації для алгоритму Качмажа. Так для класичного

алгоритму Нагумо-Ноди для нестационарного випадку ($\sigma_S^2 \neq 0$ і $\sigma_S^2 \neq 0$) з (4.26) впливає

$$\gamma_n^{opt} = \frac{2N[M\{\|\tilde{\theta}_n\|^2\} + \sigma_S^2]}{\pi N[M\{\|\tilde{\theta}_n\|^2\} + \sigma_S^2] + 2\omega\sigma_\xi^2}. \quad (4.25)$$

З цього ж виразу (4.26) впливає, що для стаціонарного випадку $\sigma_S^2 = 0$, відсутності перешкод $\sigma_S^2 = 0$ і $\delta = 0$

$$\gamma^{opt} = \frac{2}{\pi}, \quad (4.26)$$

що збігається з відомим [61]

4.3 Факторизація процедур навчання

Наявність операції віднімання при обчисленні матриці \mathbf{P}_{n+1} (2.27) призводить з зростанням n до чисельної нестійкості алгоритму (2.26) – (2.27). Для усунення розходження методу і поліпшення властивостей матриці \mathbf{P}_{n+1} використовується будь-який із методів факторизації [3,112], тобто ортогональне розкладання або подання в загальному випадку $m \times n$ – матриці \mathbf{P} рангу 1 добутком виду $\mathbf{H}\mathbf{R}\mathbf{K}^T$, де \mathbf{H} і \mathbf{K} – ортогональні матриці розмірностей $m \times m$ та $n \times n$ відповідно, а \mathbf{R} – матриця виду

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & 0 \\ 0 & 0 \end{bmatrix}, \quad (4.27)$$

де \mathbf{R}_{11} – 1×1 матриця рангу 1. Вибором \mathbf{H} і \mathbf{K} можна домогтися, щоб \mathbf{R}_{11} була верхньою або нижньою трикутною. Більш того, цю невідроджену підматрицю \mathbf{R}_{11} можна спростити так, щоб вона стала невідродженою діагональною матрицею.

На підставі того, що позитивно визначена симетрична матриця $P^T P$ допускає спектральне розкладання

$$P^T P = U D U^T, \quad (4.28)$$

де U – ортогональна $n \times n$ – матриця;

D – діагональна матриця з позитивними і незростаючими елементами, був розроблений цілий ряд методів факторизації, що відрізняються способом формування матриць і необхідними обчислювальними витратами. Серед цих методів в першу чергу слід відзначити алгоритм Хаусхолдера, ортогоналізації Грамма-Шмідта, розкладання Холесського (метод квадратного кореня) і обертання Гівенса [112-115].

4.3.1 Факторизація моделі методом квадратного кореня

Найбільшого поширення в теорії управління і теорії ідентифікації отримала факторизація з використанням розкладання Холесського, заснованого на факті існування верхньої трикутної (дійсної) $n \times n$ – матриці S такої, що

$$S S^T = P. \quad (4.29)$$

Це пояснюється, мабуть, тим, що воно допускає рекурентне обчислення, тобто може здійснюватися, як і сам процес ідентифікації, в реальному часі. Крім того, метод квадратного кореня заснований на еквівалентності завдання найменших квадратів $Ax = b$ та системи нормальних рівнянь $(A^T A)x = A^T b$ і вимагає приблизно вдвічі менше операцій, ніж алгоритм Хаусхолдера. При оцінюванні же параметрів в разі послідовного накопичування даних обидва ці методи вимагають приблизно однакового об'єму пам'яті.

В роботі [113] розглянуті дві форми розкладання Холесського - коваріаційна та інформаційна.

Так для рекурентного МНК, що забезпечує мінімум функціоналу

$$\mathbf{V}_n = \mathbf{e}_n^T \Lambda_n \mathbf{e}_n, \quad (4.30)$$

де $\mathbf{e}_n = \mathbf{y}_n^* - \hat{\mathbf{y}}_n$ – похибка оцінювання;

Λ_n - вагова матриця,

$$\Lambda_n = \begin{bmatrix} \lambda^n & 0 & \dots & 0 & 0 \\ 0 & \lambda^{n-1} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}; \quad \Lambda_n = \begin{bmatrix} \lambda \Lambda_{n-1} & 0 \\ 0 & 1 \end{bmatrix};$$

$\lambda \in (0, 1]$ – параметр зважування, коваріаційна форма дискретного методу квадратного кореня має вид

$$\left\{ \begin{array}{l} \hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_{n-1} + \boldsymbol{\gamma}_{n-1} \mathbf{e}_{n-1}; \\ \boldsymbol{\gamma}_{n-1} = \boldsymbol{\alpha}_{n-1} \mathbf{S}_{n-1}^T \mathbf{f}_{n-1}; \\ \mathbf{f}_{n-1} = \mathbf{S}_{n-1}^T \mathbf{H}_n; \\ \mathbf{S}_n = [\mathbf{S}_{n-1} - \mathbf{g}_{n-1} \boldsymbol{\gamma}_{n-1} \mathbf{f}_{n-1}^T] \lambda^{-\frac{1}{2}n-1}; \\ \boldsymbol{\alpha}_{n-1}^{-1} = \mathbf{H}_{n-1}^T \mathbf{H}_{n-1} + \lambda_{n-1}; \\ \mathbf{g}_{n-1} = (1 + \sqrt{\lambda_{n-1} \boldsymbol{\alpha}_{n-1}})^{-1}; \\ \mathbf{S}_0 = \sqrt{\boldsymbol{\alpha}} \mathbf{I}; \hat{\boldsymbol{\theta}}_0 = 0. \end{array} \right. \quad (4.31)$$

Як видно з (4.31), параметр зважування λ_n обирається змінним, але таким, щоб

$$\lim_{n \rightarrow \infty} \lambda_n = 1.$$

Співвідношення, що описують дискретний метод квадратного кореня в інформаційній формі, записуються наступним чином. Так як на n -му такті МНК оцінка має вигляд

$$\mathbf{P}_n^{-1} \hat{\boldsymbol{\theta}}_n = \mathbf{H}_n^T \mathbf{y}_n = \mathbf{f}_n, \quad (4.32)$$

а матриця \mathbf{P}_n^{-1} та вектор \mathbf{f}_n пов'язані з \mathbf{P}_{n-1}^{-1} та \mathbf{f}_{n-1} рекурентними співвідношеннями відповідно

$$\begin{aligned}\mathbf{P}_n^{-1} &= \lambda_n \mathbf{P}_{n-1}^{-1} + \mathbf{H}_{n-1} \mathbf{H}_{n-1}^T; \\ \mathbf{f}_n &= \lambda_n \mathbf{f}_{n-1} + \mathbf{H}_{n-1} \mathbf{y}_n,\end{aligned}\tag{4.33}$$

то з урахуванням виду розкладання (4.29) можна записати

$$\mathbf{S}_n^{-1} \hat{\boldsymbol{\theta}}_n = \mathbf{b}_n.\tag{4.34}$$

Це впливає з ортогональної трансформації матриці ($\mathbf{T}^T \mathbf{T} = \mathbf{I}$)

$$\mathbf{H}^T \mathbf{T}^T \mathbf{T} \mathbf{H} \hat{\mathbf{Q}} = \mathbf{H}^T \mathbf{T}^T \mathbf{T} \mathbf{y},\tag{4.35}$$

звідки матриця

$$\mathbf{T} \mathbf{H} = \begin{bmatrix} \mathbf{S}^{-1} \\ \dots \\ 0 \end{bmatrix},$$

є верхньою трикутною.

Позначимо

$$\mathbf{T} \mathbf{y} = \begin{bmatrix} \mathbf{b} \\ \dots \\ W \end{bmatrix}.$$

Тоді з (4.35) для n – го такту отримуємо

$$\mathbf{T}_n \mathbf{H}_n \hat{\boldsymbol{\theta}}_n = \mathbf{T}_n \mathbf{y}_n,$$

звідки маємо рекурентне співвідношення

$$\begin{bmatrix} \mathbf{S}_n^{-1} \\ \mathbf{0}^T \end{bmatrix} = \mathbf{T}_n \begin{bmatrix} \sqrt{\lambda} \mathbf{S}_{n-1}^{-1} \\ \mathbf{H}_n^T \end{bmatrix}; \quad (4.36)$$

$$\begin{bmatrix} \mathbf{b}_n \\ W_n \end{bmatrix} = \mathbf{T}_n \begin{bmatrix} \sqrt{\lambda} \mathbf{b}_{n-1} \\ \mathbf{y}_n \end{bmatrix}, \quad (4.37)$$

яке служить для визначення \mathbf{S}_n^{-1} та \mathbf{b}_n . \mathbf{S}_n^{-1} та \mathbf{b}_n , які отримуються в результаті рішення (4.3) та (4.37) використовуються для знаходження $\hat{\boldsymbol{\theta}}_n$ у відповідності з (4.34).

Крім розглянутого розкладання Холесського досить широко використовується UD – факторизація, що використовує представлення виду (4.28) і яка розроблена в [114]. У цьому розкладанні U – верхня трикутна матриця з одиничними елементами на діагоналі, а D – діагональна матриця

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & d_n \end{bmatrix}. \text{ З огляду на співвідношення між } \mathbf{P}_n \text{ і } \mathbf{P}_{n-1}, \text{ для даного}$$

розкладання можна записати

$$\mathbf{U}_n \mathbf{D}_n \mathbf{U}_n = \frac{1}{\lambda} [\mathbf{U}_{n-1} \mathbf{D}_{n-1} \mathbf{U}_{n-1} - \gamma_{n-1} \mathbf{H}_n \mathbf{U}_{n-1} \mathbf{D}_{n-1} \mathbf{U}_{n-1}^T], \quad (4.38)$$

або

$$\mathbf{U}_n \mathbf{D}_n \mathbf{U}_n = \frac{1}{\lambda} \mathbf{U}_{n-1} \left[\mathbf{D}_{n-1} - \frac{1}{\boldsymbol{\alpha}_{n-1}} \mathbf{V}_{n-1} \mathbf{f}_{n-1}^T \mathbf{D}_{n-1} \right] \mathbf{U}_{n-1}^T,$$

де

$$\begin{aligned} \mathbf{f}_{n-1} &= \mathbf{U}_{n-1}^T \mathbf{H}_n; \\ \mathbf{V}_{n-1} &= \mathbf{D}_{n-1} \mathbf{f}_{n-1}; \\ \boldsymbol{\alpha}_{n-1} &= \lambda + \mathbf{f}_{n-1}^T \mathbf{V}_{n-1}. \end{aligned}$$

При цьому параметр γ_{n-1} , що входить в (4.40), обчислюється так

$$\gamma_{n-1} = \frac{1}{\alpha_{n-1}} \mathbf{U}_{n-1} \mathbf{V}_{n-1}.$$

В [114] були запропоновані наступні рекурентні співвідношення для обчислення елементів матриць \mathbf{U} , \mathbf{D} та γ :

$$\begin{aligned} \alpha_j &= \alpha_{j-1} + \mathbf{V}_j \mathbf{f}_j; \\ \alpha_j(k) &= \alpha_j(k-1) \alpha_{j-1} / (\alpha_j - \lambda); \\ \mathbf{b}_j &= \mathbf{V}_j; \\ \mathbf{V}_j &= -\mathbf{f}_j / \alpha_{j-1}; \quad j = 2, \dots, 2n \end{aligned} \tag{4.39}$$

з початковими умовами

$$\begin{aligned} \alpha_1 &= \lambda + \mathbf{V}_1 \mathbf{f}_1; \\ \alpha_1(k) &= \alpha_1(k-1) / (\alpha_1 - \lambda); \\ \mathbf{b}_1 &= \mathbf{V}_1. \end{aligned}$$

На кожному такті j елементи матриці \mathbf{U} обчислюються за формулами

$$\begin{aligned} u_{ij,n} &= u_{ij,n-1} + r_i b_i; \\ b_i &:= b_i + u_{ij} V_j, \quad i = 1, \dots, j-1. \end{aligned} \tag{4.40}$$

Сама же процедура обчислення факторизованої оцінки приймає вид

$$\hat{\theta}_{n+1} = \hat{\theta}_n + \gamma_n \mathbf{e}_{n+1},$$

де $\mathbf{e}_{n+1} = \mathbf{y}_{n+1} + \mathbf{H}_{n+1}^T \hat{\theta}_n$;

$\gamma_n = (\alpha(2n))^{-1} \mathbf{b}$ (обчислення $\alpha(2m)$ здійснюється з використанням співвідношень (4.39), (4.40)).

Всі розглянуті методи факторизації призводять до чисельно стійких алгоритмів навчання, так як в використовуваних співвідношеннях вже

відсутня трансформація Гауса, що є причиною нестійкості, а обумовленість матриці P знижується вдвічі.

4.3.2 Факторизація за допомогою алгоритма Хаусхолдера

Як зазначається в [115], хоча метод Холесского вимагає вдвічі менше операцій, ніж алгоритм Хаусхолдера, для отримання ним того ж якості рішення, що і в алгоритмі Хаусхолдера при відносній точності, тут необхідно працювати з точністю η^2 . Крім того, якщо в якості одиниці зберігання використовується машинне слово однаковою для обох методів довжини, алгоритм Хаусхолдера зможе обробляти більш широкий клас задач, ніж алгоритм нормальних рівнянь.

При використанні проєкційних методів ми маємо справу з недовизначеною задачею повного рангу, а це в свою чергу призводить до деяких особливостей обчислення факторизованої оцінки $\hat{\theta}_n$. Тому видається більш ефективним для підвищення обчислювальної стійкості процедур навчання використовувати алгоритм Хаусхолдера [115], що дозволяє побудувати ортогональну матрицю

$$Q = I_N + b^{-1}uu^T,$$

де u – вектор $N \times 1$ такий, що $\|u\| \neq 0$;

$$b = -\|u\|^2/2.$$

В цьому випадку проєкційний алгоритм навчання (3.20) приймає вид

$$\hat{\theta}_n = \hat{\theta}_{n-1} + \gamma \bar{\Theta}, \quad (4.41)$$

де $\bar{\Theta}$ обчислюється за наступним алгоритмом:

$$\mathbf{H}_{n,S}^T \mathbf{Q}_{n,S} = [\mathbf{L}_{n,S} : 0]; \quad (4.42)$$

$$\mathbf{L}_{n,S} \mathbf{z}_n = \mathbf{E}_{n,S}; \quad (4.43)$$

$$\mathbf{z}_n = \bar{\mathbf{R}}_{n,S}^{-1} \mathbf{E}_{n,S}; \quad (4.44)$$

$$\bar{\Theta}_n = \mathbf{Q}_{n,S} \begin{bmatrix} \mathbf{z}_n \\ 0 \end{bmatrix} = \mathbf{Q}_{n,S} \mathbf{z}_n. \quad (4.45)$$

Тут $\bar{\mathbf{R}}_{n,S}$ – $s \times s$ -матриця рангу s ;

$\mathbf{Q}_{n,S}$ – ортогональна $N \times N$ – матриця;

$\mathbf{L}_{n,S}$ - нижня трикутна $s \times s$ -матриця;

\mathbf{z}_n - єдине рішення рівняння $\bar{\mathbf{R}}_{n,S} \mathbf{z}_n = \mathbf{E}_{n,S}$.

Ортогональна матриця $\mathbf{Q}_{n,S}$ визначається із умови

$$\mathbf{Q}_{n,i} \mathbf{H}_i = \begin{vmatrix} \mathbf{H}_{i,1} \\ \dots \\ \mathbf{H}_{i,p-1} \\ -\sigma(\mathbf{H}_{i,p}^2 + \sum_{j=l}^N \mathbf{H}_{i,j}^2)^{\frac{1}{2}} \\ \mathbf{H}_{i,p+1} \\ \dots \\ \mathbf{H}_{i,l-1} \\ 0 \\ \dots \\ 0 \end{vmatrix} = \mathbf{u}_i,$$

де $1 \leq p \leq N$; $p > 1$.

Кроки чисельного процесу, який призводить до ортогональної $N \times N$ матриці, що відповідає цілим параметрам p, l, N , можна звести в наступну схему:

$$\mathbf{Q}_n^i = \begin{cases} \mathbf{I}_N + \mathbf{b}^{-1} \mathbf{v}_n^i (\mathbf{v}_n^i)^T, & \mathbf{b} \neq 0, \quad i = 1, \dots, s; \\ \mathbf{I}_N, & \mathbf{b} = 0; \end{cases}$$

$$\mathbf{b} = \mathbf{g}^i \mathbf{v}_{n,p}^i;$$

$$\mathbf{v}_{n,j}^i = 0, \quad j = 1, \dots, p-1, \quad 1 \leq p \leq N;$$

$$\begin{aligned}
\mathbf{v}_{n,p}^i &= \mathbf{H}_{i,p} - \mathbf{g}^i; \\
\mathbf{v}_{n,j}^i &= 0, \quad j = p+1, \dots, l-1, \quad p < l; \\
\mathbf{v}_{n,j}^i &= \mathbf{H}_{i,j}, \quad j = l, \dots, N; \\
\mathbf{g}^i &= -\sigma \left(\mathbf{H}_{i,p}^2 + \sum_{j=1}^N \mathbf{H}_{i,j}^2 \right)^{1/2}; \\
\sigma &= \begin{cases} 1, & \mathbf{H}_{i,p} \geq 0; \\ -1, & \mathbf{H}_{i,p} < 0. \end{cases}
\end{aligned}$$

Зазначені дії перетворюють вектор \mathbf{H}_n наступним чином:

1. Компоненти вектора \mathbf{H}_n з номерами 1, ..., p-1 не змінюються при p > 1.
2. Головна компонента вектора \mathbf{H}_n з номером p може не змінюватися.
3. Компоненти вектора \mathbf{H}_n з номерами p+1, ..., l-1 не змінюються за умови p < l-1.
4. Компоненти вектора \mathbf{H}_n з номерами l, ..., N повинні бути анульовані при l ≤ N.

Таким чином, матриця $\mathbf{Q}_{n,s}$ може бути представлена у вигляді добутку s перетворень Хаусхолдера

$$\mathbf{Q}_{n,s} = \mathbf{Q}_n^1 \dots \mathbf{Q}_n^s.$$

Нижня трикутна матриця $\mathbf{L}_{n,s}$ обчислюється наступним чином:

$$\begin{aligned}
\mathbf{H}_{n,s} \mathbf{Q}_{n,s} &= \mathbf{H}_{n,s} \mathbf{Q}_n^1 \dots \mathbf{Q}_n^s = \begin{bmatrix} \bar{\mathbf{H}}_{n,1} & 0 & \dots & 0 \\ \bar{\mathbf{H}}_{n-1,1} & \bar{\mathbf{H}}_{n-1,2} & 0 & \dots 0 \\ \dots & \dots & \dots & 0 \\ \bar{\mathbf{H}}_{n-s+1,1} & \dots & \bar{\mathbf{H}}_{n-s+1,s} & 0 \dots 0 \end{bmatrix} = \\
&= (\bar{\mathbf{H}}_n, \bar{\mathbf{H}}_{n-1}, \dots, \bar{\mathbf{H}}_{n-s+1}) = [\mathbf{L}_{n,s} \vdots 0].
\end{aligned}$$

Перетворення Хаусхолдера векторів \mathbf{H}_j виконуються відповідно до правила

$$\bar{\mathbf{H}}_j = \bar{\mathbf{H}}_j + \mathbf{c}_j \mathbf{v}_n^i;$$

$$\mathbf{c}_j = \mathbf{b}^{-1} \mathbf{H}_j^T \mathbf{v}_n^i, \quad j = i + 1, \dots, s.$$

Для реалізації машинної програмою процес побудови ортогонального $N \times N$ – перетворення, що складається в обертанні рядків матриці $\mathbf{H}_{n,S}^T$, можна переписати в наступній алгоритмічній формі:

$$\mathbf{g}^i = -\sigma \left(\mathbf{H}_{i,i}^2 + \sum_{l=i+1}^N \mathbf{H}_{i,l}^2 \right)^{1/2}, \quad i = n, n-1, \dots, n-S+1;$$

$$\sigma = \begin{cases} 1, & \mathbf{H}_{i,i} \geq 0; \\ -1, & \mathbf{H}_{i,i} < 0; \end{cases}$$

$$\mathbf{h}_{n,i} = \mathbf{H}_{i,i} - \mathbf{g}^i;$$

$$\mathbf{H}_{i,i} = \mathbf{g}^i.$$

Побудова перетворення вектора \mathbf{H}_i закінчена. Застосування отриманого ортогонального перетворення до векторів \mathbf{H}_j ($j = i + 1, \dots, S$) здійснюється відповідно до алгоритму

$$\mathbf{b} = \mathbf{H}_{i,i} \mathbf{h}_{n,i}, \quad \text{при } \mathbf{b} \neq 0;$$

$$\mathbf{d} = \frac{\mathbf{H}_{j,i} \mathbf{h}_{n,i} + \sum_{l=i+1}^N \mathbf{H}_{j,l} \mathbf{H}_{i,l}}{\mathbf{b}}, \quad j = 1, \dots, s-i;$$

$$\mathbf{H}_{j,i} = \mathbf{H}_{j,i} + \mathbf{d} \mathbf{h}_{n,i};$$

$$\mathbf{H}_{j,l} = \mathbf{H}_{j,l} + \mathbf{d} \mathbf{H}_{n,l}, \quad l = i + 1, \dots, s.$$

Таким чином, алгоритмічна форма обчислень (3.45) – (3.48), зручна для реалізації машинною програмою, може бути записана так:

$$\hat{\Theta}_{n,1} = \frac{\hat{\Theta}_{n,1}}{\mathbf{H}_{n,1}};$$

$$\hat{\Theta}_{n,i} = \mathbf{H}_{i,i}^{-1} \left(\hat{\Theta}_{n,i} - \sum_{j=1}^{i-1} \mathbf{H}_{i,i} \hat{\Theta}_{n,j} \right), \quad j = n-1, \dots, n-s+1;$$

$$\bar{\mathbf{Q}}_n = \begin{bmatrix} \mathbf{E}_{n,s} \\ 0 \end{bmatrix};$$

$$\mathbf{b} = \mathbf{H}_{j,j} \mathbf{h}_{n,j}, \quad j = n-s+1, n-s, \dots, 1;$$

$$\mathbf{d} = \frac{\bar{\mathbf{Q}}_{n,j} \mathbf{h}_{n,j} + \sum_{i=j+1}^N \bar{\mathbf{Q}}_{n,i} \mathbf{H}_{j,i}}{\mathbf{b}};$$

$$\bar{\mathbf{Q}}_{n,i} = \bar{\mathbf{Q}}_{n,i} + \mathbf{d} \mathbf{H}_{j,i}, \quad i = j+1, \dots, N.$$

4.3.3 Ортогоналізація Грама-Шмідта

Нарешті, ще одне перетворення від косокутної системи до ортонормованої здійснюється за s кроків за наступним правилом [116]:

нормується вектор \mathbf{H}_{n-s+1}

$$\gamma_{11} = \|\mathbf{H}_{n-s+1}\|^{-1};$$

$$\bar{\mathbf{H}}_1 = \mathbf{H}_{n-s+1} \|\mathbf{H}_{n-s+1}\|^{-1};$$

ортогоналізуються вектори \mathbf{H}_{n-s+2} та $\bar{\mathbf{H}}_1$, потім нормується і т.д.

$$\beta_{ik} = \mathbf{H}_{n-s+i}^T \bar{\mathbf{H}}_k;$$

$$\bar{\mathbf{H}}_i = \left(\mathbf{H}_{n-s+i} - \sum_{n=1}^{i-1} \beta_{in} \bar{\mathbf{H}}_n \right) \left\| \mathbf{H}_{n-s+i} - \sum_{n=1}^{i-1} \beta_{in} \bar{\mathbf{H}}_n \right\|^{-1}.$$

Елементи матриці Γ визначаються за формулою

$$\gamma_{ij} = \begin{cases} 0 & \text{при } i < j; \\ \frac{1}{\left\| \bar{\mathbf{H}}_{n-s+i} - \sum_{n=1}^{i-1} \beta_{in} \bar{\mathbf{H}}_n \right\|} & \text{при } i = j; \\ \frac{\sum_{n=1}^{i-1} \beta_{in} \gamma_{ni}}{\left\| \mathbf{H}_{n-s+i} - \sum_{n=1}^{i-1} \beta_{in} \bar{\mathbf{H}}_n \right\|} & \text{при } i = j. \end{cases}$$

Алгоритм може бути записаний і в термінах матриць проєкцювання. З наведених вище співвідношень випливає, що алгоритм будується за наступною схемою:

$$\begin{aligned} 1) \quad & \bar{\mathbf{H}}_{n-s+1} = \frac{\mathbf{H}_{n-s+1}}{\|\mathbf{H}_{n-s+1}\|}; \\ & \mathbf{P}_{n-s+1}^{(1)} = \bar{\mathbf{H}}_{n-s+1} \bar{\mathbf{H}}_{n-s+1}^T; \\ 2) \quad & \bar{\mathbf{H}}_{n-s+2} = \frac{(\mathbf{I} - \mathbf{P}_{n-s+1}^{(1)}) \mathbf{H}_{n-s+2}}{\|(\mathbf{I} - \mathbf{P}_{n-s+1}^{(1)}) \mathbf{H}_{n-s+2}\|}; \\ & \mathbf{P}_{n-s+2}^{(1)} = \bar{\mathbf{H}}_{n-s+2} \bar{\mathbf{H}}_{n-s+2}^T; \\ 3) \quad & \bar{\mathbf{H}}_{n-s+3} = \frac{(\mathbf{I} - \mathbf{P}_{n-s+1}^{(1)} - \mathbf{P}_{n-s+2}^{(1)}) \mathbf{H}_{n-s+3}}{\|(\mathbf{I} - \mathbf{P}_{n-s+1}^{(1)} - \mathbf{P}_{n-s+2}^{(1)}) \mathbf{H}_{n-s+3}\|}; \\ & \mathbf{P}_{n-s+3}^{(1)} = \bar{\mathbf{H}}_{n-s+3} \bar{\mathbf{H}}_{n-s+3}^T; \\ s) \quad & \bar{\mathbf{H}}_n = \frac{\left(\mathbf{I} - \sum_{i=1}^{s-1} \mathbf{P}_{n-s+i}^{(1)} \right) \mathbf{H}_n}{\left\| \left(\mathbf{I} - \sum_{i=1}^{s-1} \mathbf{P}_{n-s+i}^{(1)} \right) \mathbf{H}_n \right\|}; \quad \mathbf{P}_n^{(1)} = \bar{\mathbf{H}}_n \bar{\mathbf{H}}_n^T, \end{aligned}$$

де $\mathbf{P}_i^{(1)}$ – проектор на одновимірний простір, заданий вектором \mathbf{H}_i .

Звідси видно, що черговий вектор \mathbf{H}_i отримується з вихідного \mathbf{H}_{n-s+i} як його нормована проекція на ортогональне доповнення до підпростору вже побудованих $(i-1)$ векторів. Однак, як зазначається в [113], такий спосіб побудови ортонормованих векторів дуже чутливий до помилок округлення. Тому для підвищення чисельної стійкості слід використовувати наступний модифікований метод Грама-Шмідта, в якому на кожному кроці обчислюється тільки одна проекція:

$$1) \quad \bar{\mathbf{H}}_{n-s+1} = \frac{\mathbf{H}_{n-s+1}}{\|\mathbf{H}_{n-s+1}\|};$$

$$\mathbf{P}_{n-s+1}^{(1)} = \bar{\mathbf{H}}_{n-s+1} \bar{\mathbf{H}}_{n-s+1}^T;$$

$$\mathbf{H}_{n-s+i}^{(1)} = (\mathbf{I} - \mathbf{P}_{n-s+1}^{(1)}) \mathbf{H}_{n-s+i} \quad (i = 2, 3, \dots, s);$$

$$2) \quad \bar{\mathbf{H}}_{n-s+2} = \frac{(\mathbf{I} - \mathbf{P}_{n-s+1}^{(1)}) \mathbf{H}_{n-s+2}}{\|(\mathbf{I} - \mathbf{P}_{n-s+1}^{(1)}) \mathbf{H}_{n-s+2}\|};$$

$$\mathbf{P}_{n-s+2}^{(1)} = \bar{\mathbf{H}}_{n-s+2} \bar{\mathbf{H}}_{n-s+2}^T;$$

$$\mathbf{H}_{n-s+i}^{(2)} = (\mathbf{I} - \mathbf{P}_{n-s+2}^{(1)}) (\mathbf{I} - \mathbf{P}_{n-s+1}^{(1)}) \mathbf{H}_{n-s+i} \quad (i = 3, 4, \dots, s);$$

.....

$$s) \quad \bar{\mathbf{H}}_n = \frac{\prod_{j=1}^{s-1} (\mathbf{I} - \mathbf{P}_{n-s+j}^{(1)}) \mathbf{H}_n}{\left\| \prod_{j=1}^{s-1} (\mathbf{I} - \mathbf{P}_{n-s+j}^{(1)}) \mathbf{H}_n \right\|}.$$

Таким чином, s-кроковий алгоритм може быть представлено у вигляді

$$\boldsymbol{\theta}_n = \boldsymbol{\theta}_{n-1} - \gamma_n \sum_{i=1}^s \boldsymbol{\Omega}_i \mathbf{H}_{n-s+i},$$

де

$$\mathbf{\Omega}_i = \sum_{n,j=1}^s \gamma_{nj} \gamma_{ni} \mathbf{e}_i ;$$

$$\mathbf{e}_i = \mathbf{y}_{n-s+j} - \mathbf{\theta}_{n-1}^T \mathbf{H}_{n-s+j}.$$

Несложно бачити, що в оцінці (3.15) використовується процес ортогоналізації векторів $\mathbf{H}_n, \mathbf{H}_{n-1}, \dots, \mathbf{H}_{n-s+1}$ саме методом Грама-Шмідта. Дійсно, при $s = 1$

$$\bar{\mathbf{H}}_n = \frac{\mathbf{H}_n}{\|\mathbf{H}_n\|}$$

та

$$\mathbf{P}_n^1 = \bar{\mathbf{H}}_n \bar{\mathbf{H}}_n^T = \mathbf{H}_n \mathbf{H}_n^T \|\mathbf{H}_n\|^{-2},$$

тобто отримуємо алгоритм Уідроу-Хоффа.

Аналогічно при $s = 2$ здійснюється ортогоналізація векторів \mathbf{H}_n і \mathbf{H}_{n-1} , при $s = 3$ - векторів $\mathbf{H}_n, \mathbf{H}_{n-1}, \mathbf{H}_{n-2}$ і т.д. Як зазначається в [79], чисельні властивості модифікованого методу ортогоналізації Грама-Шмідта схожі з властивостями алгоритму Хаусхолдера, однак для своєї реалізації цей метод вимагає більше пам'яті. Крім того, число арифметичних ітерацій в методі Грама-Шмідта дещо більше, тому що в ньому всі операції проводяться над векторами довжини N , в той час як в методі Хаусхолдера стовпці послідовно коротшають. З цієї ж причини програми, необхідні для реалізації методу Грама-Шмідта, вимагають більше пам'яті, ніж програми Хаусхолдера, тому що немає зручного способу отримувати трикутну матрицю з одиничними діагональними елементами на тому місці, де зберігалася вихідна матриця. Для підвищення ефективності перетворень можуть бути розроблені спеціальні схеми послідовного накопичення рядків або групи рядків матриці даних, проте складність обчислень при цьому зростає. Як наприклад, в методі сполучених градієнтів [44] або в методі Форсайта [116]. Таким чином, найбільш

ефективним способом підвищення стійкості алгоритму навчання є застосування перетворення Хаусхолдера.

4.4 Висновки за четвертим розділом

1. Для підвищення ефективності функціоналів, що застосовуються для отримання процедур навчання, запропоновано здійснювати оцінювання їх параметрів, зокрема параметра масштабу як в режимі off-line, так і в режимі on-line, причому останній є досить привабливим в обчислювальному аспекті, тому що оцінки визначаються за допомогою простих рекурентних формул і не використовують в явному вигляді даних минулих спостережень.

2. З метою оптимізації процедур навчання розглянуто питання вибору оптимальних параметрів цих процедур. У зв'язку з тим, що метою використання параметра регуляризації δ є підвищення обчислювальної стійкості процедур оцінювання, виникає проблема ефективного вибору цього параметра. Як показали результати досліджень, використання регуляризуючої добавки, покращуючи стійкість алгоритмів, призводить до деякого уповільнення процесу побудови моделі. Розглянуто існуючі підходи до вирішення даного питання і запропоновано процедуру рекурентного налаштування параметра регуляризації для навчання АДАЛПНИ.

3. Для зменшення часу навчання мережі, який визначається величиною параметра швидкості навчання γ , розглядалася задача вибору оптимального значення цього параметра, яке забезпечує максимальну швидкість збіжності процедури. Отримано вирази для оптимальних значень параметрів швидкості навчання для алгоритмів Качмажа та Нагумо-Ноди, які є досить загальними і залежать від статистичних характеристик корисних сигналів і завад. Тому для їх практичного застосування слід скористатися оцінками цих параметрів. Слід зазначити, що ці оцінки дозволяють досліднику при вирішенні практичних завдань попередньо оцінити можливості досліджуваного алгоритму і ефективність його застосування.

4. З метою поліпшення обчислювальних властивостей багатокрокових процедур навчання розроблені їх факторизовані форми, засновані на перетвореннях Холеського, Хаусхолдера і ортогоналізації Грама-Шмідта. Проведений аналіз їх властивостей показав, що найбільш ефективним способом підвищення стійкості алгоритму навчання є застосування перетворення Хаусхолдера.

Список використаних джерел у даному розділі наведено у повному списку використаних джерел під номерами: 3, 61, 79, 104-116.

5 ЕКСПЕРЕМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ВИРІШЕННЯ ПРАКТИЧНИХ ЗАДАЧ

Імітаційне моделювання є необхідним етапом дослідження ефективності як обраної архітектури ШНМ, так і вживаного алгоритму навчання мережі.

В даний час найбільш масовим напрямком нейрокомп'ютингу є моделювання нейронних мереж на персональних комп'ютерах. Успіхи в моделюванні дозволяють створювати навчальні програми, що дають користувачеві повне уявлення про ШНМ і їх можливості. Однак незалежно від використання ваного пакету результатом експериментального дослідження є визначення областей найбільш ефективного застосування того чи іншого типу ШНМ і розробка рекомендацій щодо їх практичного використання.

В даний час популярні і широко використовуються наступні програмні нейросимулятори: MATLAB Neural Network Toolbox, Scilab 5.5.2, NeurophStudio 2.92, Torch 7, Theano, Caffe та інші [117,118].

У даній роботі нейронні мережі прямого поширення були реалізовані як розширення пакету NeurophStudio і всі дослідження проводилися в середовищі SciLab.

5.1 Рішення задачі ідентифікації

Експеримент 1.

Вирішувалося завдання ідентифікації нестационарного об'єкта, що описується рівнянням

$$y_{n+1} = \theta_n^T x_{n+1}, \quad (5.1)$$

з наступними параметрами $N = 10$, причому з 10 параметрів 4 були нестационарними, представленими марківською моделлю (2.53) з $\sigma_S^2 = 0,5$.

Решта шість параметрів приймалися рівними

$\theta_5 = 0,8; \theta_6 = 0,4; \theta_7 = -0,2; \theta_8 = 0,55; \theta_9 = -0,67; \theta_{10} = 0,15$. В якості вхідного сигналу x_{n+1} і адитивної завади ξ_{n+1} обиралися послідовності нормально розподілених величин $x_{n+1} \sim N(0;1)$, $\xi_{n+1} \sim N(0;3)$.

В якості критерію порівняння роботи алгоритмів використовувалася величина

$$MSE = M \left\{ \left\| \tilde{\theta}_{n+1} \right\|^2 \right\}.$$

На рисунку 5.1 а, б наведені графіки зміни величини для регуляризованих алгоритмів Качмажа і Нагумо-Ноди відповідно при різному виборі параметра релаксації γ . Так як при імітаційному моделюванні величина $\left\| \tilde{\theta}_{n+1} \right\|^2$ відома, це робить можливим порівняння граничних (оптимальних) можливостей алгоритмів. Тому криві без маркерів на малюнках відповідають теоретично оптимальному вибору параметрів γ^{opt} відповідно до (4.22) та (4.26), а решта - практичному вибору цих параметрів. Так криві з кружками відповідають вибору (завданню) $\gamma = 0,7$ і $\delta = 0$, криві з трикутниками - вибору $\gamma = 0,7$ і $\delta = 0,01$, криві з квадратами - вибору $\gamma = 0,7$ і $\delta = 0,02$, криві з ромбами - вибору $\gamma = 0,6$ і $\delta = 0,02$.

На рисунку 5.1 в показані результати порівняно аналізу роботи досліджуваних алгоритмів при виборі відповідних γ^{opt} (криві без маркерів) і завданні $\gamma = 0,7$ і $\delta = 0,01$ (криві з трикутниками).

Аналізуючи результати моделювання, можна зробити висновок про те, що введення параметра регуляризації $\delta \neq 0$ приводить до уповільнення швидкості збіжності алгоритмів. При цьому, однак, не виникає проблеми розподілу на нуль, тобто стійкість алгоритмів підвищується.

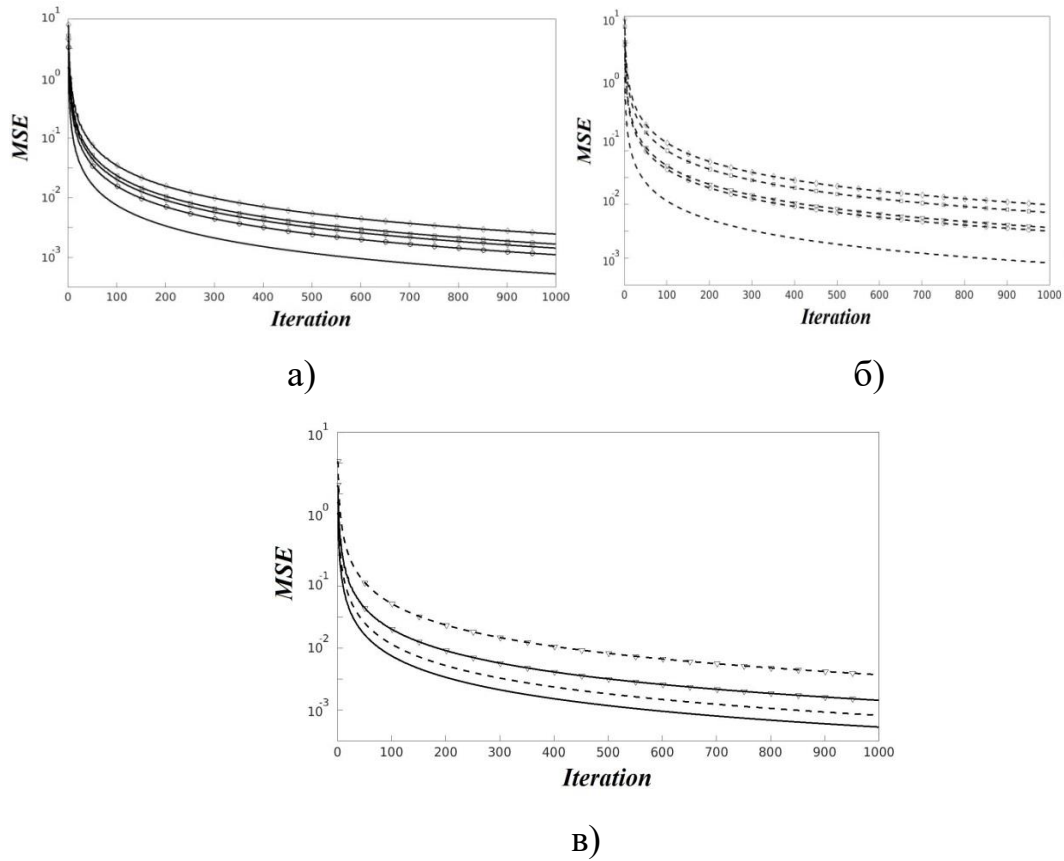


Рисунок 5.1 – Результати порівняльного аналізу

Експеримент 2 (дослідження робастних властивостей).

Розглядалася задача ідентифікації стаціонарного лінійного об'єкта, описуваного рівнянням (5.1) з параметрами $\theta = (1, 2, 4, 8, 16, 32, 64, 128, 256)^T$. В якості вхідного сигналу x_n вибиралися послідовності нормально розподілених величин $x_n \sim N(0;1)$. При тестуванні робастності алгоритмів в вихідний сигнал об'єкта додавався незалежний шум с розподілом Релея ($\gamma = 1.0$). Гістограма такої завади показана на рисунку 5.2.

Результати моделювання представлені на рисунках – 5.3, 5.4. На рисунку 5.3 показано графік налаштування параметрів моделі, а на рисунку 5.4 помилки ідентифікації.

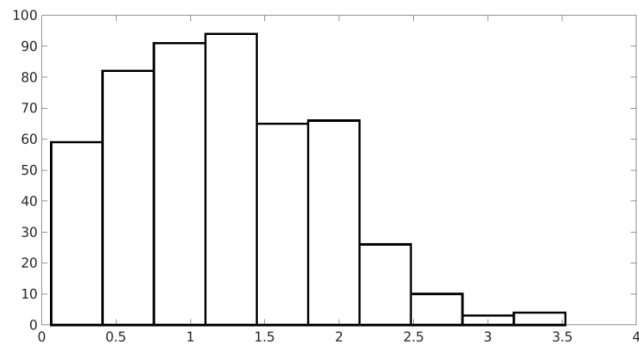


Рисунок 5.2 – Гістограма завади

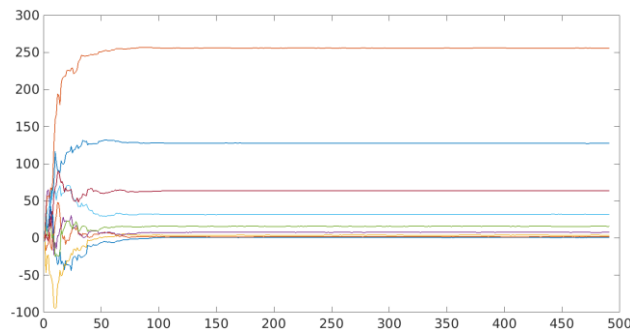


Рисунок 5.3 – Графіки налаштування параметрів моделі

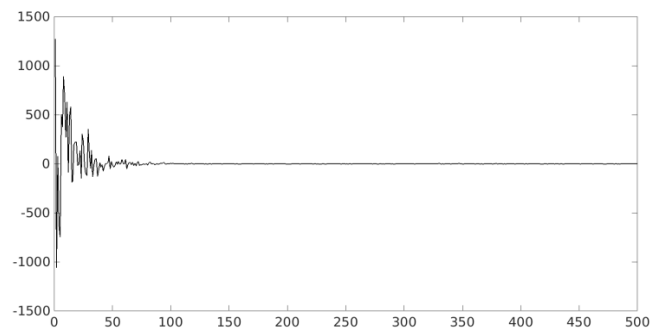


Рисунок 5.4 – Помилка ідентифікації

Експеримент 3.

Розглядалася задача ідентифікації стаціонарного лінійного об'єкта, описуваного рівнянням (5.1) з наступними параметрами:

$$\theta = (-5; -9; -40; -68; 16; 32; 64; 128; 256)^T.$$

В якості вхідного сигналу x_n вибиралися послідовності нормально розподілених величин $x_n \sim N(0;1)$. При тестуванні робастності алгоритмів в

вихідний сигнал об'єкта додавався незалежний шум з розподілом Релея ($\gamma=1.0$). Гістограма такої завади показана на рисунку 5.5.

Результати моделювання при різних завданнях параметра змішування λ представлені на рисунках 5.6 – 5.9. На рисунках 5.6 – 5.7 показано графіки налаштування параметрів моделі при виборі $\lambda=0,5$ та $\lambda=0,7$ відповідно, а на рисунках 5.8 – 5.9 – відповідні помилки ідентифікації.

Експеримент 4.

Для експериментального дослідження можливостей алгоритму (3.30) проводилася ідентифікація лінійного об'єкта (КІХ-фільтра), що описується рівнянням (5.1) з $\theta=(1, 2, 3, 4, 5, 4, 3, 2, 1)^T$. При тестуванні робастності алгоритмів в вихідний сигнал об'єкта додавався незалежний гаусівський шум зі значно більшою амплітудою для моделювання «викидів» в системі (імпульсна завада). Приклад такої завади показаний на рисунку 5.10. Результати моделювання представлені на рисунках 5.11 – 5.12. При цьому зліва показані графіки налаштування параметрів моделі, а праворуч - зміна помилки ідентифікації. Рисунок 5.11 відображає результат ідентифікації лінійної системи при виборі $\lambda=1$ і відсутності завад вимірів. На рисунках 5.12 5..наведені результати ідентифікації при наявності імпульсних завад з використанням алгоритму (3.30) при $\lambda=1, 0.8, 0.6$ відповідно. При $\lambda=0$, тобто при використанні знакового алгоритму, процес ідентифікації не збігається.

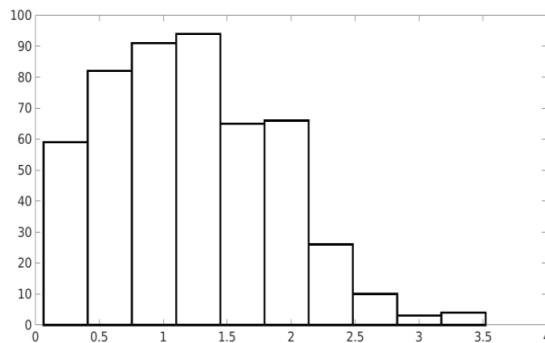


Рисунок 5.5 – Гістограма завади

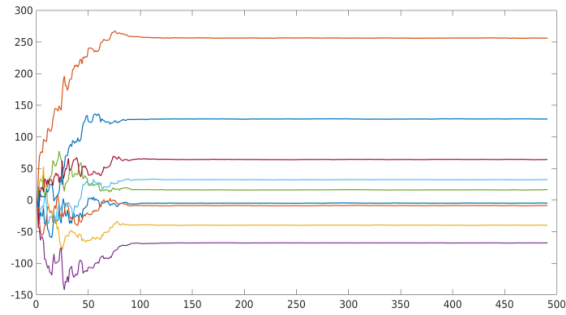


Рисунок 5.6 – Графіки налаштування параметрів моделі при виборі $\lambda = 0,5$

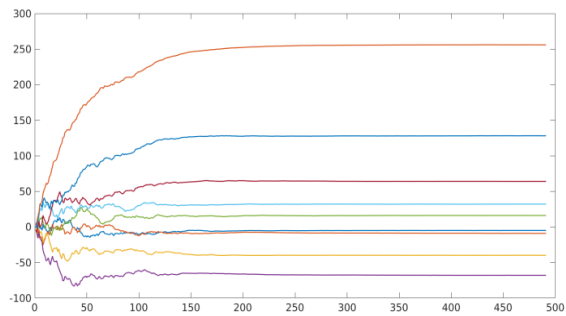


Рисунок 5.7 – Графіки налаштування параметрів моделі при виборі $\lambda = 0,7$

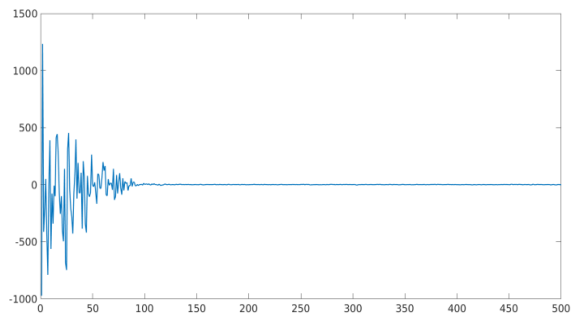


Рисунок 5.8 – Помилки ідентифікації при виборі $\lambda = 0,5$

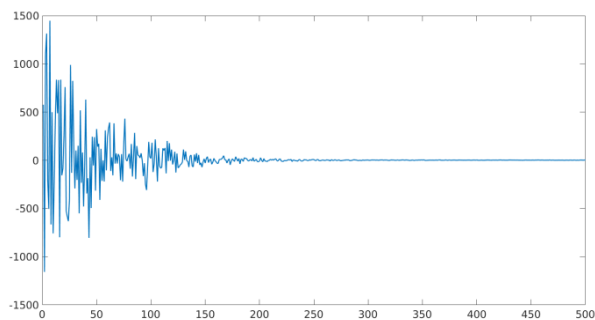


Рисунок 5.9 – Помилки ідентифікації при виборі $\lambda = 0,7$

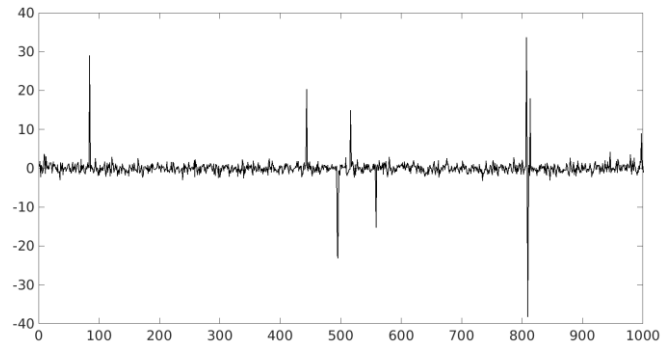
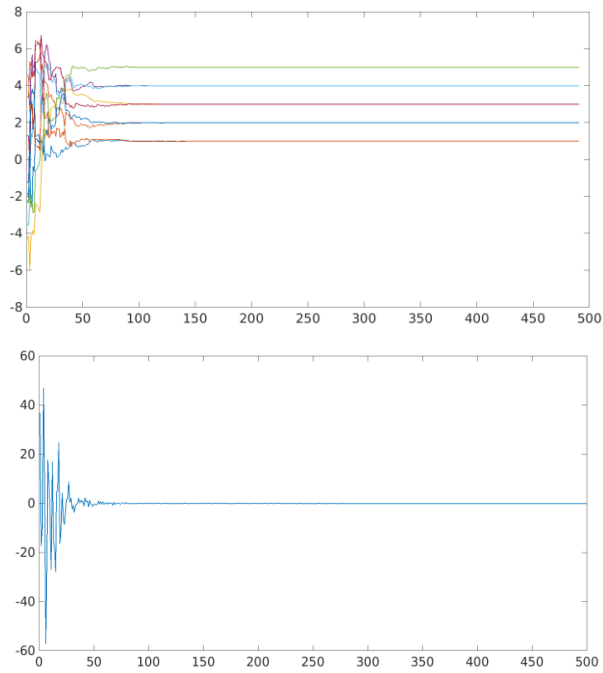
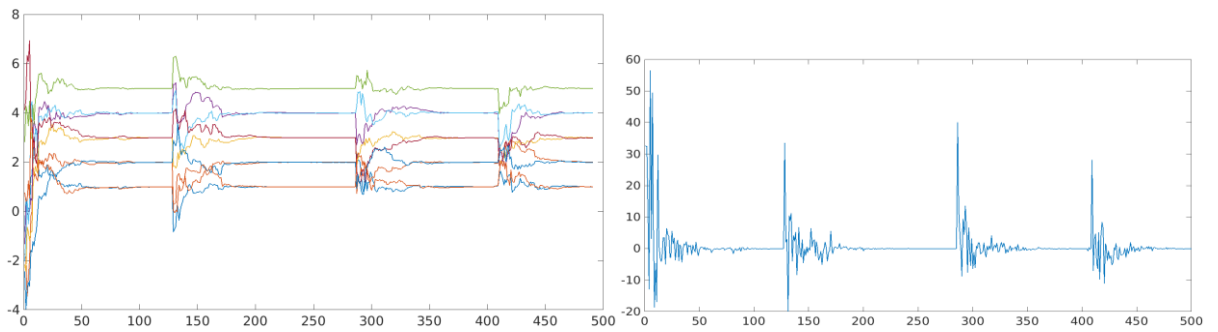


Рисунок 5.10 – Імпульсна завада

Рисунок 5.11 – Результати ідентифікації при виборі $\lambda = 1$ Рисунок 5.12 – Результати ідентифікації при виборі $\lambda = 0.8$

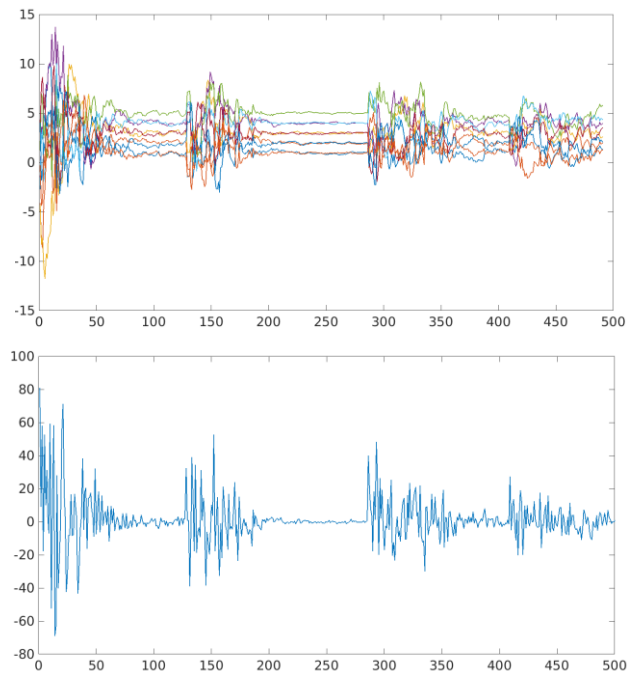


Рисунок 5.13 – Результати ідентифікації при виборі $\lambda = 0.6$

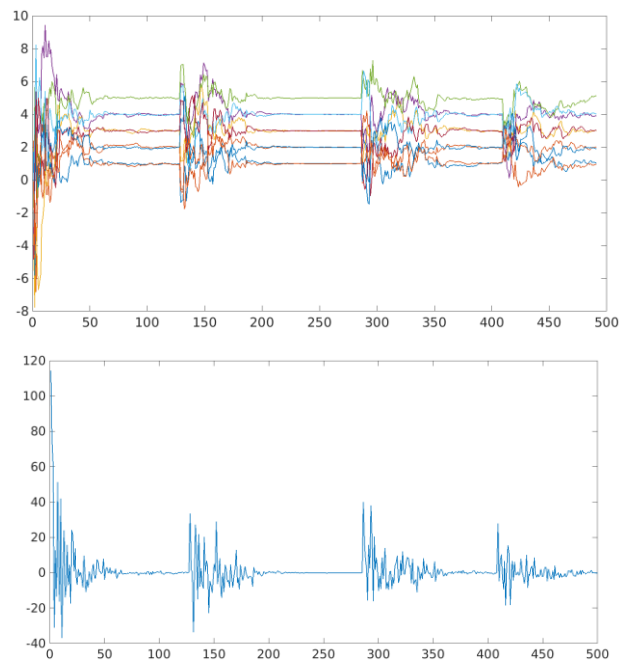


Рисунок 5.14 – Результати ідентифікації при виборі $\lambda = 0$

Експеримент 5.

Розглядалася задача ідентифікації нелінійного динамічного об'єкта, що описується рівнянням:

$$y_n = \max \left[e^{-10u_n^2}; e^{-50y_{n-1}^2}; 1.25e^{-5(u_n^2 + y_{n-1}^2)} \right], \quad (5.2)$$

де u_n^2 – вхідний сигнал, який представляє собою стаціонарну випадкову послідовність з рівномірним законом розподілу в інтервалі $[-1, 1]$, що генерується датчиком випадкових чисел. При дослідженні даного об'єкта використовувалося популяція мереж СМАС, що складаються з 100 особин, яка еволюціонувала протягом 500 епох. Всі параметри мереж (включаючи R і ρ) визначалися за допомогою ЕА. Слід зазначити, що величини R і ρ визначають обсяг використовуваної пам'яті для зберігання параметрів мережі і значно впливають на точність апроксимації. На рисунках 5.15 показані графіки зміни фітнес-функції мережі-переможця (ліворуч) і обсягу необхідної пам'яті для зберігання її параметрів (праворуч).

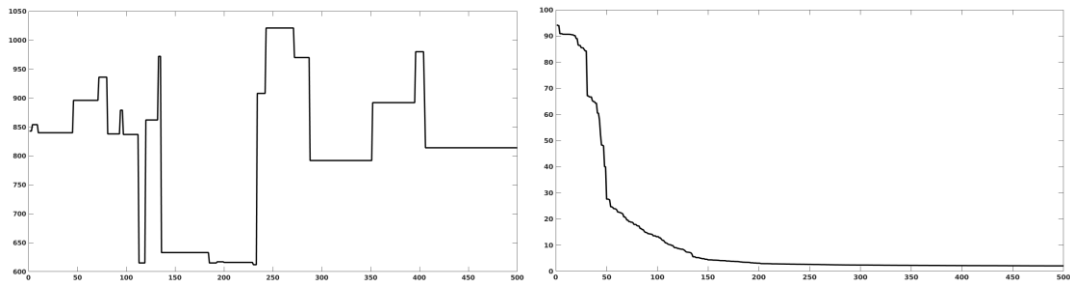


Рисунок 5.15 – Результати моделювання

Результати ідентифікації стаціонарного об'єкта (5.2) представлені на рисунках 5.16 -5.17. На рисунку 5.16 показана сама поверхня, що описується даним рівнянням, а на рисунку 5.17 - поверхня, відновлена за допомогою ЕСМАС.

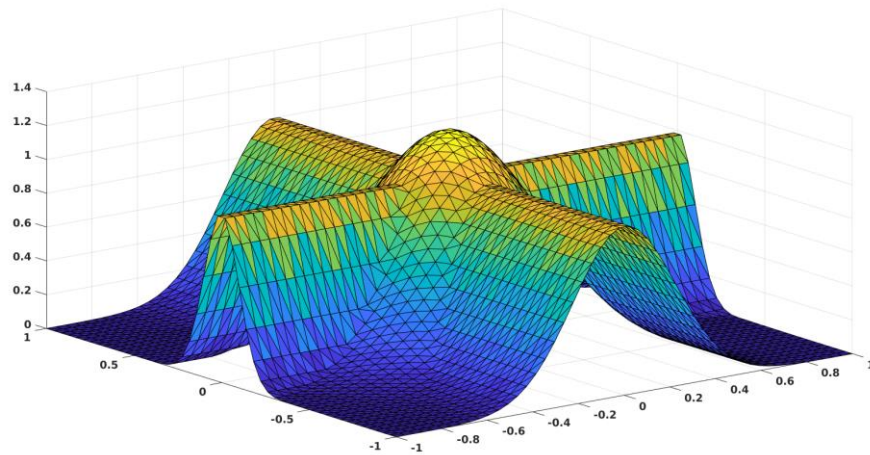


Рисунок 5.16 – Поверхня, що моделюється

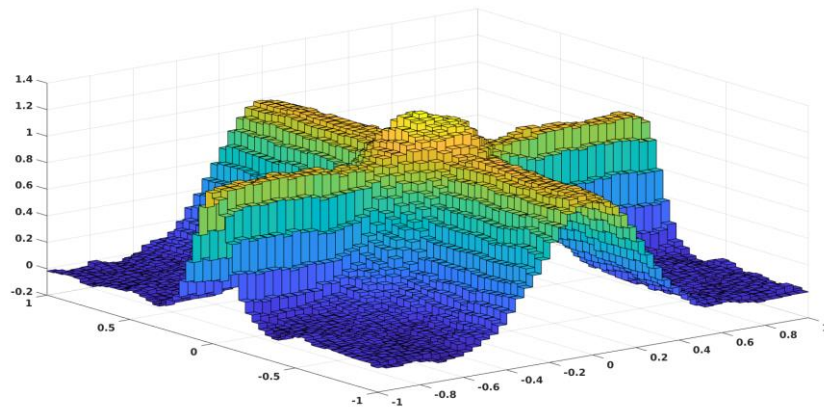


Рисунок 5.17 – Отримана поверхня

Мережа-переможець містила 814 вагових параметрів при $R = 186$ і $\rho = 95$, а також використовувала прямокутні функції активації. На рисунку 5.17 наведені результати ідентифікації об'єкта (12) при наявності нормально розподіленої на інтервалі $[-0.3, 0.3]$ випадкової завади ξ_n . В даному випадку мережа-переможець використовувала косинусоїдальні функції активації.

5.2 Рішення задачі прогнозування

Експеримент 1.

Розглядалась задача прогнозування об'єму пасажиропотоку у аеропортах Німеччини. Для цього було використано данні з офіційного сайту

Eurostat [119] – статистичної служба Європейського союзу, що займається збором статистичної інформації по країнам-членам ЄС і гармонізацією статистичних методів використовуваних даними країнами.

Слід зазначити, що часові ряди, які відображають пасажиропотік будь-якого транспорту є надзвичайно популярні при тестуванні різноманітних систем прогнозування, оскільки вони мають усі необхідні складові часового ряду – тренд, сезонність та циклічність. Отже, для прогнозування об'єму пасажиропотоку у аеропортах Німеччини використовувався багатосаровий перцептрон (БП) з трьома входами та одним виходом (для прогнозування кількості пасажирів у наступному місяці використовувались данні за три минулих місяця), БП мав два прихованих шара по 20 нейронів кожний. Для навчання нейронної мережі було сформовано навчаючу вибірку, у якій було відображено зміни пасажиропотоку Німеччини за останні 16 років. Результати моделювання з використанням різних значень параметру δ у алгоритмі навчання мережі наведено на рисунку 5.18–5.19. На рисунках сполушною лінією показано реальний пасажиропотік, а пунктирною – спрогнозований за допомогою БП. Результати на рисунку 5.18 відповідають налаштуванню відповідно з алгоритмом Качмажа при $\delta = 0.01$, а на рисунку 5.19 – при $\delta = 0.05$.

Як видно з результатів моделювання, при виборі значення параметру $\delta = 0.01$ наприкінці часового ряду помилка прогнозування істотно зросла і результати прогнозування істотно відрізнялись від реальних значень. При виборі ж параметру $\delta = 0.05$ вдалося отримати результати прогнозування, які суттєво не відрізнялись від реальних.

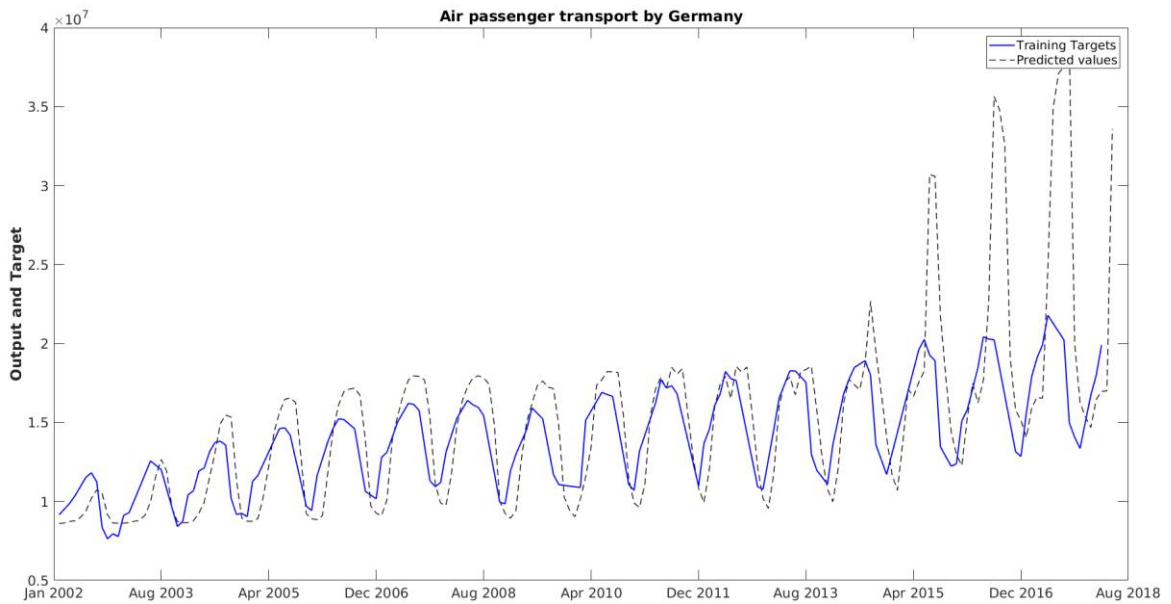


Рисунок 5.18 – Результати прогнозування пасажиропотоку ($\delta = 0.01$)

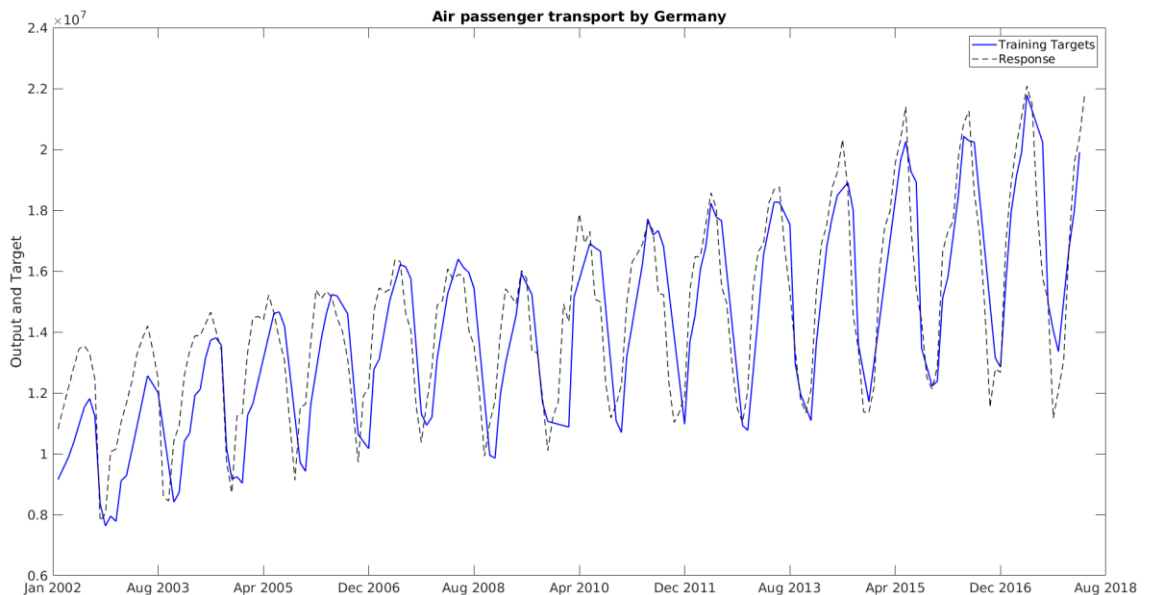


Рисунок 5.19 – Результати прогнозування пасажиропотоку ($\delta = 0.05$)

Експеримент 2.

Вирішувалася задача прогнозування енергоспоживання деревопереробним підприємством Німеччини за допомогою БШП. Для побудови моделі процесу енергоспоживання використовувалася статистична вибірка за останні 3 роки. На рисунку 5.20 показані результати побудови прогнозуючої моделі. Результати прогнозування на півроку вперед представлені на рисунку 5.21.

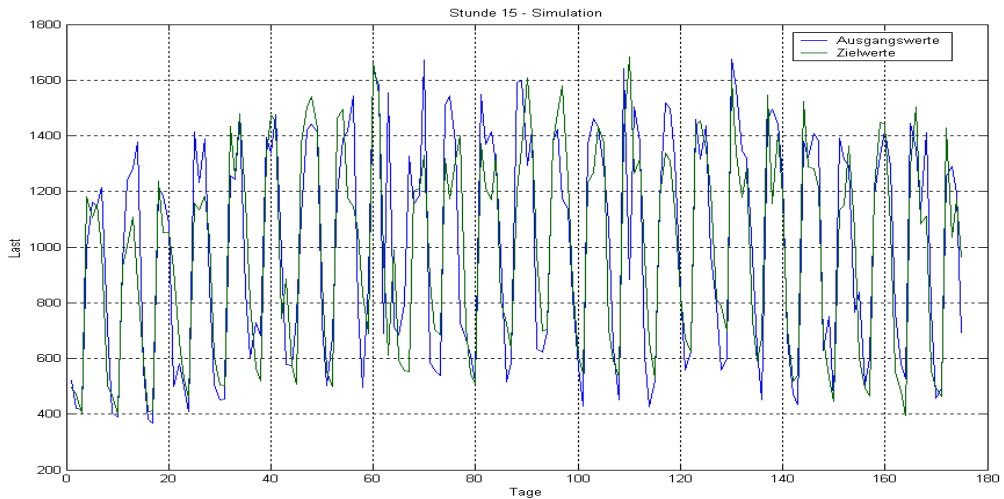


Рисунок 5.20 – Результати побудови прогнозуючої моделі

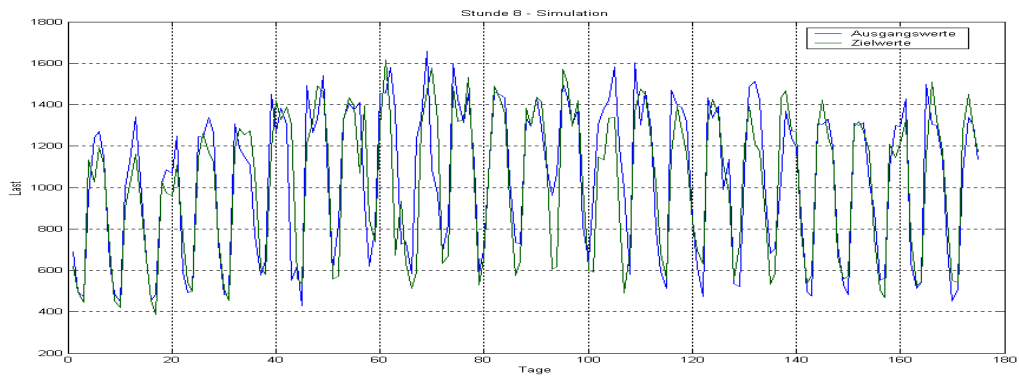


Рисунок 5.21 – Результати прогнозування енергоспоживання на півроку вперед

Досліджувались властивості процедур навчання, побудованих на основі стохастичного градієнта, наведених в Додатку Д.

Результати досліджень показують, що для задачі оптимізації досить ефективним є простий алгоритм градієнтного спуску. Використання імпульсів в розглянутих модифікаціях призводить до деякого поліпшення процесу розпізнавання, але збільшує вартість обчислення.

Таблиця 5.1

<i>SGD</i>	100	73%	0.6247	0.6139
	250	72%	0.6280	0.6195
	500	73%	0.6130	0.6193
<i>Momentum</i>	100	82%	0.4133	0.4076
	250	88%	0.2496	0.2356
	500	88%	0.2491	0.2320
<i>NAG</i>	100	95%	0.1293	0.1152
	250	95%	0.1301	0.1149
	500	97%	0.0891	0.0803
<i>Adagrad</i>	100	98%	0.0298	0.0285
	250	98%	0.0254	0.0227
	500	99%	0.0110	0.0059
<i>RMSProp</i>	100	56%	1.0334	1.0304
	250	57%	1.0301	0.9879
	500	56%	1.0013	1.0060
<i>Adadelta</i>	100	67%	0.7688	0.7597
	250	66%	0.7601	0.7818
	500	67%	0.7686	0.7605
<i>Adam</i>	100	62%	0.8718	0.8621
	250	62%	0.8623	0.8580
	500	68%	0.7218	0.7153
<i>Adamax</i>	100	97%	0.0889	0.0789
	250	99%	0.0091	0.0086
	500	99%	0.0083	0.0071
<i>Nadam</i>	100	75%	0.5847	0.5630
	250	83%	0.4114	0.3993
	500	89%	0.2682	0.2376
<i>AMSGrad</i>	100	74%	0.6048	0.5956
	250	79%	0.5073	0.4820
	500	83%	0.4154	0.3952
<i>WNGrad</i>	100	78%	0.4875	0.4749
	250	81%	0.4546	0.4403
	500	87%	0.3228	0.2985
<i>s - step (s = 10)</i>	100	80%	0.4933	0.4918
	250	83%	0.4196	0.3856
	500	85%	0.3481	0.3319
<i>s - step (s = 20)</i>	100	82%	0.4129	0.4023
	250	86%	0.3287	0.3382
	500	87%	0.3219	0.3130

У розглянутих задачах найбільш ефективним алгоритмом є Adamax. В [120] рекомендується завжди починати з Adam, незалежно від архітектури нейронної мережі та проблемних областей, в яких він використовується. На нашу думку, при вирішенні завдань розпізнавання слід використовувати алгоритм Adamax. Однак, проблема вибору оптимальних значень параметрів

алгоритмів, що забезпечують максимальну швидкість навчання, залишається відкритою.

5.3 Побудова нейромережевої системи ідентифікації процесу виробництва кальцинованої соди

Технологічна схема виробництва кальцинованої соди (ВКС) включає кілька циклів, найбільшими з яких є цикл аміаку, що включає відділення абсорбції-десорбції, карбонізації та фільтрації, і цикл CO_2 , що включає в себе відділення карбонізації, фільтрації і кальцинації. Цикли обумовлюють тісний зв'язок між режимами, які об'єднуються цими відділеннями. Найбільш жорстко зазначений зв'язок проявляється між режимами відділень абсорбції-десорбції, карбонізації та фільтрації, тобто в аміачному циклі. Це визначає першочерговість автоматизації відділень аміачного циклу, що забезпечує, з одного боку, вирішення першочергових завдань зниження собівартості готового продукту і охорони навколишнього середовища, а з іншого - можливість подальшого природного розвитку системи управління всім виробництвом як ієрархічної системи більш високого рівня.

Труднощі оптимізації процесів ВКС обумовлені наступними факторами [120]:

- складністю фізико-хімічних процесів переробки сировини, що супроводжуються цілою низкою реакцій хімічного перетворення, міжфазного міжфазового обміну, змішування і поділу, нагрівання та охолодження;
- відсутністю достатньо точних математичних моделей цих процесів;
- нелінійністю і нестационарністю характеристик процесів (наприклад, інкрустацією внутрішніх поверхонь карбонізаційної колони в відділенні карбонізації, нестійкістю розрахунку концентрацій аміаку тощо);
- істотним впливом збурень технологічних режимів, що виникають внаслідок зміни параметрів потоків газів, що несуть аміак, а також внаслідок зміни теплового режиму холодильників;

- наявністю досить жорстких обмежень, які накладаються на технологічні процеси (у вигляді норм його поведінки), на потоки сировини і енергії, а також обумовлених роботою суміжних відділень ВКС.

При моделюванні технологічних процесів, що протікають в типовому виробництві кальцинованої соди, розглядалася технологічна схема, наведена на рисунку 5.22.

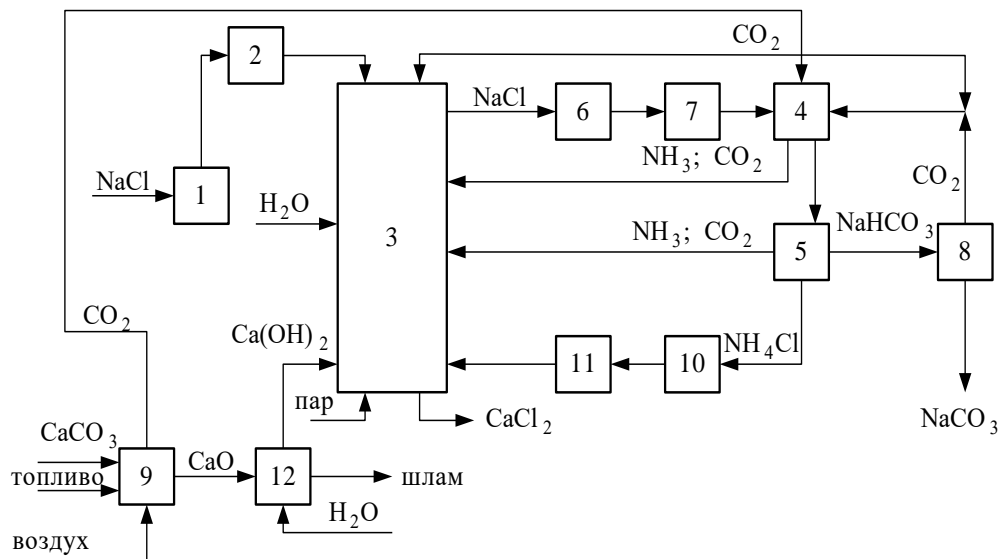
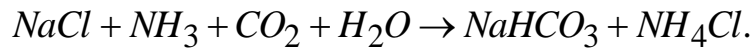


Рисунок 5.22 – Технологічна схема ВКС

Хлорнатрієва сировина у вигляді розсолу надходить у відділення розсолоочістки 1 для видалення сполук кальцію і магнію. Після очищення розсіл подається в напірний бак 2, з якого він надходить на чотири (для типового ПКС) паралельно працюючих елемента (технологічні нитки) відділення абсорбції-десорбції 3, що складаються з апаратів колонного типу абсорбційної і десорбційної частин. В апарати абсорбційної частини противотоком розсолу надходять гази з десорбційної частини, а також з відділень карбонізації 4 і фільтрації 5 для насичення розсолу аміаком і частково діоксидом вуглецю. Отриманий аммонізований розсіл після охолодження в зрошувальному холодильнику 6 надходить до збірника 7, потім - в відділення карбонізації 4 і розподіляється між паралельно працюючими

групами карбонізаційних колон для насичення діоксидом вуглецю, який подається після компримування з відділень кальцинації 8, абсорбції-десорбції 3 і шахтних вапняно випалювальних печей 9. З хімічної точки зору процес отримання бікарбонату натрію описується однією реакцією:



В результаті карбонізації утворюється суспензія бікарбонату натрію і хлористого амонію, яка надходить до відділення фільтрації 5. Тут її тверда фаза (кристали бікарбонату натрію) на паралельно працюючих обертових барабанних вакуум-фільтрах відділяється від маткової рідини, промивається, а потім направляється у відділення кальцинації 8, в якому бікарбонат натрію розкладається прокалюванням з утворенням готового продукту - карбонату натрію (соди кальцинованої) і діоксиду вуглецю.

Після сепарації фільтрова рідина, що представляє собою суміш маткової і промивної рідини і містить хлористий амоній, деяка кількість вуглеамонійних солей і хлористого натрію, який не прореагував, надходить до збірника 10, з якого її подають в напірний бак 11. Далі вона надходить в апарати десорбційної частини для десорбції з неї аміаку і діоксиду вуглецю за допомогою пари. Для розкладання хлористого амонію подають гідроксид кальцію (вапняне молоко) з відділення гасіння 12. Утворений в результаті парогазовий потік направляють в абсорбційну частину відділення, а рідкі відходи - в накопичувачі.

При вивченні технологічних процесів відділення абсорбції-десорбції виробництва кальцинованої соди, були виділені 64 параметра, що характеризують ці процеси. Як показали дослідження, в якості вхідних змінних для одного такого відділення доцільно вибрати такі:

$$x_1 = G_{ф.ж} \text{ – витрата фільтрової рідини, (м}^3\text{/ч);}$$

$$x_2 = G_{ДС}^n \text{ – витрата пари в ДС, (т/ч);}$$

$x_3 = G_{CM}^{IM}$ – витрата вапняного молока в змішувачі, (м³/ч);

$x_4 = G_{IIIKLI}^{o.p.}$ – витрата очищеного розсолу на промивач газу колон, (м³/ч);

$x_5 = P_{ДС}^c$ – надлишковий тиск газу на вході в дистиляр (кг с/см²);

$x_6 = t_{ф.ж}$ – температура фільтрової рідини (°С);

$x_7 = P_{KIIIА}^c$ – абсолютний тиск газу в колекторі перед промивачем газу абсорбції (мм рт.ст.);

$x_8 = P_{KIIА-2}^c$ – абсолютний тиск газу в колекторі перед другим абсорбером (мм рт.ст.);

$x_9 = C_{o.p}^{Cl}$ – зміст хлор-іона в очищеному розсолі;

$x_{10} = t_{IM}$ – температура вапняного молока;

$x_{11} = C_{CaO}^{IM}$ – зміст СаО в вапняному молоці.

Основними вихідними характеристиками відділення абсорбції-десорбції є наступні:

$y_1 = t_{AB-1}^c$ – температура газу на виході з першого абсорбера;

$y_2 = t_{AB-2}^{AP}$ – температура рідини (амонізованого розсолу) на виході з другого абсорбера (°С);

$y_3 = C_{AB-2}^{NH_3}$ – прямий титр аміаку в рідині на виході з другого абсорбера; прямой титр аммиака в жидкости на выходе из второго абсорбера;

$y_4 = C_{AB-2}^{CO_2}$ – вміст діоксиду вуглецю в рідині на виході з другого абсорбера.

Серед вхідних змінних $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$ є регульованими, інші нерегульовані.

При побудові нейромережевої моделі об'єкта [39] попередньо була проведена стандартизація змінних, що полягає в центруванні і нормуванні вхідних і вихідних змінних

$$\bar{x}_{jk} = \frac{x_{jk} - m_{x_j}}{\sigma_{x_j}}, \quad \bar{y}_{ik} = \frac{y_{ik} - m_{y_i}}{\sigma_{y_i}}, \quad j = \overline{1,11}, \quad k = \overline{1,4000}, \quad i = \overline{1,4},$$

де m_{x_j} – середнє вибірки x_{jk} , $k = \overline{1,4000}$;

m_{y_i} – середнє вибірки y_{ik} , $k = \overline{1,4000}$;

σ_{x_j} , σ_{y_i} – дисперсії вибірок x_{jk} і y_{ik} відповідно.

В якості нейромережевої моделі використовувалася ШНМ типу МП 8-8-4, що складається з трьох шарів, з 8 нейронами у вхідному шарі, 8 нейронами в прихованому шарі і 4 нейронами у вихідному шарі. Це відповідає тому, що в якості вхідних вибиралися регульовані параметри процесу виступають $x_1 - x_8$, вихідних - $y_1 - y_4$.

В якості функції активації нейронів вхідного і прихованого шарів використовувався гіперболічний тангенс, а в вихідному шарі - лінійні функції активації. Налаштування мережі здійснювалася на підставі 4000 навчальних пар, необхідна точність задавалася на рівні 0.001. Для навчання мережі використовувався алгоритм Левенберга-Марквардта. Результати ідентифікації об'єкта представлені на рисунках 5.22 -5.23. Тут суцільною лінією показані виходи нейронної мережі, а пунктирною - виходи об'єкта. Як видно з малюнка, мережа досить добре навчилася і здатна з високою точністю відтворювати поведінку об'єкта.

Як впливає з наведених результатів, застосування нейромережевих моделей для ідентифікації технологічних процесів відділення абсорбції-десорбції ПКС виявилось досить ефективним.

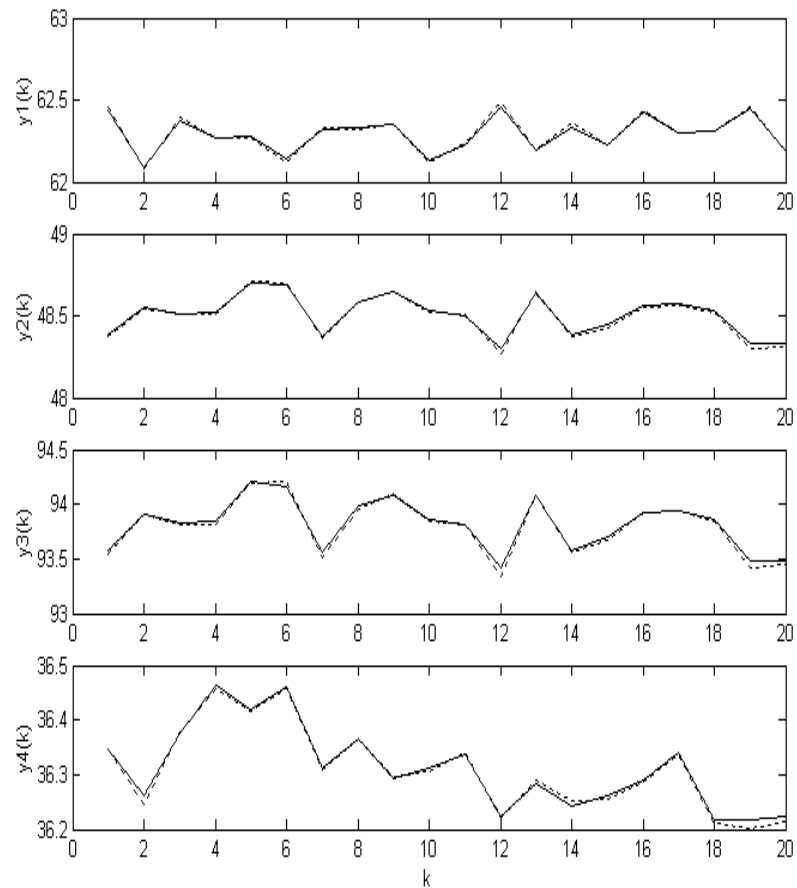


Рисунок 5.23 – Результати ідентифікації технологічного процесу відділення абсорбції – десорбції ВКС

На рис. 5.24 показано результати прогнозування вихідних параметрів технологічного процесу відділення абсорбції-десорбції ВКС.

Акт використання результатів дисертаційної роботи наведено в додатку Ж

5.4 Висновки за п'ятим розділом

1. У зв'язку з тим, що імітаційне моделювання є необхідним етапом дослідження ефективності як обраної архітектури ШНМ, так і вживаного алгоритму навчання мережі, розглянуто найбільш популярні і широко використовуємі на цей час програмні нейросимулятори: MATLAB Neural Network Toolbox, Scilab 5.5.2, NeurophStudio 2.92, Torch 7, Theano, Caffe. Для реалізації нейронних мереж прямого поширення обрано розширення пакету NeurophStudio, а для дослідження мереж – середовище SciLab.

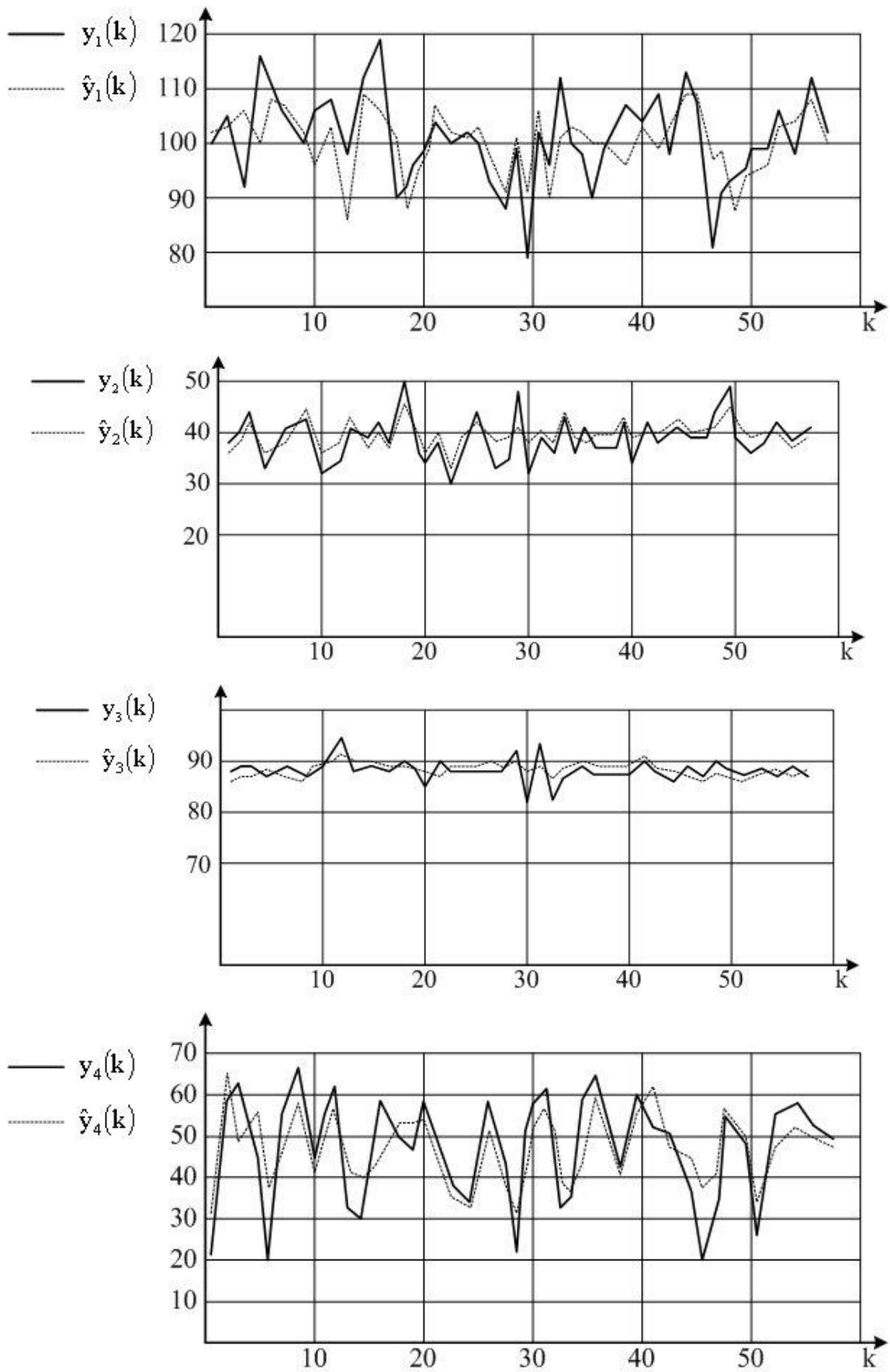


Рисунок 5.24 – Результати прогнозування вихідних параметрів технологічного процесу відділення абсорбції-десорбції ПКС

2. Здійснено імітаційне моделювання нейромережових методів ідентифікації стаціонарних та нестаціонарних об'єктів за умов наявності гаусівських та не гаусівських завад за допомогою ШНМ АДАЛІНА. Дослідження властивостей регуляризованих алгоритмів навчання Качмажа і Нагумо-Ноди при різному виборі параметра релаксації γ підтвердило ефективність використання цих алгоритмів для оцінювання параметрів, які описуються марківською моделлю першого порядку

3. Досліджено використання для прогнозування нестаціонарних часових рядів еволюційної мережі СМАС, всі параметри якої визначалися за допомогою еволюційного алгоритму, за умов наявності завад. При дослідженні даного об'єкта використовувалася популяція мереж СМАС, що складалися з 100 особин, яка еволюціонувала протягом 500 епох. Результати дослідження свідчать про ефективність використання цієї мережі для побудови нелінійних моделей прогнозування за наявністю завад.

4. На основі результатів імітаційного моделювання розроблена нейромережева система прогнозування, що дозволяє збільшити точність прогнозування і підвищити її ефективність. Запропонована і програмно реалізована нейромережева система прогнозування параметрів технологічних процесів виробництва кальцинованої соди, що дозволяє збільшити точність прогнозування та оперативно корегувати керуючі дії з метою дотримання технологічних режимів.

Список використаних джерел у даному розділі наведено у повному списку використаних джерел під номерами: 39, 117-120.

ВИСНОВКИ

У дисертаційній роботі на основі отриманих теоретичних і експериментальних досліджень вирішена актуальна науково-практична задача побудови нейромережових методів прогнозування нестационарних часових рядів, які на відміну від існуючих, зменшують помилку і збільшують точність прогнозування в умовах апріорної і поточної невизначеності і наявності завад.

При цьому отримано наступні основні результати:

1. Проведено аналіз стану проблеми прогнозування нестационарних часових рядів. Розглянуто питання застосування нейронних мереж для рішення цих завдань і описані основні типи ШНМ, сформульована мета і задачі дослідження.

2. Розглянуто БШП в якості нейромережевої моделі для прогнозування. Описано процедуру вибору його структури та запропоновано процедуру навчання. Вивчено умови її збіжності та отримано вираз для оптимального значення параметра збіжності, що дозволяє підвищити зменшити час навчання мережі.

3. Вперше запропоновано використання для навчання АДАЛІНИ в задачі оцінювання параметрів, які описуються марківською моделлю першого порядку, регуляризованих алгоритмів Качмажа та Нагумо-Ноди. Визначено умови збіжності та отримано неасимптотичні оцінки точності цих алгоритмів, що дозволяє встановити потенційно можливий виграш від їх використання.

4. Вперше запропоновано використання в задачах прогнозування комбінованого функціоналу, що складається з критерію четвертого ступеня та модульного, та градієнтна процедура його мінімізації. Досліджено умови її збіжності в середньому і середньоквадратичному. Отримано аналітичні оцінки неасимптотичних та асимптотичних значень помилки оцінювання параметрів і точності ідентифікації моделі. що забезпечило отримання робастних оцінок шуканих параметрів мережі.

5. Отримав подальший розвиток нейромережевий метод прогнозування з використанням рекурентної форми l -крокової проекційної процедури навчання, що дозволило підвищити швидкість навчання ШНМ.

6. Удосконалено метод вибору оптимальних значень параметрів процедур навчання, що дозволило забезпечити максимальну швидкість збіжності процедур та їх стійкість. а також дає можливість досліднику при вирішенні практичних завдань попередньо оцінити можливості процедур, що використовуються, та ефективність їх застосування.

7. Отримав подальший розвиток метод факторизації багатокрокових процедур навчання, що дозволило поліпшити їх обчислювальні властивості та підвищити стійкість.

8. Здійснено імітаційне моделювання нейромережевих методів прогнозування нестационарних часових рядів, на основі результатів яких розроблена нейромережева система прогнозування, що дозволяє збільшити точність прогнозування і підвищити її ефективність.

9. Запропонована і програмно реалізована нейромережева система прогнозування параметрів технологічних процесів виробництва кальцинованої соди, що дозволяє збільшити точність прогнозування та оперативно корегувати керуючі дії з метою дотримання технологічних режимів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Rudenko O. Analysis of convergence of adaptive single-step algorithms for the identification of non-stationary objects / O. Rudenko, O. Bezsonov, V. Lebediev, A. Romanyk // Eastern-European Journal of Enterprise Technologies. 2019. – № 1/4 (97). – P. 6–14. (Належить до категорії А, входить до міжнародної наукометричної бази Scopus).
2. Руденко О.Г. Робастна ідентифікація об'єктів на основі мінімізації комбінованого функціоналу / О.Г. Руденко О.Г., О.О. Безсонов, Н.М. Сердюк, К.О. Олійник, О.С. Романюк // Системи обробки інформації. – 2020. – №1 (160). – С.80–88 (Належить до категорії Б).
3. Бессонов А.А. Факторизованные алгоритмы обучения персептрона в задаче построения нелинейной модели / А.А. Бессонов, К.О. Олейник, С.А. Романюк, О.Г. Руденко, Н.Н. Сердюк / Бионика интеллекта.-2020. – №1(94). – С. 23-29 (Належить до категорії Б).
4. Руденко О.Г. Критерії вибору персептронної моделі для прогнозування: аналіз та практичні рекомендації щодо їх використання / О.Г. Руденко, О.О. Безсонов, О.Г. Лебедев, О.С. Романюк // Бионика интеллекта. –2018. –№ 2 (91). – С.31– 40.
5. Руденко О.Г. Про один алгоритм навчання нейронної мережі в задачі прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.Г. Лебедев, О.С. Романюк // Бионика интеллекта. – 2018. – №1(90). –С.79–83.
6. Руденко О.Г. Робастна ідентифікація об'єктів за наявністю негаусівських завад / О.Г. Руденко О.Г., О.О. Безсонов, Н.М. Сердюк, К.О. Олійник, О.С.Романюк // Бионика интеллекта. –2019. – №2(93). – С.7–12.
7. Руденко О.Г. Нейромережеве прогнозування часових рядів на основі багатосарового персептрона / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Development Management. – 2019. –Vol. 5 – Issue 1. – С.23–34.
8. Олійник К.О. Градієнтні алгоритми навчання згортальних нейронних мереж / К.О. Олійник, О.С. Романюк // Інформаційні технології та

системи: монографія / за заг. ред. В. С. Пономаренка. – Харків. : Видавництво «Стиль-іздат», 2020. – С. 37–51.

9. Rudenko O. Analysis of the influence of interference on the statistical properties of estimates of non-stationary parameters / O. Rudenko, A. Romanyk // Інформаційні системи і технології: монографія / за заг. ред. В. С. Пономаренка. – Харків: ФОП Бровін О.В., 2019. – С.97–107.

10. Романюк О.С. Нейроеволюційний підхід до прогнозування нестационарних послідовностей / О.С. Романюк // Збірник матеріалів І Міжнар. наук.-практ. конф. молодих вчених, аспірантів і студентів; «Сучасні інформаційні технології та системи в управлінні». [Електронний ресурс] 19–20 квітня 2018 р. – Київ : КНЕУ, 2018. – С.204 – 205.

11. Руденко О.Г. Прогнозування нестационарних послідовностей за допомогою коеволюціонуючих штучних нейромереж/ О.Г. Руденко, О.С. Романюк // Збірник наукових праць за матеріалами ІІ міжнародної науково-практичної конференції. «Синергетика, мехатроніка, телематика дорожніх машин і систем у навчальному процесі та науці». 29 травня 2018 р. – Харків, ХНАДУ, 2018. – С.159–161.

12. Руденко О.Г. Дослідження особливостей навчання штучних нейронних мереж в задачі прогнозування / О.Г. Руденко, О.С. Романюк // Матеріали 8-ої міжнародної науково-технічної конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління». 26–27 квітня 2018 р., Полтава, Баку, Харків, Жиліна. – 2018. – С. 32.

13. Rudenko O. Time series prediction based on evolving neural network CMAC / O. Rudenko, O. Bezsonov, A. Romanyk // 3-d International Conference on Computer Algebra and Information Technologies. – Odessa, August 20-25, 2018. – P. 49 –51.

14. Руденко О.Г. Про оцінювання нестационарних параметрів, що описуються марківською моделлю / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Тези доповідей шостої міжнародної науково-практичної

конференції «Проблеми інформатизації». 14 – 16 листопада 2018 р., Черкаси, Баку, Бельско-Бяла, Харків. – 2018. – С.48.

15. Руденко О.Г. Прогнозування часових рядів на основі коеволюційних нейронних мереж / О.Г. Руденко, О.С. Романюк // Тези доповідей міжнародної науково–практичної конференції «Інформаційні технології та системи». 10-11 квітня 2019, Харків. – 2019. – С.15-16.

16. Руденко О.Г. Критерії вибору перцептронної моделі для прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Тези доповідей дев'ятої міжнародної науково-практичної конференції «Сучасні напрями розвитку інформаційно-комунікаційних систем та засобів управління», 11-12 квітня 2019 р., Баку, Харків, Жиліна. – 2019. – С.90 – 91.

17. Руденко О.Г. Про вибір математичної моделі для прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Матеріали XVII міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем». (MSSIS-2019) 20-22 листопада 2019 р., Дніпро. – 2019. – С. 221–222.

18. Олейник К.О. Дослідження спрощених алгоритмів навчання штучних нейронних мереж прямого поширення / К.О. Олейник, О.С. Романюк // Матеріали міжнародної науково-практичної конференції «Інформаційні технології та системи». 9-10 квітня 2020, Харків :ХНЕУ імені Семена Кузнеця. – 2020. – С. 19.

19.Кашьяп Р.Л. Построение динамических стохастических моделей по экспериментальным данным / Р.Л. Кашьяп, А.П. Рао. – М.: Наука, 1983. – 384 с.

20.Chen S. Representations of nonlinear systems; The NARMAX model / S. Chen, S.A. Billings // Int. J. Control. – 1983. – V. 49(3). – P. 1013–1032.

21.Hyndman R. J. Another look at measures of forecast accuracy / R. J. Hyndman, A. B. Koehler. / Int. J. of Forecasting. – 2006. – 22 (4). – Pp. 679–688.

22.Бодянский Е.В. Искусственные нейронные сети: архитектура, обучение, применение / Е.В. Бодянский, О.Г. Руденко. – Харьков: ТЕЛТЕХ,

2004. – 372 с.

23. Box G.E.P. Time Series Analysis: Forecasting and Control / G.E.P. Box, G.M. Jenkins, G.C. Reinsel, G.M. Ljung. – 5th Edition. – Wiley, 2015. – 712 p.

24. Афанасьев В.Н. Анализ временных рядов и прогнозирование. Учебник / В.Н. Афанасьев, М.М. Юзбашев. – М.: Финансы и статистика, 2001. – 228 с.

25. Бідюк П.І. Методи прогнозування / П.І. Бідюк, О.С. Меняйленко, О.В. Половцев. – Луганськ: Альма-матер 1, 2008. – 308 с.

26. Снитюк В.Е. Прогнозирование. Модели, методы, алгоритмы: учебное пособие. / В.Е. Снитюк. – К.: «Маклаут», 2008. – 364 с.

27. Олійник А. О. Інтелектуальний аналіз даних : навчальний посібник / А.О. Олійник, С.О. Субботін, О.О. Олійник. – Запоріжжя: ЗНТУ, 2011. – 271 с.

28. Осовский С. Нейронные сети для обработки информации / С. Осовский. – М.: Финансы и статистика, 2002. – 344 с.

29. Shcherbakov M.V. A Survey of Forecast Error Measures / M.V. Shcherbakov, A. Brebels, N.L. Shcherbakova, A.P. Tyukov, T.A. Janovsky, V. Kamaev. // World Applied Sciences Journal 24 (Information Technologies in Modern Industry, Education & Society). – 2013. – P.171–176.

30. Brockwell P.J. Introduction to Time Series and Forecasting. 3rd ed. / P.J. Brockwell, R.A. Davis R.A. – Springer, 2016. – 428 p.

31. Четыркин Е.Н. Статистические методы прогнозирования. Издание 2-е. / Е.Н. Четыркин. – М.: Статистика, 1977. – 200 с.

32. Gooijer J.G.D. 25 Years of Time Series Forecasting. / J.G.D. Gooijer, R.J. Hyndman / Int. Journal of Forecasting.- 2006.- 22(4).- Pp. 679-688.

33. Айвазян С.А. Прикладная статистика. Исследование зависимостей / С.А. Айвазян, И.С. Енюков, Л.Д. Мешалкин. – М.: Финансы и статистика, 1985. – 432 с.

34. Ивахненко А. Г. Долгосрочное прогнозирование и управление сложными системами / А. Г. Ивахненко. – К.: «Техніка», 1975. – 312 с.

35. Billings S.A. Identification of nonlinear systems: a survey / S.A. Billings

// IEEE Proc. – 1980. – V. 127. – N1. – P. 272–285.

36. Бідюк П. І. Аналіз часових рядів: навчальний посібник / П. І. Бідюк, В.Д. Романенко, О.Л. Тимошук. – Київ: НТУУ «КПІ», 2010. – 230 с.

37. Adhikari R. An Introductory Study on Time series Modeling and Forecasting / R. Adhikari, R. K. Agrawal. – LAP Lambert Academic Publishing, 2013. – 77 p. [Електроний ресурс], – Режим доступу: https://www.researchgate.net/publication/235219651_An_Introductory_Study_on_Time_series_Modeling_and_Forecasting

38. Руденко О.Г. Основы теории искусственных нейронных сетей / О.Г. Руденко, Е.В. Бодянский. Харьков: ТЕЛІТЕХ, 2002. – 317 с.

39. Хайкин С. Нейронные сети: полный курс / С. Хайкин. – М.: Вильямс, 2006. – 1104 с.

40. Patterson D. Artificial Neural Networks, Theory and Application / D. Patterson. – Singapur: Prentice Hall Inc., 1996. – 497 p.

41. Hornik K. Multilayer feedforward networks are universal approximators / K. Hornik, M. Stinchcombe, H. White // Neural Networks. – 1989. – Vol.1. – N2. – P. 359–366.

42. Specht D.F. A general regression neural network / D.F. Specht // IEEE Trans. on Neural Networks. – 1991. – V. 2. – N 6. – P. 568–576.

43. Zell A. Simulation Neuronaler Netze / A. Zell. – Bonn: Addison-Wesley, 1994. – 624 p.

44. Поляк Б.Т. Введение в оптимизацию. – М.: Наука, 1983. – 384 с.

45. Shepherd A. J. Second-order methods for neural networks: Fast and reliable training methods for multi-layer perceptrons / Shepherd A.J. – New York: Springer, 1997. – 146 p.

46. Costa M. A. Improving generalization of MLPs with sliding mode control and the Levenberg-Marquardt algorithm / M.A. Costa, A. Braga, B.R. Menezes // Neurocomp. – 2007. – V. 70. – № 7–9. – P. 1342–1347.

47. Widrow B. Adaptive switching circuits. / B. Widrow, M. Hoff // IRE WESCON Convention Record. Part 4. New York: Institute of Radio Engineers. –

1960. –P. 96–104.

48. Albus J.S. A new approach to manipulator control: the cerebellar model articulation controller (CMAC) // ASME Trans., J. Dynamic Systems, Measurement and Control. – 1975. – Vol. 97. – N3. – P.220–227.

49. Shenker B. Dynamic modeling using neural networks / B. Shenker, M. Agarwal / Int. J. of Systems Science. –1997. – 28(12). – P. 1285–1298.

50. Nelles O. Nonlinear system identification / O. Nelles. – Berlin: Springer – Verlag, 2000. – 785 p.

51. Ham F.M. Principles of Neurocomputing for Science and Engineering / F.M. Ham, I. Kostanic. – N.Y.: Mc Graw-Hill Inc., 2001. – 468 p.

52. Hagan M.T. Training feedforward networks with the Marquardt algorithm / M.T. Hagan, M.B. Menhaj // IEEE Tr. Neural Networks. – 1994. – V. 5(6). – P. 989-993.

53. Sayed A.H. Adaptation, Learning, and Optimization over Networks. – Boston – Delft, 2014. – 501 p.

54. Kaczmarz S. Angenäherte Auflösung von Systemen linearer Gleichungen. Bull. Int. Acad. Polon. Sci. Lett., C 1, Sci. Math. Nat., 1937. – Ser. A. – P. 355–357. English translation: Kaczmarz S. Approximate solution of systems of linear equations. / S. Kaczmarz / Int. J. of Control. – 1993. – 57. – P. 1269–1271.

55. Widrow B. 30 years of adaptive neural networks: perceptron, madaline, and backpropagation/ B. Widrow, M. Lehr // Proc.IEE. – 1990. –vol.78. – 9. – P. 1415–1442.

56. Abbas, O.M. Business Forecasting Among Neural Networks and Statistical Methods, 2017. [Электроний ресурс]. – Режим доступу: <https://www.researchgate.net/publication/319665773>

57. Круглов В.В. Искусственные нейронные сети. Теория и практика / В.В. Круглов, В.В. Борисов. – М.: Горячая линия – Телеком, 2001. – 382с.

58. Чадеев В.М. Определение динамических характеристик объектов в процессе их нормальной эксплуатации для целей самонастройки / В.М. Чадеев // Автоматика и телемеханика. – 1964. – Т. 25. – 9. – С. 1302–1306.

59. Райбман Н.С. Адаптивные модели в системах управления / Н.С. Райбман, В.М. Чадеев. – М.: Сов. Радио, 1966. – 156 с.
60. Лелашвили Ш.Г. Применение одного итерационного метода для анализа многомерных автоматических систем / Лелашвили Ш.Г. // Схемы автоматического управления. – Тбилиси: Мецниереба. – 1965. – С.19–33.
61. Либероль Б.Д. Исследование сходимости одношаговых адаптивных алгоритмов идентификации / Б.Д. Либероль, О.Г. Руденко, А.А. Бессонов // Проблемы управления и информатики. – 2018 – №5. – С.19–32.
62. Либероль Б.Д. Модифицированный алгоритм Качмажа для оценивания параметров нестационарных объектов. / Б.Д. Либероль, О.Г. Руденко, В.А. Тимофеев // Проблемы управления и информации. – 1995. – № 4. – С. 81–89.
63. Ciochina S. An optimized NLMS algorithm for system identification. / S. Ciochină, C. Paleologu, J. Benesty // Signal Processing. – 2016. – 118. – P. 115–121. [Электронный ресурс]. – Режим доступа: <https://www.semanticscholar.org/paper/An-optimized-NLMS-algorithm-for-system-Ciochina-Paleologu/9c8d161c9fd4b86db7f40879167fdea079149455>.
64. Khong A. W. H. Selective-tap adaptive filtering with performance analysis for identification of timevarying systems, / A.W.H. Khong, P.A. Naylor // IEEE Trans. Audio, Speech, and Language Processing. – 2007. – v. 15. – P. 1681–1695.
65. Benesty J. On regularization in adaptive filtering / J. Benesty, C. Paleologu, S. Ciochină // IEEE Trans. Audio, Speech, Language Process. – 2011. – Vol. 19. – P. 1734–1742.
66. Paleologu C. An overview on optimized NLMS algorithms for acoustic echo cancellation / C. Paleologu, S. Ciochină, J. Benesty, S.L. Grant // EURASIP J. Adv. Sig. Proc. – 2015. – 19 p.
67. Bershad N.J. Performance comparison of RLS and LMS algorithms for tracking a first order Markov communications channel / N.J. Bershad,

S. McLaughlin, C. F. N. Cowan // Proc. IEEE Int. Symposium on Circuits and Syst. – 1990. – P. 266–270.

68. Nagumo I. A learning method for system identification. / I. A. Nagumo // IEEE Trans. Autom. Control. – 1967.–AC–12, 3. – №1(17) – P. 282–287.

69. Wagner K. Towards analytical convergence analysis of proportion-type NLMS algorithms / K. Wagner, M. Doroslovacki // Proc. IEEE Int. Conf. Acoustics Speech Signal Processing. – 2008. – P. 3825–3828.

70. Yao X. A review of evolutionary artificial neural networks / X. Yao / Int. J. Intell. Syst. – 1993. – № 8(4). – P. 539–567.

71. Yao X. Evolving Artificial Neural Networks / X. Yao // Proc. of the IEEE. – 1999. – Vol.87. - №9. – P. 1423–1447.

72. Smith R. E. Cooperative versus competitive system elements in coevolutionary systems / R.E. Smith, H. Cribbs // In Proc. of the Fourth International Conference on Simulation of Adaptive Behavior, North Falmouth, MA (USA). – MIT Press, Cambridge, MA. – 1996. – P. 9–13.

73. Chand S. Cooperative coevolution of feed forward neural networks for financial time series problem /S. Chand, R. Chandra // In Int. Joint Conf. on Neural Networks (IJCNN), Beijing, China. – 2014. – P. 202–209.

74. Armstrong J.S. Error measures for generalizing about forecasting methods: empirical comparisons. / J.S. Armstrong, F. Collopy. // Int. J. Forecasting. – 1992. – 8. – P. 69–80.

75. Хьюбер П. Робастность в статистике./ П. Хьюбер. – М.: Мир, 1984. – 304 с.

76. Мудров В.И. Методы обработки измерений. (Квазиподобные оценки) / В.И. Мудров, В.Л. Кушко – М.: Сов. радио, 1976. – 192 с.

77. Смоляк С.А. Устойчивые методы оценивания: статистическая обработка неоднородных совокупностей / С.А. Смоляк, Б.П. Титаренко – М.: Статистика, 1980. – 208 с.

78. Цыпкин Я.З. Огрубленный метод максимального правдоподобия / Я.З. Цыпкин, Б.Т. Поляк // Динамика систем. – Горький, 1977. – Вып. 12. –

C. 22–46.

79. Цыпкин Я.З. Информационная теория идентификации / Я.З. Цыпкин. – М.: Наука. Физматлит, 1995. – 336 с.

80. Hampel F.R. The influence curve and its role in robust estimation / F.R. Hampel // J. Amer. Statist. Assoc. – 1974. – June. – 69. – P. 383–393.

81. Hampel F.R. Robust Statistics. The Approach Based on Influence Functions / F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, W.A. Stahel. – N.Y.: John Wiley and Sons, 1986. – 526 p.

82. Andrews D.F. A robust method for multiple linear regression / D.F. Andrews // Technometrics. – 1974. – 16. – P. 523–531.

83. Mosteller F. Data analysis and regression: a second course in statistics / F. Mosteller, J.W. Tukey. – Addison Wesley, 1977. – 588 p.

84. Deng G. Sequential and adaptive learning algorithms for M-estimation / G. Deng // EURASIP J. Adv. in Signal Proc. – 2008, ID 459586.

85. Хогг Р.В. Введение в помехоустойчивое оценивание / Р.В. Хогг // «Устойчивые стохастические методы оценки данных». – М.: Машиностроение, 1984. – С. 12–25.

86. Хьюбер П.Дж. Помехоустойчивое сглаживание / П.Дж. Хьюбер // «Устойчивые стохастические методы оценки данных». – М.: Машиностроение, 1984. – С. 36–46.

87. Adamczyk T. Application of the Huber and Hampel M-estimation in real estate value modeling. / T. Adamczyk // Geomatics and Environmental Engineering. – 2017. – 11 (1). – 15 p.

88. Руденко О.Г. Робастная идентификация нелинейных объектов / О.Г. Руденко, А.А. Бессонов, С.О. Руденко // Проблеми інформаційних технологій. – 2015. – №1(17) – С. 83–86.

89. Zhao F. A Robust M-Shaped Error Weighted Algorithms for Censored Regression/ F. Zhao, H. Zhao, W. Wang // Circuits, Systems, and Signal Processing. – 2020. – Vol. 39. – P. 324–343. [Электронный ресурс]. – Режим доступа: <https://link.springer.com/journal/34/volumes-and-issues/39-2>

90. Zhang, S. A family of robust m-shaped error weighted least mean square algorithms: performance analysis and echo cancellation application // S. Zhang, W. Zheng, J. Zhang, H. Han // *IEEE Access*. – 2017. – P. 14716–14727
91. Walach E. The least mean fourth (LMF) adaptive algorithm and its family / E. Walach, D. Widrow // *IEEE Trans.* – 1984. – IT 30. – P. 275–283..
92. Bershada, N. Mean-square stability of the normalized least mean fourth algorithm for white Gaussian Inputs / N. Bershada, J.C.M. Bermudez // *Digit. Signal Process.* – 2011. – 21(6). – P. 694–700.
93. Eweda, E. A. New insights into the normalization of the least mean fourth algorithm / E. Eweda, A. Zerguine // *Signal Image Video Process.* – 2013. – 7(2). – P. 255–262.
94. Eweda E. Global stabilization of the least mean fourth algorithm / E. Eweda // *IEEE Trans. Signal Process.* – 2012. – 60(3). – P. 1473–1477.
95. Eweda E. Stochastic analysis of a stable normalized least mean fourth algorithm for adaptive noise canceling with a white Gaussian reference / E. Eweda, N. Bershada // *IEEE Trans. Signal Process.* – 2012. – 60(12). – P. 6235–6244.
96. Hübscher P.I. A mean-square stability analysis of the least mean fourth adaptive algorithm / P.I. Hübscher, J.C.M. Bermudez, V.H. Nascimento // *IEEE Trans. Signal Process.* – 2007. – 55(8). – P. 4018–4028.
97. Chambers J. Least mean mixed-norm adaptive filtering / J. Chambers, O. Tanrikulu, A.G. Constantinides // *Electronics letters*. – 1984. – 30. – 19. – P. 1574–1575.
98. Chambers J. A Robust Mixed-Norm Adaptive Filter Algorithm / J. Chambers, A. Avlonitis // *IEEE Signal Processing Letters*. – 1997. – V. 4. – №2. – P.46-48.
99. Zerguine A. A variable-parameter normalized mixed-norm (VPNMN) adaptive algorithm / A. Zerguine // *EURASIP Journal on Advances in Signal Processing*. – 2012. – №55. [Электронный ресурс]. – Режим доступа: <http://asp.eurasipjournals.com/content/2012/1/55>
100. Sayin M.O. The Krylov-proportionate normalized least mean fourth

approach: Formulation and performance analysis / M.O. Sayin, Y. Yilmaz, A. Demir, S.S. Kozat // *Signal Process.* – 2015. – 109. – P. 1–13.

101. Mathews V.J. Improved Convergence Analysis of Stochastic Gradient Adaptive Filters Using the Sign Algorithm / V.J. Mathews, S.H. Cho // *IEEE Trans. on Acoustic, Speech, and Signal Processing*, 1987. – V.35. – №4. – P. 450–454.

102. Goodwin G.C. A Globally Convergent Adaptive Predictor / G.C. Goodwin, P.J. Ramage, P.E. Caines / *Automatica.* – 1981. – Vol. 17. – №1. – P.135–140.

103. Вазан М. Стохастическая аппроксимация / М. Вазан. – М.: Мир, 1972. – 296 с.

104. Безсонов О.О. Оптимізація процесу навчання штучних нейронних мереж / О.О. Безсонов, К.О.Олійник, О.С. Романюк, О.Г.Руденко // Матеріали статей 9-ї Міжнародної науково-технічної конференції «Інформаційні системи та технології ICT-2020», Харків, 17-20 листопада 2020 р. –Х.: Друкарня Мадрид, 2020. – С.300–302. [Електроний ресурс]. – Режим доступу: <https://openarchive.nure.ua/handle/document/16177>

105. Hogg R.V. Statistical robustness: one view of its use in applications / R.V. Hogg // *Amer. Statist.* –1979. – 33. – P.108-111.

106. Мартин Р.Д. Устойчивый авторегрессионный анализ временных рядов / Р.Д. Мартин // *Устойчивые стохастические методы оценки данных.* – М.: Машиностроение, 1984. – С.121–146.

107. Martin R.D. Determining the character of time series outliers / R.D. Martin, J.E. Zeh // *Proc. Amer. Statist. Assoc. «Business and Economics» Section.* – 1977.

108. Martin R.D. Robust estimation of signal amplitude / R.D. Martin // *IEEE Tr. Inf. Theory.* – 1972. –18. – P. 596-606.

109. Martin R.D. Robust estimation via stochastic approximation / R.D. Martin, C.J. Masreliez // *IEEE Trans. Inf. Theory.* – 1975. –21. – P. 263–271.

110. Benesty J. On Regularization in Adaptive Filtering / J. Benesty, C. Paleologu, S. Ciochina // *IEEE Trans. on Audio, Speech, and Language*

Processing. – 2011. – Vol. 19. – 6. – P. 1734–1742.

111. Демиденко Е.З. Линейная и нелинейная регрессии / Е.З. Демиденко. – М.: Финансы и статистика, 1981. – 304 с.

112. Лоусон И. Численное решение задач метода наименьших квадратов. Пер. с англ. / И. Лоусон, Р. Хенсон Р. – М.: Наука, 1986. – 232с.

113. Браммер К. Фильтр Калмана-Бьюси. Пер. с нем. / К. Браммер, Г. Зиффлинг. – М.: Наука, 1982. – 200 с.

114. Bierman J. Factorization methods for discrete sequential estimation / J. Bierman. – N.Y.: Academic press, 1977. – 241 p.

115. Householder A.S. The theory of matrices in numerical analysis / A.S. Householder. – N.Y.: Blasedell Publ. Co., 1964. – 236 p.

116. Forsythe G.E. Today's methods of linear algebra / G.E. Forsythe // SIAM Rev. – 1967. – Vol.9. – P.489–515.

117. Нейронные сети. STATISTICA Neural Networks. Пер. с англ. – М.: Горячая линия – Телеком, 2001. – 122 с.

118. Медведев В.С. Нейронные сети. MATLAB 6 / В.С. Медведев, В.Г. Потемкин. – М.: Диалог. – МИФИ, 2002. – 496 с.

119. Eurostat (n.d.). [Электроний ресурс]. – Режим доступа: http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=avia_paoc&lang=en

120. Зайцев И.Д. Теория и методы авторизованного проектирования химических производств / И.Д. Зайцев. – К.: Наукова думка, 1981. – 308 с.

121. Растринин Л.А. Экстраполяционные методы проектирования и управления / Л.А.Растринин, Ю.П. Пономарев. – М.: Машиностроение, 1986. – 120 с.

122. Montgomery D.C. Introduction to time series analysis and forecasting / D.C. Montgomery, C.L. Jennings, M. Kulahci. – New Jersey: John Wiley & Sons. Inc., 2008. – 469 p.

123. Бокс Дж. Анализ временных рядов прогноз и управление / Дж. Бокс, Г. Дженкинс. – М.: Мир, 1974. – Кн. 1. – 406 с. – Кн. 2. – 197 с.

124. Bowerman B. Forecasting: methods and applications. 4th ed. /

B. Bowerman, R. O'Connell, A. Koehler. – Belmont, CA.: Thomson Brooks, 2005. – 384 p.

125. Hyndman R.J. Forecasting: principles and practice. / R.J. Hyndman, G. Athanasopoulos // OTexts, Melbourne, Australia, 2017. [Электронный ресурс]: <http://otexts.com/fpp>.

126. Арженовский С.В. Статистические методы прогнозирования Учебное пособие / С.В. Арженовский, И.Н. Молчанов. – Ростов-на-Дону, 2001. – 74 с.

127. Андерсон Т. Статистический анализ временных рядов /Т. Андерсон. – М.: Мир, 1976. – 756 с.

128. Ruder S. An overview of gradient descent optimization algorithms. / S. Ruder / arXiv:1609.04747v2 [cs.LG] 15 Jun 2017

129. Oppermann A. [Optimization Algorithms in Deep Learning](#) / A. Oppermann [Электронный ресурс]. – Режим доступа: <https://www.deeplearning-academy.com/p/ai-wiki-optimization-algorithms>. – Oct. 2019

130. Le Roux N. A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets / N. Le Roux, M. Schmidtmark, F. Bach [Электронный ресурс]. – Режим доступа: arXiv:1202.6258v4 [math.OC] 11 Mar 2013

131. Johnson R. Accelerating stochastic gradient descent using predictive variance reduction / R. Johnson, T. Zhang // In Advances in Neural Information Processing Systems. – 2013. – P. 315–323.

132. Defazio A., SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives / A. Defazio, F. Bach, S. Lacoste-Julien. [Электронный ресурс]. – Режим доступа: arXiv:1407.0202v3 [cs.LG] 16 Dec 2014.

133. Нестеров Ю.Е. Метод минимизации выпуклых функций со скоростью сходимости / Ю.Е. Нестеров // Докл. АН СССР. – 1983. – Т. 269. – Вып. 3. – С. 543–547.

134. Duchi J. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization / J. Duchi, E. Hazan, Y. Singer // Journal of Machine Learning Research. – 2011. – 12. – P. 2121–2159. [Электроний ресурс]. – Режим доступа: <http://jmlr.org/papers/v12/duchi11a.html>

135. Tieleman A. Lecture 6.5 – «rmsprop: Divide the gradient by a running average of its recent magnitude» / A. Tieleman, T. Hinton // COURSERA: Neural Networks Mach. Learn. – 2012. – 4. – P. 26–31.

136. Zeiler M. D. ADADELTA: An Adaptive Learning Rate Method / M.D. Zeiler. – 2012. [Электроний ресурс]. – Режим доступа: <http://arxiv.org/abs/1212.5701>

137. Kingma D. P. Adam: a Method for Stochastic Optimization / D.P. Kingma, J.L. Ba //International Conference on Learning Representations. – 2015. – P. 1–13.

138. Jiang Z. On Higher-order Moments in Adam / Z. Jiang, A. Balu, S.Y. Tan, Y.M. Lee, C.S. Sarkar. [Электроний ресурс]. – Режим доступа: arXiv:1910.06878v1 [cs.LG] 15 Oct 2019

139. Dozat. T. Incorporating Nesterov Momentum into Adam. / Dozat // ICLR Workshop. – 2016. – 1. – P. 2013–2016.

140. Price R. A useful theorem for nonlinear devices having Gaussian inputs / R. Price // IREN Trans. Inform Theory. – 1958. – 4. – P. 69–72.

Додаток А

Критерії точності прогнозування [4]

Критерії, засновані на абсолютних помилках

Mean Forecast Error, MFE:

$$MFE = \frac{1}{N} \sum_{i=1}^N e_i. \quad (\text{A.1})$$

Mean Absolute Error, MAE:

$$MAE = \frac{1}{N} \sum_{i=1}^N |e_i| = \text{mean } |e_i| \quad (\text{A.2})$$

визначає величину помилки прогнозу в цілому.

Як і MFE, дана метрика також залежить від шкали вимірювання і використання перетворень вихідних даних.

У метрики MAE є два недоліки, які проявляються тільки в разі аналізу прогнозної точності моделей:

- 1) її значення важко інтерпретувати;
- 2) за допомогою цієї метрики можна проводити порівняння тільки між моделями з одного ряду даних. На практиці ж часто стоїть завдання зрозуміти, як себе веде модель з кількох рядів даних в порівнянні з іншими моделями. Однак отримані з різних рядів значення MAE складати один з одним не можна.

Mean Squared Error, MSE:

$$MSE = \frac{1}{N} \sum_{i=1}^N e_i^2. \quad (\text{A.3})$$

Дана метрика враховує наявність рідкісних екстремальних помилок.

Sum of Squared Error, SSE:

$$SSE = \sum_{i=1}^N e_i^2. \quad (A.4)$$

Дана метрика має схожі з MSE характеристики.

Signed Mean Squared Error, SMSE:

$$SMSE = \frac{1}{N} \sum_{i=1}^N \left(\frac{e_i}{|e_i|} \right) e_i^2. \quad (A.5)$$

Дана метрика залежить від шкали вимірювання і перетворень вхідних даних і нормалізує отримане значення MSE.

Root Mean Squared Error, RMSE:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N e_i^2}. \quad (A.6)$$

Дана метрика має схожі з MSE характеристиками.

Fourth Root Mean Quadrupled Error, FRMQE, R4MS4E [21/35]:

$$FRMQE = R4MS4E = \sqrt[4]{\sum_{i=1}^N e_i^4}. \quad (A.7)$$

Normilized Mean Squared Error (NMSE):

$$NMSE = \frac{MSE}{\sigma^2} = \frac{1}{N\sigma^2} \sum_{i=1}^N e_i^2. \quad (A.8)$$

Дана метрика нормалізує отримане значення MSE.

Median Absolute Error, MdAE:

$$MdAE = \underset{i=1, N}{\text{median}} |e_i| \quad (A.9)$$

Standard deviation (SD):

$$SD = \sum_{i=1}^N \sqrt{\frac{(y_i - ME)^2}{N}}. \quad (\text{A.10})$$

Абсолютні вимірювання помилок мають наступні недоліки:

1) залежність від масштабу [31] (тому якщо задача прогнозу включає в себе об'єкти з різними масштабами або величинами, абсолютна помилка в якості метрики не може застосовуватися);

2) високий вплив викидів в даних на оцінку ефективності прогнозу [38/43] (тому якщо дані містять викиди, то абсолютні значення помилок є неефективними).

RMSE, MSE мають низьку надійність (робастність): результати можуть бути різними залежно від різної частки даних [41].

Критерії, засновані на обчисленні процентних відношень

Mean Absolute Percentage Error, MAPE:

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{e_i}{y_i} \right| \times 100. \quad (\text{A.11})$$

Як видно, MAPE не має розмірності, з дуже простою інтерпретацією. Її можна вимірювати в частках або відсотках, легко складати з різних рядів. Можна навіть розрахувати MAPE і вивчити його розподіл, використовуючи інструменти статистичного аналізу.

Дана метрика застосовується для часових рядів, фактичні значення яких значно більше 1 (наприклад, оцінки помилки прогнозування енергоспоживання майже у всіх статтях наводяться як значення MAPE). Вона по-різному ставиться до позитивних і негативних помилок: жорсткіше до випадків завищених прогнозів, ніж занижених. У випадку з агрегуванням прогнозів по різних значеннях це призводить до викривлення – дослідник не отримує достовірної інформації про якість його прогнозу.

В результаті детального порівняння MAPE з іншими метриками при виборі моделі прогнозування в роботі [23] показано, що MAPE є ненадійним

критерієм при використанні цієї метрики для вибору між конкуруючими прогнозними моделями. Зокрема, існує висока ймовірність того, що MAPE вибере модель, яка дає оцінку нижче середнього (тобто «недооцінює») моделі, що прогнозує середнє значення.

Аналогічний результат отримано в роботі [20], де показано, що MAPE систематично і послідовно високо оцінює моделі, які дають занадто оптимістичний прогноз. Через поділ на фактичні значення ця метрика виявилася чутливою до масштабу, і при малому значенні фактичного у значення MAPE може бути дуже великим.

Mean Arctangent Absolute Percentage Error (MAAPE) [36]:

$$MAAPE = \frac{1}{N} \sum_{i=1}^N AAPE_i, \quad (A.12)$$

$$\text{де } AAPE_i = \arctg \left(\left| \frac{y_i - \hat{y}_i}{y_i} \right| \right).$$

Ця метрика дозволяє дещо поліпшити властивості MAPE.

Mean Percentage Error, MPE:

$$MPE = \frac{1}{N} \sum_{i=1}^N \left(\frac{e_i}{y_i} \right) \times 100 \quad (A.13)$$

покликана оцінювати не точність прогнозу, а його зміщення.

Можна помітити, що вона схожа на MAPE, має ті ж перевагами і недоліками, що і MAPE, але відрізняється лише відсутністю модулів в чисельнику. MPE показує відсоток зсуву прогнозу. Позитивні значення цього критерію вказують на систематичне заниження прогнозу, а негативні – на завищення. Значення MPE так само, як і MAPE може вимірюватися у відсотках.

Дана метрика є досить хорошим показником повної помилки прогнозу, але вона не настільки інтуїтивно зрозуміла, на відміну від інших метрик.

Median Absolute Percentage Error, MdAPE:

$$MdAPE = \underset{i=1,N}{\text{median}}|\bar{e}_i|, \quad (\text{A.14})$$

де $\bar{e}_i = e_i / y_i$.

Root Median Square Percentage Error, RMSPE:

$$RMSPE = \sqrt{\underset{i=1,N}{\text{median}}(\bar{e}_i)^2}. \quad (\text{A.15})$$

Для даної групи критеріїв характерними є наступні недоліки:

- 1) з'являється поділ на нуль, коли фактичне значення дорівнює нулю;
- 2) несиметричність – значення помилки відрізняються, якщо прогнозоване значення більше або менше фактичного [22,121];
- 3) викиди значно впливають на результат, особливо, якщо відхилення має значно більшу величину, ніж максимальне значення «нормальних» випадків [38];
- 4) обчислення помилки є необ'єктивним, що може призвести до неправильної оцінки моделі прогнозування [30].

Симетричні критерії

Symmetric MAPE, SMAPE:

$$SMAPE = \frac{1}{N} \sum_{i=1}^N \frac{2|e_i|}{|y_i| + |\hat{y}_i|} \quad (\text{A.16})$$

використовує середнє арифметичне фактичного і прогнозованого значень, застосовується для часових рядів, фактичні значення яких значно більше 1.

Незважаючи на те, що метрика повинна б бути симетричною, але такою вона до кінця не є. SMAPE демонструє зміщення тільки цього разу в бік завищених прогнозів: завищені прогнози призводять до меншої помилки, ніж занижені.

Symmetric Median APE, SMdAPE:

$$SMdAPE = \underset{i=1,N}{\text{median}} \left| \frac{y_i - \hat{y}_i}{(y_i + \hat{y}_i)} \right|. \quad (\text{A.17})$$

Modified sMAPE, msMAPE:

$$msMAPE = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{\frac{|y_i| + |\hat{y}_i|}{2} + s_i} \quad (\text{A.18})$$

$$s_i = \frac{1}{i-1} \sum_{k=1}^{i-1} |y_i - \bar{y}_i|, \bar{y}_i = \frac{1}{i-1} \sum_{k=1}^{i-1} y_k. \quad (\text{A.19})$$

Дана група критеріїв має такі недоліки:

- 1) незважаючи на назву, вони є несиметричними [34] ;
- 2) якщо фактичне значення дорівнює прогнозованому, але з протилежним знаком, або обидва ці значення дорівнюють нулю, тоді відбувається поділ на нуль;
- 3) на значення критеріїв суттєво впливають викиди аналогічно з відсотковими помилками;
- 4) існує проблема інтерпретації результатів.

Відносні критерії

Mean Relative Absolute Error, MRAE:

$$MRAE = \text{mean} \frac{|e_i|}{(y_i - \hat{y}_i^*)}, \quad (\text{A.20})$$

де \hat{y}_i^* - прогнозна величина базової (еталонної) моделі (на практиці, як правило, обирають $\hat{y}_i^* = y_{i-1}$).

Median Relative Absolute Error, MdRAE:

$$MdRAE = \underset{i=1,N}{\text{median}} \left| \frac{|e_i|}{(y_i - \hat{y}_i^*)} \right|. \quad (\text{A.21})$$

Geometric Mean Relative Absolute Error, GMRAE [25]:

$$GMRAE = \sqrt{\prod_{i=1}^m \left| \frac{y_i - \hat{y}_i}{y_i - \tilde{y}_i} \right|} \quad (A.22)$$

де $y_i - \tilde{y}_i$ - помилка за деякою другою моделлю.

GMRAE показує, у скільки разів побудована модель виявилася гірше (або краще) за обрану для порівняння (модель-бенчмарк). Якщо $GMRAE > 1$, то обрана модель виявилася менш точною, в протилежній ситуації – більш точною. Однак питання про вибір другої моделі залишається. GMRAE не може використовуватися в ситуаціях, коли хоча б одна з помилок однієї з моделей виявилася рівною нулю – в цьому випадку GMRAE стає рівним або нулю або нескінченності.

Виходячи з наведених формул, поділ на нульову помилку відбувається, якщо прогностичне значення отримане за допомогою моделі, дорівнює дійсному значенню.

Якщо обрана наївна модель, то поділ на нуль помилка виникає у випадку безперервної послідовності ідентичні значення часового ряду.

Масштабовані помилки

Mean Absolute Scaled Error, MASE:

$$MASE = \frac{T-1}{N} \frac{\sum_{i=1}^N |e_{T+i}|}{\sum_{j=2}^T |y_j - y_{j-1}|} \quad (A.23)$$

Сума в чисельнику, відповідає тестовій вибірці, а в знаменнику – навчальній. Друга фактично являє собою середню абсолютну помилку прогнозу за методом Naive. Вона ж відповідає середньому абсолютному відхиленню ряду в перших різницях. Ця величина, по суті, показує, наскільки навчальна вибірка передбачувана і може дорівнювати нулю тільки в тому випадку, коли всі значення в навчальній вибірці рівні один одному, що

відповідає відсутності будь-яких змін в ряді даних, тобто ситуації, на практиці майже неможливій.

Слід зазначити, що модуль в знаменнику () потрібен для ситуацій, коли при прогнозуванні присутні як позитивні, так і негативні значення змінних, середня величина яких може бути близька до нуля. Якщо в такій ситуації брати просто середнє, виникає все та ж проблема масштабу.

sMAE – scaled MAE, іноді позначається як MAE / mean:

$$sMAE = \frac{1}{N} \sum_{i=1}^N \frac{|e_i|}{y_i}. \quad (\text{A.24})$$

Зауважимо, знаменник sMAE може дорівнювати нулю тільки в одному випадку – якщо всі фактичні значення в навчальній вибірці дорівнюють нулю. Критерій sMAE – практично ідеальний показник. У нього є лише один невеликий недолік – він не обмежений зверху. У загальному випадку значення sMAE буде показувати, який відсоток в середньому становлять помилки прогнозу. Ця величина буде незміщеною (в порівнянні з MAPE і SMAPE) і може легко складатися з іншими такими ж величинами.

Root Mean Square Scaled Error, RMSSE:

$$RMSSE = \sqrt{\text{mean}\left(\frac{|e_i|}{\frac{1}{T-1} \sum_{j=2}^T |y_j - y_{j-1}|}\right)^2}. \quad (\text{A.25})$$

Відносні критерії є симетричними та стійкими до викидів, однак мають наступні недоліки:

-якщо у прогнозованому горизонті реальні значення є рівними, тоді відбувається поділ на нуль.

- можливе невелике зміщення оцінок [122].

Критерії, засновані на обчисленні відносних величин

Relative Mean Absolute Error, RelMAE:

$$RelMAE = \frac{MAE}{MAE^*}, \quad (A.26)$$

де MAE і MAE * середні абсолютні помилки для аналізованої моделі прогнозування та базової (еталонної) моделі відповідно, які обчислюють за формулою (2).

Relative Root Mean Squared Error, RRMSE:

$$RRMSE = \frac{RMSE}{RMSE^*}. \quad (A.27)$$

Log mean squared error ratio, LMR:

$$LMR = \log\left(\frac{RMSE}{RMSE^*}\right). \quad (A.28)$$

Ці критерії визначають кількість випадків, коли аналізована модель прогнозування перевершує базову, але не оцінюють значення різниці.

Недоліками цих критеріїв є можливість поділу на нульову помилку, якщо базова прогнозна помилка дорівнює нулю.

Інші критерії

Direction accuracy, DA) [25]:

$$DA = \frac{1}{N} \sum_{i=1}^N a_i, \quad (A.29)$$

$$\text{де } a_i = \begin{cases} 1, & \text{якщо } (y_{i+1} - y_i)(\hat{y}_{i+1} - y_i) > 0, \\ 0 & \text{в іншому випадку.} \end{cases}$$

Modified direction accuracy (MDA) [25]:

$$MDA = \begin{cases} \frac{\sum_{i=0}^{N-1} D_i}{N-1}, & \\ \end{cases} \quad (A.30)$$

$$D_i = (A_i - F_i)^2$$

$$A_i = 1, \text{ якщо } y_{i+1} - y_i \leq 0,$$

$$A_i = 0, \text{ якщо } y_{i+1} - y_i > 0$$

$$F_i = 0, \text{ якщо } \hat{y}_{i+1} - \hat{y}_i \leq 0,$$

$$F_i = 0, \text{ якщо } \hat{y}_{i+1} - \hat{y}_i > 0.$$

Directional symmetry, *DS* [123]:

$$DS = \frac{100}{N} \sum_{i=1}^N d_i, \quad (\text{A.31})$$

$$\text{де } d_i = \begin{cases} 1, & \text{якщо } (y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}) \geq 0, \\ 0 & \text{в іншому випадку.} \end{cases}$$

DS оцінює точність прогнозованого напрямку.

Weighted directional symmetry, *WDS*) [123]:

$$WDS = \frac{\sum_{i=1}^N d_i |y_i - \hat{y}_i|}{\sum_{i=1}^N d'_i |y_i - \hat{y}_i|}, \quad (\text{A.32})$$

$$\text{де } d'_i = \begin{cases} 1(y_i - y_{i-1})(\hat{y}_i - \hat{y}_{i-1}) \geq 0, \\ 0 \text{ в іншому випадку.} \end{cases}$$

WDS вимірює величини та напрямки похибки прогнозування. При цьому штрафуються помилки, пов'язані з неправильно спрогнозованим напрямком і заохочується правильно передбачений напрямок. Чим менше значення *WDS*, тим краще прогнозування з точки зору його точності та напрямку.

Sign Criteria:

$$Sign = \frac{1}{N} \sum_{i=1}^N z_i, \quad (\text{A.33})$$

$$\text{де } z_i = \begin{cases} 1, & \text{якщо } (y_{i+1})(\hat{y}_{i+1}) > 0, \\ 0 & \text{в іншому випадку.} \end{cases}$$

Akaike Information Criteria, AIC [27]:

$$AIC = \log MSE + \frac{2q}{N}, \quad (\text{A.34})$$

де q -загальна кількість параметрів моделі; N - кількість спостережень.

Особливості AIC:

- 1) він залежить від припущення нормальності даних;
- 2) цей критерій не є спроможним; тобто при збільшенні даних він схильний до так званої перепідгонці;
- 3) критерій AIC в загальному випадку є асимптотично ефективним.

Bayesian Information Criteria, BIC [29]:

$$BIC = \log MSE + q \frac{\log N}{N}. \quad (\text{A.35})$$

Особливостями даного критерію є наступне:

- 1) цей критерій вводить більш строгі штрафні характеристики, ніж AIC; тобто BIC «штрафує» моделі з великою кількістю параметрів більшою мірою, ніж це робить AIC;
- 2) критерій BIC є строго слухним критерієм: якщо дані в дійсності згенеровані моделлю, адекватною реальній, то BIC вибере справжню модель з одиничною ймовірністю з урахуванням того, що розмір вхідних даних буде прагнути до нескінченності;
- 3) BIC менш ефективний, ніж AIC.

KIC - Kullback Information Criterion [74]

$$KIC = N \ln \left[\frac{1}{N} \sum_{i=1}^N e_i^2 \right] + 3S \ln N.$$

Цей критерій є менш використовуваним за інші.

Критерій Хеннана-Куїнна (HQ) [40]

$$HQ = N \ln \left[\frac{1}{N} \sum_{i=1}^N e_i^2 \right] + 2S \ln(\ln N). \quad (\text{A.36})$$

Слід зазначити, що в окремому випадку нормального розподілу критерій *AIC* приймає вид

$$AIC(S) = RSS(S) + 2S, \quad (\text{A.37})$$

званому критерієм Акаїке-Меллоуса.

Weighted Information Criteria, *WIC*.

В роботі [41] було описано спроби поєднання статистичних та інформаційних критеріїв в одному зваженому критерії (Weighted Information Criterion, *WIC*)

$$WIC = 0,1(AIC + BIC) + 0,2(RMSE + MAPE) + 0,2((1 - DA) + MDA). \quad (\text{A.38})$$

Слід зазначити, що коефіцієнти, які використовувалися в цьому критерії задавалися евристично не зрозуміло з яких міркувань.

Adaptive Weighted Information Criterion, *AWIC*.

В роботі [41] замість фіксованих використовувалися коефіцієнти, які обчислювалися шляхом вирішення деякої оптимізаційної задачі (при цьому критерії *AIC* та *BIC* бралися з вагою 0,1). Відповідний критерій було названо адаптивним (Adaptive WIC, *AWIC*)

$$AWIC = 0,1(AIC + BIC) + w_1 RMSE + w_2 MAPE + w_3(1 - DA) + w_4 MD. \quad (\text{A.39})$$

Слід, однак, зазначити, що при цьому

- 1) не визначено критерій для оцінки вагових коефіцієнтів *AIC* и *BIC*;
- 2) немає жодних вказівок або рекомендацій щодо використання конкретних статистичних критеріїв, які б враховували поведінку часового ряду;
- 3) евристичні методи не були успішними в оцінці вагів для комбінованих методів;

4) цей критерій не враховує характеристик часових рядів.

U-статистика Тейла

Ця важлива міра розраховується за формулою [42]:

$$U = \frac{\sqrt{\sum_{i=1}^N (FPE_i - APE_i)^2}}{\sqrt{APE_i^2}}, \quad (\text{A.40})$$

де $FPE_i = \frac{\hat{y}_i - y_i}{y_i}$ – відносна зміна прогнозних даних; $APE_i = \frac{y_{i+1} - y_i}{y_i}$

-відносна зміна фактичних даних, або

$$U = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N e_i^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N \hat{y}_i^2} \sqrt{\frac{1}{N} \sum_{i=1}^N y_i^2}}. \quad (\text{A.41})$$

Цей показник дозволяє здійснювати порівняння формальних методів прогнозування з найпростішими («найвними») методами.

Як видно з виразу (A.10), шляхом піднесення відносних похибок до квадрата ліквідується ефект взаємної компенсації помилок і забезпечується ефект, коли значні помилки мають більшу вагу, ніж дрібні.

Як базис для порівняння використовуються два прості методи:

а) метод «випадкового блукання» (random walk) – це математична модель процесу випадкових змін – кроків у дискретні моменти часу. При цьому передбачається, що зміна на кожному кроці не залежить від попередніх змін і від часу;

б) метод урахування сезонності – цей метод розглядає можливість наявності в часових рядах сезонності. Оскільки високий відсоток флуктуацій даних у межах часового ряду зумовлений саме сезонністю, цей метод може мати кращі результати, ніж перший, при цьому залишаючись достатньо простим.

U-статистика Тейла коливається в межах діапазону $[0; \infty]$ і є простою для інтерпретації:

- якщо $U = 1$, найпростіші базові методи є настільки точними, як і формальний метод прогнозування, що застосовується;

- якщо $U < 1$, формальний метод прогнозування, який застосовується, є кращим від базових методів. Наближення значення U-статистики до 0 свідчить про підвищення якості прогнозування порівняно з базовими методами;

- якщо $U > 1$, немає сенсу використовувати відповідний формальний метод прогнозування, оскільки базові методи дають кращі результати.

U-статистика Тейла може використовуватися при оцінці якості прогнозів, отриманих на основі різних методів та моделей. Може використовуватися при оцінці якості прогнозів, отриманих на основі різних методів та моделей.

Недоліки:

- 1) є залежною від зміни масштабу та перетворення даних;
- 2) її значення суттєво залежать від дисперсії ряду, що прогнозується (при прогнозуванні процесів з низькою дисперсією при явно невдалому прогнозі U-статистика може мати мале значення, що обумовлюється низькою варіацією признаку).

Модифікована U-статистика Тейла [42]:

$$U = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N e_i^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N \hat{y}_i^2} \sqrt{\frac{1}{N} \sum_{i=1}^N y_i^2}} \frac{\sigma_y^2}{\sigma_{\hat{y}}^2}, \text{ якщо } \sigma_y^2 > \sigma_{\hat{y}}^2,$$

$$U = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N e_i^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N \hat{y}_i^2} \sqrt{\frac{1}{N} \sum_{i=1}^N y_i^2}} \frac{\sigma_{\hat{y}}^2}{\sigma_y^2}, \text{ якщо } \sigma_y^2 < \sigma_{\hat{y}}^2$$

має за мету ліквідувати недолік традиційної статистики.

Крім того, існує ще достатньо велика кількість метрик, що представляють собою деякі модифікації розглянутих вище, наприклад, середня відносна абсолютна помилка (Mean Relative Absolute Error, MRAE), медіанна абсолютна процентна помилка (Median Absolute Percentage Error, MdAPE), медіанна відносна абсолютна помилка (Median Relative Absolute Error, MdRAE), процентна середньоквадратична помилка (Root Mean Square Percentage Error, RMSPE), відносна середня абсолютна помилка (Relative Mean Absolute Error, RelMAE), відносна середньоквадратична помилка (Relative Root Mean Square Error, RelRMSE), симетрична медіанна APE (Symmetric Median Mean Absolute Percentage Error, sMdAPE) та ін. [124-127].

Додаток Б

Навчання на основі стохастичного градієнта

Для навчання згорткових нейронних мереж може застосовуватися як стандартний метод зворотного поширення помилки, так і його різні модифікації. В основі цього методу лежить алгоритм стохастичного градієнтного спуску (Stochastic Gradient Descent).

Стохастичний градієнтний спуск (SGD) і його варіанти є найбільш широко вивченими алгоритмами для оптимізації в задачах машинного навчання і стохастичної апроксимації. В літературі по обробці сигналів SGD носить назву Least MeanSquares (LMS) алгоритм.

Звичайний градієнтний спуск описується співвідношеннями

$$\theta(k+1) = \theta(k) - \eta \nabla_{\theta} J(\theta(k)), \quad (\text{Б.1})$$

де θ - параметри мережі,

$J(\theta(k))$ - цільова функція (функція втрат);

η - параметр швидкості навчання (learning rate).

Збіжність алгоритму (Б.1) в загальному випадку не гарантується, проте встановлено [103], що в разі опуклої функції $J(\theta(k))$ і при виконанні наступних умов:

$$\lim_{k \rightarrow \infty} \eta(k) = 0; \quad \sum_{k=1}^{\infty} \eta(k) = \infty; \quad \sum_{k=1}^{\infty} \eta^2(k) < \infty. \quad (\text{Б.2})$$

процес градієнтного спуску буде збігатися.

Можливо два основні підходи до реалізації градієнтного спуску:

– пакетний (batch), коли на кожній ітерації навчальна вибірка проглядається цілком, і тільки після цього змінюється. Це вимагає великих обчислювальних витрат;

– стохастичний (stochastic/online), коли на кожній ітерації алгоритму з навчальної вибірки якимось (випадковим) чином обирається тільки один об'єкт. Таким чином, вектор θ налаштовується на кожний знову обираємий об'єкт.

Даному алгоритму притаманні такі недоліки:

– застрягання в локальних мінімумах або сідлових точках мінімізованого функціоналу;

– повільна збіжність внаслідок складного ландшафту цільової функції, коли плато чергуються з регіонами сильної нелінійності (похідна на плато практично дорівнює нулю, а раптовий обрив, навпаки, може занадто змінити оцінку параметра);

– деякі параметри оновлюються значно рідше інших, особливо коли в даних зустрічаються інформативні, але рідкісні ознаки, що погано позначається на нюансах узагальнюючого правила мережі. З іншого боку, надання занадто великої значимості взагалі всім рідко зустрічається ознаками може привести до перенавчання;

– занадто малі значення параметра η призводять до повільної збіжності і застрягання в локальних мінімумах, занадто великі - до «проскакування» вузьких глобальних мінімумів або зовсім розбіжність.

Використання методів другого порядку, розглянутих вище, вимагає обчислення гессіан - матриці похідних по кожній парі параметрів, а, для методу Ньютона-ще й зворотний до неї, тобто реалізація даних методів пов'язана зі значними обчислювальними витратами.

У зв'язку з цим на практиці широкого поширення набули методи, засновані на методі стохастичного градієнта і володіють в порівнянні з ним цілу низку переваг [128,129].

Розглянемо ці методи більш докладно.

Стохастичний середній градієнт (SAG) [128,129].

Метод стохастичного середнього градієнта (SAG) - це метод зменшення дисперсії, запропонований для поліпшення швидкості збіжності. Ітерації SAG приймають форму

$$\theta(k+1) = \theta(k) - \frac{\alpha(k)}{n} \sum_{i=1}^n y_i(k),$$

$$y_i = \begin{cases} \nabla_{\theta} J_i(\theta(k)) & \text{if } i = i_k, \\ y_{i-1} & \text{otherwise.} \end{cases} \quad (\text{Б.3})$$

де $\alpha(k)$ - швидкість навчання,

i_k - випадковий індекс, що обирається на кожній ітерації.

По суті, SAG зберігає в пам'яті градієнт по кожній функції в сумі функцій, що оптимізуються. На кожній ітерації значення градієнта тільки однієї такої функції обчислюється і оновлюється в пам'яті. Метод SAG оцінює загальний градієнт на кожній ітерації шляхом усереднення значень градієнта, що зберігаються в пам'яті. Як і методи SG, вартість кожної ітерації не залежить від кількості функцій в сумі.

Витрати на обчислення не відрізняються від SGD, але накладні витрати пам'яті набагато більше. Це типовий спосіб використання простору для економії часу. Було показано, що SAG є алгоритмом лінійної збіжності, який набагато швидше, ніж SGD, і має великі переваги в порівнянні з іншими алгоритмами стохастичного градієнта.

Однак метод SAG застосовують тільки в разі, коли функція втрат є гладкою, а цільова функція є опуклою, наприклад, проблеми опуклого лінійного прогнозування.

В цьому випадку SAG досягає більш високої швидкості збіжності, ніж SGD. Крім того, при деяких специфічних проблемах він може навіть забезпечити кращу збіжність, ніж при стандартному градієнтному спуску.

Стохастичний градієнт зменшення дисперсії (SVRG) [130].

Оскільки метод SAG застосовують тільки до гладких і опуклих функцій і він повинен зберігати градієнт кожного зразка, його незручно застосовувати

в неопуклих нейронних мережах. Метод стохастичного зменшення градієнта дисперсії (SVRG) був запропонований для підвищення ефективності оптимізації в складних моделях. Алгоритм SVRG підтримує середній інтервальний градієнт, обчислюючи градієнти всіх вибірок на кожній ітерації:

$$\tilde{\mu} = \frac{1}{N} \sum_{i=1}^N g_i(\tilde{\theta}), \quad (\text{Б.4})$$

де $\tilde{\theta}$ -параметр поновлення інтервалу.

Параметр інтервалу $\tilde{\theta}$ містить середню пам'ять всіх градієнтів вибірки за минулий час для кожного інтервалу часу. SVRG вибирає форму випадковим чином і виконує поновлення градієнта для поточних параметрів:

$$\theta(k+1) = \theta(k) - \eta(g_i(k)(\theta(k) - g_i(k)(\tilde{\theta}) + \tilde{\mu}). \quad (\text{Б.5})$$

Градієнт розраховується до двох разів на кожному оновленні. Після w ітерацій виконується $\tilde{\theta} \leftarrow \theta_w$ і починаються такі ітерації. Завдяки цим оновленням $\theta(k+1)$ і інтервалу оновлення параметр $\tilde{\theta}$ буде збігатися до оптимального, а потім θ^* і $\tilde{\mu} \rightarrow 0$.

$$g_i(k)(\theta(k) - g_i(k)(\tilde{\theta}) + \tilde{\mu} \rightarrow g_i(k)(\theta(k) - g_i(k)(\theta^*) \rightarrow 0. \quad (\text{Б.6})$$

SVRG пропонує життєво важливу концепцію, яка називається зменшенням дисперсії. Ця концепція пов'язана з аналізом збіжності SGD, в якому необхідно передбачити, що існує постійна верхня межа для дисперсії градієнтів. Ця постійна верхня межа означає, що SGD не може досягти лінійної збіжності. Однак в SVRG верхня межа дисперсії може безперервно зменшуватися завдяки спеціальному елементу оновлення, завдяки чому досягається лінійна збіжність.

Стратегії SAG і SVRG пов'язані зі зменшенням дисперсії. У порівнянні з SAG SVRG не потрібно тримати всі градієнти в пам'яті, що означає, що ресурси пам'яті зберігаються, і це може ефективно застосовуватися для

вирішення складних завдань. Експерименти показали гарну продуктивність SVRG для неопуклої нейронної мережі [131]. Існує також багато варіантів таких алгоритмів стохастичної оптимізації лінійної збіжності, таких як, наприклад, алгоритм SAGA [131].

Momentum [132]

Замість того щоб залежати тільки від поточного градієнта для поновлення ваги, в градієнтному спуску з імпульсом поточний градієнт замінюється на (означає швидкість), експонентну ковзаючу середню поточного і минулих градієнтів (тобто до часу

$$\theta(k+1) = \theta(k) - \eta v(k+1); \quad (\text{Б.7})$$

$$v(k+1) = \gamma v(k) + (1-\gamma) \nabla_{\theta} J(\theta(k)), \quad (\text{Б.8})$$

де $\gamma = 0,9$.

Пізніше це оновлення імпульсу стає стандартним оновленням для компонента градієнта

NAG (Nesterov Accelerated Gradient) [133]

Даний алгоритм реалізує ідею накопичення імпульсу, використовуючи інформацію про зміну кожного параметра у вигляді експоненційного змінного середнього

$$v(k+1) = \gamma v(k) + (1-\gamma) x. \quad (\text{Б.9})$$

В алгоритмі накопичується градієнт цільової функції мережі

$$v(k+1) = \gamma v(k) + \eta \nabla_{\theta} J(\theta(k)), \quad (\text{Б.10})$$

$$\theta(k+1) = \theta(k) - v(k). \quad (\text{Б.11})$$

який і використовується при корекції параметрів

Більш точні співвідношення для алгоритму NAG мають вигляд

$$\theta(k+1) = \theta(k) - v(k+1). \quad (\text{Б.12})$$

$$v(k+1) = \gamma v(k) + \eta \nabla_{\theta} J(\theta(k) - \gamma v(k)); \quad (\text{Б.13})$$

Adagrad [134].

Adagrad (adaptive gradient) враховує частоту активації нейронів шляхом зберігання для кожного параметра мережі суми квадратів його оновлень. При цьому використовується модифікована формула поновлення:

$$M\{g^2(k+1)\} = \gamma M\{g^2(k)\} + (1-\gamma)g^2(k), \quad (\text{Б.14})$$

а корекція параметрів здійснюється за правилом

$$\theta(k+1) = \theta(k) - \frac{\eta}{\sqrt{G(k) + \varepsilon}} g(k), \quad (\text{Б.15})$$

де $G(k)$ - сума квадратів оновлень, а ε - згладжує параметр, необхідний для того, щоб уникнути поділу на 0. У часто оновлялись в минулому величина параметра $G(k)$ буде великою (великий знаменник в (15)), тобто параметр зміниться незначно. Рідко змінювався параметр зміниться суттєво. Параметр зазвичай вибирають порядку 10^{-6} - 10^{-8} .

RMSProp [135]

Недолік Adagrad в тому, що в (15) може збільшуватися скільки завгодно, в результаті чого через деякий час оновлень стають занадто маленькими, що призводить до паралічу алгоритму. RMSProp і Adadelta покликані виправити цей недолік. RMSprop - це неопублікований метод адаптивної швидкості навчання, запропонований Дж. Хінтоном в лекції 6 його курсу Coursera.

З цією метою Adagrad модифікується таким чином, щоб часто оновлювані ваги корегувалися рідше. Для цього замість повної суми оновлень використовується усереднений по історії квадрат градієнта, тобто ковзне середнє виду

$$M\{g^2(k+1)\} = \gamma M\{g^2(k)\} + (1-\gamma)g^2(k), \quad (\text{Б.16})$$

тоді замість (Б,15) отримаємо

$$\theta(k+1) = \theta(k) - \frac{\eta}{\sqrt{M\{g^2(k)\} + \varepsilon}} g(k). \quad (\text{Б.17})$$

Так як знаменник являє собою корінь із середнього квадратів градієнтів

$$RMS\{g(k)\} = \sqrt{M\{g^2(k)\} + \varepsilon}, \quad (\text{Б.18})$$

алгоритм отримав назву RMSProp - root mean square propagation.

Adadelata [136]

Adadelata є продовженням Adagrad, метою якого є недопусканіє монотонного зменшення швидкості навчання. Замість того, щоб накопичувати все градієнти минулого квадрата, Adadelata обмежує вікно накопичених градієнтів минулого фіксованим раз мером.

Adadelata відрізняється від RMSProp тим, що в чисельник (Б.17) додається стабілізуючий член пропорційний RMS від $\Delta\theta(k)$. На кроці $k+1$ значення ще не відомо, тому оновлення параметрів відбувається в три етапи, а не в два: спочатку накопичується квадрат градієнта, потім оновлюється θ , після чого оновлюється $RMS\{\Delta\theta(k)\}$, тобто

$$\theta(k+1) = \theta(k) - \frac{RMS\{\Delta\theta(k)\}}{RMS\{g(k)\}} g(k); \quad (\text{Б.19})$$

$$M\{\|\Delta\theta(k+1)\|^2\} = \gamma M\{\|\Delta\theta(k)\|^2\} + (1-\gamma)\|\Delta\theta(k)\|^2; \quad (\text{Б.20})$$

$$RMS\{\Delta\theta(k)\} = \sqrt{M\{\|\Delta\theta(k)\|^2\} + \varepsilon}. \quad (\text{Б.21})$$

Для RMSProp, Adadelata і для Adagrad не потрібно дуже точно підбирати швидкість навчання - досить його наближене значення. Зазвичай радять

почати підгін η з 0,1-1, а γ так і залишити 0,9. Чим ближче γ до 1, тим довше RMSProp і Adadelta з великим $RMS\{\Delta\theta(k)\}$ будуть сильно оновлювати мало використовувані ваги. Якщо ж $\gamma \approx 1$ і $RMS\{\Delta\theta(k)\} = 0$, то Adadelta буде довго «з недовірою» ставитися до рідко використовуваних ваг, що може привести як до паралічу алгоритму, так і викликати навмисно «жадібну» поведінку, коли алгоритм спочатку оновлює нейрони, які кодують найкращі ознаки.

Adam [137]

Adam (adaptive moment estimation) - ще один оптимізаційний алгоритм. Він поєднує в собі і ідею накопичення руху, і ідею слабшого поновлення ваг для типових ознак.

За аналогією з (9) маємо:

$$m(k+1) = \beta_1 m(k) + (1 - \beta_1) g(k). \quad (\text{Б.22})$$

Від алгоритму Нестерова Adam відрізняється тим, що в ньому накопичується не $\Delta\theta$, а значення градієнта. Для отримання інформації про зміну градієнта в [135] запропоновано оцінювати ще й середню нецентрованого дисперсію:

$$v(k+1) = \beta_2 v(k) + (1 - \beta_2) g^2(k). \quad (\text{Б.23})$$

Так як це, по суті, являє собою вираз (Б.16), то даний алгоритм не відрізняється від RMSProp. Автори Adam пропонують в якості значень за замовчуванням $\beta_1 = 0,9, \beta_2 = 0,999, \varepsilon = 10^{-8}$ і стверджують, що алгоритм працює краще або приблизно так само, як і всі попередні алгоритми на широкому наборі датасета за рахунок початкового калібрування.

Adamax [138]

Алгоритм Adamax є модифікацією алгоритму Adam, в якому замість дисперсії використовується інерційний момент розподілу градієнтів

довільного ступеня p . Хоча це може привести до нестабільності обчислень, на практиці випадок $p \rightarrow \infty$, працює на диво добре

$$v(k+1) = \beta_2^p v(k) + (1 - \beta_2^p) |g(k)|^p. \quad (\text{Б.24})$$

Nadam [139]

Алгоритм Nadam (Nesterov-accelerated Adaptive Moment Estimation) є модифікацією алгоритму Нестерова з адаптацією параметрів імпульсу

$$\theta(k+1) = \theta(k) - \frac{\eta}{\sqrt{G(k) + \varepsilon}} \left(\beta_1 \hat{g}(k) + \frac{1 - \beta_1}{1 - \beta_2} \nabla_{\theta} J(\theta(k)) \right), \quad (\text{Б.25})$$

$$\hat{g}(k) = \frac{g(k)}{1 - \beta_1}; \quad (\text{Б.26})$$

$$\hat{G}(k) = \frac{G(k)}{1 - \beta_2}; \quad (\text{Б.27})$$

$$g(k+1) = \beta_1 g(k) + (1 - \beta_1) \nabla_{\theta} J(\theta(k)); \quad (\text{Б.28})$$

$$G(k+1) = \beta_2 G(k) + (1 - \beta_2) \left[\nabla_{\theta} J(\theta(k)) \right]^2; \quad (\text{Б.29})$$

$$\alpha = 0.002; \beta_1 = 0.9; \beta_2 = 0.999 \quad \varepsilon = 10^{-7}.$$

AMSGrad

Іншим варіантом Adam є AMSGrad. Цей варіант переглядає компонент адаптивної швидкості навчання в Adam і змінює його, щоб гарантувати, що поточне значення G завжди було б більше, ніж попередній часовий крок.

$$\theta(k+1) = \theta(k) - \frac{\eta}{\sqrt{\hat{G}(k) + \varepsilon}} g(k); \quad (\text{Б.30})$$

$$\hat{G}(k+1) = \max(\hat{G}(k), G(k+1)); \quad (\text{Б.31})$$

$$g(k+1) = \beta_1 g(k) + (1 - \beta_1) \nabla_{\theta} J(\theta(k)); \quad (\text{Б.32})$$

$$G(k+1) = \beta_2 G(k) + (1 - \beta_2) \left[\nabla_{\theta} J(\theta(k)) \right]^2; \quad (\text{Б.33})$$

$$G(k+1) = \beta_2 G(k) + (1 - \beta_2) \left[\nabla_{\theta} J(\theta(k)) \right]^2; \quad (\text{Б.34})$$

$$\alpha = 0.001; \beta_1 = 0.9; \beta_2 = 0.999 \quad \varepsilon = 10^{-7}.$$

WNGrad

В алгоритмі WNGrad (weight normalization Grad) використовується метод динамічного оновлення швидкості навчання відповідно до отриманих досі градієнтів

$$\theta(k+1) = \theta(k) - \frac{1}{b(k)} \nabla_{\theta} J(\theta(k)), \quad (\text{Б.35})$$

$$b(k+1) = b(k) + \frac{1}{b(k)} \left\| \nabla_{\theta} J(\theta(k)) \right\|. \quad (\text{Б.36})$$

За результатами численних експериментів WNGrad є конкурентом простому стохастичному градієнтному спуску з точки зору стійкості і помилки узагальнення при навчанні нейронних мереж. Існують модифікації WNGrad, в яких використовується імпульс (WN-Adam and WNGrad-Momentum).

Додаток В

Аналіз впливу завади ξ_{n+1} на статистичні властивості оцінки Качмажа

Проаналізуємо вплив завади ζ_{n+1} на статистичні властивості оцінки Качмажа.

Обчислимо $M\{\tilde{\theta}_{n+1}\}$

$$M\{\tilde{\theta}_{n+1}\} = M\left\{\left(I - \gamma \frac{\tilde{x}_{n+1}\tilde{x}_{n+1}^T}{\|\tilde{x}_{n+1}\|^2 + \delta}\right)(\tilde{\theta}_n + S_{n+1})\right\} + \gamma M\left\{\frac{\zeta_{n+1}\zeta_{n+1}^T}{\|\tilde{x}_{n+1}\|^2 + \delta}\theta_n\right\} - \gamma M\left\{\frac{\tilde{x}_{n+1}}{\|\tilde{x}_{n+1}\|^2 + \delta}\xi_{n+1}\right\}. \quad (\text{B.1})$$

Тут I - одинична матриця $N \times N$.

Враховуючи статистичну незалежність сигналів і завад і той факт, що [61]

$$M\left\{\frac{x_{n+1}x_{n+1}^T}{\|x_{n+1}\|^2 + \delta}\right\} = \frac{1}{N}I - M\left\{\frac{\delta}{\|x_{n+1}\|^2 + \delta}\right\}I; \quad (\text{B.2})$$

$$M\left\{\frac{1}{\|x_{n+1}\|^2 + \delta}\right\} = \frac{1}{(N-2)\sigma_x^2 + \delta} \left(1 - \frac{2\delta\sigma_x^2}{((N-2)\sigma_x^2 + \delta)^2}\right), \quad (\text{B.3})$$

обчислимо

$$\begin{aligned}
M\{\tilde{\theta}_{n+1}\} &= \\
&= \left(1 - \frac{\gamma}{N} \left(\frac{\left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^3 - \delta N \left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^2 + 2\delta N \sigma_{\tilde{x}}^2}{\left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^3} \right) \right) M\{\tilde{\theta}_n + S_{n+1}\} + \\
&\quad + \frac{\gamma}{N} \frac{\sigma_{\tilde{x}}^2 - N\sigma_{\xi}^2 + \delta}{\left(\sigma_{\tilde{x}}^2 + \delta \right)} \theta_n
\end{aligned} \tag{B.4}$$

З отриманого виразу (B.4) випливає, що для збіжності алгоритму (2.52) необхідно задовольнити нерівність

$$\left| 1 - \frac{\gamma}{N} \left(\frac{\left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^3 - \delta N \left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^2 + 2\delta N \sigma_{\tilde{x}}^2}{\left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^3} \right) \right| < 1, \tag{B.5}$$

звідки маємо

$$\lim_{n \rightarrow \infty} M\{\hat{\theta}_{n+1}\} \neq 0. \tag{B.6}$$

Таким чином, при виконанні (B.5) алгоритм (2.52) в середньому збігається, але оцінка буде зміщеною $\lim_{n \rightarrow \infty} M\{\hat{\theta}_{n+1}\} \neq 0$.

Іншими словами, вектор оцінки буде пов'язаний з вектором шуканих параметрів співвідношенням

$$M\{\hat{\theta}_{n+1}\} = \frac{(\sigma_{\tilde{x}}^2 - N\sigma_{\xi}^2 + \delta) \left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^3}{\left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^3 - \delta N \left((N-2)\sigma_{\tilde{x}}^2 + \delta \right)^2 + 2N\delta\sigma_{\tilde{x}}^2} \theta_{n+1}. \tag{A.7}$$

Як видно з вищезазначених результатів, наявність завади ξ_{n+1} не призводить до зміщення в оцінці. Проаналізуємо, як ця завада впливає на дисперсію оцінки.

Для вивчення питань збіжності у середньоквадратичному помножимо

(2.56) зліва на $\tilde{\theta}_{n+1}^T$

$$\begin{aligned} \|\tilde{\theta}_{n+1}\|^2 &= \|\tilde{\theta}_n\|^2 + \|S_{n+1}\|^2 - 2\gamma \frac{\tilde{\theta}_n^T x_{n+1}}{\|x_{n+1}\|^2 + \delta} [\xi_{n+1} + \tilde{\theta}_n^T x_{n+1}] - \\ &- 2\gamma \frac{x_{n+1}^T S_{n+1}}{\|x_{n+1}\|^2 + \delta} [\xi_{n+1} + \tilde{\theta}_n^T x_{n+1}] + \gamma^2 \frac{\|x_{n+1}\|^2}{(\|x_{n+1}\|^2 + \delta)^2} [\xi_{n+1} + \tilde{\theta}_n^T x_{n+1}]^2 \end{aligned} \quad (\text{B.8})$$

Обчислимо $M \left\{ \|\tilde{\theta}_{n+1}\|^2 \right\}$

$$\begin{aligned} M \left\{ \|\theta_{n+1}\|^2 \right\} &= M \left\{ \|\theta_n\|^2 \right\} + M \left\{ \|S_{n+1}\|^2 \right\} - 2\gamma M \left\{ \frac{\theta_n^T x_{n+1} x_{n+1}^T \theta_n}{\|x_{n+1}\|^2 + \delta} \right\} - \\ &2\gamma M \left\{ \frac{S_{n+1}^T x_{n+1} x_{n+1}^T S_{n+1}}{(\|x_{n+1}\|^2 + \delta)^2} \|x_{n+1}\|^2 \right\} + \gamma^2 M \left\{ \frac{\xi_{n+1}^2 \|x_{n+1}\|^2}{(\|x_{n+1}\|^2 + \delta)^2} \right\} \end{aligned} \quad (\text{B.9})$$

Враховуючи (B.2), (B.3) і те, що [61]

$$\begin{aligned} M \left\{ \frac{(\tilde{\theta}_n^T x_{n+1})^2}{\|x_{n+1}\|^2 + \delta} \right\} &= \\ &= \left[\frac{1}{N} I - \frac{\delta}{(N-2)\sigma_x^2 + \delta} \left(1 - \frac{2\sigma_x^2 \delta}{((N-2)\sigma_x^2 + \delta)^2} \right) \right] M \left\{ \|\tilde{\theta}_n\|^2 \right\}; \\ M \left\{ \frac{\|x_{n+1}\|^2}{(\|x_{n+1}\|^2 + \delta)^2} \xi_{n+1}^2 \right\} &= \frac{(N-2)\sigma_x^2 \sigma_\xi^2}{((N-2)\sigma_x^2 + \delta)^2}; \end{aligned}$$

$$\begin{aligned}
M \left\{ \frac{\|x_{n+1}\|^4}{(\|x_{n+1}\|^2 + \delta)^2} \right\} &= \\
&= -2\delta \left(\frac{1}{(N-2)\sigma_x^2 + \delta} \left(1 - \frac{2\sigma_x^2\delta}{((N-2)\sigma_x^2 + \delta)^2} \right) - \frac{\delta^2}{((N-2)\sigma_x^2 + \delta)^2} \right) = \\
&= \frac{\delta^2((N-2)\sigma_x^2 + \delta) - 2\delta((N-2)\sigma_x^2 + \delta)^2 + 4\delta^2\sigma_x^2}{((N-2)\sigma_x^2 + \delta)^3},
\end{aligned}$$

після нескладних перетворювань отримуємо

$$\begin{aligned}
&M \left\{ \tilde{\theta}_{n+1}^2 \right\} = \\
&\left\{ 1 - \frac{\gamma}{N} \left[(2-\gamma) - \frac{2(\gamma-1)(\delta(N-2)\sigma_x^2 + \delta)^2 + 2\sigma_x^2\delta}{((N-2)\sigma_x^2 + \delta)^3} - \frac{\gamma^2\delta^2}{((N-2)\sigma_x^2 + \delta)^2} \right] \right\} \times \\
&\times M \left\{ \tilde{\theta}_n^2 + S_{n+1}^2 \right\} + \gamma^2 \frac{(N-2)\sigma_x^2\sigma_\xi^2}{((N-2)\sigma_x^2 + \delta)^2}.
\end{aligned} \tag{B.10}$$

Додаток Г

Дослідження робастної процедури навчання (3.25)

Розглянемо питання збіжності процедури (3.25).

Помноживши обидві частини (3.26) зліва на $\tilde{\theta}_{n+1}^T$ урахуванням того, що похибки відсутні, маємо

$$\begin{aligned} \|\tilde{\theta}_{n+1}\|^2 &= \|\tilde{\theta}_n\|^2 - 4\gamma\lambda e_{n+1}^2 - 2\gamma(1-\lambda)e_{n+1} \operatorname{sign} e_{n+1} + 4\gamma^2\lambda^2 e_{n+1}^2 \|x_{n+1}\|^2 + \\ &+ 4\gamma^2\lambda(1-\lambda)e_{n+1} \operatorname{sign} e_{n+1} \|x_{n+1}\|^2 + \gamma^2(1-\lambda)^2 \|x_{n+1}\|^2. \end{aligned} \quad (\Gamma.1)$$

З (Г.1) видно, що так як $\gamma > 0$, приріст функції Ляпунова

$$\Delta V_{n+1} = \|\tilde{\theta}_{n+1}\|^2 - \|\tilde{\theta}_n\|^2, \quad (\Gamma.2)$$

буде негативним, якщо

$$2\left(2\lambda e_{n+1}^2 + (1-\lambda)|e_{n+1}|\right) > \gamma\left(2\lambda e_{n+1} + (1-\lambda)|e_{n+1}|\right)^2 \|x_{n+1}\|^2. \quad (\Gamma.3)$$

Таким чином, умови збіжності процедури (3.25) будуть виконуватися, якщо параметр γ задовольняє нерівності

$$0 < \gamma < \frac{2e_{n+1}}{\left(\lambda e_{n+1} + (1-\lambda)\operatorname{sign} e_{n+1}\right)\|x_{n+1}\|^2}. \quad (\Gamma.4)$$

Дослідимо статистичні властивості процедури навчання (3.25) при наявності завад виміру, тобто $y_{n+1}^* = \theta^{*T} x_{n+1} + \xi_{n+1}$, $\xi_{n+1} \square N\left(0, \sigma_{\xi}^2\right)$.

Припустимо, що на завада не корельована з корисними сигналами. Записавши (3.25) щодо помилок навчання, маємо (3.26).

Розглянемо математичне сподівання $M\{\tilde{\theta}_{n+1}\}$. Усереднюючи обидві частини (3.26), отримуємо

$$\begin{aligned} M\{\tilde{\theta}_{n+1}\} &= M\left\{\tilde{\theta}_n - 2\gamma\lambda\left(\tilde{\theta}_n^T x_{n+1} + \xi_{n+1}\right)x_{n+1} - \right. \\ &\quad \left. - \gamma(1-\lambda)\text{sign}\left(\tilde{\theta}_n^T x_{n+1}\right)x_{n+1}\right\}. \end{aligned} \quad (\text{Г.5})$$

Легко бачити, що

$$M\left\{\tilde{\theta}_n^T x_{n+1} x_{n+1}\right\} = M\left\{x_{n+1} x_{n+1}^T \tilde{\theta}_n\right\} = M\left\{x_{n+1} x_{n+1}^T\right\} M\left\{\tilde{\theta}_n\right\} = R_{xx} M\left\{\tilde{\theta}_n\right\}, \quad (\text{Г.6})$$

де R_{xx} – кореляційна матриця вхідного сигналу.

Детально розглянемо вираз $M\left\{x_{n+1} \text{sign}\left(\tilde{\theta}_n^T x_{n+1}\right)x_{n+1}\right\}$. В цьому випадку, якщо сигнал $x_{n+1} \square N\left(0, \sigma_x^2\right)$, маємо

$$\begin{aligned} M\left\{x_{n+1} \text{sign}\left(\tilde{\theta}_n^T x_{n+1}\right)\right\} &= M\left\{M\left\{x_{n+1} \text{sign}\left(\tilde{\theta}_n^T x_{n+1}\right) \mid \tilde{\theta}_n\right\}\right\} = \\ &= M\left\{\sqrt{\frac{2}{\pi}} \frac{1}{\sigma_e \mid \theta_n} M\left\{x_{n+1} \left(\theta_n^T x_{n+1}\right) \mid \theta_n\right\}\right\} = \\ &= M\left\{\sqrt{\frac{2}{\pi \sigma_e^2}} M\left\{x_{n+1} x_{n+1}^T \theta_n \mid \theta_n\right\}\right\} = \sqrt{\frac{2}{\pi \sigma_e^2}} M\left\{x_{n+1} x_{n+1}^T \theta_n\right\} = \\ &= \sqrt{\frac{2}{\pi \sigma_e^2}} R_{xx} M\left\{\theta_n\right\}, \end{aligned} \quad (\text{Г.7})$$

де σ_e - середньоквадратичне значення неузгодженості e_{n+1} .

Вираз (Г.7) отримано з використанням теореми Прайса [140], згідно з якою для двох випадкових гаусівських величин x і y з нульовими математичними сподіваннями справедливо

$$M \{x \sin y\} = \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_y} M \{xy\},$$

де σ_y – середньоквадратичне значення y .

З урахуванням властивостей завади $M \{\xi_{n+1} x_{n+1}\} = 0$ маємо

$$M \{\tilde{\theta}_{n+1}\} = \{I - 2\gamma\lambda R_{xx} - \gamma(1-\lambda)\beta R_{xx}\} M \{\tilde{\theta}_n\}. \quad (\Gamma.8)$$

Тут $\beta = \sqrt{\frac{1}{\pi\sigma_e^2}}$; trR_{xx} – слід матриці R_{xx} .

З (Г.8) випливає, що процедура (3.25) буде збігатися в середньому, якщо параметр γ задовольняє нерівності

$$0 < \gamma < \frac{2}{(2\lambda + (1-\lambda)\beta)trR_{xx}}. \quad (\Gamma.9)$$

З огляду на властивості вхідних сигналів, можемо записати

$$0 < \gamma < \frac{2}{(2\lambda + (1-\lambda)\beta)N\sigma_x^2}. \quad (\Gamma.10)$$

Помножимо обидві частини (3.26) зліва на $\tilde{\theta}_{n+1}^T$. Розглянемо величину $M \left\{ \|\tilde{\theta}_{n+1}\|^2 \right\}$.

Взявши від обох частин (3.30) математичне сподівання, маємо

$$\begin{aligned} M \left\{ \|\tilde{\theta}_{n+1}\|^2 \right\} = & M \left\{ \|\tilde{\theta}_n\|^2 - 4\gamma\lambda (\tilde{\theta}_n^T x_{n+1})^2 - \right. \\ & - 2\gamma(1-\lambda)\tilde{\theta}_n^T x_{n+1} \text{sign}(\tilde{\theta}_n^T x_{n+1}) + 4\gamma^2\lambda^2 (\tilde{\theta}_n^T x_{n+1})^2 \|x_{n+1}\|^2 + \\ & \left. + 4\gamma^2\lambda(1-\lambda)(\tilde{\theta}_n^T x_{n+1}) \text{sign}(\tilde{\theta}_n^T x_{n+1}) \|x_{n+1}\|^2 + \gamma^2(1-\lambda)^2 \|x_{n+1}\|^2 \right\}. \end{aligned} \quad (\Gamma.11)$$

Розглянемо кожний доданок в правій частині (Г.11) з урахуванням статистичних властивостей сигналів і завад. У нашому випадку

$$M \left\{ \left(\tilde{\theta}_n^T x_{n+1} \right)^2 \right\} = \sigma_\xi^2 + \sigma_x^2 M \left\{ \left\| \theta_n \right\|^2 \right\}; \quad (\text{Г.12})$$

$$M \left\{ \left(\tilde{\theta}_n^T x_{n+1} \right)^2 \left\| x_{n+1} \right\|^2 \right\} = 3\sigma_x^4 M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} + N\sigma_x^2 \sigma_\xi^4; \quad (\text{Г.13})$$

$$M \left\{ \left\| x_{n+1} \right\|^2 \right\} = N\sigma_x^2. \quad (\text{Г.14})$$

Вираз для $M \left\{ \tilde{\theta}_n^T x_{n+1} \text{sign} \left(\theta_n^T x_{n+1} \right) \right\}$ проаналізуємо таким чином:

$$\begin{aligned} M \left\{ \tilde{\theta}_n^T x_{n+1} \text{sign} \left(\tilde{\theta}_n^T x_{n+1} \right) \right\} &= M \left\{ M \left\{ \tilde{\theta}_n^T x_{n+1} \text{sign} \left(\tilde{\theta}_n^T x_{n+1} \right) \mid \tilde{\theta}_n \right\} \right\} = \\ &= M \left\{ \tilde{\theta}_n^T \sqrt{\frac{2}{\pi \sigma_x^2 e^{\tilde{\theta}_n^T \tilde{\theta}_n}}} M \left\{ x_{n+1} x_{n+1}^T \tilde{\theta}_n \mid \tilde{\theta}_n \right\} \right\} \approx M \left\{ \tilde{\theta}_n^T \beta \tilde{\theta}_n \right\} R_{xx} = \\ &= \beta \text{tr} R_{xx} M \left\{ \tilde{\theta}_n^2 \right\} = \beta \sigma_x^2 M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}. \end{aligned} \quad (\text{Г.15})$$

Аналогічно отримуємо

$$M \left\{ \tilde{\theta}_n^T x_{n+1} \text{sign} \left(\tilde{\theta}_n^T x_{n+1} \right) \left\| x_{n+1} \right\|^2 \right\} = 3\beta \sigma_x^4 M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}. \quad (\text{Г.16})$$

Підстановка виразів (Г.7), (Г.12)-(Г.16) в (Г.11) і нескладні перетворення дають

$$\begin{aligned} M \left\{ \left\| \tilde{\theta}_{n+1} \right\|^2 \right\} &= \left(1 - 4\gamma\lambda\sigma_x^2 + 2\gamma(1-\lambda)\beta\sigma_x^2 + 12\gamma^2\lambda^2\sigma_x^4 + 12\gamma^2(1-\lambda)\sigma_x^4 \right) \times \\ &\times M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} + \gamma^2(1-\lambda)^2 N\sigma_\xi^2\sigma_x^2. \end{aligned} \quad (\text{Г.17})$$

З (Г.17) випливає, що процедура (3.25) буде збігатися в середньоквадратичному при виконанні умови

$$\left| 1 - 4\gamma\lambda\sigma_x^2 + 2\gamma(1-\lambda)\beta\sigma_x^2 + 12\gamma^2\lambda^2\sigma_x^4 + 12\gamma^2(1-\lambda)\sigma_x^4 \right| < 1.$$

Додаток Д

Дослідження статистичних властивостей процедури навчання

Припустимо, що завада не корельована з корисними сигналами. Записавши (3.31) щодо помилок навчання, маємо (3.35).

Розглянемо математичне сподівання $M\{\tilde{\theta}_{n+1}\}$. Усереднюючи обидві частини (3.35), отримуємо

$$M\{\tilde{\theta}_{n+1}\} = M\{\tilde{\theta}_n - 2\gamma\lambda(\tilde{\theta}_n^T x_{n+1} + \xi_{n+1})x_{n+1} - \gamma(1-\lambda)\text{sign}(\tilde{\theta}_n^T x_{n+1})x_{n+1}\}. \quad (\text{Д.1})$$

Легко бачити, що

$$M\{\tilde{\theta}_n^T x_{n+1} x_{n+1}\} = M\{x_{n+1} x_{n+1}^T \tilde{\theta}_n\} = M\{x_{n+1} x_{n+1}^T\} M\{\tilde{\theta}_n\} = R_{xx} M\{\tilde{\theta}_n\}, \quad (\text{Д.2})$$

Вираз $M\{x_{n+1} \text{sign}(\tilde{\theta}_n^T x_{n+1}) x_{n+1}\}$ для випадку, коли сигнал $x_{n+1} \sim N(0, \sigma_x^2)$, було розглянуто вище.

З урахуванням властивостей завади $M\{\xi_{n+1} x_{n+1}\} = 0$ маємо

$$M\{\tilde{\theta}_{n+1}\} = \left\{ I - 3\gamma\lambda\sigma_e^2 R_{xx} - \gamma(1-\lambda)\beta R_{xx} \right\} M\{\tilde{\theta}_n\}, \quad (\text{Д.3})$$

звідки випливає, що процедура (3.31) буде збігатися в середньому, якщо параметр γ задовольняє нерівності

$$0 < \gamma < \frac{2}{(3\lambda\sigma_e^2 + (1-\lambda)\beta) \text{tr} R_{xx}}. \quad (\text{Д.4})$$

З огляду на властивості вхідних сигналів, можемо записати умову (Д.4) наступним чином:

$$0 < \gamma < \frac{2}{(3\sigma_e^2\lambda + (1-\lambda)\beta)N\sigma_x^2}. \quad (\text{Д.5})$$

Для дослідження збіжності алгоритму в середньоквадратичному розглянемо функцію Ляпунова $M \left\{ \left\| \tilde{\theta}_{n+1} \right\|^2 \right\}$.

Помноживши обидві частини (3.35) зліва на $\tilde{\theta}_{n+1}^T$, отримаємо

$$\begin{aligned} \left\| \tilde{\theta}_{n+1} \right\|^2 = & \left\| \tilde{\theta}_n \right\|^2 - 2\gamma\lambda e_{n+1}^3 \left(\tilde{\theta}_n^T x_{n+1} \right) - 2\gamma(1-\lambda) \tilde{\theta}_n^T x_{n+1} \text{sign} e_{n+1} + \\ & \gamma^2 \lambda^2 e_{n+1}^6 \left\| x_{n+1} \right\|^2 + 2\gamma^2 \lambda (1-\lambda) e_{n+1}^3 \text{sign} e_{n+1} \left\| x_{n+1} \right\|^2 + \gamma^2 (1-\lambda)^2 \left\| x_{n+1} \right\|^2. \end{aligned} \quad (\text{Д.6})$$

В даний вираз входять величини e_{n+1}^3 та e_{n+1}^6 , які відповідно дорівнюють

$$\begin{aligned} e_{n+1}^3 = & (e_{a,n+1} + \xi(k))(e_{a,n+1}^2 + 2e_{a,n+1}\xi_{n+1} + \xi_{n+1}^2); \\ e_{n+1}^6 = & (e_{a,n+1} + \xi_{n+1})^6 = e_{a,n+1}^6 + 5e_{a,n+1}^5 \xi_{n+1} + 6e_{a,n+1}^4 \xi_{n+1}^2 + \\ & + e_{a,n+1}^3 \xi_{n+1}^3 + 15e_{a,n+1}^2 \xi_{n+1}^4 + \xi_{n+1}^6. \end{aligned} \quad (\text{Д.7})$$

Взявши від обох частин (Д.6) математичне сподівання, маємо

$$\begin{aligned} M \left\{ \left\| \tilde{\theta}_{n+1} \right\|^2 \right\} = & M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} - 2\gamma\lambda M \left\{ e_{n+1}^3 e_{a,n+1} \right\} - 2\gamma(1-\lambda) M \left\{ e_{a,n+1} \text{sign} e_{n+1} \right\} + \\ & + \gamma^2 \lambda^2 M \left\{ e_{n+1}^6 \left\| x_{n+1} \right\|^2 \right\} + 2\gamma^2 \lambda (1-\lambda) M \left\{ e_{n+1}^3 \text{sign} e_{n+1} \left\| x_{n+1} \right\|^2 \right\} + \\ & + \gamma^2 (1-\lambda)^2 M \left\{ \left\| x_{n+1} \right\|^2 \right\}. \end{aligned} \quad (\text{Д.8})$$

Розглянемо кожний доданок в правій частині (Д.8) з урахуванням (Д.7) і статистичних властивостей сигналів і завад. У нашому випадку

$$M \left\{ e_{n+1}^3 e_{a,n+1} \right\} = M \left\{ \left(\tilde{\theta}_n^T x_{n+1} \right)^4 \right\} \approx 3\sigma_x^4 \left(M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} \right)^2;$$

$$\begin{aligned}
M \left\{ \left(\tilde{\theta}_n^T x_{n+1} \right)^2 \right\} M \left\{ \xi_{n+1}^4 \right\} &\approx \sigma_x^2 M \left\{ \xi_{n+1}^4 \right\} M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}; \\
M \left\{ \left(\tilde{\theta}_n^T x_{n+1} \right)^6 \left\| x_{n+1} \right\|^2 \right\} &\approx (15N + 90) \sigma_x^8 \left(M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} \right)^3; \\
M \left\{ \left(\tilde{\theta}_n^T x_{n+1} \right)^4 \left\| x_{n+1} \right\|^2 \right\} &= (3N + 2) \sigma_x^4 M \left\{ \xi_{n+1}^4 \right\} M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}; \\
M \left\{ \left(\tilde{\theta}_n^T x_{n+1} \right)^2 \left\| x_{n+1} \right\|^2 \right\} &= (N + 2) \sigma_x^4 M \left\{ \xi_{n+1}^4 \right\} M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}; \\
M \left\{ \left(\tilde{\theta}_n^T x_{n+1} \right)^2 \right\} &= \sigma_x^2 M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}; \\
M \left\{ \left\| x_{n+1} \right\|^2 \xi^6 \right\} &= M \left\{ \left\| x_{n+1} \right\|^2 \right\} M \left\{ \xi^6 \right\} = N \sigma_x^2 M \left\{ \xi^6 \right\}; \\
M \left\{ \left\| x_{n+1} \right\|^2 \right\} &= N \sigma_x^2.
\end{aligned}$$

Вираз для $M \left\{ \tilde{\theta}_n^T x_{n+1} \operatorname{sign} \left(\tilde{\theta}_n^T x_{n+1} \right) \right\}$ також було проаналізовано раніше.

При обчисленні $M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}$ враховано статистичні властивості завади,

тобто $M \left(\xi_{n+1} \right) = M \left(\xi_{n+1}^3 \right) = M \left(\xi_{n+1}^5 \right) = 0$.

Підстановка отриманих виразів в (Д.8) і нескладні перетворення дають

$$\begin{aligned}
M \left\{ \left\| \tilde{\theta}_{n+1} \right\|^2 \right\} &= \left(\begin{array}{l} 1 - 6\gamma\lambda\sigma_x^2\sigma_\xi^2 + 15(N+2)\gamma^2\lambda^2\sigma_x^4 M \left\{ \xi_{n+1}^4 \right\} - \\ - 2\gamma(1-\lambda)\beta\sigma_x^2 \end{array} \right) \times \\
&\times M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} + \left(\begin{array}{l} 15(3N+12)\gamma^2\lambda^2\sigma_x^6\sigma_\xi^2 + \\ + 6\gamma^2\lambda(1-\lambda)\sigma_x^4\beta - 6\gamma\lambda\sigma_x^4 \end{array} \right) \times \\
&\times \left(M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} \right)^2 + (15N+90)\gamma^2\lambda^2\sigma_x^8 \left(M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} \right)^3 + \\
&+ \gamma^2\sigma_x^2 N \left(\lambda^2 M \left\{ \xi_{n+1}^6 \right\} + (1-\lambda)^2 \right).
\end{aligned} \tag{Д.9}$$

Якщо алгоритм збігається, величина $M \left\{ \left\| \tilde{\theta}_{n+1} \right\|^2 \right\}$ буде малою. Тому для аналізу сталого стану вираз (Д.9) можна спростити, нехтуючи величинами $\left(M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} \right)^2$ та $\left(M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\} \right)^3$ і обмежувачись розглядом величини

$$\begin{aligned} M \left\{ \left\| \tilde{\theta}_{n+1} \right\|^2 \right\} &= (1 - 6\gamma\lambda\sigma_x^2\sigma_\xi^2 + 15(N+2)\gamma^2\lambda^2\sigma_x^4) M \left\{ \xi_{n+1}^4 \right\} - \\ &- 2\gamma(1-\lambda)\beta\sigma_x^2 M \left\{ \left\| \tilde{\theta}_n \right\|^2 \right\}. \end{aligned} \quad (\text{Д.10})$$

З (Г.10) випливає, що процедура (3.31) буде збігатися в середньоквадратичному (приріст функції Ляпунова буде негативним) за виконання умови

$$\begin{aligned} &|1 - 6\gamma\sigma_x^2 M \left\{ \xi^2(k) \right\} + 15\gamma^2\lambda^2(N+2)\sigma_x^4 M \left\{ \xi^4(k) \right\} + \\ &+ 6\gamma^2\lambda(1-\lambda)\beta(N+2)\sigma_x^4 M \left\{ \xi^2(k) \right\} + 2\gamma(1-\lambda)\beta\sigma_x^2| < 1, \end{aligned}$$

тобто якщо параметр γ задовольняє нерівності

$$0 < \gamma < \frac{2(3\lambda\sigma_\xi^2 + (1-\lambda)\beta)}{15(N+2)\lambda^2\sigma_x^2 M \left\{ \xi_{n+1}^4 \right\}}. \quad (\text{Д.12})$$

Додаток Е

Список публікацій здобувача

1. Rudenko O. Analysis of convergence of adaptive single-step algorithms for the identification of non-stationary objects / O. Rudenko, O. Bezsonov, V. Lebediev, A. Romanyk // Eastern-European Journal of Enterprise Technologies. 2019. – № 1/4 (97). – P. 6–14. (Належить до категорії А, входить до міжнародної наукометричної бази Scopus).
2. Руденко О.Г. Робастна ідентифікація об'єктів на основі мінімізації комбінованого функціоналу / О.Г. Руденко О.Г., О.О. Безсонов, Н.М. Сердюк, К.О. Олійник, О.С. Романюк // Системи обробки інформації. – 2020. – №1 (160). – С.80–88 (Належить до категорії Б).
3. Бессонов А.А. Факторизованные алгоритмы обучения перцептрона в задаче построения нелинейной модели / А.А. Бессонов, К.О. Олейник, С.А. Романюк, О.Г. Руденко, Н.Н. Сердюк / Бионика интеллекта.- 2020. –№1(94). – С. 23-29 (Належить до категорії Б).
4. Руденко О.Г. Критерії вибору перцептронної моделі для прогнозування: аналіз та практичні рекомендації щодо їх використання / О.Г. Руденко, О.О. Безсонов, О.Г. Лебедев, О.С. Романюк // Бионика интеллекта. – 2018. –№ 2 (91). – С.31– 40.
5. Руденко О.Г. Про один алгоритм навчання нейронної мережі в задачі прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.Г. Лебедев, О.С. Романюк // Бионика интеллекта. – 2018. – №1(90). –С.79–83.
6. Руденко О.Г. Робастна ідентифікація об'єктів за наявністю негаусівських завад / О.Г. Руденко О.Г., О.О. Безсонов, Н.М. Сердюк, К.О. Олійник, О.С.Романюк // Бионика интеллекта. –2019. – №2(93). – С.7–12.
7. Руденко О.Г. Нейромережеве прогнозування часових рядів на основі багатосарового перцептрона / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Development Management. – 2019. –Vol. 5 – Issue 1. – С.23–34.

8. Олійник К.О. Градієнтні алгоритми навчання згортальних нейронних мереж / К.О. Олійник, О.С. Романюк // Інформаційні технології та системи: монографія / за заг. ред. В. С. Пономаренка. – Харків. : Видавництво «Стиль-іздат», 2020. – С. 37–51.

9. Rudenko O. Analysis of the influence of interference on the statistical properties of estimates of non-stationary parameters / O. Rudenko, A. Romanyk // Інформаційні системи і технології: монографія / за заг. ред. В. С. Пономаренка. – Харків: ФОП Бровін О.В., 2019. – С.97–107.

10. Романюк О.С. Нейроеволюційний підхід до прогнозування нестационарних послідовностей / О.С. Романюк // Збірник матеріалів І Міжнар. наук.-практ. конф. молодих вчених, аспірантів і студентів; «Сучасні інформаційні технології та системи в управлінні». [Електронний ресурс] 19–20 квітня 2018 р. – Київ : КНЕУ, 2018. – С.204 – 205.

11. Руденко О.Г. Прогнозування нестационарних послідовностей за допомогою коеволюціонуючих штучних нейромереж/ О.Г. Руденко, О.С. Романюк // Збірник наукових праць за матеріалами ІІ міжнародної науково-практичної конференції. «Синергетика, мехатроніка, телематика дорожніх машин і систем у навчальному процесі та науці». 29 травня 2018 р. – Харків, ХНАДУ, 2018. – С.159–161.

12. Руденко О.Г. Дослідження особливостей навчання штучних нейронних мереж в задачі прогнозування / О.Г. Руденко, О.С. Романюк // Матеріали 8-ої міжнародної науково-технічної конференції «Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління». 26-27 квітня 2018 р., Полтава, Баку, Харків, Жиліна. – 2018. – С. 32.

13. Rudenko O. Time series prediction based on evolving neural network CMAC / O. Rudenko, O. Bezsonov, A. Romanyk // 3-d International Conference on Computer Algebra and Information Technologies. – Odessa, August 20-25, 2018. – P. 49 –51.

14. Руденко О.Г. Про оцінювання нестационарних параметрів, що описуються марківською моделлю / О.Г. Руденко, О.О. Безсонов, О.С.

Романюк // Тези доповідей шостої міжнародної науково-практичної конференції «Проблеми інформатизації». 14 – 16 листопада 2018 р., Черкаси, Баку, Бельско-Бяла, Харків. – 2018. – С.48.

15. Руденко О.Г. Прогнозування часових рядів на основі коеволюційних нейронних мереж / О.Г. Руденко, О.С. Романюк // Тези доповідей міжнародної науково-практичної конференції «Інформаційні технології та системи». 10-11 квітня 2019, Харків. – 2019. – С.15-16.

16. Руденко О.Г. Критерії вибору персептронної моделі для прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Тези доповідей дев'ятої міжнародної науково-практичної конференції «Сучасні напрями розвитку інформаційно-комунікаційних систем та засобів управління», 11-12 квітня 2019 р., Баку, Харків, Жиліна. – 2019. – С.90 – 91.

17. Руденко О.Г. Про вибір математичної моделі для прогнозування часових рядів / О.Г. Руденко, О.О. Безсонов, О.С. Романюк // Матеріали XVII міжнародної науково-практичної конференції «Математичне та програмне забезпечення інтелектуальних систем». (MSSIS-2019) 20-22 листопада 2019 р., Дніпро. – 2019. – С. 221–222.

18. Олейник К.О. Дослідження спрощених алгоритмів навчання штучних нейронних мереж прямого поширення / К.О. Олейник, О.С. Романюк // Матеріали міжнародної науково-практичної конференції «Інформаційні технології та системи». 9-10 квітня 2020, Харків :ХНЕУ імені Семена Кузнеця. – 2020. – С. 19.

Додаток Ж
Акт впровадження

«ЗАТВЕРДЖУЮ»

Директор державної установи
«Державний науково-дослідний і
проектний інститут основної хімії»
(ДУ «НІОХІМ»), канд. наук з
держ. управління



І.Г. Довгалюк

2020 р.

АКТ

використання результатів дисертаційної роботи аспіранта Романюка Олександра Сергійовича на тему «Нейромережеве прогнозування нестационарних послідовностей», виконаної на кафедрі Комп'ютерних інтелектуальних технологій та систем Харківського національного університету радіоелектроніки


Комісія у складі: від ДУ «НІОХІМ» – заступника директора з розвитку і науки, к.т.н., Малакей З.А.; начальника науково-технічного відділу (НТВ), д.т.н., проф. Панасенка В.О.; зав. лабораторії метрології, автоматизації та сушки Барановського А.О.; від ХНУРЕ – зав. кафедри Комп'ютерних інтелектуальних технологій та систем (КІТС), д.т.н., проф. Руденка О.Г., д.т.н., проф. Корабльова М.М., к.т.н., доц. Сердюк Н.М. підтверджують факт впровадження результатів дисертаційної роботи Романюка О.С. у програмно-апаратні системи управління технологічними процесами виробництва кальцинованої соди (ТП ВКС), а саме:

- у ДУ «НІОХІМ» розроблена багаторівнева комп'ютерно-інтегрована система управління технологічними процесами виробництва кальцинованої соди (ТП ВКС), яка являє собою складний програмно-апаратний комплекс;
- розроблені у дисертаційній роботі Романюка О.С. програмні моделі були включені до пакету підпрограм, що використовуються в системі управління ТП ВКС;
- впроваджено нейромережеві структури, розроблені особисто Романюком О.С., з чим згодні співавтори його наукових праць;
- для регулювання та стабілізації ТП ВКС використовуються методики, засновані на складних математичних моделях, які потребують значних витрат ресурсів управляючих ЕОМ. Розроблені Романюком О.С. методи та структури дозволяють значно поліпшити стабільність вихідних параметрів ТП ВКС та прогнозування ходу технологічних режимів. Це призвело до поліпшення характеристик процесів виробництва.

Розробки, виконані в дисертаційній роботі Романюка О.С., що спрямовані на підвищення енергоефективності виробництва кальцинованої соди аміачним способом за рахунок створення оптимальної структури та системи управління відділень виробництва кальцинованої соди, кафедра КІТС передає ДУ «НІОХІМ» для використання щодо модернізації та проектування комп'ютерно-інтегрованої технології абсорбції та дистиляції виробництва кальцинованої соди.

Від ДУ «НІОХІМ»:

Заступник директора
з розвитку і науки, к.т.н.



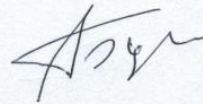
З.А. Малакей

Начальник НТВ, д.т.н., проф.



В.О. Панасенко

Зав. лабораторії метрології,
автоматизації та сушки



А.О. Барановський

Від ХНУРЕ:

Зав. кафедри КІТС д.т.н., проф.



О.Г. Руденко

д.т.н., проф.

М.М. Корабльов

к.т.н., доц.

Н.М. Сердюк