

Министерство образования и науки Украины  
Харьковский национальный университет радиоэлектроники

На правах рукописи

БОЙКО ЕЛЕНА АЛЕКСАНДРОВНА

УДК 004.032.26

ЭВОЛЮЦИОНИРУЮЩИЕ НЕЙРО-ФАЗЗИ СИСТЕМЫ В ЗАДАЧАХ  
ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ДАННЫХ

05.13.23 – системы и средства искусственного интеллекта

Диссертация на соискание учёной степени  
кандидата технических наук

Научный руководитель:  
Бодянский Евгений Владимирович,  
доктор технических наук, профессор

Цей примірник дисертації ідентичний  
за змістом з іншими, що подані до  
спеціалізованої вченої ради Д64.052.01

Учений секретар спецради Д64.052.01

О.А. Винокурова

Харьков – 2016

## СОДЕРЖАНИЕ

Введение	5
Раздел 1. Обзор состояния проблемы и постановка задачи исследования	12
1.1 Методы обучения и самообучение искусственных нейронных сетей	12
1.2 Нечёткая логика и нейро-фаззи системы	15
1.3 Эволюционирующие системы	18
1.4 Постановка задачи исследования	33
Выводы к разделу 1	34
Раздел 2. Эволюционирующие нейро-фаззи системы для обучения с учителем	36
2.1 Система Такаги – Сугено – Канга	36
2.2 Эволюционирующая многослойная нейро-фаззи система с гибридным обучением	38
2.2.1 Архитектура эволюционирующей многослойной нейро-фаззи системы	38
2.2.2 Самообучение-эволюция первого скрытого слоя	42
2.2.3 Адаптивное обучение синаптических весов	44
2.3 Настройка параметров функций принадлежности нейро-фаззи узла	47
2.4 Эволюционирующая МГУА-нейро-фаззи система с малым числом настраиваемых параметров	49
2.4.1 Архитектура эволюционирующей многослойной МГУА-нейро-фаззи системы	49
2.4.2 Нейро-фаззи система с малым числом настраиваемых параметров в качестве узла МГУА-системы	52
2.5 Эволюционирующая каскадная нейро-фаззи система	55

2.6 Адаптивное прогнозирование нестационарных нелинейных последовательностей на основе эволюционирующей нейро(нео)-фаззи-WANARX-модели	56
2.6.1 Архитектура ANARX-модели	58
2.6.2 Нео-фаззи-ANARX-модель	59
2.6.3 Взвешенная ANARX-модель (WANARX-модель)	63
Выводы к разделу 2	68
Раздел 3. Эволюционирующие нейро-фаззи системы для обучения без учителя	69
3.1 Самоорганизующаяся карта Кохонена	69
3.2 Ансамбль адаптивных нейро-фаззи сетей Кохонена для online нечёткой кластеризации потоков данных	74
3.2.1 Вероятностная нечёткая кластеризация	74
3.2.2 Нечёткая кластеризация с переменным фаззификатором	76
3.2.3 Ансамбль адаптивных нейро-фаззи сетей Кохонена	78
3.3 Эволюционирующая адаптивная нейро-фаззи сеть Кохонена и online алгоритм её самообучения в задачах нечёткой кластеризации	81
3.3.1 Эволюционирующая нечёткая кластеризация на основе возможностного подхода (EPCM)	81
3.3.2 Эволюционирующая адаптивная нечёткая кластеризующая нейронная сеть Кохонена (EAFCKN)	85
Выводы к разделу 3	89
Раздел 4. Моделирование и практическое применение разработанных архитектур и методологий	90
4.1 Моделирование эволюционирующей многослойной нейро-фаззи системы с гибридным обучением	90
4.2 Моделирование эволюционирующей МГУА-нейро-фаззи системы	93

	4
4.3 Моделирование эволюционирующей каскадной системы	95
4.4 Моделирование взвешенной ANARX-системы (WANARX-системы)	102
4.5 Моделирование эволюционирующей адаптивной нечёткой нейронной сети Кохонена	107
4.6 Решение практической задачи прогнозирования медицинских показателей	114
4.7 Решение практической задачи кластеризации земельных участков	120
Выводы к разделу 4	127
Выводы	129
Список использованных источников	131
Приложение А – Акты о внедрении	142

## ВВЕДЕНИЕ

На пересечении теорий искусственных нейронных сетей и нечётких систем сформировалось новое направление в теории вычислительного интеллекта, известное как нейро-фаззи системы [1-4], которые обладают как универсальными аппроксимирующими свойствами, так и возможностями нечёткого вывода. Они получили широкое распространение для решения различных задач интеллектуального анализа данных, идентификации, эмуляции, прогнозирования, интеллектуального управления и т. п. в условиях нелинейности, неопределённости, стохастичности и хаотичности, различного рода возмущений и помех.

В случаях, когда необходимо обрабатывать информацию в адаптивном online режиме, на первый план выходит вопрос скорости сходимости процесса обучения. Эти условия существенно ограничивают класс нейро-фаззи систем, которые пригодны для обработки данных в таком режиме. Наиболее эффективными с этой точки зрения в этой ситуации оказываются ANFIS и TSK-системы [1, 2, 5]. Поскольку их выходной сигнал линейно зависит от настраиваемых параметров, для их обучения можно использовать как множество известных линейных алгоритмов обучения [6], так и известные методы адаптивной идентификации [7, 8], которые обладают высоким быстродействием.

**Актуальность темы.** Несмотря на эффективность обработки данных ANFIS и TSK-системами, известно, что такие системы обучаются, как правило, только на уровне синаптических весов и параметров функций принадлежности, причём количество этих параметров, а также архитектура нейро-фаззи системы в целом в процессе обучения не изменяются. В то же время априори невозможно выбрать оптимальную структуру такой системы. Поэтому необходимо кроме обучения таких параметров как синаптические веса и параметры функций принадлежности, обучать собственно архитектуру системы. Это направление исследований является предметом,

так называемых, эволюционирующих нейро-фаззи систем [9-13]. Но основным отличием большинства известных на данный момент систем такого типа является то, что вся обучающая выборка задана априори, а само обучение происходит в пакетном режиме.

Таким образом, на сегодняшний день актуальной является задача разработки таких эволюционирующих нейро-фаззи систем, которые в процессе поступления новой информации в последовательном online режиме изменяют как свои синаптические веса и параметры функций принадлежности, так и их количество и саму архитектуру системы. Такие системы позволяют обрабатывать данные, поступающие с объектов, которые являются нелинейными, нестационарными и могут быть искажены помехами априори неизвестной природы. Стоит отметить, что актуальными являются задачи разработки как систем, которые обучаются с учителем, так и систем, которые обучаются без учителя.

**Связь работы с научными программами, планами, темами.** Диссертационная работа выполнена в рамках госбюджетных НИР «Нейро-фаззи системы для текущей кластеризации и классификации последовательностей данных в условиях их искаженности отсутствующими и аномальными наблюдениями» (№ ГР 0113U000361) и «Динамический интеллектуальный анализ последовательностей нечёткой информации в условиях существенной неопределённости на основе гибридных систем вычислительного интеллекта» (№ ГР 0116U002539), которые выполнялись согласно приказам Министерства образования и науки Украины по результатам конкурсного отбора проектов научных исследований. В рамках указанных НИР соискателем в качестве исполнителя разработаны адаптивные методы обучения эволюционирующих нейро-фаззи систем для решения задач прогнозирования, эмуляции и кластеризации в online режиме.

**Цели и задачи исследования.** Целью исследования является разработка эволюционирующих нейро-фаззи систем и методов их обучения, которые позволяют в online режиме настраивать не только синаптические

веса и параметры функций принадлежности, но и архитектуру системы в целом.

Для достижения поставленной цели необходимо решить такие научные задачи:

- проанализировать известные архитектуры нейро-фаззи систем и методы их обучения;

- разработать архитектуру и методы обучения многослойной нейро-фаззи системы, основанной на системе Ванга–Менделя и сочетающей в себе процессы эволюции архитектуры, самообучения функций принадлежности и контролируемого обучения синаптических весов;

- разработать методы обучения для настройки синаптических весов и параметров функций принадлежности нейро-фаззи узлов для улучшения аппроксимирующих свойств эволюционирующих нейро-фаззи систем;

- разработать методы обучения системы с нейро-фаззи узлами, основанной на методе группового учёта аргументов (МГУА), для обработки коротких выборок данных;

- разработать архитектуру и методы обучения каскадной нейро-фаззи и каскадной нео-фаззи системы для обработки выборок, которые содержат небольшое количество данных;

- разработать архитектуру и методы обучения эволюционирующей нейро-фаззи системы, основанной на аддитивной нелинейной авторегрессионной модели с экзогенными входами (ANARX-модели) для адаптивного прогнозирования нестационарных нелинейных временных рядов;

- разработать архитектуру и методы самообучения эволюционирующей нейро-фаззи системы для кластеризации данных, поступающих на обработку в последовательном online режиме, в условиях неопределённости относительно количества кластеров;

- выполнить имитационное моделирование разработанных методов и моделей и решить с их помощью практические задачи.

*Объект исследования* – процесс интеллектуального анализа данных в последовательном режиме.

*Предмет исследования* – эволюционирующие нейро-фаззи системы, предназначенные для интеллектуального анализа данных в последовательном режиме.

*Методы исследования.* Для проведения исследований применяются методы вычислительного интеллекта: теория искусственных нейронных сетей для синтеза новых архитектур эволюционирующих систем; теория нечёткой логики для реализации нечёткого вывода на основе разработанных архитектур; теория оптимизации для разработки методов настройки синаптических весов, параметров центра и ширины функций принадлежности; аппарат математической статистики для систематизации и использования полученных в результате работы данных для научных и практических выводов; имитационное моделирование для определения эффективности применения разработанных систем для различных наборов данных.

### **Научная новизна полученных результатов.**

1. Впервые предложен метод гибридного обучения эволюционирующей многослойной нейро-фаззи системы, сочетающий в себе процессы эволюции архитектуры, самообучения функций принадлежности и контролируемого обучения синаптических весов, что позволяет обрабатывать данные, поступающие последовательно в online режиме.

2. Впервые предложен метод обучения эволюционирующих систем на основе МГУА и каскадных систем вычислительного интеллекта с использованием в качестве узлов двухвходовых нейро-фаззи систем Ванга–Менделя с настройкой всех параметров функций принадлежности и двухвходовых нео-фаззи узлов, что позволяет обрабатывать данные в условиях коротких обучающих выборок.



3. Впервые предложен метод адаптивного прогнозирования нестационарных нелинейных временных рядов на основе эволюционирующей взвешенной нейро(нео)-фаззи аддитивной нелинейной авторегрессионной модели с экзогенными входами, позволяющий настраивать как синаптические веса, так и архитектуру, и предназначенный для обработки данных, поступающих в online режиме.

4. Усовершенствованы эволюционирующая нейро-фаззи кластеризирующая сеть Кохонена и метод её настройки, отличающиеся тем, что задача нечёткой кластеризации потока данных решается в online режиме в условиях априорной неопределённости относительно количества кластеров.

**Практическое значение полученных результатов.** Использование комплекса предложенных архитектур и методов обучения позволяет повысить эффективность применения эволюционирующих нейро-фаззи систем для решения задач прогнозирования и идентификации данных различной физической природы и кластеризации в online режиме. Полученные теоретические результаты были исследованы экспериментально на тестовых и реальных данных, где доказали своё преимущество над известными методами. Предложенные в работе методы обучения эволюционирующих нейро-фаззи систем обеспечивают оптимальную точность выходного сигнала в условиях априорной и потоковой неопределённости и могут быть использованы в разных областях, где данные представлены в числовой форме в виде таблиц «объект-свойство» или числовых последовательностей в режиме последовательной или пакетной обработки.

Предложенная взвешенная ANARX-система подтвердила свою эффективность в задаче прогнозирования медицинских показателей. Результаты исследований внедрены в ООО «Инфобуд», г. Харьков (акт внедрения от 03.10.2016). Предложенная эволюционирующая нейро-фаззи система использовалась для решения задачи кластеризации земельных

участков. Результаты исследований внедрены в ЧП «НВП «Амальгама-Строй», г. Харьков (акт внедрения от 06.10.2016).

**Личный вклад соискательницы.** Все положения, которые выносятся на защиту, основные результаты теоретических и экспериментальных исследований получены соискательницей лично. Вклад автора в публикациях, написанных в соавторстве, такой: [14] – предложен метод эволюционирующего самообучения первого слоя гибридной системы; [15] – разработан метод адаптивного обучения нейро-фаззи узлов эволюционирующей каскадной системы; [16] – разработан адаптивный метод обучения для возможностной нечёткой кластеризации; [17] – разработан метод обучения эволюционирующей каскадной системы с нео-фаззи узлами; [18] – предложен метод обучения взвешенной ANARX-системы для прогнозирования временных рядов; [19] – предложен метод обучения параметров центра и ширины функций принадлежности в нейро-фаззи узлах; [20] – разработана модификация правила самообучения Д. Хебба – Э. Оя; [21] – разработана модификация метода адаптивного самообучения; [24] – разработан метод обучения эволюционирующей ANARX-системы с нео-фаззи узлами для прогнозирования временных рядов; [25] – разработан метод настройки синаптических весов нейро-фаззи системы для решения задачи диагностики.

**Апробация результатов диссертации.** Основные результаты диссертационной работы были представлены и обсуждались на Международной научной конференции «Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта» (Херсон, 2014), VII и VIII Международной школе-семинаре «Теория принятия решений» (Ужгород, 2014, 2016), научной конференции физико-технического факультета Донецкого национального университета (Винница, 2015), 19-м Международном молодёжном форуме «Радиоэлектроника и молодёжь в XXI веке» (Харьков, 2015).

**Публикации.** Основные положения диссертационной работы опубликованы в 12 научных работах: 6 статьях в периодических специализированных изданиях по техническим наукам (из них 3 статьи опубликованы в специализированных изданиях, включенных в перечни МОН Украины, 3 статьи издано за границей); 6 публикаций в сборниках работ научных конференций. Все 6 статей входят в международные наукометрические базы, в том числе две – в SCOPUS, одна – в Web of Science.

# РАЗДЕЛ 1

## ОБЗОР СОСТОЯНИЯ ПРОБЛЕМЫ И ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ

### 1.1 Методы обучения и самообучения искусственных нейронных сетей

Одной из основных особенностей интеллектуальных систем является их способность проводить обобщения, основываясь на уже имеющихся и новых данных, или – другими словами – их способность к обучению.

Обучение – это процесс, в котором свободные параметры нейронной сети настраиваются посредством моделирования среды, в которую эта сеть встроена. Тип обучения определяется способом подстройки этих параметров [3]. Если же говорить с точки зрения математики, то обучение нейронных сетей можно определить как параметрическую задачу нелинейной оптимизации.

Естественно, что не существует такого универсального алгоритма обучения, который подходил бы для всех нейронных сетей. Поэтому следует рассмотреть разные парадигмы и модели, которые по-разному реализуют обучение систем.

Существуют три парадигмы обучения, на основании которых производится настройка параметров нейронных сетей [6, 26]:

- обучение с учителем (supervised learning). При применении этой парадигмы считается, что для каждого вектора  $x(k)$  из заданной обучающей выборки уже известен желаемый результат  $y(k)$ . Таким образом, к примеру, обучается персептрон. Настройка параметров сети осуществляется с использованием обучающего вектора и сигнала ошибки. Одним из недостатков этой парадигмы является то, что для обучения сети не всегда имеется достаточное количество образов с ответами;

- обучение без учителя (unsupervised learning). При применении этой парадигмы считается, что желаемые результаты работы нейронной сети не известны, а алгоритм обучения настраивает веса самостоятельно. Важно отметить то, что, в отличие от обучения «с учителем», в данном случае входные образы, по которым проводится обучение нейронной сети, не маркированы, т. е. для них не задан желаемый выход системы. При этом подходе осуществляется поиск закономерностей в обучающей выборке и проводится группирование похожих входных векторов по неявным признакам. Отличительной особенностью этой парадигмы является наличие меры качества представления, которому должна научиться система, а параметры нейронной сети должны оптимизироваться по отношению к этой мере. Одним из примеров применения этой парадигмы является алгоритм обучения Кохонена;

- обучение «с подкреплением» (обучение методом критики, reinforcement learning). При применении этой парадигмы считается, что есть возможность оценивать правильность работы сети и указывать желаемое направление обучения, т. е. формировать сигнал подкрепления. Этот подход можно обозначить как промежуточный между обучением «с учителем» и обучением «без учителя».

Архитектура нейронных сетей связана с алгоритмами, которые используются для её обучения. Согласно классификации, данной Хайкином [6], выделяют 3 основных класса архитектур нейронных сетей:

- однослойные сети прямого распространения. Они содержат единственный слой нейронов. Информация при этом передаётся от входного слоя на выходной;

- многослойные сети прямого распространения. Они содержат один или несколько скрытых слоёв. Обычно нейроны каждого из слоёв сети в качестве входных сигналов используют только выходные сигналы нейронов предыдущего слоя. Также выделяют полносвязные нейронные сети (когда

все узлы одного слоя соединены со всеми узлами смежных слоёв) и неполносвязные (когда некоторые синаптические связи отсутствуют);

- рекуррентные сети. Их отличительной особенностью является наличие одной или нескольких обратных связей.

- По Хайкину [6] существует 5 основных моделей обучения:

- обучение на основе коррекции ошибок (реализация метода оптимальной фильтрации);

- обучение с использованием памяти (с явным использованием обучающих данных);

- Хеббовское обучение (основано на нейробиологических принципах);

- конкурентное обучение (также основано на нейробиологических принципах);

- обучение по методу Больцмана (основано на идеях статистической механики).

Одним из самых распространённых является обучение, основанное на коррекции ошибок. Оно использует специальный механизм управления: к синаптическим весам нейрона применяется последовательность корректировок с целью приблизить выходной сигнал системы к желаемому. Корректировка зависит от полученного сигнала ошибки

$$e(k) = y(k) - \hat{y}(k), \quad (1.1)$$

где  $k$  – текущее дискретное время или порядковый номер наблюдения из обучающей выборки,

$y(k)$  – желаемый выход нейрона,

$\hat{y}(k)$  – фактический выходной сигнал нейрона.

Ещё одной часто используемой моделью обучения является конкурентное обучение. Как следует из названия, в данном случае в

нейронной сети выходные нейроны конкурируют между собой за «право» быть активизированными. В отличие от нейронной сети, основанной на обучении Хебба, в каждый момент времени в конкурентной сети только один нейрон может быть активным. Его называют нейроном-победителем, а принцип конкурентного обучения обычно формулируется как «победитель получает всё» (winner-takes-all, WTA) [28].

## 1.2 Нечёткая логика и нейро-фаззи системы

Одной из актуальных проблем современной науки является построение и использование систем, работа которых основана на модели рассуждения человека. Лотфи А. Заде осуществил значительное продвижение в этом направлении [29, 30]. Он ввёл понятие нечёткого множества (fuzzy set), для которого функция принадлежности элемента множеству (характеристическая функция) может принимать значения в интервале  $[0,1]$ . Он определил ряд операций над нечёткими множествами, а также ввёл понятие лингвистической переменной, значениями которой являются нечёткие множества. Таким образом, Л. Заде и его последователи основали новую теорию и создали аппарат для описания интеллектуальной деятельности, включающую неопределённость и нечёткость.

Формально чёткое подмножество  $A_c$  универсального множества  $E$  можно описать как множество упорядоченных пар

$$A_c = \{\mu_{A_c}(x)/x\}, \quad (1.2)$$

где  $\mu_{A_c}(x)$  – характеристическая функция, которая принимает значение 1, если  $x$  удовлетворяет некоторому свойству  $R$ , или 0 – в противном случае.

Аналогично нечёткое подмножество  $A_f$  универсального множества  $E$  можно определить как множество упорядоченных пар

$$A_f = \{ \mu_{A_f}(x) / x \}, \quad (1.3)$$

где  $\mu_{A_f}(x)$  – функция принадлежности, которая принимает значения из множества принадлежностей  $M = [0,1]$ .

Следует отметить, что характеристическая функция принадлежности определяет степень принадлежности элемента  $x$  нечёткому подмножеству  $A$ .

На основе правил нечёткой логики строятся системы нечёткого логического вывода. Более подробно они будут рассмотрены далее. Условно нечёткий логический вывод можно разбить на 4 этапа:

1. Введение нечёткости (фаззификация). На этом этапе функции принадлежности, определённые на входных переменных, применяются к их фактическим значениям с целью определить степень истинности каждой предпосылки каждого правила.

2. Логический вывод. На этом этапе вычисленное значение истинности для предпосылок каждого правила применяется к заключениям каждого правила. Каждой переменной вывода для каждого правила будет назначено нечёткое подмножество. В качестве правил логического вывода обычно используются операции  $\min$  (минимума) или  $\text{prod}$  (умножения). Для  $\min$  функция принадлежности вывода «отсекается» по высоте, соответствующей вычислительной степени истинности предпосылки правила (нечёткая логика «и»). Для  $\text{prod}$  функция принадлежности вывода масштабируется при помощи вычисленной степени истинности предпосылки правила.

3. Композиция. На этом этапе все нечёткие подмножества, назначенные к каждой переменной вывода, объединяются с целью



сформировать одно нечёткое подмножество для всех переменных вывода. При таком объединении обычно используются операции  $\max$  (максимума) или  $\text{sum}$  (суммы). При  $\max$  комбинированный вывод нечёткого подмножества конструируется как поточечный максимум по всем нечётким подмножествам (нечёткая логика «или»). При  $\text{sum}$  комбинированный вывод нечёткого подмножества формируется как поточечная сумма по всем нечётким подмножествам, назначенным переменной вывода правилами логического вывода.

4. Приведение к чёткости (дефаззификация). Оно производится, если нужно преобразовать нечёткий набор выводов в чёткое число.

Среди преимуществ применения аппарата нечётких множеств и нечёткой логики можно выделить следующие:

- условия и методы решения задачи могут быть описаны на языке, близком к естественному;
- системы, основанные на нечёткой логике, являются универсальными: согласно теореме о нечёткой аппроксимации (fuzzy approximation theorem), любая математическая система может быть аппроксимирована системой, основанной на нечёткой логике;
- системы, основанные на нечёткой логике, являются эффективными, что доказано рядом теорем, которые аналогичны теоремам о полноте для искусственных нейронных сетей.

Вместе с тем, у нечётких систем можно выделить и определённые недостатки:

- набор нечётких правил может оказаться неполным или противоречивым;
- функции принадлежности, их вид и параметры могут не вполне отражать реальную действительность, поскольку они выбираются субъективно.

Для того чтобы устранить указанные недостатки, нечёткие системы можно сделать адаптивными, т. е. нечёткие правила и параметры функций принадлежности будут настраиваться по мере работы такой системы. Одними из самых распространённых и эффективных вариантов таких систем являются гибридные нейронные системы.

Гибридные нейронные системы по своей структуре похожи на многослойные нейронные сети, но скрытые слои в них соответствуют этапам функционирования нечёткой системы. В общем виде первый слой гибридных нейронных систем вводит нечёткость на основе заданных функций принадлежности. Далее вводится совокупность нечётких правил. И, наконец, последние слои приводят результат к чёткому ответу. Следует отметить, что каждый из представленных слоёв имеет набор параметров, подлежащих настройке. К этим параметрам относятся параметры функций принадлежности, активационных функций, а также синаптические веса.

### 1.3 Эволюционирующие системы

Чтобы решить какую-либо задачу с применением искусственной нейронной сети или нейро-фаззи системы, необходимо сначала выбрать или спроектировать архитектуру системы, соответствующую поставленной задаче. Для этого следует выбрать, в частности, количество слоёв в сети, количество узлов в каждом слое и определить необходимые связи между слоями и между узлами. В связи с этим определение термина «обучения», данное выше, можно расширить, добавив в него также настройку архитектуры сети.

Количество узлов во входном слое системы обычно связано с размерностью входного вектора  $x(k)$ . Количество нейронов в выходном слое обычно принимается равным размерности ожидаемого на выходе вектора. Более сложной и существенной проблемой является выбор

количества скрытых слоёв системы, а также выбор количества узлов в каждом из этих слоёв.

Одним из подходов к получению оптимальной архитектуры нейронной сети является редукция. Алгоритмы редукции начинают работу с архитектуры сети, которая является избыточной. Далее она упрощается благодаря исключению наименее значимых весов, которые слабо влияют на функционирование и результаты работы системы.

Редукция системы проводится с целью уменьшения количества скрытых узлов, а также связей между слоями и отдельными узлами системы. Важным фактором является то, что редукция сети усиливает способность системы к обобщению [31].

Большинство методов редукции сети, которые применяются в настоящее время, можно разделить на две группы:

- методы, которые связаны с оценкой «чувствительности» целевой функции к удалению нейрона или отдельной связи (веса) [32]. В этих методах происходит удаление весов, которые оказывают наименьшее воздействие на значение целевой функции. Достаточно часто редукция происходит при сравнении значения анализируемого веса со средним значением весов всей системы или отдельных слоёв. Другой способ заключается в учёте чувствительности системы к вариациям весов: из системы исключаются только те связи, чувствительность к изменениям которых является минимальной. После этого уже на редуцированной варианте продолжается процесс обучения системы;

- методы, которые связаны с изменением целевой функции [33]. Целевая функция модифицируется таким образом, что в ней появляются специальные новые компоненты, которые «штрафуют» за неэффективную архитектуру системы. Например, отмечаются элементы, которые усиливают малые значения амплитуды весов. Но стоит отметить, что такие значения весов не всегда и не обязательно ослабляют их влияние на функционирование и результат работы системы. Поэтому, проводя

сравнение с методами первой группы, можно сказать, что методы второй группы являются менее эффективными.

Существует также другой подход к подбору архитектуры системы, принцип которого противоположен принципам подхода, рассмотренного выше. Речь идёт о таком способе, когда сначала для обучения берётся система, содержащая минимальное (иногда равное нулю) количество нейронов. Далее происходит последовательное добавление новых узлов и постепенное наращивание архитектуры до достижения требуемой точности и определённого уровня натренированности системы.

Именно этот подход лежит, в частности, в основе алгоритма каскадной корреляции Фальмана [34, 35]. Среди недостатков такого подхода можно выделить то, что многие методы, реализующие его, не всегда эффективно работают с данными, представленными векторами большой размерности.

В целом, этот подход (часто называемый конструктивным) реализует поиск архитектуры путем итеративного изменения сложности выбранной модели. При этом стоит отметить, что решение о способе модификации сети принимается в соответствии с некоторым заданным алгоритмом построения. Важным является то, что алгоритм построения сети задается таким образом, чтобы добавление новых узлов или связей на каждом этапе гарантированно уменьшало (или не увеличивало) значение ошибки работы сети.

Применение рассмотренных подходов к построению архитектуры нейронной сети очень тесно связано с тем, что большинство процессов в реальном мире имеет эволюционный характер.

Эволюционирующий процесс – это процесс, который непрерывно развивается, изменяется с течением времени [12]. Моделирование эволюционирующих процессов является достаточно сложной задачей, поскольку некоторые законы эволюции могут быть априори неизвестными и могут динамически изменяться. Система, моделирующая реальный эволюционирующий процесс, должна быть способна отслеживать динамику

этого процесса и адаптироваться к изменениям в этом процессе. Например, система распознавания речи должна уметь адаптироваться к новым акцентам и новым языкам. Система, которая моделирует решение когнитивных задач мозга человека, должна быть адаптивной, поскольку все когнитивные процессы по своей природе являются эволюционирующими. В биоинформатике система моделирования экспрессии генов должна быть способна адаптироваться к новой информации, которая может определять, как на один ген влияет другой ген, на который, в свою очередь, влияет третий ген. Существует огромное количество задач, в которых процессы эволюционируют с течением времени.

Зная правила, по которым эволюционирует процесс, можно создать модель, эволюционирующую так же, как и реальный процесс, а затем использовать эту модель для понимания и прогнозирования реального процесса. Но без знания этих правил существует также возможность вывести их из данных-результатов этого процесса.

В зависимости от типа эволюционирующих правил, поведение эволюционирующего процесса можно определить как [12]:

- случайное: развитие процесса невозможно описать никакими правилами и процесс невозможно спрогнозировать;

- хаотическое: процесс является прогнозируемым, но только на небольшой промежуток времени, поскольку процесс в заданный момент времени нелинейно зависит от процесса в предыдущие моменты времени. Обычно хаотические процесс описываются математическими уравнениями, которые используют некоторые параметры для оценки следующего состояния процесса в зависимости от его предыдущих состояний;

- квазипериодическое: процесс является прогнозируемым с определённой ошибкой прогноза. Эволюционирующие правила меняются незначительно со временем;

- периодическое: процесс повторяет шаблон поведения через определённые промежутки времени и является полностью прогнозируемым.

Процесс можно описать фиксированным набором правил, которые не меняются с течением времени.

Системы, моделирующие реальные эволюционирующие процессы, должны непрерывно настраивать свою структуру и параметры для улучшения качества своей работы.

Среди достаточно распространённых методов вычислительного интеллекта можно выделить искусственные нейронные сети, системы нечёткого вывода и гибридные системы.

Несмотря на достаточно успешное их применение для решения широкого круга задач интеллектуального анализа данных, эти системы обладают рядом недостатков, связанных с их использованием для анализа и работы с эволюционирующими процессами. Среди наиболее существенных недостатков можно выделить такие:

1. Сложности в выборе архитектуры системы. Как правило, модель, основанная на принципах вычислительного интеллекта, имеет фиксированную архитектуру. В контексте искусственных нейронных сетей (ИНС) это означает, что нейронная сеть имеет фиксированное количество нейронов и связей. В связи с этим, адаптация системы к новым поступающим на обработку данным, имеющим отличающуюся от предыдущих данных природу, может оказаться затруднительным.

2. Обучение в пакетном режиме и обучение в течение нескольких эпох требует значительных временных ресурсов. Такие системы не приспособлены для работы в online режиме с достаточно высоким темпом поступления новых данных на обработку.

3. Многие из существующих систем вычислительного интеллекта не могут извлекать эволюционирующие правила, по которым развивается эта система, а также могут представлять результаты своей работы в терминах естественного языка.

Таким образом, актуальной является задача разработки новых архитектур и методов обучения, которые позволят избежать указанных трудностей.

Эволюционирующая интеллектуальная система (evolving intelligent system, EIS) – это информационная система, которая развивает свою архитектуру, функциональность и знания непрерывно, самоорганизованно, адаптивно и интерактивно на основании поступающей информации и которая выполняет интеллектуальные задачи, типичные для человека (например, адаптивное распознавание образов, интеллектуальное управление), улучшая таким образом свою производительность [11-13].

Эволюционирующая коннекционистская система (evolving connectionist system, ECOS) – это адаптивная система с инкрементным обучением и представлением знаний, которая изменяет свою структуру и функциональность на протяжении своей работы и которая основывается на коннекционистской архитектуре, состоящей из искусственных нейронов и связей между ними [12]. Другими словами можно сказать, что эволюционирующая коннекционистская система – это система вычислительного интеллекта, основанная на нейронных сетях, которая непрерывно адаптирует свою структуру при поступлении на обработку данных.

Одной из первых и наиболее эффективных систем данного типа можно назвать систему, основанную на эволюционирующем методе кластеризации.

Эволюционирующий метод кластеризации (evolving clustering method) – это быстрый алгоритм для динамической кластеризации входного потока данных [36]. В этом методе количество кластеров заранее не задано, а процесс настройки центров основывается на вычислении расстояний. Для каждого кластера максимальное расстояние между вектором  $x$  и ближайшим центром кластера не может превышать заданное пороговое значение  $D_{thr}$ .

Центры кластеров представлены эволюционирующими узлами. Важной особенностью является то, что этот метод обрабатывает данные за одну эпоху.

Процесс кластеризации начинается с пустого множества кластеров. Когда создается новый кластер, определяется его центр  $C_c$ . Начальное значение радиуса  $R_i$  равно 0.

При поступлении новых обучающих примеров некоторые уже созданные кластеры будут модифицироваться, при этом изменяя положение центра и увеличивая радиус.

Для кластеризации задается пороговое значение  $D_{thr}$ . Это максимальное расстояние между элементом выборки, который принадлежит одному кластеру и наиболее удален от его центра, и центром этого кластера. Этот параметр влияет на размер кластеров и их количество.

Пороговое значение может быть настраиваемым в процессе адаптивной кластеризации в зависимости от некоторого критерия (например, текущая ошибка, количество кластеров и т. п.)

Эволюционирующий метод кластеризации можно представить в виде последовательности шагов:

Шаг 1: создаётся первый кластер  $C_1$ , используя первый образ из потока входных данных. Его позиция обозначается как центр первого кластера  $C_{c_1}$ , а начальное значение радиуса кластера  $R_{i_1}$  устанавливается равным 0.

Шаг 2: если рассмотрены все примеры из потока данных, процесс кластеризации завершается. Иначе вычисляется нормализованное евклидово расстояние  $d(i, j)$  между текущим входным образом и всеми  $n$  центрами кластеров:

$$d(i, j) = \|x_i - C_{c_j}\|, j = 1, 2, \dots, n, \quad (1.4)$$



где  $x_i$  – текущий входной образ,

$Cc_j$  – центры кластеров.

Шаг 3: если существует кластер  $C_m$ , для которого выполняются условия (1.5) и (1.6), то образ  $x_i$  принадлежит кластеру  $C_m$ .

$$d(i, j) = \min_j d(i, j) = \min_j (\|x_i - Cc_j\|), j = 1, 2, \dots, n, \quad (1.5)$$

$$d(i, m) \leq Ru_m, \quad (1.6)$$

где  $Cc_m$  – центр кластера  $C_m$ ,

$Ru_m$  – радиус кластера  $C_m$ ,

$d(i, m)$  – расстояние между центром кластера  $C_m$  и примером  $x_i$ .

В этом случае не создается новый кластер и ни один из существующих кластеров не изменяется. Затем осуществляется переход к шагу 2. Если условие не выполняется, то осуществляется переход к шагу 4.

Шаг 4: найти кластер  $C_a$  с центром  $Cc_a$ . Для этого необходимо вычислить значения расширенного расстояния по формуле (1.7), используя радиус кластера  $Ru_a$  и расстояние  $d(i, a)$  до всех  $n$  кластеров, а затем выбрать кластер  $C_a$  с минимальным значением величины  $s(i, a)$  по формуле (1.8).

$$s(i, j) = d(i, j) + Ru_j, j = 1, 2, \dots, n, \quad (1.7)$$

$$s(i, a) = d(i, a) + Ru_a = \min_j s(i, j), j = 1, 2, \dots, n, \quad (1.8)$$

где  $Ru_a$  – радиус кластера  $C_a$ ,

$d(i, a)$  – расстояния до каждого кластера,

$s(i, a)$  – расширенное расстояние до каждого кластера.

Шаг 5: если выполняется неравенство (1.9), то пример  $x_i$  не может принадлежать ни к одному из уже сформированных кластеров.

$$s(i, a) > 2Dthr. \quad (1.9)$$

В этом случае необходимо создать новый кластер. Для этого нужно выполнить все действия, которые описаны на шаге 1. Затем осуществляется переход к шагу 2.

Если условие не выполняется, то осуществляется переход к шагу 6.

Шаг 6: если выполняется неравенство (1.10), то кластер  $C_a$  изменяется.

$$s(i, a) \leq 2Dthr. \quad (1.10)$$

Значение модифицированного радиуса  $Ru_a^{new}$  вычисляется по формуле:

$$Ru_a^{new} = \frac{s(i, a)}{2}. \quad (1.11)$$

Новый центр  $Cc_a^{new}$  размещается на прямой, которая соединяет входной вектор  $x_i$  и предыдущий центр кластера  $Cc_a$ , таким образом, чтобы расстояние между новым центром  $Cc_a^{new}$  и точкой примера  $x_i$  равнялось  $Ru_a^{new}$ . Затем осуществляется переход к шагу 2.

Максимальное расстояние от любого центра кластера к наиболее удаленному примеру из этого кластера находится в пределах порогового

значения  $Dthr$ , несмотря на то, что алгоритм не хранит информацию об уже рассмотренных примерах.

Целевая функция в представленном алгоритме простая. Она гарантирует, что для каждого образа  $x$  существует центр кластера  $C_j$  такой, что расстояние между  $x$  и центром кластера  $C_j$  меньше, чем предопределенное пороговое значение  $Dthr$ .

Эволюционирующие правила для эволюционирующего метода кластеризации включают в себя:

- правило для создания нового кластера;
- правило для модификации существующего кластера.

Рассмотрим пример работы эволюционирующего метода кластеризации в двумерном пространстве [36].

На вход алгоритма последовательно представлены образы 9 образов: от  $x_1$  до  $x_9$ .

На рис. 1.1 представлен первый этап работы алгоритма ЭМК. Для создания первого кластера  $C_1^0$  используется первый образ  $x_1$  из входного потока данных. Его положение определяет центр первого кластера  $C_{C_1^0}$ . Значение радиуса этого кластера  $Ru_1^0$  равно 0.

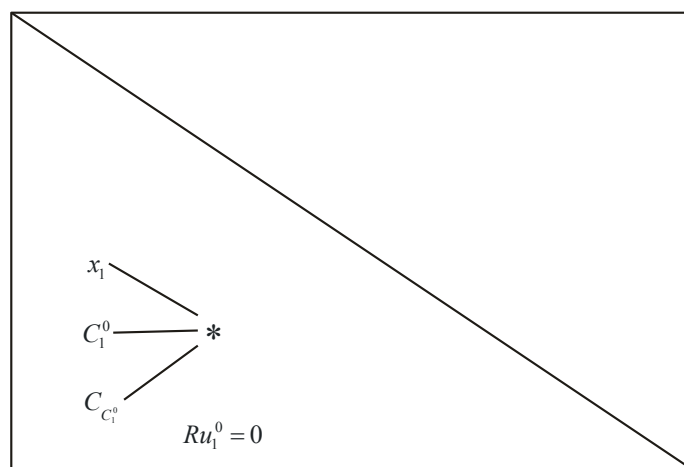


Рисунок 1.1 – Первый этап работы алгоритма ЭМК

Обработка образа  $x_2$  влечет за собой модификацию кластера  $C_1^0 \rightarrow C_1^1$  как представлено на рис. 1.2 (смещение центра данного кластера в положение  $C_{c_1}^1$  и соответствующее изменение его радиуса  $Ru_1^1$ ).

Обработка образа  $x_3$  влечет за собой создание нового кластера  $C_2^0$ . Его центр  $C_{c_2}^0$  определяется положением  $x_3$ . Значение радиуса кластера  $Ru_2$  равно 0.

Образ  $x_4$  принадлежит кластеру  $C_1^1$ , поскольку выполняются условия (1.5) и (1.6). Модификации кластеров нет.

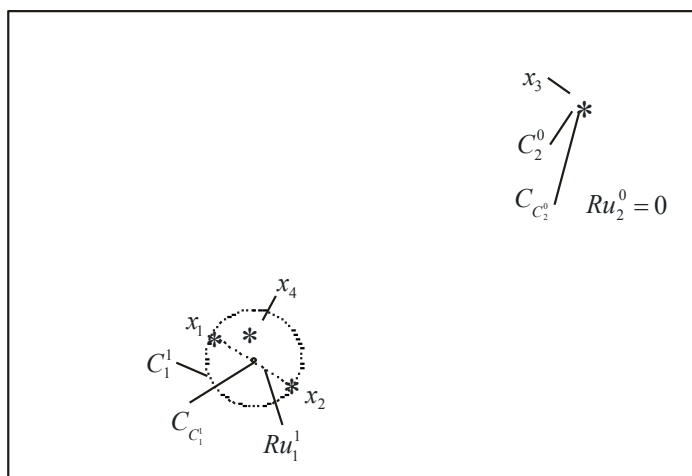


Рисунок 1.2 – Пример работы алгоритма ЭМК

Обработка образа  $x_5$  влечет за собой модификацию кластера  $C_1^1 \rightarrow C_1^2$  как представлено на рис. 1.3 (смещение центра данного кластера в положение  $C_{c_1}^2$  и соответствующее изменение его радиуса  $Ru_1^2$ ).

Образ  $x_6$  принадлежит кластеру  $C_1^2$ , поскольку выполняются условия (1.5) и (1.6). Модификации кластеров нет.

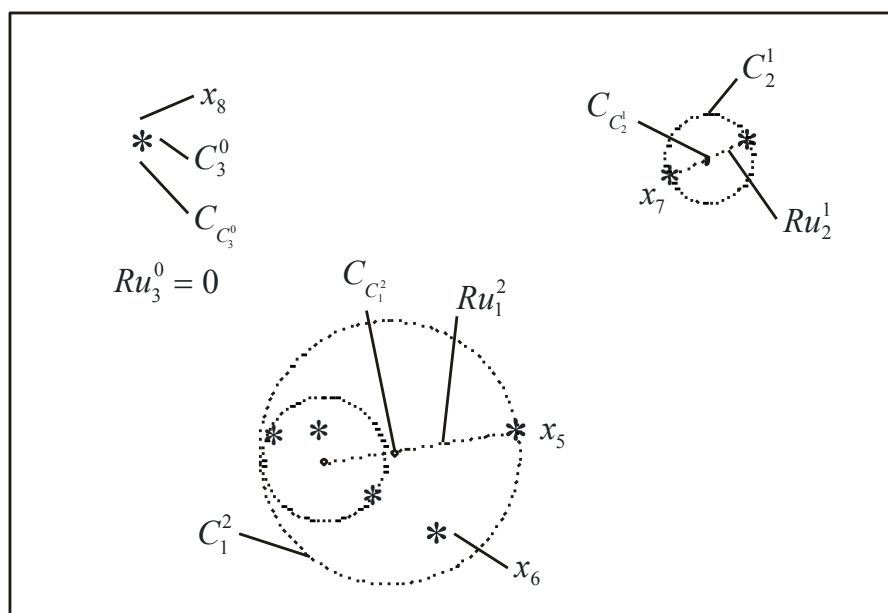


Рисунок 1.3 – Пример работы алгоритма ЭМК

Обработка образа  $x_7$  влечет за собой модификацию кластера  $C_2^0 \rightarrow C_2^1$  как представлено на рис. 1.3 (смещение центра данного кластера в положение  $Cc_2^1$  и соответствующее изменение его радиуса  $Ru_2^1$ ).

Обработка образа  $x_8$  влечет за собой создание нового кластера  $C_3^0$ . Его центр  $Cc_3^0$  определяется положением  $x_8$ . Значение радиуса кластера  $Ru_3$  равно 0.

Обработка образа  $x_9$  влечет за собой модификацию кластера  $C_1^2 \rightarrow C_1^3$  как представлено на рис. 1.4 (смещение центра данного кластера в положение  $Cc_1^3$  и соответствующее изменение его радиуса  $Ru_1^3$ ).

Рассмотренный подход также исследуется в рамках так называемых эволюционирующих нейро-фаззи систем.

Эволюционирующие нейро-фаззи системы – это нейро-фаззи системы, которые изменяют свою структуру согласно принципам эволюционирующих коннекционистских систем.

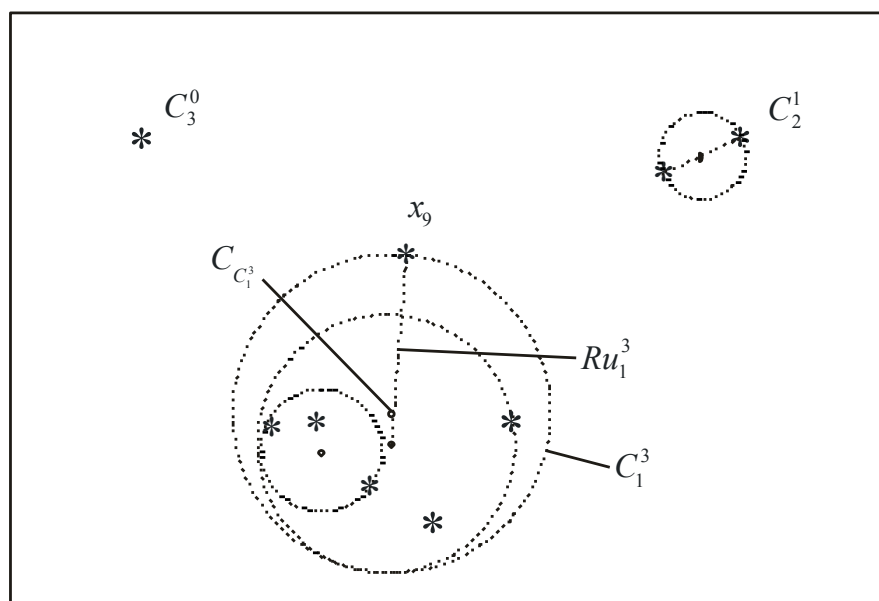


Рисунок 1.4 – Пример работы алгоритма ЭМК

Узлы в эволюционирующих нейро-фаззи системах создаются в процессе обучения. При этом параметры функций принадлежности, которые представляют существующие узлы, также могут модифицироваться и настраиваться в процессе обучения. Существует целый ряд функций (например, треугольные, гауссианы и т. д.), которые могут быть использованы в качестве функций принадлежности в этих системах. Новые узлы эволюционируют в системе согласно определённым правилам. Среди таких правил часто используется проверка условия, которое использует заданное пороговое значение. С этим пороговым значением сравниваются значения уровней принадлежности обрабатываемого вектора. В зависимости от соблюдения или несоблюдения заданных условий производится добавление новых узлов системы. Кроме значений уровней принадлежности вектора, в качестве критерия условия добавления или изменения узлов системы также может использоваться расстояние от этого вектора до центров функций принадлежности.

Примером такой системы является эволюционирующая нечёткая нейронная сеть (Evolving Fuzzy Neural Network, EFuNN) [9, 10].

EFuNN является сетью прямого распространения, содержащей пять слоёв нейронов. При этом, каждый из этих слоёв выполняет определённую функцию.

Первый слой – это входной слой (input layer).

Второй слой – это слой условия (condition layer). Каждый узел в этом слое представляет одну функцию принадлежности (треугольную, гауссиан или др.). Для каждого входа из первого слоя есть своё множество функций принадлежности во втором слое. Количество и тип функций принадлежности могут динамически изменяться. Второй слой выполняет фаззификацию входных переменных. Вес связи между условным нейроном и входом определяет центр соответствующей функции принадлежности. В то же время, нижняя и верхняя граница этой функции определяется центрами соседних функций принадлежности.

Третий слой – это эволюционирующий слой (evolving layer), также называемый слоем правил. На этом слое вычисляется расстояние между фаззифицированным входным вектором и вектором весов. Этот слой эволюционирует благодаря методам обучения «с учителем» и/или обучения «без учителя».

Четвёртый слой – это слой действия (action layer). Нейроны в этом слое представляют функции принадлежности, связанные с выходными нейронами. Вес связи, соединяющий узел из этого слоя, с узлом выходного слоя определяет центр соответствующей функции принадлежности.

Пятый слой – это выходной слой системы. Он производит дефаззификацию нечётких значений, полученных на выходе предыдущего слоя.

На рисунке представлен пример EFuNN с тремя входными нейронами. Второй слой содержит 7 узлов: 2 связаны с первым нейроном первого слоя, 3 – со вторым, 2 – с третьим. Третий слой содержит три узла с правилами. Система имеет 2 выхода, каждый из которых связан с двумя узлами четвёртого слоя.

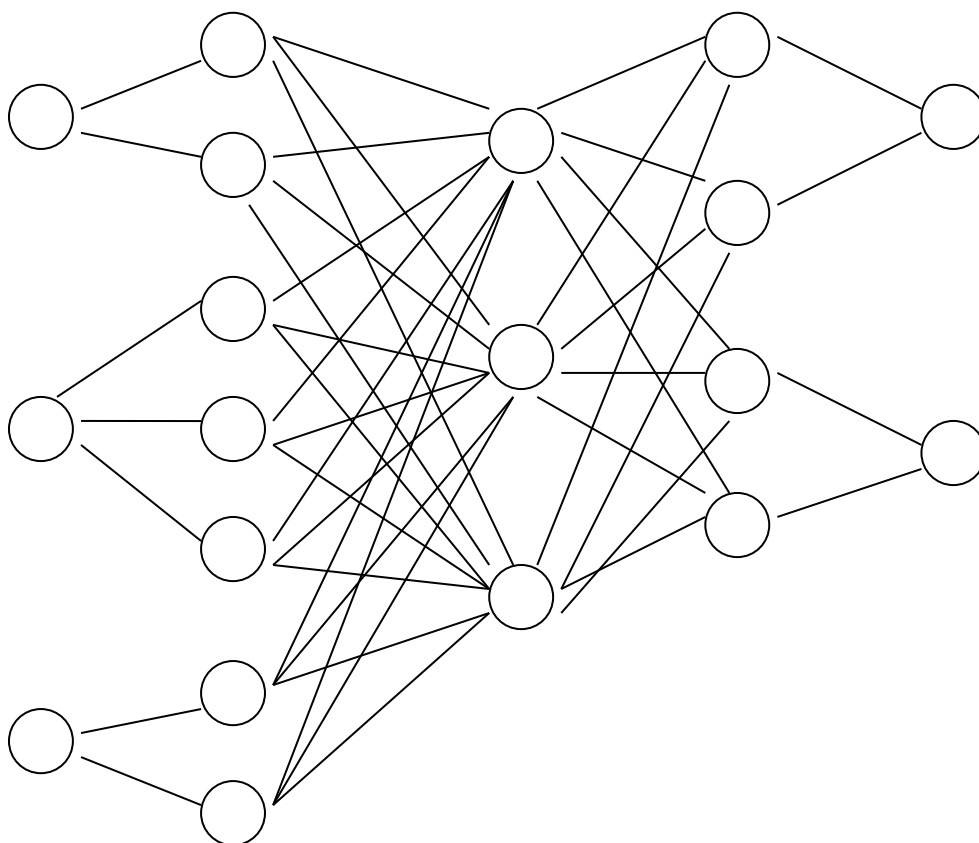


Рисунок 1.5 – Упрощённая схема примера EFuNN

Известно, что EFuNN нашла широкое применение для решения ряда задач, причём как самостоятельная система, так и как часть ансамбля систем [72].

Среди других систем, которые для настройки архитектуры применяют эволюционирующий подход, можно также выделить Simple Evolving Connectionist System (SECoS), Evolving Self-Organising Map (ESOM), Dynamic Evolving Neural-Fuzzy Inference System (DENFIS), Evolving Fuzzy Inference System (EFIS).

В целом, можно сказать, что преимущества эволюционирующих коннекционистских систем заключаются в том, что они помогают избежать проблем, связанных с применением традиционных архитектур нейронных



сетей, в частности – многослойного персептрона. Среди основных преимуществ можно выделить следующие [37]:

- практически отсутствует проблема переобучения благодаря конструктивной природе алгоритмов обучения и постепенному усложнению архитектуры системы;

- обучение происходит быстро, поскольку алгоритмы настройки параметров такой системы требуют представления обучающих данных только один раз (в отличие от многих систем, которые требуют многократную обработку входных данных в течение нескольких эпох);

- для большинства таких систем существует возможность непрерывного обучения и настройки архитектуры в течение всего времени работы системы, что улучшает аппроксимирующие способности системы в условиях, когда данные имеют нелинейный и нестационарный характер;

- их можно применять для решения широкого круга задач интеллектуального анализа данных.

#### 1.4 Постановка задачи исследования

Проведенный обзор состояния проблемы позволяет сделать вывод, что на сегодняшний день актуальной и важной является задача разработки эволюционирующих нейро-фаззи систем и методов их обучения, предназначенных для решения задач интеллектуального анализа данных. Поскольку существующие архитектуры и методы имеют ряд недостатков, которые затрудняют работу этих систем в online режиме, то очевидно, что существует необходимость в разработке новых методов, которые должны решить описанные выше проблемы.

Для достижения поставленной цели необходимо рассмотреть ряд вопросов и решить следующие задачи:

- выполнить обзор и анализ известных на данное время архитектур нейро-фаззи систем и методов их обучения;

- разработать архитектуру и методы обучения многослойной нейро-фаззи системы, основанной на системе Такаги–Сугено–Канга нулевого порядка (системе Ванга–Менделя) с гибридным обучением;
- разработать методы обучения для настройки синаптических весов и параметров функций принадлежности нейро-фаззи узлов;
- разработать методы обучения системы для обработки коротких выборок данных, которая основывается на методе группового учёта аргументов (МГУА);
- разработать методы обучения эволюционирующей каскадной системы, которая в качестве узлов использует двухвходовые нейро-фаззи или нео-фаззи системы;
- разработать архитектуру и методы обучения эволюционирующей нейро-фаззи системы, основанной на ANARX-модели и предназначенной адаптивного прогнозирования нестационарных нелинейных временных рядов;
- разработать архитектуру и методы самообучения эволюционирующей кластеризирующей нейро-фаззи системы для обработки данных в online режиме с наращиванием количества кластеров;
- выполнить имитационное моделирование разработанных методов и решить с их помощью практические задачи.

#### Выводы к разделу 1

1. Проведен обзор парадигм обучения, основных классов архитектур искусственных нейронных сетей, методов обучения и самообучения нейронных систем.
2. Проведен обзор теории нечётких множеств, нечёткого логического вывода, рассмотрены особенности нейро-фаззи систем, гибридного обучения и известных систем нечёткого вывода.

3. Проанализированы существующие на данный момент подходы к выбору и настройке архитектур искусственных нейронных сетей и нейро-фаззи систем. Показаны преимущества и недостатки редукции сети. Выполнен обзор эволюционирующего подхода, приведены его характеристики и преимущества по сравнению с традиционными архитектурами нейронных сетей. Проведен обзор существующих эволюционирующих коннекционистских систем.

4. Сформулированы задачи исследования.

## РАЗДЕЛ 2

### ЭВОЛЮЦИОНИРУЮЩИЕ НЕЙРО-ФАЗЗИ СИСТЕМЫ ДЛЯ ОБУЧЕНИЯ С УЧИТЕЛЕМ

#### 2.1 Система Такаги – Сугено – Канга

Одной из наиболее популярных нечётких систем адаптивного типа является модель вывода Такаги – Сугено – Канга (TSK) [31, 76, 77]. В этой модели на выходе системы нет дефаззификатора, поскольку функция заключения определяется нечётким, но точечным образом. Благодаря этому модель вывода в такой системе значительно упрощается.

Модель TSK в общей форме может быть представлена в виде

$$\text{If } x_1 = A_1 \text{ and } x_2 = A_2 \text{ and } \dots \text{ and } x_N = A_N, \text{ then } y = f(x_1, x_2, \dots, x_N), \quad (2.1)$$

где  $f(x_1, x_2, \dots, x_N)$  – некоторая чёткая функция,

$N$  – количество правил.

Функциональная зависимость чаще всего представляется в виде полиномиальной функции нескольких переменных, например, в виде полинома первого порядка

$$y = f(x) = p_0 + \sum_{i=1}^N p_i x_i, \quad (2.2)$$

где  $p_0, p_1, \dots, p_N$  – веса, настраиваемые в процессе обучения.

Также на практике довольно часто применяется более простая модель вывода TSK, в которой полином нулевого порядка имеет вид

$$y = f(x) = p_0. \quad (2.3)$$

Выход системы, реализующей модель вывода TSK, в которой используется несколько правил, определяется как их средневзвешенное значение. Выходной сигнал такой системы можно представить в виде

$$y = \frac{\sum_{i=1}^M w_i y_i}{\sum_{j=1}^M w_j} = \sum_{i=1}^M \frac{w_i}{\sum_{j=1}^M w_j} y_i. \quad (2.4)$$

Важно отметить, что для весов  $w_i$  соблюдается условие нормализации

$$\sum_{i=1}^M \frac{w_i}{\sum_{j=1}^M w_j} = 1. \quad (2.5)$$

Таким образом, если для каждого  $j$ -го правила вывода ( $j=1,2,\dots,M$ ) реализуется функция TSK – полином первого порядка

$$y_i = p_{i0} + \sum_{j=1}^N p_{ij} x_j, \quad (2.6)$$

то на выходе модели TSK формируется сигнал, который может быть записан в виде

$$y = \sum_{i=1}^M \frac{w_i}{\sum_{j=1}^M w_j} \left( p_{i0} + \sum_{j=1}^N p_{ij} x_j \right). \quad (2.7)$$

Если в модели TSK используется  $M$  правил вывода и  $N$  переменных  $x_j$ , то обобщённая схема вывода в такой модели имеет вид

$$\begin{aligned}
 & \text{If } (x_1 \text{ is } A_1^{(1)}) \text{ and } (x_2 \text{ is } A_2^{(1)}) \text{ and } \dots \text{ and } (x_N \text{ is } A_N^{(1)}), \text{ then } y_1 = p_{10} + \sum_{j=1}^N p_{1j} x_j; \\
 & \dots \\
 & \text{If } (x_1 \text{ is } A_1^{(M)}) \text{ and } (x_2 \text{ is } A_2^{(M)}) \text{ and } \dots \text{ and } (x_N \text{ is } A_N^{(M)}), \text{ then } y_M = p_{M0} + \sum_{j=1}^N p_{Mj} x_j.
 \end{aligned} \tag{2.8}$$

## 2.2 Эволюционирующая многослойная нейро-фаззи система с гибридным обучением

2.2.1 Архитектура эволюционирующей многослойной нейро-фаззи системы. Архитектура рассматриваемой системы, приведенная на рис. 2.1, состоит из пяти последовательных слоев. На входной (нулевой) слой нейро-фаззи системы подается  $(n \times 1)$ -мерный вектор сигналов-образов  $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$  (здесь  $k = 1, 2, \dots$  – текущее дискретное время), подлежащих обработке. Первый скрытый слой содержит  $nh$  функций принадлежности  $\mu_{li}(x_i)$ ,  $i = 1, 2, \dots, n$ ;  $l = 1, 2, \dots, h$ . Таким образом, для каждого входа задается  $h$  функций принадлежности. Первый скрытый слой выполняет фаззификацию входного пространства. При этом подразумевается, что в процессе обучения-эволюции должны настраиваться как собственно параметры этих функций, так и их число: на рис. 1 – количество узлов первого скрытого слоя  $\mu_h$ . Второй скрытый слой обеспечивает агрегирование уровней принадлежности, рассчитанных в первом скрытом слое, и состоит из  $h$  блоков умножения. Третий скрытый слой – это слой настраиваемых синаптических весов  $w_1, w_2, \dots, w_h$ , подлежащих определению в процессе контролируемого обучения.

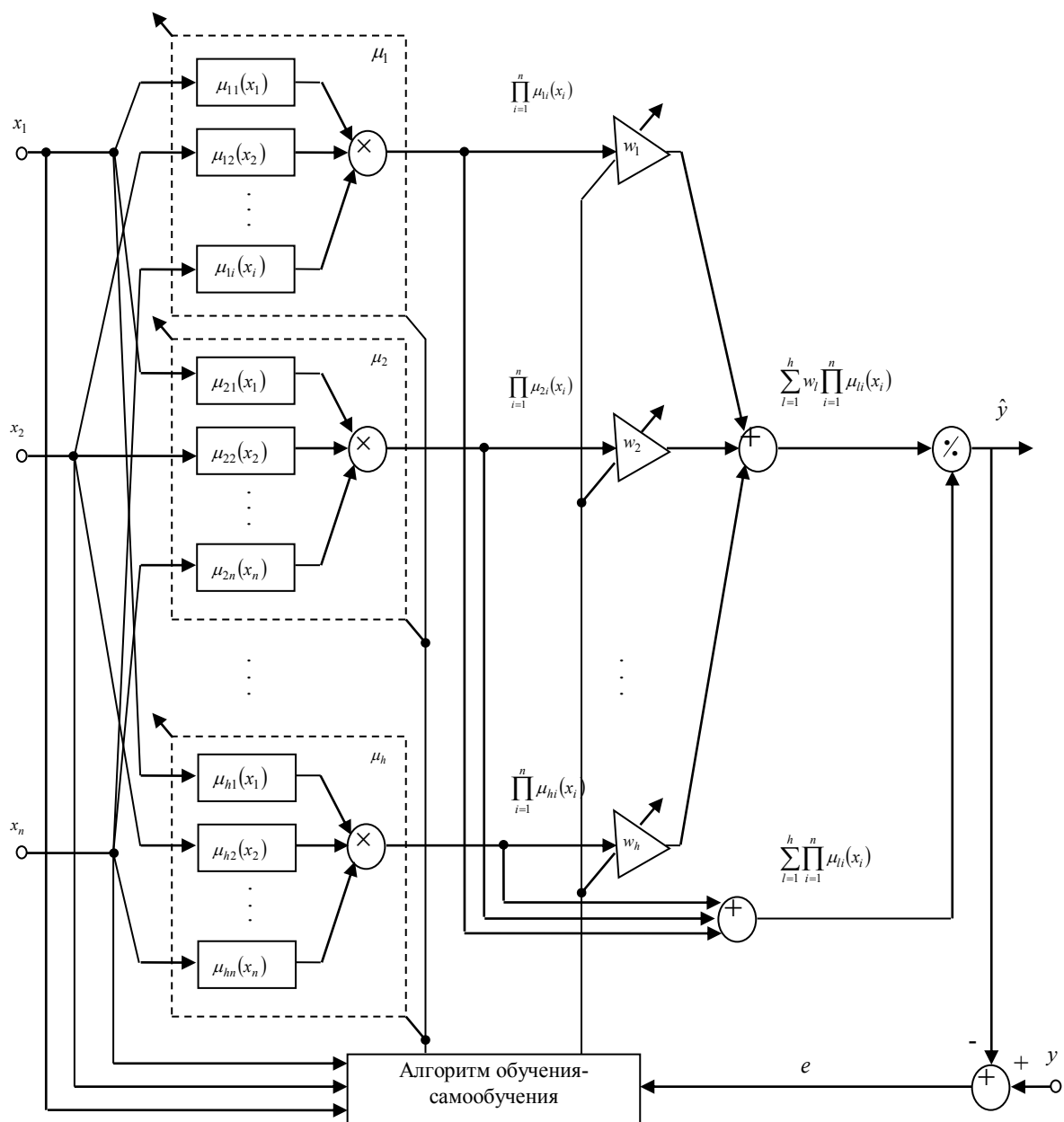


Рисунок 2.1 – Архитектура эволюционирующей многослойной нейро-фаззи системы

Четвертый скрытый слой образован двумя сумматорами и вычисляет суммы выходных сигналов второго и третьего скрытого слоев.

И, наконец, в пятом (выходном) слое производится дефаззификация, в результате которой вычисляется выходной сигнал NFS  $\hat{y}(k)$ .

Таким образом, если на вход нейро-фаззи системы подан векторный сигнал  $x(k)$ , элементы первого скрытого слоя производят его фаззификацию, вычисляя уровни принадлежности  $0 < \mu_{li}(x_i(k)) \leq 1$ . Обычно в качестве функций принадлежности используются традиционные гауссианы

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma_i^2}\right), \quad (2.9)$$

где  $c_{li}$  – параметр центра  $l$ -й функции принадлежности  $i$ -го входа,

$\sigma_i$  – параметр ширины функции принадлежности  $i$ -го входа.

Стоит отметить, что предварительное нормирование данных на некоторый интервал, например,  $-1 \leq x_i(k) \leq 1$  позволяет упростить расчеты, поскольку параметры ширины  $\sigma_i$  могут быть приняты одинаковыми для всех входов, т. е.  $\sigma_i = \sigma$ .

Кроме гауссианов (2.9) могут быть использованы и иные ядерные функции, например,  $B$ -сплайны, отвечающие условию единичного разбиения, четные вэйвлеты, гибкие функции активации-принадлежности [38, 39] и т. д.

Во втором скрытом слое вычисляются агрегированные значения

$\prod_{i=1}^n \mu_{li}(x_i(k))$ , при этом для гауссианов с одинаковыми параметрами ширины



$$\prod_{i=1}^n \mu_{li}(x_i(k)) = \prod_{i=1}^n \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = \exp\left(-\frac{\|x(k) - c_l\|^2}{2\sigma^2}\right), \quad (2.10)$$

где  $c_l = (c_{l1}, c_{l2}, \dots, c_{ln})^T$ .

Таким образом, сигналы на выходах блоков умножения второго скрытого слоя аналогичны сигналам на выходах нейронов первого скрытого слоя обычных радиально-базисных нейронных сетей (RBFN) [6].

Выходами третьего скрытого слоя есть значения  $w_l \prod_{i=1}^n \mu_{li}(x_i(k))$ , четвертого –  $\sum_{l=1}^h w_l \prod_{i=1}^n \mu_{li}(x_i(k))$  и  $\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))$  и, наконец, на выходе системы (пятого выходного слоя) появляется сигнал

$$\begin{aligned} \hat{y}(x(k)) &= \frac{\sum_{l=1}^h w_l \prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \sum_{l=1}^h w_l \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))} = \\ &= \sum_{l=1}^h w_l \varphi_l(x(k)) = w^{hT} \varphi^h(x(k)) \end{aligned} \quad (2.11)$$

$$\text{где } \varphi_l(x(k)) = \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{l=1}^h \prod_{i=1}^n \mu_{li}(x_i(k))},$$

$$w^h = (w_1, w_2, \dots, w_h)^T,$$

$$\varphi^h(x(k)) = (\varphi_1(x(k)), \varphi_2(x(k)), \dots, \varphi_h(x(k)))^T.$$

Несложно заметить, что рассматриваемая система реализует нелинейное отображение входного пространства в скалярный выходной сигнал ( $R^n \rightarrow R^1$ ) подобно нормализованной RBFN [40], а по архитектуре

(при фиксированном  $h$ ) совпадает с TSK-системой нулевого порядка, т. е. архитектурой Ванга–Менделя [41].

2.2.2 Самообучение-эволюция первого скрытого слоя. Процесс настройки параметров и количества функций принадлежности происходит следующим образом. Пусть на вход системы, у которой в исходном состоянии в первом скрытом слое отсутствуют функции принадлежности, поступает первое наблюдение обучающей выборки  $x(1) = (x_1(1), x_2(1), \dots, x_n(1))^T$ . Это наблюдение формирует первый узел первого скрытого слоя  $\mu_1$  так, что  $c_{1i}(1) = x_i(1)$ . Таким образом создается  $n$  функций принадлежности и формируется единственный синаптический вес  $w_1(0)$ , задаваемый случайно в интервале  $-1 \leq w_1(0) \leq 1$ .

Далее для этой функции принадлежности  $\mu_1$  задается радиус соседства  $r$ , определяемый максимально возможным числом функции принадлежности в NFS  $h$ . Так, если функции принадлежности по осям распределены равномерно, то

$$r = \frac{2}{h-1}. \quad (2.12)$$

Далее при поступлении второго наблюдения  $x(2)$  производится проверка условия

$$\max_i |c_{1i} - x_i(2)| \leq r. \quad (2.13)$$

Если это условие выполняется, то производится коррекция центров функций принадлежности узла  $\mu_1$  согласно правилу

$$c_{1i}(2) = c_{1i}(1) + \eta(2)(x_i(2) - c_{1i}(1)), \quad (2.14)$$

где  $\eta$  – параметр шага обучения.

Таким образом, при  $\eta(2) = 0,5$

$$c_{1i}(2) = \frac{c_{1i}(1) + x_i(2)}{2}. \quad (2.15)$$

В том случае, если условие (2.13) не выполняется, то формируется второй узел  $\mu_2$  функций принадлежности первого скрытого слоя, центры которых

$$c_{2i}(2) = x_i(2). \quad (2.16)$$

Одновременно с узлом  $\mu_2$  в NFS вводится второй синаптический вес  $w_2$ , также задаваемый случайным образом.

Пусть к моменту поступления на вход NFS наблюдения  $x(k)$  сформировано  $p$  узлов функций принадлежности  $\mu_1, \mu_2, \dots, \mu_p$ ,  $p < h$  с центрами  $c_{li}(k-1)$ ,  $l = 1, 2, \dots, p$ ,  $i = 1, 2, \dots, n$ . С приходом  $x(k)$  производится проверка условия

$$\max_i |c_{li} - x_i(k)| \leq r \quad \forall l = 1, 2, \dots, p. \quad (2.17)$$

Если это условие выполняется, то производится коррекция центров функций принадлежности, ближайших к соответствующим компонентам  $x_i(k)$  согласно правилу

$$c_{ii}(k) = c_{ii}(k-1) + \eta(k)(x_i(k) - c_{ii}(k-1)). \quad (2.18)$$

Несложно заметить, что (2.18) есть не что иное, как правило самообучения Т. Кохонена «Победитель получает все» [28] с той лишь разницей, что самообучение карты Кохонена реализуется на гиперсфере

$$\|x(k)\|_2 = 1, \quad (2.19)$$

а правило (2.18) – на гиперкубе

$$\|x(k)\|_\infty = 1. \quad (2.20)$$

В том случае, если условие (2.17) не выполняется, в системе формируется  $(p+1)$ -й ( $p+1 \leq h$ ) узел  $\mu_{p+1}$  с центрами функций принадлежности

$$c_{p+1,i}(k) = x_i(k). \quad (2.21)$$

Одновременно с узлом  $\mu_{p+1}$  формируется синаптический вес  $w_{p+1}$ .

Как видно, данная процедура есть гибрид эволюционирующего алгоритма Н. Касабова [12] и самоорганизующейся карты Т. Кохонена [28], при этом процесс эволюции архитектуры-самообучения функций принадлежности может протекать как непрерывно, так и до достижения числа функций принадлежности предельного значения  $nh$ .

**2.2.3 Адаптивное обучение синаптических весов.** Как уже отмечалось, для настройки синаптических весов нейро-фаззи системы могут быть использованы известные алгоритмы обучения-идентификации [78, 79]:

- экспоненциально-взвешенный рекуррентный метод наименьших квадратов

$$\left\{ \begin{aligned} w^h(k) &= w^h(k-1) + \frac{P^h(k-1)(y(k) - w^{hT}(k-1)\varphi^h(x(k)))}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \varphi^h(x(k)) = \\ &= w^h(k-1) + \frac{P^h(k-1)(y(k) - \hat{y}^h(x(k)))}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \varphi^h(x(k)), \\ P^h(k) &= \frac{1}{\beta} \left( P^h(k-1) - \frac{P^h(k-1)\varphi^h(x(k))\varphi^{hT}(x(k))P^h(k-1)}{\beta + \varphi^{hT}(x(k))P^h(k-1)\varphi^h(x(k))} \right) = \\ &= \left( \sum_{\tau=1}^k \varphi^h(x(\tau))\varphi^{hT}(x(\tau)) \right)^{-1}, \quad 0 < \beta \leq 1, \end{aligned} \right. \quad (2.22)$$

где  $y(k)$  – внешний обучающий сигнал,

$\beta$  – параметр забывания устаревшей информации;

- оптимальный по быстродействию градиентный одношаговый алгоритм Качмажа–Уидроу–Хоффа

$$w^h(k) = w^h(k-1) + \frac{y(k) - w^{hT}(k-1)\varphi^h(x(k))}{\|\varphi^h(x(k))\|^2} \varphi^h(x(k)); \quad (2.23)$$

- алгоритм обучения, обладающий как следящими, так и сглаживающими свойствами [42,43]

$$\left\{ \begin{aligned} w^h(k) &= w^h(k-1) + (\beta^h(k))^{-1} (y(k) - w^{hT}(k-1)\varphi^h(x(k))) \varphi^h(x(k)), \\ \beta^h(k) &= \beta \beta^h(k-1) + \|\varphi^h(x(k))\|^2, \quad 0 \leq \beta \leq 1 \end{aligned} \right. \quad (2.24)$$

и тому подобные процедуры.

Здесь интересно заметить, что процедура (2.24) связана с алгоритмом (2.22) соотношением

$$\beta^h(k) = \text{Tr } P^h(k), \quad (2.25)$$

а при  $\beta = 0$  приобретает форму алгоритма (2.23).

Процесс настройки синаптических весов может протекать одновременно с самообучением-эволюцией первого скрытого слоя.

Пусть к моменту поступления наблюдения  $x(k)$  сформировано  $p$  узлов функций принадлежности  $\mu_1, \mu_2, \dots, \mu_p$  и рассчитан вектор синаптических весов  $w^p(k-1)$ . Пусть условие (2.17) не выполняется, что сразу же ведет к формированию узла  $\mu_{p+1}$  и заданию произвольного начального значения синаптического веса  $w_{p+1}$ . При этом выходной сигнал NFS может быть представлен в виде

$$\begin{aligned} \hat{y}^{p+1}(x(k)) &= (w^{p+1}(k-1))^T \varphi^{p+1}(x(k)) = \\ &= w^{pT}(k-1) \varphi^p(x(k)) + w_{p+1} \varphi_{p+1}(x(k)), \end{aligned} \quad (2.26)$$

а алгоритм (2.24) –

$$\begin{cases} w^{p+1}(k) = \begin{pmatrix} w^p(k-1) \\ \text{-----} \\ w_{p+1} \end{pmatrix} + (\beta^{p+1}(k))^{-1} (y(k) - \hat{y}^{p+1}(x(k))) \begin{pmatrix} \varphi^p(x(k)) \\ \text{-----} \\ \varphi_{p+1}(x(k)) \end{pmatrix}, \\ \beta^{p+1}(k) = \beta \beta^p(k-1) + \|\varphi^p(x(k))\|^2 + \varphi_{p+1}^2(x(k)). \end{cases} \quad (2.27)$$

Как можно видеть, процесс одновременных эволюции-самообучения-контролируемого обучения не вызывает никаких вычислительных проблем.

### 2.3 Настройка параметров функций принадлежности нейро-фаззи узла

Настройка параметров центра и ширины функций принадлежности может проводиться по алгоритму с учителем, основанном на минимизации целевой функции [31], которая, как правило, задается с использованием евклидовой нормы и для одной пары обучающих данных  $(x(k), y(k))$  принимает вид

$$E(k) = \frac{1}{2}(y(k) - \hat{y}(k))^2 = \frac{1}{2}(y(k) - w^T \varphi(x(k)))^2. \quad (2.28)$$

При применении метода наискорейшего спуска соответствующие формулы адаптации в общем случае для  $(n \times 1)$ -мерного вектора входных сигналов принимают вид

$$c_{rj}(k+1) = c_{rj}(k) - \eta_c \frac{\partial E(k)}{\partial c_{rj}}, \quad (2.29)$$

$$\sigma_{rj}(k+1) = \sigma_{rj}(k) - \eta_\sigma \frac{\partial E(k)}{\partial \sigma_{rj}}, \quad (2.30)$$

где  $\eta_c$  – параметр шага обучения для параметров центра функции принадлежности,

$\eta_\sigma$  – параметр шага обучения для параметра ширины функции принадлежности,

$$r = 1, 2, \dots, h,$$

$$j = 1, 2, \dots, n.$$

Для упрощения расчета производных и ускорения вычисления значения функции принадлежности формула адаптации параметра ширины может быть записана в виде

$$-0.5\sigma_{rj}^{-2}(k+1) = \sigma_{rj}^{-2}(k) - \eta_{\sigma} \frac{\partial E(k)}{\partial(-0.5\sigma_{rj}^{-2})}. \quad (2.31)$$

При использовании в качестве функций принадлежности традиционных гауссианов (2.9) соответствующие формулы градиента целевой функции (2.28) для одной пары  $(x(k), y(k))$  обучающих данных принимают вид

$$\frac{\partial E(k)}{\partial c_{rj}} = (w^T \varphi(x(k)) - y(x(k))) \sum_{l=1}^h w_l \frac{\partial \varphi_l(x(k))}{\partial c_{rj}}, \quad (2.32)$$

$$\frac{\partial E(k)}{\partial(-0.5\sigma_{rj}^{-2})} = (w^T \varphi(x(k)) - y(x(k))) \sum_{l=1}^h w_l \frac{\partial \varphi_l(x(k))}{\partial(-0.5\sigma_{rj}^{-2})}, \quad (2.33)$$

где

$$\varphi_l(x(k)) = \frac{\prod_{i=1}^n \mu_{li}(x_i(k))}{\sum_{p=1}^h \prod_{i=1}^n \mu_{pi}(x_i(k))}. \quad (2.34)$$

Производные  $\frac{\partial \varphi_l(x)}{\partial c_{rj}}$  и  $\frac{\partial \varphi_l(x)}{\partial(-0.5\sigma_{rj}^{-2})}$ , определенные на основе (2.9) и

(2.34), можно записать как



$$\frac{\partial \varphi_l(x)}{\partial c_{rj}} = \frac{\delta_{lr} m(x) - t_l(x)}{(m(x))^2} \prod_{i=1, i \neq j}^n \mu_{ri}(x_i) \frac{\partial \mu_{rj}(x_j)}{\partial c_{rj}}, \quad (2.35)$$

$$\frac{\partial \varphi_l(x)}{\partial (-0.5\sigma_{rj}^{-2})} = \frac{\delta_{lr} m(x) - t_l(x)}{(m(x))^2} \prod_{i=1, i \neq j}^n \mu_{ri}(x_i) \frac{\partial \mu_{rj}(x_j)}{\partial (-0.5\sigma_{rj}^{-2})}, \quad (2.36)$$

где  $\delta_{lr}$  – дельта Кронекера,

$$t_l(x) = \prod_{i=1}^n \mu_{li}(x_i),$$

$$m(x) = \sum_{p=1}^h \prod_{i=1}^n \mu_{pi}(x_i).$$

Производные  $\frac{\partial \mu_{rj}(x_j)}{\partial c_{rj}}$  и  $\frac{\partial \mu_{rj}(x_j)}{\partial (-0.5\sigma_{rj}^{-2})}$ , определенные на основе (2.9),

можно записать как

$$\frac{\partial \mu_{rj}(x_j)}{\partial c_{rj}} = \frac{x_j - c_{rj}}{\sigma_{rj}^2} \exp\left(-\frac{(x_j - c_{rj})^2}{2\sigma_{rj}^2}\right), \quad (2.37)$$

$$\frac{\partial \mu_{rj}(x_j)}{\partial (-0.5\sigma_{rj}^{-2})} = (x_j - c_{rj})^2 \exp\left(-\frac{(x_j - c_{rj})^2}{2\sigma_{rj}^2}\right). \quad (2.38)$$

2.4 Эволюционирующая МГУА-нейро-фаззи система с малым числом настраиваемых параметров

2.4.1 Архитектура эволюционирующей многослойной МГУА-нейро-фаззи системы. Метод группового учёта аргументов (МГУА)

разрабатывался академиком А. Г. Ивахненко и его школой [44-54]. МГУА является методом индуктивного моделирования и одним из наиболее эффективных методов структурно-параметрической идентификации сложных объектов, процессов и систем по данным наблюдений в условиях неполноты информации.

Среди основных преимуществ метода группового учёта аргументов можно выделить следующие:

- он хорошо работает на коротких выборках данных, т. е. когда количество настраиваемых параметров больше количества наблюдений;
- система, основанная на МГУА, конструируется сама в процессе работы и не требует точного задания количества слоёв.

Архитектура эволюционирующей МГУА-системы приведена на рис 2.2.

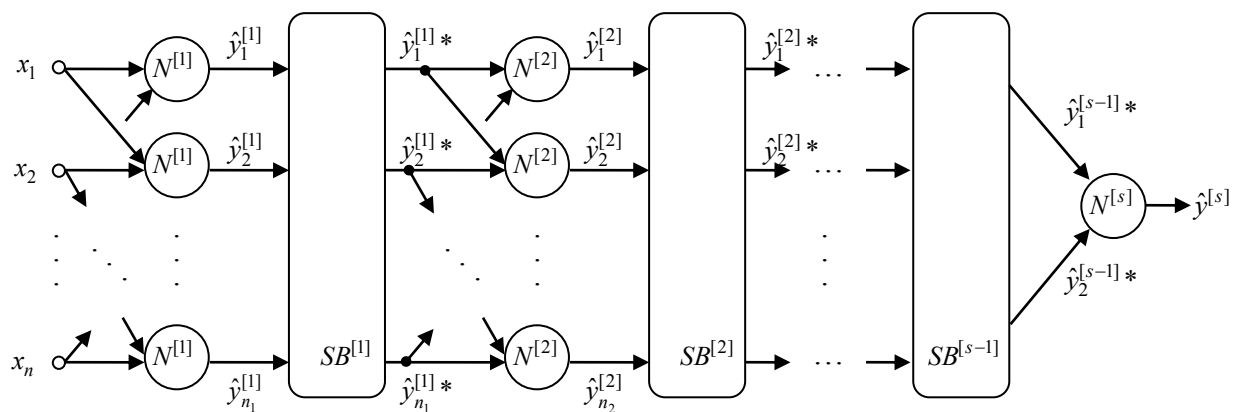


Рисунок 2.2 – Эволюционирующая МГУА-система

На нулевой (рецепторный) слой системы поступает  $(n \times 1)$ -мерный вектор входных сигналов  $x = (x_1, x_2, \dots, x_n)^T$ , который далее передается на первый скрытый слой, содержащий  $n_1 = c_n^2$  узлов-нейронов, каждый из которых имеет всего два входа. На выходах узлов  $N^{[1]}$  первого скрытого

слоя формируются выходные сигналы  $\hat{y}_l^{[1]}$ ,  $l = 1, 2, \dots, 0,5n(n-1) = c_n^2$ . Далее эти сигналы поступают на блок селекции первого скрытого слоя  $SB^{[1]}$ , который отбирает из множества выходных сигналов  $\hat{y}_l^{[1]}$   $n_1^*$  ( $n_1^* \leq n$ ) наиболее точных из них в смысле принятого критерия, чаще всего среднего квадрата ошибки  $\sigma_{y_l^{[1]}}^2$ .

Из этих  $n_1^*$  наилучших выходов первого скрытого слоя  $\hat{y}_l^{[1]*}$  формируется  $n_2$  (обычно  $n \leq n_2 \leq 2n$ ) попарных сочетаний  $\hat{y}_l^{[1]*}, \hat{y}_p^{[1]*}$ , которые подаются на второй скрытый слой, образованный нейронами  $N^{[2]}$ , аналогичными нейронам  $N^{[1]}$ .

Из выходных сигналов этого слоя  $\hat{y}_l^{[2]}$  блок селекции второго скрытого слоя  $SB^{[2]}$  отбирает только те, которые по точности, например, в смысле  $\sigma_{y_l^{[2]}}^2$ , превосходят наилучший сигнал первого скрытого слоя  $\hat{y}_l^{[1]*}$ .

Третий скрытый слой формирует сигналы, превосходящие по точности наилучший сигнал  $\hat{y}_l^{[2]*}$  и т. д.

Процесс эволюции системы происходит до тех пор, пока блок селекции  $SB^{[s-1]}$  не сформирует на своем выходе всего два сигнала  $\hat{y}_1^{[s-1]*}$  и  $\hat{y}_2^{[s-1]*}$ . Именно эти два сигнала подаются на единственный выходной нейрон  $N^{[s]}$ , вычисляющий выходной сигнал системы  $\hat{y}^{[s]}$ .

В качестве узлов МГУА-систем можно использовать самые различные типы нейронов, например, N-адалины, активные [49-51], R- [55-57], Q- [58], спайк-, вэйвлет-, нео-фаззи-нейроны [59-63] и другие подобные блоки систем вычислительного интеллекта, обладающие необходимыми аппроксимирующими возможностями и способностью к обучению. При этом, однако, может быть потеряно главное достоинство оригинального МГУА – возможность работы в условиях очень коротких обучающих выборок.

### 2.4.2 Нейро-фаззи система с малым числом настраиваемых параметров в качестве узла МГУА-системы

В качестве нейрона рассматриваемой МГУА-системы предлагается использовать узел, архитектура которого представлена на рис. 2.3.

Данная архитектура представляет собой по сути нейро-фаззи систему Ванга–Менделя [64,65] с двумя входами  $x_i(k)$  и  $x_j(k)$ , пятью последовательно соединенными слоями обработки информации и одним выходом  $\hat{y}_l(k)$ .

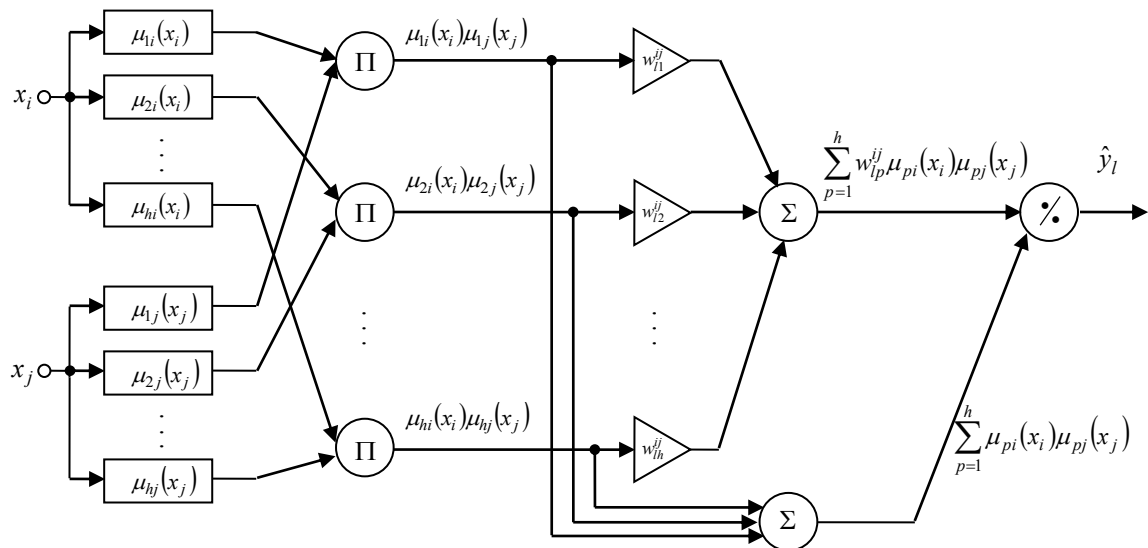


Рисунок 2.3 – Архитектура узла МГУА-нейро-фаззи системы

На вход узла подается двумерный вектор входных сигналов  $x(k) = (x_i(k), x_j(k))^T$ , подлежащих обработке, где  $k = 1, 2, \dots, N$  – номер наблюдения в обучающей выборке или текущее дискретное время.

Первый слой узла содержит  $2h$  функций принадлежности  $\mu_{pi}(x_i(k))$ ,  $\mu_{pj}(x_j(k))$ ,  $p = 1, 2, \dots, h$  и реализует фаззификацию входных переменных.

Второй слой обеспечивает агрегирование уровней принадлежности, вычисляемых в первом слое, содержит  $h$  блоков умножения и формирует двумерные радиально-базисные функции активации  $\mu_{pi}(x_i(k))\mu_{pj}(x_j(k))$ .

Третий слой – это слой синаптических весов, подлежащих настройке в процессе обучения, при этом выходами слоя являются значения  $w_{lp}^{ij}\mu_{pi}(x_i(k))\mu_{pj}(x_j(k))$ , а количество весов определяется числом функций принадлежности на каждом входе  $h$ .

Четвертый слой образован двумя сумматорами и вычисляет суммы выходных сигналов второго и третьего слоев.

И, наконец, в пятом (выходном) слое нейрона производится нормализация, в результате которой вычисляется выходной сигнал узла  $\hat{y}_l(k)$ .

На выходе узла (пятого слоя) формируется сигнал

$$\begin{aligned} \hat{y}_l(k) &= \frac{\sum_{p=1}^h w_{lp}^{ij} \mu_{pi}(x_i(k)) \mu_{pj}(x_j(k))}{\sum_{p=1}^h \mu_{pi}(x_i(k)) \mu_{pj}(x_j(k))} = \frac{\sum_{p=1}^h w_{lp}^{ij} \tilde{x}_p(k)}{\sum_{p=1}^h \tilde{x}_p(k)} = \\ &= \sum_{p=1}^h w_{lp}^{ij} \frac{\tilde{x}_p(k)}{\sum_{p=1}^h \tilde{x}_p(k)} = \sum_{p=1}^h w_{lp}^{ij} \varphi_p^{ij}(x(k)) = (w_{lp}^{ij})^T \varphi^{ij}(x(k)) \end{aligned} \quad , \quad (2.39)$$

где  $\varphi^{ij}(x(k)) = (\varphi_1^{ij}(x(k)), \dots, \varphi_p^{ij}(x(k)), \dots, \varphi_p^{ij}(x(k)))^T$ ,

$$\varphi_p^{ij}(x(k)) = \mu_{pi}(x_i(k)) \mu_{pj}(x_j(k)) \left( \sum_{p=1}^h \mu_{pi}(x_i(k)) \mu_{pj}(x_j(k)) \right)^{-1},$$

$$w_{lp}^{ij} = (w_{l1}^{ij}, \dots, w_{lp}^{ij}, \dots, w_{lh}^{ij})^T.$$

Несложно заметить, что узел реализует нелинейное отображение входных сигналов в выходной подобно нормализованной радиально-

базисной нейронной сети, однако содержит существенно меньшее число  $h$  настраиваемых параметров по сравнению с нейронной сетью.

Если, пользуясь введенными обозначениями, записать преобразования, реализуемые в каждом узле стандартного МГУА, в виде

$$\hat{y}_l(k) = w_{l0}^{jj} + w_{l1}^{jj}x_i(k) + w_{l2}^{jj}x_j(k), \quad (2.40)$$

содержащем три неизвестных параметра, несложно заметить, что при трех функциях принадлежности на каждом входе предлагаемого узла, мы приходим к тем же трем синаптическим весам, подлежащим уточнению-обучению.

Оценивание этих синаптических весов может быть реализовано в простейшем случае с помощью обычного метода наименьших квадратов (МНК), традиционно используемого в МГУА. При этом, если обучающая выборка задана в полном объеме, можно использовать пакетную форму МНК в виде

$$w_i^{jj}(N) = \left( \sum_{k=1}^N \varphi^{jj}(x(k)) (\varphi^{jj}(x(k)))^T \right)^+ \sum_{k=1}^N \varphi^{jj}(x(k)) y(k) \quad (2.41)$$

(здесь  $y(k)$  – внешний обучающий сигнал), если же данные поступают на обработку последовательно в online режиме, используется рекуррентная форма МНК

$$\begin{cases} w_i^{jj}(k) = w_i^{jj}(k-1) + \frac{P^{jj}(k-1) \left( y(k) - (w_i^{jj}(k-1))^T \varphi^{jj}(x(k)) \right) \varphi^{jj}(x(k))}{1 + (\varphi^{jj}(x(k)))^T P^{jj}(k-1) \varphi^{jj}(x(k))}, \\ P^{jj}(k) = P^{jj}(k-1) - \frac{P^{jj}(k-1) \varphi^{jj}(x(k)) (\varphi^{jj}(x(k)))^T P^{jj}(k-1)}{1 + (\varphi^{jj}(x(k)))^T P^{jj}(k-1) \varphi^{jj}(x(k))}. \end{cases} \quad (2.42)$$

## 2.5 Эволюционирующая каскадная нейро-фаззи система

Архитектура эволюционирующей каскадной системы [66-68] представлена на рис. 2.4.

На нулевой слой системы поступает  $(n \times 1)$ -мерный вектор входных сигналов  $x(k) = (x_1(k), x_2(k), \dots, x_n(k))^T$ , который далее передается на первый скрытый слой, содержащий  $c_n^2$  узлов-нейронов, каждый из которых имеет два входа. На выходах узлов  $N^{[1]}$  первого скрытого слоя формируются выходные сигналы  $\hat{y}_s^{[1]}$ ,  $s = 1, 2, \dots, 0,5n(n-1) = c_n^2$ . Далее эти сигналы поступают на блок селекции  $SB$ , который выполняет сортировку узлов первого скрытого слоя по принятому критерию (чаще всего по значению среднего квадрата ошибки  $\sigma_{y_s^{[1]}}^2$ ) так, что  $\sigma_{\hat{y}_1^{[1]*}}^2 < \sigma_{\hat{y}_2^{[1]*}}^2 < \dots < \sigma_{\hat{y}_s^{[1]*}}^2$ . Выходы блока селекции  $\hat{y}_1^{[1]*}$  и  $\hat{y}_2^{[1]*}$  поступают на вход единственного узла-нейрона второго слоя, на выходе которого формируется выходной сигнал  $\hat{y}^{[2]}$ . Этот выходной сигнал вместе с выходным сигналом блока селекции  $\hat{y}_3^{[1]*}$  поступает на вход узла-нейрона следующего слоя. Процесс наращивания каскадов продолжается до достижения требуемой точности.

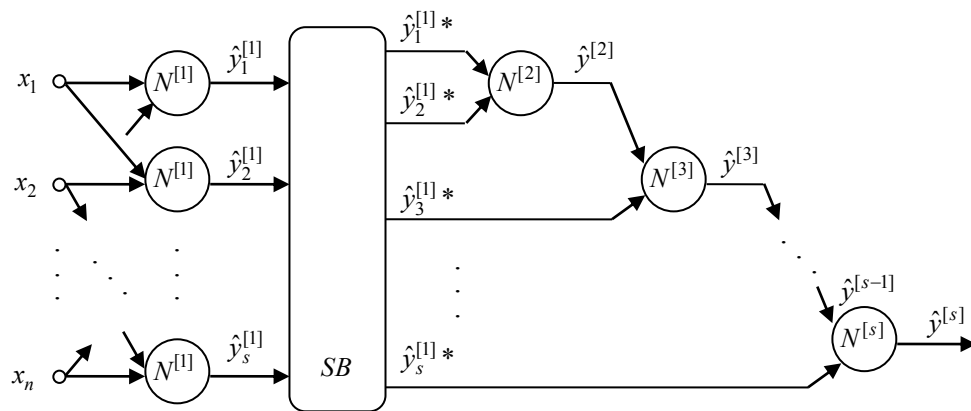


Рисунок 2.4 – Архитектура эволюционирующей каскадной системы

В качестве узлов рассматриваемой эволюционирующей каскадной системы могут использоваться двухвходовые нейро-фаззи системы, рассмотренные ранее, а также двухвходовые неο-фаззи узлы, архитектура которых будет рассмотрена далее.

## 2.6 Адаптивное прогнозирование нестационарных нелинейных последовательностей на основе эволюционирующей нейро(неο)-фаззи-WANARX-модели

Задача математического прогнозирования последовательностей данных (временных рядов) на сегодня достаточно хорошо изучена, а количество публикаций, посвященных этой проблеме, практически необозримо. Существует огромное количество методов решения этой задачи – от простейших регрессионных, корреляционных, спектральных, экспоненциального сглаживания и т. п. до продвинутых интеллектуальных, требующих порой использования достаточно сложных математических методов и высокой квалификации пользователя. Проблема резко усложняется, если исследуемые ряды нестационарны и нелинейны, содержат тренды неизвестного характера, квазипериодические компоненты, стохастические и хаотические составляющие. В этих условиях с наилучшей стороны показали себя нелинейные прогнозирующие модели, основанные на математическом аппарате вычислительного интеллекта [2–4], и, прежде всего, нейро-фаззи системы [1], благодаря высоким аппроксимирующим и экстраполирующим свойствам, способности к обучению, прозрачности и интерпретируемости получаемых результатов. Здесь особенно следует выделить, так называемые, NARX-модели [8], имеющие вид

$$\hat{y}(k) = f(y(k-1), y(k-2), \dots, y(k-n_y), x(k-1), \dots, x(k-n_x)), (2.43)$$



где  $\hat{y}(k)$  – оценка (прогноз) прогнозируемой последовательности в момент дискретного времени  $k = 1, 2, \dots$ ;

$f(\bullet)$  – некоторое нелинейное преобразование, реализуемое нейромфаззи системой,

$x(k)$  – наблюдаемый экзогенный фактор, определяющий поведение  $y(k)$ .

Можно заметить, что в рамки описания (2.43) укладываются как популярные AR-, ARX-, ARMAX-модели Бокса–Дженкинса, так и нелинейные NARMA-модели. Такие модели хорошо изучены, существует достаточно большое количество архитектур и алгоритмов обучения, их реализующих, однако при этом предполагается, что порядки модели  $n_y, n_x$  некоторым образом заданы заранее.

В случае структурной нестационарности исследуемых рядов эти порядки априори неизвестны и также должны настраиваться в процессе обучения. В этой ситуации целесообразно воспользоваться аппаратом эволюционирующих коннекционистских систем [6–9], в которых предусмотрена не только настройка синаптических весов и функций активации-принадлежности, но и собственно архитектуры.

На сегодня известен целый ряд таких алгоритмов, позволяющих реализовать такое обучение как в пакетном, так и последовательном режиме. Ситуация существенно усложняется, если информация на обработку поступает с достаточно высокой частотой в форме потока данных [69].

В этом случае наиболее популярные эволюционирующие системы оказываются слишком громоздкими, чтобы обучаться и обрабатывать информацию в online режиме.

### 2.6.1 Архитектура ANARX-модели

В качестве такой альтернативной достаточно простой и эффективной архитектуры может быть рассмотрена, так называемая, ANARX-модель (Additive NARX), имеющая вид [70,71]

$$\hat{y}(k) = f_1(y(k-1), x(k-1)) + f_2(y(k-2), x(k-2)) + \dots + f_n(y(k-n), x(k-n)) = \sum_{l=1}^n f_l(y(k-l), x(k-l)) \quad (2.44)$$

(здесь  $n = \max\{n_y, n_x\}$ ), при этом исходная задача синтеза прогнозирующей системы декомпозируется на множество локальных задач параметрической идентификации моделей-узлов с двумя входами  $y(k-l)$ ,  $x(k-l)$ ,  $l = 1, 2, \dots, n, \dots$ .

В качестве таких узлов авторы [70,71] использовали элементарные перцептроны Розенблатта с сигмоидальными функциями активации. ANARX-модель обеспечила высокое качество прогнозирования, но в общем случае зачастую требует включения в архитектуру большого количества узлов [72].

В настоящей работе рассматриваются некоторые вопросы синтеза прогнозирующих нейро-фаззи- и нео-фаззи систем [73-75] на основе ANARX-моделей, лишенных отмеченных выше недостатков.

На рис. 2.5 приведена архитектура ANARX-системы, образованная двумя линиями элементов чистого запаздывания  $z^{-1}$  ( $z^{-1}y(k) = y(k-1)$ ) и  $n$  параллельно включенными узлами  $N^{[l]}$ , при этом, как видно, обучение этих узлов производится независимо друг от друга, а введение новых узлов или исключение избыточных никак не влияет на работу всех остальных нейронов, т. е. эволюция такой системы реализуется элементарной манипуляцией количества узлов.

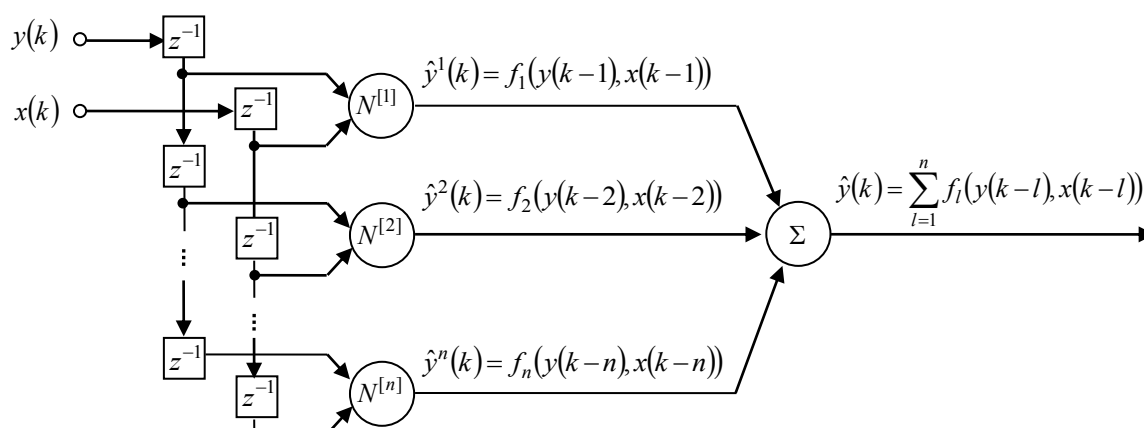


Рисунок 2.5 – Архитектура ANARX-системы

В качестве узла рассматриваемой ANARX-системы могут использоваться двухвходовые нейро-фаззи системы, рассмотренные ранее, а также двухвходовые нео-фаззи узлы.

### 2.6.2 Нео-фаззи-ANARX-модель

В случае необходимости обработки очень больших объемов информации в рамках концепции “Big Data” [80], когда на первый план выходят скорость обработки данных и простота численной реализации системы вычислительного интеллекта, вместо нейро-фаззи-узлов ANARX-модели целесообразно использовать нео-фаззи-нейроны, введенные Т. Ямакавой с коллегами [73–75] и являющиеся нелинейными обучаемыми системами. Архитектура нео-фаззи-нейрона как узла ANARX-системы приведена на рис. 2.6. К достоинствам нео-фаззи-нейрона можно отнести высокую скорость обучения, вычислительную простоту, хорошие аппроксимирующие свойства, возможность нахождения глобального минимума критерия обучения в online режиме.

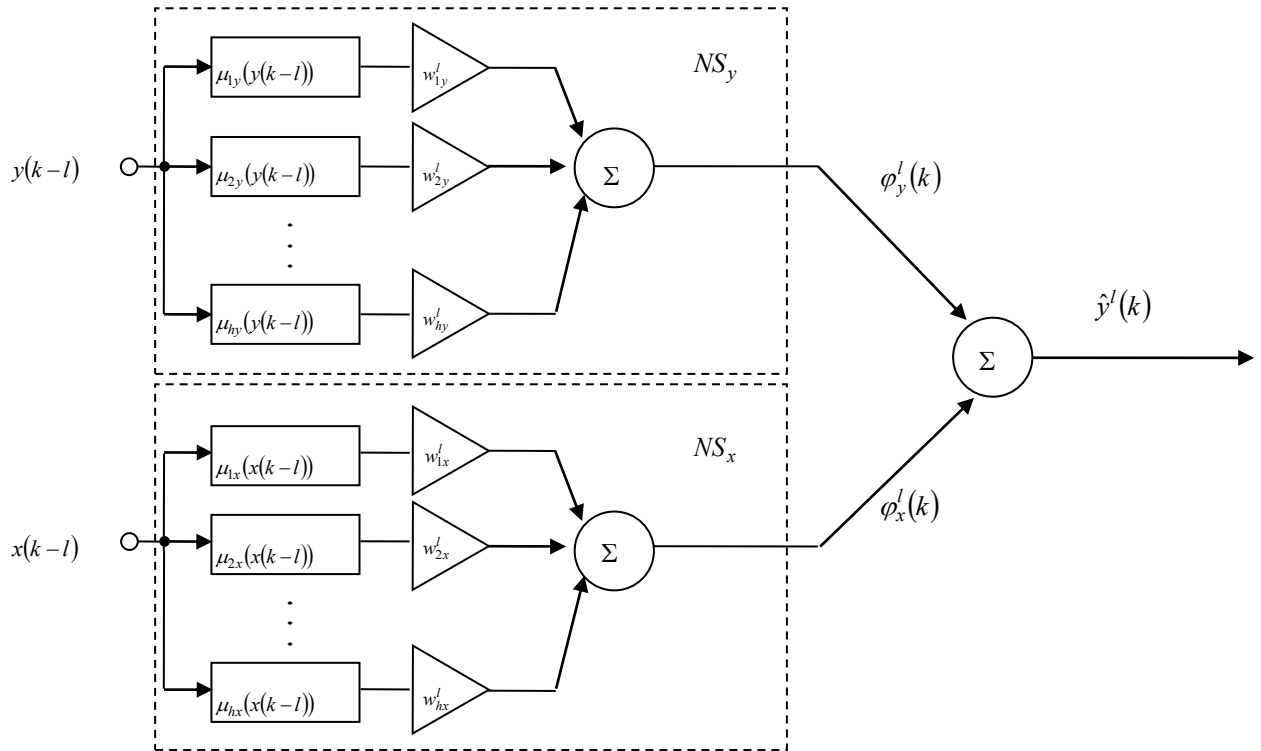


Рисунок 2.6 – Нео-фаззи узел ANARX-системы

Составными элементами нео-фаззи-нейрона являются нелинейные синапсы  $NS_y$ ,  $NS_x$ , в которых реализуются правила нечеткого вывода Такаги–Сугено нулевого порядка, однако, как несложно заметить, конструктивно нео-фаззи-нейрон гораздо проще нейро-фаззи узла, приведенного на рис. 2.3.

При подаче на вход такого узла сигналов  $y(k-l)$ ,  $x(k-l)$ , на его выходе формируется выходное значение

$$\hat{y}^l(k) = \varphi_y^l(k) + \varphi_x^l(k) = \sum_{i=1}^h w_{iy}^l \mu_{iy}(y(k-l)) + \sum_{i=1}^h w_{ix}^l \mu_{ix}(x(k-l)), \quad (2.45)$$

а на выходе ANARX-модели в целом

$$\hat{y}(k) = \sum_{l=1}^n \left( \sum_{i=1}^h w_{iy}^l \mu_{iy}(y(k-l)) + \sum_{i=1}^h w_{ix}^l \mu_{ix}(x(k-l)) \right), \quad (2.46)$$

т. е., поскольку нео-фаззи-нейрон также является аддитивной моделью [81], ANARX-модель на нео-фаззи-нейронах является дважды аддитивной.

В качестве функций принадлежности в нео-фаззи-нейроне обычно используются треугольные конструкции, отвечающие условиям единичного разбиения

$$\sum_{i=1}^h \mu_{iy}(y(k-l)) = 1; \quad \sum_{i=1}^h \mu_{ix}(x(k-l)) = 1, \quad (2.47)$$

что позволяет упростить конструкцию узла, исключив из нее слой нормализации.

В [82] в качестве функций принадлежности нео-фаззи-нейрона было предложено использовать  $B$ -сплайны, обеспечивающие более высокое качество аппроксимации и также отвечающие условиям единичного разбиения. При этом для  $B$ -сплайна  $q$ -го порядка можно записать

$$\mu_{iy}^q(y(k-l)) = \begin{cases} \left. \begin{array}{l} 1, \text{ если } c_{iy} \leq y(k-l) < c_{i+1,y}, \\ 0 \text{ в противном случае} \end{array} \right\} \text{ для } q = 1, \\ \frac{y(k-l) - c_{iy}}{c_{i+q-1,y} - c_{iy}} \mu_{iy}^{q-1}(y(k-l)) + \frac{c_{i+q,y} - y(k-l)}{c_{i+q,y} - c_{i+1,y}} \times \\ \times \mu_{i+1,y}^{q-1}(y(k-l)), \text{ для } q > 1, \\ i = 1, \dots, h - q, \end{cases} \quad (2.48)$$

$$\mu_{ix}^q(x(k-l)) = \begin{cases} \left. \begin{array}{l} 1, \text{ если } c_{ix} \leq x(k-l) < c_{i+1,x}, \\ 0 \text{ в противном случае} \end{array} \right\} \text{ для } q=1, \\ \frac{x(k-l) - c_{ix}}{c_{i+q-1,x} - c_{ix}} \mu_{ix}^{q-1}(x(k-l)) + \frac{c_{i+q,x} - x(k-l)}{c_{i+q,x} - c_{i+1,x}} \times \\ \times \mu_{i+1,x}^{q-1}(x(k-l)), \text{ для } q > 1, \\ i = 1, \dots, h-q. \end{cases} \quad (2.49)$$

Следует отметить, что  $B$ -сплайны являются своего рода обобщенными функциями принадлежности: так, при  $q=2$  получаем традиционные треугольные функции принадлежности, при  $q=4$  – кубические сплайны и т. п.

Вводя далее векторные переменные  $w^l = (w_{1y}^l, \dots, w_{hy}^l, w_{1x}^l, \dots, w_{hx}^l)^\top$ ,  $\varphi^l(k) = (\mu_{1y}(y(k-l)), \dots, \mu_{hy}(y(k-l)), \mu_{1x}(x(k-l)), \dots, \mu_{hx}(x(k-l)))^\top$ , можно переписать (2.45) в виде

$$\hat{y}^l(k) = w^{\top} \varphi^l(k) \quad (2.50)$$

и использовать для обучения нео-фаззи-нейрона либо алгоритмы (2.23), (2.24), либо процедуру [83]

$$\begin{cases} w^l(k) = w^l(k-1) + r_l^{-1}(k)(y(k) - w^{\top}(k-1)\varphi^l(k))\varphi^l(k) \\ r_l(k) = \alpha r_l(k-1) + \varphi^{\top}(k)\varphi^l(k), 0 \leq \alpha \leq 1, \end{cases} \quad (2.51)$$

обладающую как фильтрующими, так и следящими свойствами. Можно заметить также, что при  $\alpha = 1$  (2.51) полностью совпадает с оптимальным алгоритмом Качмажа–Уидроу–Хоффа (2.23).

Эволюционирующие системы на нео-фаззи-нейронах продемонстрировали свою эффективность при решении ряда задач, включая и прогнозирование [57, 82–85], однако с точки зрения простоты реализации и быстродействия аддитивная система наилучшим образом приспособлена для обработки потоков данных в задачах Data Stream Mining [69].

### 2.6.3 Взвешенная ANARX-модель (WANARX-модель)

Поскольку каждый узел  $N^{[l]}$  ANARX-модели настраивается независимо от других и представляет собой по сути отдельную нейро(нео)-фаззи систему, для улучшения качества получаемых прогнозов можно воспользоваться идеей комбинирования ансамбля нейронных сетей [86]. Этот подход естественно ведет к архитектуре взвешенной (weighted ANARX) нейро-нео-фаззи-WANARX системы, приведенной на рис. 2.7.

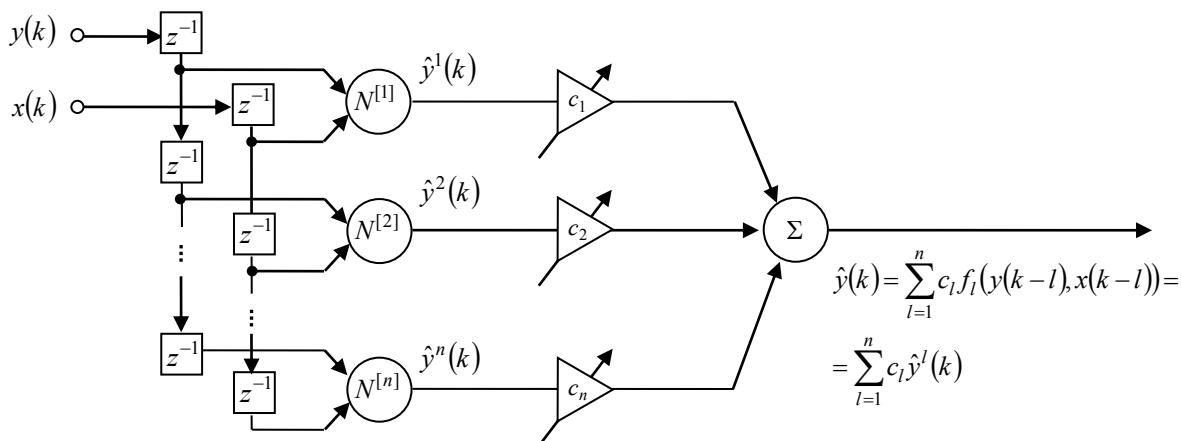


Рисунок 2.7 – Архитектура WANARX-системы

При этом выходной сигнал такой системы может быть записан в виде

$$\hat{y}(k) = \sum_{l=1}^n c_l \hat{y}^l(k) = c^T \hat{\hat{y}}(k), \quad (2.52)$$

где  $\hat{\hat{y}}(k) = (\hat{y}^1(k), \hat{y}^2(k), \dots, \hat{y}^n(k))^T$ ,

$c = (c_1, c_2, \dots, c_n)^T$  – вектор настраиваемых весовых коэффициентов, определяющих близость сигналов  $\hat{y}^l(k)$  к прогнозируемому процессу  $y(k)$  и отвечающих условиям несмещенности

$$\sum_{l=1}^n c_l = c^T I_n = 1, \quad (2.53)$$

где  $I_n$  –  $(n \times 1)$ -вектор, образованный единицами.

Для нахождения вектора  $c$  в пакетном режиме можно воспользоваться методом неопределенных множителей Лагранжа. Для этого в рассмотрение вводится последовательность ошибок прогнозирования

$$\begin{aligned} v(k) &= y(k) - \hat{y}(k) = y(k) - c^T \hat{\hat{y}}(k) = \\ &= c^T I_n y(k) - c^T \hat{\hat{y}}(k) = c^T (I_n y(k) - \hat{\hat{y}}(k)) = c^T V(k), \end{aligned} \quad (2.54)$$

функция Лагранжа

$$L(c, \lambda) = \sum_k c^T V(k) V^T(k) c + \lambda (c^T I_n - 1) = c^T R c + \lambda (c^T I_n - 1), \quad (2.55)$$

где  $\lambda$  – неопределенный множитель Лагранжа,



$R = \sum_k V(k)V^T(k)$  – корреляционная матрица ошибок.

Также в рассмотрение вводится система уравнений Каруша–Куна–Таккера

$$\begin{cases} \nabla_c L(c, \lambda) = 2Rc + \lambda I_n = \vec{0}, \\ \frac{\partial L}{\partial \lambda} = c^T I_n - 1 = 0. \end{cases} \quad (2.56)$$

Решение системы (2.56) ведет к известному результату

$$\begin{cases} c = R^{-1} I_n (I_n^T R^{-1} I_n)^{-1}, \\ \lambda = -2 I_n^T R^{-1} I_n, \end{cases} \quad (2.57)$$

при этом лагранжиан (2.55) в седловой точке приобретает значение

$$L^*(c, \lambda) = (I_n^T R^{-1} I_n)^{-1}. \quad (2.58)$$

Реализация алгоритма (2.57) может наталкиваться на существенные трудности при обработке информации в online режиме и высоком уровне корреляции сигналов  $\hat{y}^l(k)$ , ведущем к плохой обусловленности матрицы  $R$ , которую необходимо обращать на каждом такте реального времени  $k$ .

Запишем функцию Лагранжа (2.55) в виде

$$L(c, \lambda) = \sum_k (y(k) - c^T \hat{y}(k))^2 + \lambda (c^T I_n - 1) \quad (2.59)$$

и градиентный алгоритм отыскания ее седловой точки на основе процедуры Эрроу–Гурвица [87,88]:

$$\begin{cases} c(k) = c(k-1) - \eta_c(k) \nabla_c L(c, \lambda), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k) \frac{\partial L(c, \lambda)}{\partial \lambda}, \end{cases} \quad (2.60)$$

или

$$\begin{cases} c(k) = c(k-1) + \eta_c(k) \left( 2(y(k) - c^T(k-1)\hat{y}(k))\hat{y}(k) - \lambda(k-1)I_n \right) = \\ \quad = c(k-1) + \eta_c(k) \left( 2v(k)\hat{y}(k) - \lambda(k-1)I_n \right) \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k) (c^T(k)I_n - 1), \end{cases} \quad (2.61)$$

где  $\eta_c(k)$ ,  $\eta_\lambda(k)$  – параметры шага обучения.

Процедура Эрроу–Гурвица сходится к седловой точке при достаточно общих предположениях относительно значений  $\eta_c(k)$ ,  $\eta_\lambda(k)$ , однако для ускорения процесса обучения, что особенно важно в задачах Data Stream Mining, эти параметры могут быть оптимизированы.

Для этого домножим первое соотношение (2.61) слева на  $\hat{y}^T(k)$

$$\hat{y}^T(k)c(k) = \hat{y}^T(k)c(k-1) + \eta_c(k) \left( 2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T(k)I_n \right) \quad (2.62)$$

и введем дополнительную функцию, характеризующую критериальную сходимость:

$$\begin{aligned} \left( y(k) - \hat{y}^T(k)c(k) \right)^2 = & v^2(k) - 2\eta_c(k)v(k) \left( 2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right) + \\ & + \eta_c^2(k) \left( 2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right)^2. \end{aligned} \quad (2.63)$$

Решение дифференциального уравнения

$$\begin{aligned} \frac{\partial \left( y(k) - \hat{y}^T(k)c(k) \right)^2}{\partial \eta_c(k)} = & -2v(k) \left( 2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right) + \\ & + 2\eta_c(k) \left( 2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n \right)^2 = 0 \end{aligned} \quad (2.64)$$

позволяет получить оптимальные значения шага обучения  $\eta_c(k)$  в форме

$$\eta_c(k) = \frac{v(k)}{2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n}, \quad (2.65)$$

подставляя которую в (2.61), окончательно можно записать

$$\begin{cases} c(k) = c(k-1) + \frac{v(k)(2v(k)\hat{y}(k) - \lambda(k-1)I_n)}{2v(k)\|\hat{y}(k)\|^2 - \lambda(k-1)\hat{y}^T I_n}, \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)(c^T(k)I_n - 1). \end{cases} \quad (2.66)$$

Несложно заметить, что при  $\lambda(k-1) = 0$  процедура (2.66) совпадает с алгоритмом Качмажа–Уидроу–Хоффа (2.23).

## Выводы к разделу 2

1. Рассмотрена модель нечёткого вывода Такаги–Сугено–Канга.
2. Предложена архитектура и методы обучения эволюционирующей многослойной нейро-фаззи системы, основанной на системе Ванга–Менделя и сочетающей в себе процессы эволюции архитектуры, самообучения функций принадлежности и контролируемого обучения синаптических весов.
3. Предложен метод обучения для настройки синаптических весов и параметров функций принадлежности нейро-фаззи узлов для улучшения аппроксимирующих свойств эволюционирующих нейро-фаззи систем.
4. Рассмотрена архитектура эволюционирующей системы на основе МГУА для обработки коротких выборок данных. Предложено в качестве узла этой системы использовать двухвходовые системы Ванга–Менделя.
5. Рассмотрена архитектура и предложены методы обучения каскадной нейро-фаззи и каскадной нео-фаззи системы для обработки выборок, которые содержат небольшое количество данных.
6. Рассмотрена архитектура аддитивной нелинейной авторегрессионной модели с экзогенными входами (ANARX-модели), предложена архитектура взвешенной ANARX-модели (WANARX-модели), в которой в качестве узлов могут быть использованы двухвходовые нейро-фаззи системы Ванга–Менделя или двухвходовые нео-фаззи системы.

### РАЗДЕЛ 3

## ЭВОЛЮЦИОНИРУЮЩИЕ НЕЙРО-ФАЗЗИ СИСТЕМЫ ДЛЯ ОБУЧЕНИЯ БЕЗ УЧИТЕЛЯ

### 3.1 Самоорганизующаяся карта Кохонена

Сеть Кохонена [28] относится к самоорганизующимся сетям. Это означает, что они не получают желаемый выходной сигнал при поступлении входного обучающего вектора, а в результате обучения сеть разделяет входные сигналы на классы, таким образом формируя топологические карты. Одним из важнейших свойств обученной сети Кохонена является её способность к обобщению. Суть работы сети Кохонена заключается в том, что происходит кластеризация входных векторов в группы схожих между собой векторов. При этом веса сети настраиваются так, чтобы входные образы, которые принадлежат одному кластеру, активизировали один и тот же выходной нейрон. Вектор каждого из нейронов сети Кохонена может заменить группу соответствующих ему классифицируемых векторов. Таким образом, это свойство позволяет использовать данный вид сети в области сжатия данных. Стоит отметить, что самоорганизующаяся карта Т. Кохонена реализует отображение входного пространства размерности  $n$  в выходное пространство размерности  $m$ .

Самоорганизующаяся карта имеет очень простую архитектуру с прямой передачей информации. Кроме нулевого (рецепторного) слоя, она содержит единственный слой нейронов, который очень часто называют слоем Кохонена [28].

Именно благодаря такой организации, каждый нейрон сети получает всю информацию об анализируемом образе и генерирует на своём выходе соответствующий отклик.

После этого в слое Кохонена возникает режим конкуренции, в результате которой определяется единственный нейрон-победитель с

максимальным выходным сигналом. Этот сигнал по латеральным связям обеспечивает возбуждение ближайших «соседей» победителя и подавление реакции далеко отстоящих узлов.

Самоорганизующиеся карты могут иметь различную топологию. Однако наиболее часто рецепторы и нейроны располагаются в узлах одно- или двумерной решётки.

Рассмотрим подробнее архитектуру самоорганизующейся карты. На вход сети поступает  $n$ -мерный входной сигнал. Сеть содержит единственный слой из  $m$  нейронов, которые образуют на плоскости прямоугольные решётки.

Нейроны характеризуются своим местоположением в сети. Каждый нейрон слоя Кохонена связан с каждым входом нулевого (входного) слоя прямыми связями, а также со всеми другими нейронами поперечными (латеральными, внутрислойными) связями.

На рис. 3.1 представлена 1D-карта Кохонена.

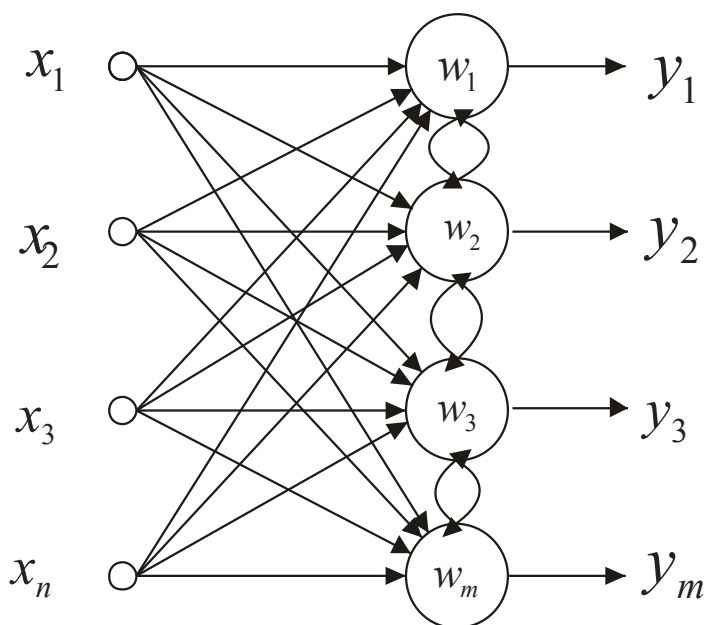


Рисунок 3.1 – 1D-карта Кохонена

В процессе обучения соседние нейроны влияют друг на друга сильнее, чем те, которые расположены дальше. Именно латеральные связи в сети обеспечивают возбуждение одних нейронов и торможение других.

Каждый нейрон из слоя Кохонена формирует взвешенную сумму сигналов  $f(x, w) = \sum_{i=1} w_i x_i$ . При этом, если синапсы ускоряющие, то  $w_{ij} > 0$ . Если же синапсы тормозящие, то  $w_{ij} < 0$ .

Перед началом работы алгоритма обучения сети Кохонена входные векторы предварительно нормализуются:

$$\tilde{x}_i = \frac{x_i}{\sqrt{\sum_i x_i^2}} = \frac{x_i}{\|x\|}, i = 1, 2, \dots, N \quad (3.1)$$

Сам алгоритм обучения сети Кохонена может быть описан как последовательность шагов:

Шаг 1: производится инициализация начальных значений синаптических весов  $w_{ij}(0)$ .

Одним из часто применяемых способов инициализации является присваивание синаптическим весам значений, равных случайно выбранным векторам из множества наблюдений. В целом, существует три основных способа инициализации начальных весов:

- инициализация случайными значениями, когда всем весам даются малые случайные величины;
- инициализация примерами, когда в качестве начальных значений задаются значения случайно выбранных примеров из обучающей выборки;
- линейная инициализация. В этом случае веса инициализируются значениями векторов, линейно упорядоченных вдоль линейного подпространства, проходящего между двумя главными собственными векторами исходного набора данных.

Шаг 2: на вход системы подаётся нормализованный вектор сигналов  $\tilde{x}$  и выбирается вектор весов (нейрон), ближайший к  $\tilde{x}$ , т. е. вектор, для которого Евклидово расстояние до  $\tilde{x}$  будет наименьшим:

$$\arg \min_j \|\tilde{x} - w_j\|, j = 1, 2, \dots, l. \quad (3.2)$$

Это выражение также может быть записано в следующем виде:

$$c = \arg \max_j (\tilde{x}^T w_j), j = 1, 2, \dots, l. \quad (3.3)$$

Шаг 3: производится коррекция (настройка) вектора синаптических весов  $w_{ij}$  по правилу

$$w_{ij}(k+1) = w_{ij}(k) + \eta(k) f_{ij}(k) (\tilde{x}(k) - w_{ij}(k)), \quad (3.4)$$

где  $w_{ij}(k+1)$  – новое значение веса,

$\eta(k)$  – изменяющийся со временем коэффициент усиления (на первой итерации  $\eta = 1$  и постепенно уменьшается до нуля, т. о.  $\eta(k) \in (0, 1]$ ),

$f_{ij}(k)$  – монотонно убывающая функция (функция соседства) вида

$$f_{ij}(k) = f(\|r_i - r_j\|, k), \quad (3.5)$$

где  $r_i$  – вектор, определяющий положение  $i$ -го нейрона в решётке,

$r_j$  – вектор, определяющий положение  $j$ -го нейрона в решётке.



Очевидно, что для нейрона-победителя  $f_c(\|r_i - r_j\|) = f_c(0) = 1$ .

Чаще всего в качестве функции соседства выбирают гауссиан, «мексиканскую шляпу», косинусоидную функцию, функции Епанечникова и т. п.

При этом выходные сигналы сети определяются так:

$$y_j(k) = \begin{cases} 1, & \text{если } w_j^T(k) \tilde{x}(k) > w_p^T(k) \tilde{x}(k) \text{ для всех } p \neq j, \\ 0 & \text{в противном случае.} \end{cases} \quad (3.6)$$

Далее шаги 1 и 2 повторяются до тех пор, пока выходные значения сети не стабилизируются с заданной точностью.

Смысл настройки (3.4) сводится к тому, что происходит минимизация разницы между входным вектором  $\tilde{x}(k)$  и вектором синаптических весов нейрона-победителя.

Другими словами можно сказать, что этот алгоритм в процессе настройки «подтягивает» вектор синаптических весов нейрона-победителя к текущему входному образу  $\tilde{x}(k)$ .

На рис. 3.2 представлена геометрическая интерпретация настройки синаптических весов для двумерного случая.

В данном случае находится вектор  $\tilde{x}(k) - w_{ij}(k)$ , который затем уменьшается на величину  $\eta(k)$ , которая задаёт скорость обучения.

Таким образом, можно сказать, что обучение сводится к вращению вектора весов нейрона в направлении входного вектора без существенного изменения его длины.

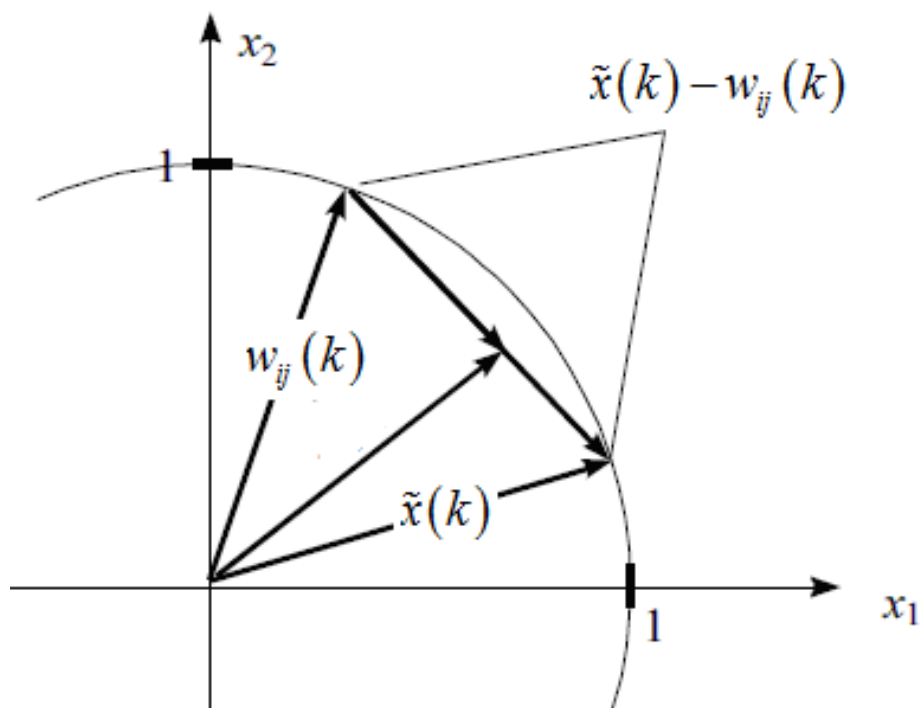


Рисунок 3.2 – Геометрическая интерпретация настройки весов сети Кохонена

3.2 Ансамбль адаптивных нейро-фаззи сетей Кохонена для online нечёткой кластеризации потоков данных

### 3.2.1 Вероятностная нечёткая кластеризация

Среди всех процедур кластеризации алгоритмы, основанные на целевых функциях, считаются математически строгими. Они решают задачу оптимизации при различных априорных предположениях. В такой ситуации наиболее часто применяется вероятностный подход, который основывается на минимизации целевой функции

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m u_j^k(k) \|x(k) - c_j\|^2 \quad (3.7)$$

при заданных условиях

$$\sum_{j=1}^m u_j(k) = 1, \quad k = 1, 2, \dots, N, \quad (3.8)$$

$$0 \leq \sum_{k=1}^N u_j(k) \leq N, \quad (3.9)$$

где  $u_j(k) \in [0, 1]$  – это уровень принадлежности вектора  $x(k)$  к  $j$ -му классу,

$c_j$  – это прототип  $j$ -го кластера,

$\beta$  – это неотрицательный параметр фаззификации (обычно называемый фаззификатором), который фактически определяет уровень размытости границ кластеров.

Результатом такой процедуры кластеризации является  $(N \times m)$ -мерная матрица  $U = \{u_j(k)\}$ , которую также называют матрицей нечёткого разбиения.

Количество кластеров  $m$  устанавливается заранее и не может быть изменено в процессе выполнения вычислительных процедур.

Следует отметить, что благодаря ограничению (3.8) элементы матрицы  $U$  могут рассматриваться как вероятности того, что векторы данных принадлежат некоторым определённым кластерам. Поэтому процедуры, основанные на минимизации (3.7) называются алгоритмами вероятностной нечёткой кластеризации.

### 3.2.2 Нечёткая кластеризация с переменным фаззификатором

Введём функции Лагранжа

$$L(u_j(k), c_j, \lambda(k)) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|x(k) - c_j\|^2 + \sum_{k=1}^N \lambda(k) \left( \sum_{j=1}^m u_j(k) - 1 \right), \quad (3.10)$$

где  $\lambda(k)$  – это неопределённый множитель Лагранжа.

Решив систему уравнений Каруша–Куна–Таккера, получаем решение в виде

$$\left\{ \begin{array}{l} u_j(k) = \frac{\left( \|x(k) - c_j\|^2 \right)^{\frac{1}{1-\beta}}}{\sum_{l=1}^m \left( \|x(k) - c_l\|^2 \right)^{\frac{1}{1-\beta}}}, \\ c_j = \frac{\sum_{k=1}^N u_j^\beta(k) x(k)}{\sum_{k=1}^N u_j^\beta(k)}, \\ \lambda(k) = - \left( \left( \sum_{l=1}^m \beta \|x(k) - c_l\|^2 \right)^{\frac{1}{1-\beta}} \right)^{1-\beta}, \end{array} \right. \quad (3.11)$$

которое при  $\beta = 2$  совпадает с алгоритмом Fuzzy C-Means (FCM) Дж. Бездека. При  $\beta \rightarrow 1$  результат этого решения близок к результатам, полученным традиционным алгоритмом чёткой кластеризации K-Means.

В качестве альтернативы процедуре, в которой используется фаззификатор  $1 < \beta < \infty$ , в [89] была предложена целевая функция для нечёткой вероятностной кластеризации (с ограничениями (3.8) и (3.9))

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1-\alpha)u_j(k)) \|x(k) - c_j\|^2, \quad (3.12)$$

где  $0 < \alpha \leq 1$  – это параметр, который определяет природу полученного решения.

Введя функцию Лагранжа

$$L(u_j(k), c_j, \lambda(k)) = \sum_{k=1}^N \sum_{j=1}^m (\alpha u_j^2(k) + (1-\alpha)u_j(k)) \|x(k) - c_j\|^2 + \sum_{k=1}^N \lambda(k) \left( \sum_{j=1}^m u_j(k) - 1 \right) \quad (3.13)$$

и решив систему уравнений Каруша–Куна–Таккера

$$\begin{cases} \frac{\partial L(u_j(k), c_j, \lambda(k))}{\partial u_j(k)} = (2\alpha u_j(k) + 1 - \alpha) \|x(k) - c_j\|^2 + \lambda(k) = 0, \\ \nabla_{c_j} L(u_j(k), c_j, \lambda(k)) = -\sum_{k=1}^N 2(\alpha u_j^2(k) + (1-\alpha)u_j(k))(x(k) - c_j) = \vec{0}, \\ \frac{\partial L(u_j(k), c_j, \lambda(k))}{\partial \lambda(k)} = \sum_{j=1}^m u_j(k) - 1 = 0, \end{cases} \quad (3.14)$$

получаем решение в виде

$$\begin{cases} u_j(k) = -\frac{1-\alpha}{2\alpha} + \frac{1+m\frac{1-\alpha}{2\alpha}}{\sum_{l=1}^m \frac{\|x(k) - c_l\|^2}{\|x(k) - c_l\|^2}}, \\ c_j = \frac{\sum_{k=1}^N (\alpha u_j^2(k) + (1-\alpha)u_j(k))x(k)}{\sum_{k=1}^N (\alpha u_j^2(k) + (1-\alpha)u_j(k))}. \end{cases} \quad (3.15)$$

При  $\alpha=1$ , эта процедура совпадает с FCM. Важно отметить, что метод (3.15) не может быть использован для решения задач Data Stream Mining, поскольку он не может обрабатывать информацию в online режиме. Поэтому адаптивная модификация выражения (7) была введена в [90]

$$\begin{cases} u_j(k+1) = -\frac{1-\alpha}{2\alpha} + \frac{1+m\frac{1-\alpha}{2\alpha}}{\sum_{l=1}^m \frac{\left(\|x(k+1)-c_j(k)\|^2\right)}{\left(\|x(k+1)-c_l(k)\|^2\right)}}, \\ c_j(k+1) = c_j(k) + \eta(k) \left( \alpha u_j^2(k+1) + (1-\alpha)u_j(k+1) \right) (x(k+1) - c_j(k)), \end{cases} \quad (3.16)$$

где  $\eta(k)$  – это параметр скорости обучения.

Несложно заметить, что второе рекуррентное выражение в (3.16) является правилом самообучения Кохонена по принципу «Победитель получает больше» с функцией соседства  $\alpha u_j^2(k+1) + (1-\alpha)u_j(k+1)$ .

### 3.2.3 Ансамбль адаптивных нейро-фаззи сетей Кохонена

Хотя значение параметра  $\alpha$  в формулах (3.15) и (3.16) лежит в значительно более узком интервале, чем фаззификатор  $\beta$ , но в настоящее время не существует формальных правил выбора и настройки этого параметра. Поэтому при решении конкретной задачи в пакетном режиме эта задача обычно решается несколько раз с помощью выражения (3.15) с различными значениями параметра  $\alpha$  (начиная с очень малого значения и увеличивая до 1).

Понятно, что такой подход не может эффективно решить задачу в online режиме. В этом случае целесообразно применить идею ансамбля параллельно работающих процедур кластеризации [91,92], где каждая

процедура кластеризации работает со значением параметра  $\alpha$ , отличным от других процедур.

Этот ансамбль легко может быть реализован с помощью адаптивных нейро-фаззи сетей Кохонена [93]. Они представляют собой двухслойную архитектуру, где центры кластеров настраиваются в слое Кохонена (который содержит  $m$  нейронов  $N_j^K$ ), а уровни принадлежности рассчитываются в выходном слое (который содержит  $m$  нейронов  $N_j^M$ ).

Ансамбль, сформированный этими сетями, представлен на рис. 3.3.

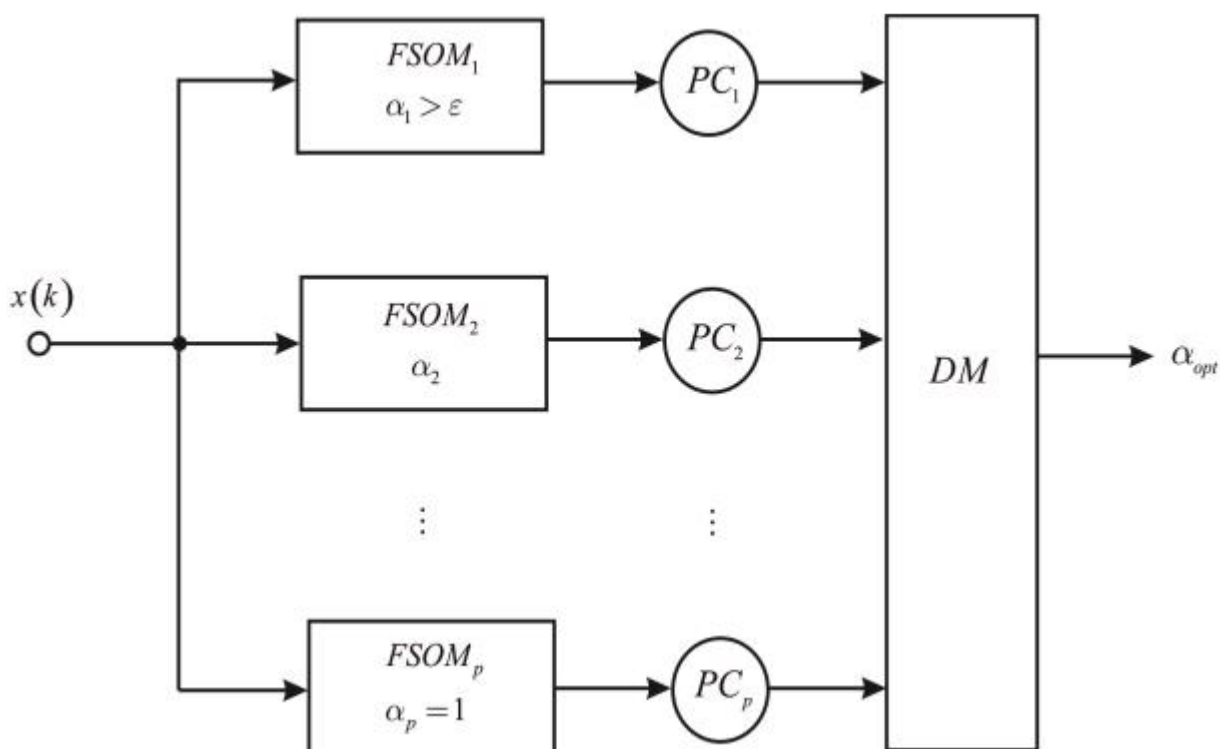


Рисунок 3.3 – Архитектура ансамбля адаптивных нейро-фаззи сетей Кохонена

Алгоритм самообучения для  $p$ -го члена ансамбля ( $p = 1, 2, \dots, q$ ), который содержит  $q$  нейро-фаззи сетей, может быть записан в виде

$$\begin{cases} c_{jp}(k+1) = c_{jp}(k) + \eta(k) \left( \alpha_p u_{jp}^2(k) + (1 - \alpha_p) u_{jp}(k) \right) (x(k+1) - c_{jp}(k)), \\ u_{jp}(k+1) = -\frac{1 - \alpha_p}{2\alpha_p} + \frac{1 + m \frac{1 - \alpha_p}{2\alpha_p}}{\sum_{l=1}^m \frac{\left( \|x(k+1) - c_{jp}(k)\|^2 \right)}{\left( \|x(k+1) - c_{lp}(k)\|^2 \right)}}, \end{cases} \quad (3.17)$$

при этом слой Кохонена настраивается с помощью первой формулы в (3.17), а выходной слой вычисляет уровни принадлежности  $u_{jp}(k+1)$  для каждого входного наблюдения  $x(k+1)$ .

Качество кластеризации, выполненной каждым членом ансамбля, может быть оценено с помощью любого индекса нечёткой кластеризации [94]. Среди них одним из наиболее простых и наиболее эффективных индексов является, так называемый, коэффициент разбиения (partition coefficient, PC), который рассчитывается по формуле

$$PC_p(k+1) = \frac{1}{k+1} \sum_{\tau=1}^{k+1} \sum_{j=1}^m u_{jp}^2(\tau). \quad (3.18)$$

Этот коэффициент имеет чёткий физический смысл: чем лучше выражены кластеры, тем выше значение  $PC_p$  (максимальное значение  $PC_p = 1$ ). Минимальное значение ( $PC_p = m^{-1}$ ) достигается, если данные равномерно принадлежат всем кластерам. Но очевидно, что этот феномен представляет наихудшее решение. Этот коэффициент является подходящим в рамках предлагаемой системы, поскольку он может быть рассчитан в online режиме.



Следует отметить, что выражение (3.18) тесно связано с традиционным FCM. Этот коэффициент должен быть модифицирован для рассматриваемого случая в форме

$$PC_p(k+1) = \frac{1}{k+1} \sum_{\tau=1}^{k+1} \sum_{j=1}^m (\alpha_p u_{jp}^2(\tau) + (1 - \alpha_p) u_{jp}(\tau)). \quad (3.19)$$

3.3 Эволюционирующая адаптивная нейро-фаззи сеть Кохонена и online алгоритм ее самообучения в задачах нечеткой кластеризации

3.3.1 Эволюционирующая нечеткая кластеризация на основе возможностного подхода (EPCM)

В [18] предложен оригинальный online эволюционирующий метод нечеткой кластеризации (evolving fuzzy clustering method, EFCM), в основе которого лежит вероятностный подход [96] к решению задачи, при этом основным параметром, который в итоге определяет конечный результат, является радиус формируемых кластеров, выбираемый из эмпирических соображений и определяющий в итоге число возможных классов. Несмотря на эффективность вероятностных алгоритмов нечеткой кластеризации (FCM), их «слабым» местом является «жесткое» ограничение

$$\sum_{j=1}^m u_j(k) = 1 \quad \forall k = 1, 2, \dots, N, \quad (3.20)$$

ведущее к тому, что наблюдение, равно принадлежащее всем классам, т. е. равноудаленное от всех центроидов, может иметь те же уровни принадлежности, что и наблюдение, также равноудаленное от центроидов, но не принадлежащее ни одному из кластеров.

Этого недостатка лишен, так называемый, возможностный (possibilistic) подход (PCM) к нечеткой кластеризации [97], связанный с оптимизацией целевой функции

$$E(u_j, c_j) = \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|x(k) - c_j\|^2 + \sum_{j=1}^m \mu_j \sum_{k=1}^N (1 - u_j(k))^\beta, \quad (3.21)$$

где  $c_j = (c_{j1}, c_{j2}, \dots, c_{jn})^T$  – центроид  $j$ -го кластера, вычисляемый в процессе обработки данных,

$\beta > 1$  – параметр фаззификации («фаззификатор»), определяющий «размытость» границ между кластерами и обычно полагаемый  $\beta = 2$ ,

$\mu_j > 0$  – скалярный параметр, определяющий расстояние, на котором принадлежность принимает значение 0,5, т. е. если

$$\|x(k) - c_j\|^2 = \mu_j, \quad (3.22)$$

то  $u_j(k) = 0,5$ .

Использование возможностного подхода ведет к эволюционирующему методу нечеткой кластеризации (EPCM), который удобно записать в форме последовательности шагов [98]:

Шаг 1: при поступлении наблюдения  $x(1)$  формируется первый кластер с центроидом  $c_1$ .

Шаг 2: при поступлении наблюдения  $x(1)$  проверяется условие

$$\|x(2) - c_1\| \leq \Delta, \quad (3.23)$$

где  $\Delta$  – некоторый априори задаваемый порог.

Если это условие выполняется, то наблюдение  $x(1)$  не формирует новый центроид, т. е. полагается, что оно принадлежит первому кластеру с уровнем принадлежности

$$u_1(2) = \left( 1 + \left( \frac{\|x(2) - c_1\|^2}{\mu_1} \right)^{\frac{1}{\beta-1}} \right)^{-1}. \quad (3.23)$$

Если же выполняется условие

$$\Delta < \|x(2) - c_1\| \leq 2\Delta, \quad (3.24)$$

то происходит коррекция центроида  $c_1$  согласно WTA-правилу самообучения Кохонена [28]:

$$c_1(2) = c_1(1) + \eta(2)(x(2) - c_1(1)), \quad (3.25)$$

где  $\eta(2)$  – параметр шага настройки.

При этом центроид  $c_1$  «подтягивается» к вектору наблюдений  $x(1)$ ; если же для  $x(1)$  выполняется неравенство

$$2\Delta < \|x(2) - c_1\|, \quad (3.26)$$

то формируется второй кластер с центроидом

$$c_2 = x(2). \quad (3.27)$$

При этом должны быть рассчитаны уровни принадлежности  $u_2(1)$  и  $u_1(2)$  по формулам, приведенным ниже.

Шаг  $N$  : итак, если имеется  $N$  наблюдений и  $m$  кластеров с центроидами  $c_j$ , расчеты всех принадлежностей и скорректированные координаты центроидов оцениваются согласно соотношениям [97]:

$$\left\{ \begin{array}{l} u_j(k) = \left( 1 + \left( \frac{\|x(k) - c_j\|^2}{\mu_j} \right)^{\frac{1}{\beta-1}} \right)^{-1}, \\ c_j = \frac{\sum_{k=1}^N u_j^\beta(k) x(k)}{\sum_{k=1}^N u_j^\beta(k)}, \\ \mu_j(k) = \frac{\sum_{k=1}^N u_j^\beta(k) \|x(k) - c_j\|^2}{\sum_{k=1}^N u_j^\beta(k)}, \end{array} \right. \quad (3.28)$$

получаемым в результате минимизации (3.21) по всем оцениваемым параметрам.

Система уравнений (3.28) представляет собой по сути пакетный алгоритм обработки информации так, что при поступлении наблюдения  $x(N+1)$  все расчеты должны быть проведены заново. Понятно, что при достаточно высокой частоте поступления данных рассмотренный подход может оказаться неэффективным.

В связи с этим представляется целесообразной разработка адаптивной нейро-фаззи системы, позволяющей обрабатывать поступающую информацию с помощью рекуррентных процедур, не требующих хранения ранее обработанных данных.

### 3.3.2 Эволюционирующая адаптивная нечеткая кластеризующая нейронная сеть Кохонена (EAFCKN)

Организовать действительно online обработку информации в темпе поступления новых данных можно, используя вместо традиционных пакетных FCM и PCM их рекуррентные аналоги [99,100], являющиеся по сути градиентными процедурами оптимизации принятого критерия самообучения (кластеризации). В [101] было показано, что эти процедуры являются по сути WTM-правилами самообучения Кохонена с сужающейся функцией соседства и с успехом могут быть использованы для обучения самоорганизующихся карт с фиксированной архитектурой.

Реализовать эволюционирующую нейро-фаззи кластеризующую систему можно на основе двухслойной адаптивной нейро-фаззи сети [93] с архитектурой, которая показана на рис. 3.4.

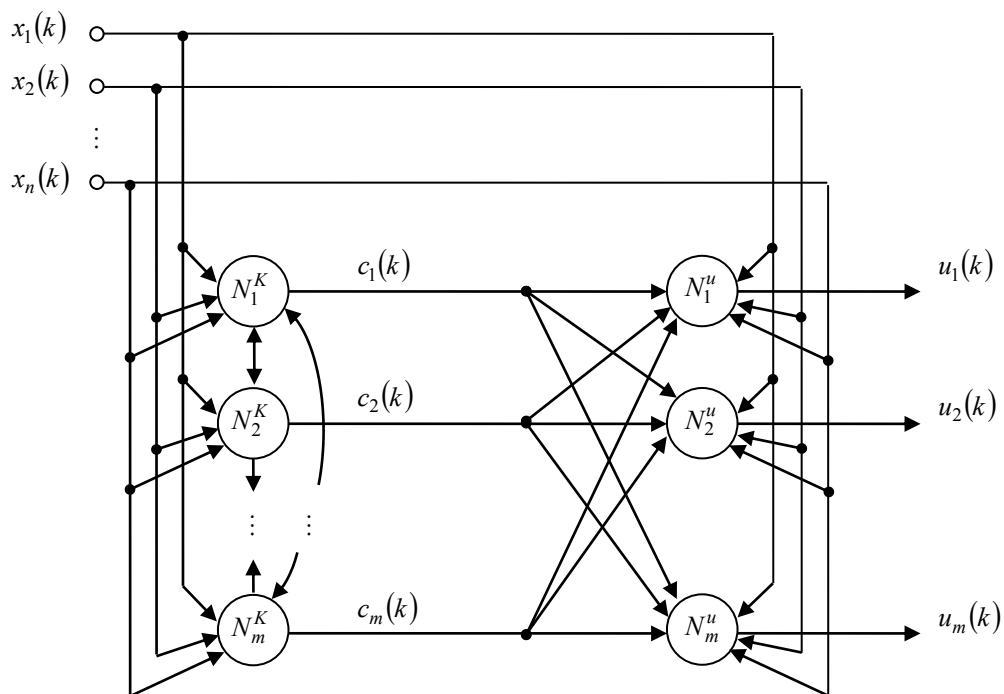


Рисунок 3.4 – Архитектура адаптивной кластеризующей нейро-фаззи сети

Первый скрытый слой сети образован обычными нейронами Кохонена  $N_j^K$ , связанными между собой латеральными связями, по которым реализуется процесс конкуренции. Выходной слой сети, образованный узлами  $N_j^u$ , предназначен для вычисления уровней принадлежности каждого наблюдения  $x(k)$  каждому  $j$ -му кластеру,  $j = 1, 2, \dots, m$ .

Для настройки центроидов кластеров используется рекуррентная процедура самообучения, имеющая вид [101]:

$$\left\{ \begin{array}{l} c_j(k+1) = c_j(k) + \frac{u_j^\beta(k)}{k+1} (x(k+1) - c_j(k)), \\ u_j(k+1) = \frac{1}{1 + \left( \frac{\|x(k+1) - c_j(k+1)\|^2}{\mu_j(k)} \right)^{\frac{1}{1-\beta}}}, \\ \mu_j(k+1) = \frac{\sum_{p=1}^{k+1} u_j^\beta(p) \|x(p) - c_j(k+1)\|^2}{\sum_{p=1}^{k+1} u_j^\beta(p)}. \end{array} \right. \quad (3.29)$$

Несложно заметить, что первое соотношение (3.29) и есть WTM-правило самообучения с сужающейся функцией соседства  $(k+1)^{-1} u_j^\beta(k)$ .

Процесс эволюции системы подобно предыдущему начинается с единственного нейрона Кохонена, который уточняет координаты первого центроида  $c_1$ . Следующий нейрон добавляется в сеть при выполнении условия (3.26), которое в данном случае принимает форму

$$2\Delta < \|x(k) - c_1(k-1)\|. \quad (3.30)$$

В этот момент формируется нейрон с центроидом  $c_2(k) = x(k)$ . Здесь следует заметить, что поскольку в нейронных сетях Кохонена данные предварительно нормируются на гипершар так, что

$$\|x(k)\|^2 = \|c_j(k)\|^2 = 1, \quad (3.31)$$

неравенство (3.26), определяющее необходимость введения в сеть новых нейронов, приобретает форму

$$-1 \leq 1 - 2\Delta^2 < c_j^T(k-1)x(k) \leq 1, \quad \forall j = 1, 2, \dots, m, \quad (3.32)$$

или

$$-1 \leq 1 - 2\Delta^2 < \cos(c_j(k-1), x(k)) \leq 1. \quad (3.33)$$

Таким образом, наращивание архитектуры происходит в результате постоянного контроля неравенств (3.32) или (3.33) и производится в случае, если эти неравенства нарушаются. Заметим также, что порог  $\Delta$  здесь имеет тот же смысл, что и в (3.26).

Благодаря использованию именно возможностного подхода, целесообразна реализация еще одной «ветви эволюции», а именно, если в какой-то момент времени обнаружится, что принадлежности наблюдения  $x(k)$  не превышают некоторого дополнительного порогового значения

$$u_j(k) < \varepsilon \quad \forall j = 1, 2, \dots, m, \quad (3.34)$$

т. е. наблюдение  $x(k)$  достаточно далеко отстоит от всех уже сформированных центроидов, это также может служить сигналом для формирования нового кластера

$$c_{m+1}(k) = x(k). \quad (3.35)$$

Для оценки качества нечеткой кластеризации может быть использован популярный индекс Кси–Бени [102] в расширенной форме [94].

Для фиксированной выборки, содержащей  $N$  наблюдений этот индекс имеет вид

$$EXB(N) = \frac{\left( \sum_{k=1}^N \sum_{j=1}^m u_j^\beta(k) \|x(k) - c_j(N)\|^2 \right) / N}{\min_{j \neq l} \|c_j(N) - c_l(N)\|^2} = \frac{NEXB(N)}{DEXB(N)}. \quad (3.36)$$

Для online обработки этот индекс подобно центроидам кластеров также может рассчитываться рекуррентно:

$$EXB(k+1) = \frac{NEXB(k+1)}{DEXB(k+1)} = \frac{NEXB(k) + \frac{1}{k+1} \left( \sum_{j=1}^m u_j^\beta(k+1) \|x(k+1) - c_j(k+1)\|^2 - NEXB(k) \right)}{\min_{j \neq l} \|c_j(k+1) - c_l(k+1)\|^2}. \quad (3.37)$$

Включение выражения (3.37) в процедуру обучения (3.29) позволяет организовать дополнительный контроль за количеством кластеров,



формируемых системой. Так, вводя третий порог  $\delta$  и проверяя на каждом шаге условие

$$EXB(k+1) > \delta, \quad (3.38)$$

можно остановить процесс наращивания нейронов в случае нарушения неравенства (3.38).

### Выводы к разделу 3

1. Рассмотрена самоорганизующаяся карта Кохонена, её архитектура и методы обучения.

2. Рассмотрен вероятностный подход к нечёткой кластеризации, различные целевые функции нечёткой кластеризации, подлежащие минимизации. Рассмотрены методы настройки системы с фаззификатором  $\beta$  и параметром  $\alpha$ , полученные в результате решения систем уравнений Каруша–Куна–Таккера.

3. Предложен ансамбль адаптивных нейро-фаззи сетей Кохонена для обработки данных в online режиме, в котором каждая отдельная нейро-фаззи сеть Кохонена работает со своим значением параметра  $\alpha$ .

4. Проанализированы недостатки вероятностного подхода к нечёткой кластеризации, рассмотрен возможностный подход к нечёткой кластеризации.

5. Модифицирован метод самообучения адаптивной нейро-фаззи сети Кохонена, основанный на возможностном подходе к нечёткой кластеризации и способный обрабатывать данные в online режиме.

РАЗДЕЛ 4  
МОДЕЛИРОВАНИЕ И ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ  
РАЗРАБОТАННЫХ АРХИТЕКТУР И МЕТОДОВ

4.1 Моделирование эволюционирующей многослойной нейро-фаззи системы с гибридным обучением

Эффективность предложенной эволюционирующей многослойной нейро-фаззи системы с гибридным обучением была продемонстрирована при решении задачи эмуляции динамического объекта [103], описываемого уравнением

$$y_p(k+1) = f(y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)). \quad (4.1)$$

Для эмуляции объекта использовалась нейро-фаззи система с количеством входов  $n = 5$ .

Обучение системы производилось на основе выборки, полученной с помощью уравнения (4.1) с управляющими сигналами

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_2^2}, \quad (4.2)$$

$$u(k) = \sin(2\pi k / 250) \quad (4.3)$$

для  $k < 500$ . В качестве проверочных данных эмулятора использовался динамический объект (4.1) с теми же управляющими сигналами для  $k \leq 500$ , а далее для  $k > 500$  вид управляющих сигналов менялся на

$$u(k) = 0.8\sin(2\pi k / 250) + 0.2\sin(2\pi k / 25). \quad (4.4)$$

В качестве критерия качества прогноза была использована среднеквадратичная ошибка (mean squared error, MSE):

$$MSE = \frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k))^2 \quad (4.5)$$

Фазовый портрет для тестовой части выборки представлен на рис. 4.1.

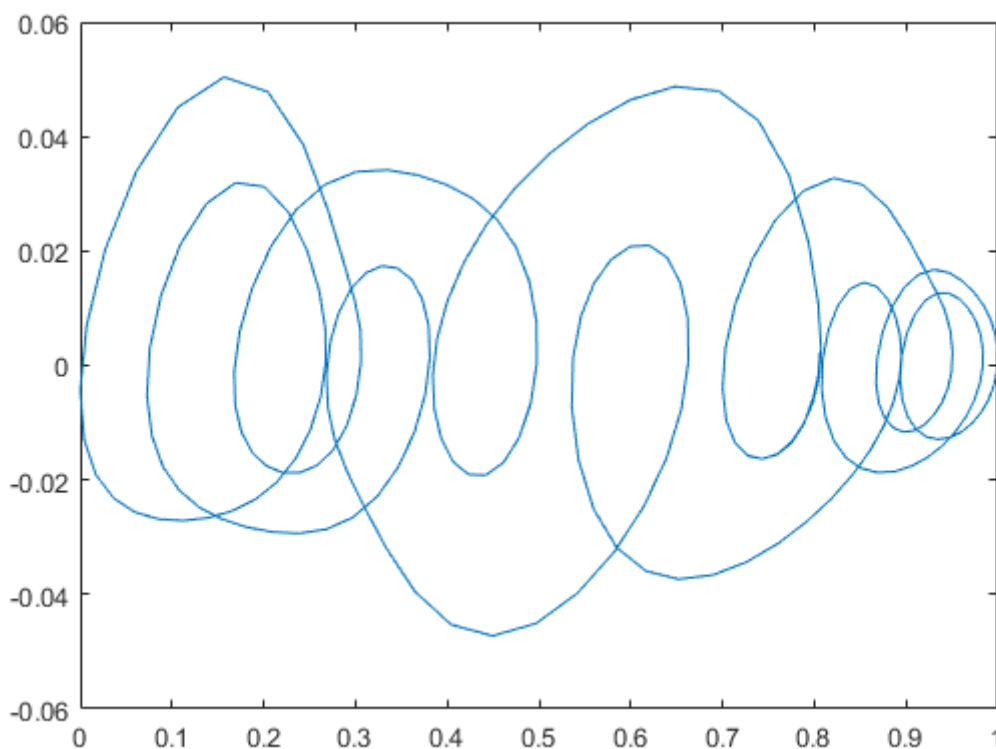


Рисунок 4.1 – Фазовый портрет для тестовой части выборки

Результаты эмуляции динамического объекта (4.1) приведены на рис. 4.2.

Фактические значения на рисунке представлены сплошной линией, а значения эмуляции – пунктирной линией.

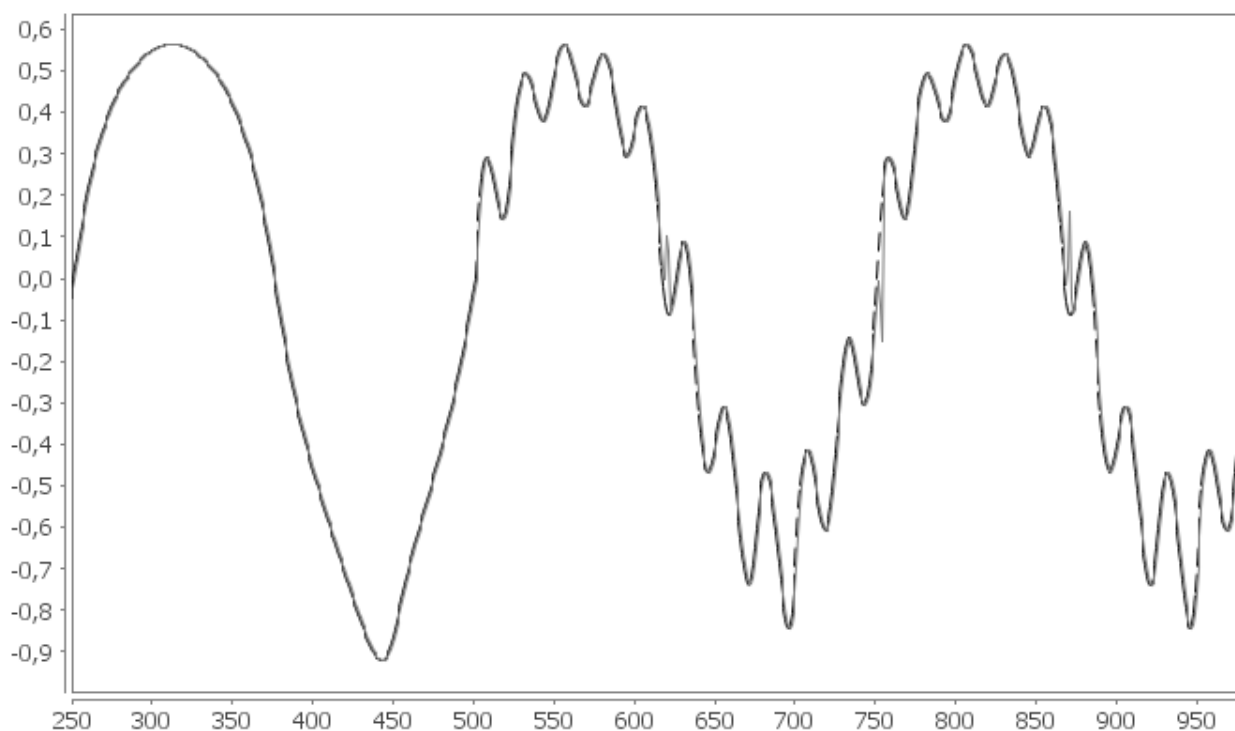


Рисунок 4.2 – Фрагмент результатов эмуляции динамического объекта

Как видно, две кривые, представляющие фактические значения и значения эмуляции, практически идентичны.

В табл. 1 приведены результаты процесса эмуляции на основе рассматриваемой эволюционирующей многослойной нейро-фаззи системы с гибридным обучением.

Таблица 4.1 – Результаты эмуляции динамического объекта

Максимальное количество функций принадлежности $h$	MSE
5	0.00145
10	0.00122
20	0.00121

Для проведения экспериментов использовались системы с разным максимальным количеством функций принадлежности  $h$ .

Как следует из приведенных экспериментальных результатов, предложенная архитектура эволюционирующей многослойной нейро-фаззи системы обеспечивает высокую точность эмуляции.

#### 4.2 Моделирование эволюционирующей МГУА-нейро-фаззи системы

Эффективность предложенной эволюционирующей МГУА-нейро-фаззи системы и её методов обучения была продемонстрирована при решении задачи прогнозирования нестационарных сигналов.

Для эксперимента использовалась выборка данных, содержащая информацию о ежемесячном количестве пятен на солнце. Данные отображают показатели с 1749 по 1983 год.

Общий объём выборки составил 2820 наблюдений. Она была разделена на две части. Обучающая выборка содержала 2256 наблюдений, тестовая – 564 наблюдения.

В качестве критерия качества прогноза был использован квадратный корень из среднеквадратичной ошибки (root mean squared error, RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N (y(k) - \hat{y}(k))^2} \quad (4.6)$$

RMSE является по сути квадратным корнем из MSE, рассмотренной выше в (4.5).

Для сравнения качества прогнозирования использовались многослойный персептрон (MLP), радиально-базисная нейронная сеть (RBFN) и ANFIS.

В табл. 4.2 представлены результаты систем: ошибка на этапе обучения, ошибка на этапе тестирования и время обучения системы.

Таблица 4.2 – Сравнение результатов систем

Системы	RMSE (обучающая)	RMSE (тестовая)	Время, с
MLP	0.1058	0.1407	0.2813
RBFN	0.1066	0.2155	0.2219
ANFIS	0.0559	0.1965	0.8906
Предложенная система	0.0790	0.1098	0.4210

Результат прогнозирования для проверочной выборки приведен на рис. 4.3. Синей линией показан прогнозируемый ряд, сиреневой – прогноз системы, серой – значение ошибки.

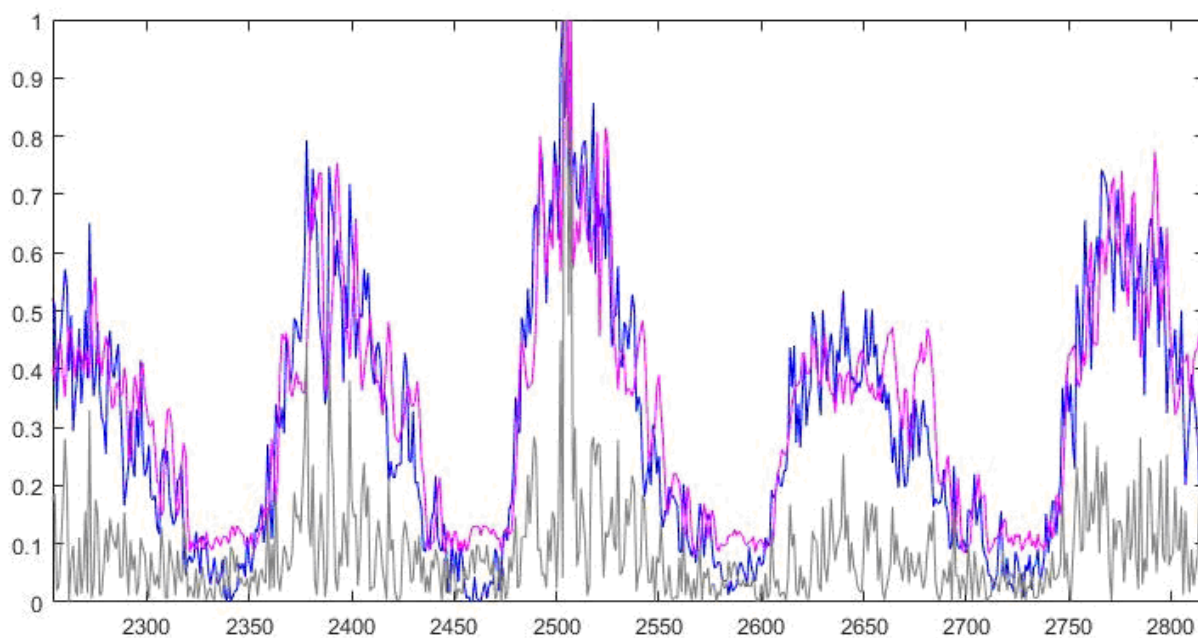


Рисунок 4.3 – Результаты прогнозирования ряда эволюционирующей МГУА-нейро-фаззи системой

### 4.3 Моделирование эволюционирующей каскадной системы

Эффективность предложенной эволюционирующей каскадной системы с нейро-фаззи узлами, а также эволюционирующей каскадной системы с нео-фаззи узлами и их методов обучения была продемонстрирована при решении задачи прогнозирования нестационарных сигналов.

В качестве тестовой выборки был использован временной ряд, описывающий ежемесячное давление над уровнем моря за период с 1882 по 1998 год (Darwin sea level pressure, выборка получена из хранилища данных DataMarket).

На рис. 4.4 представлен фазовый портрет прогнозируемого ряда.

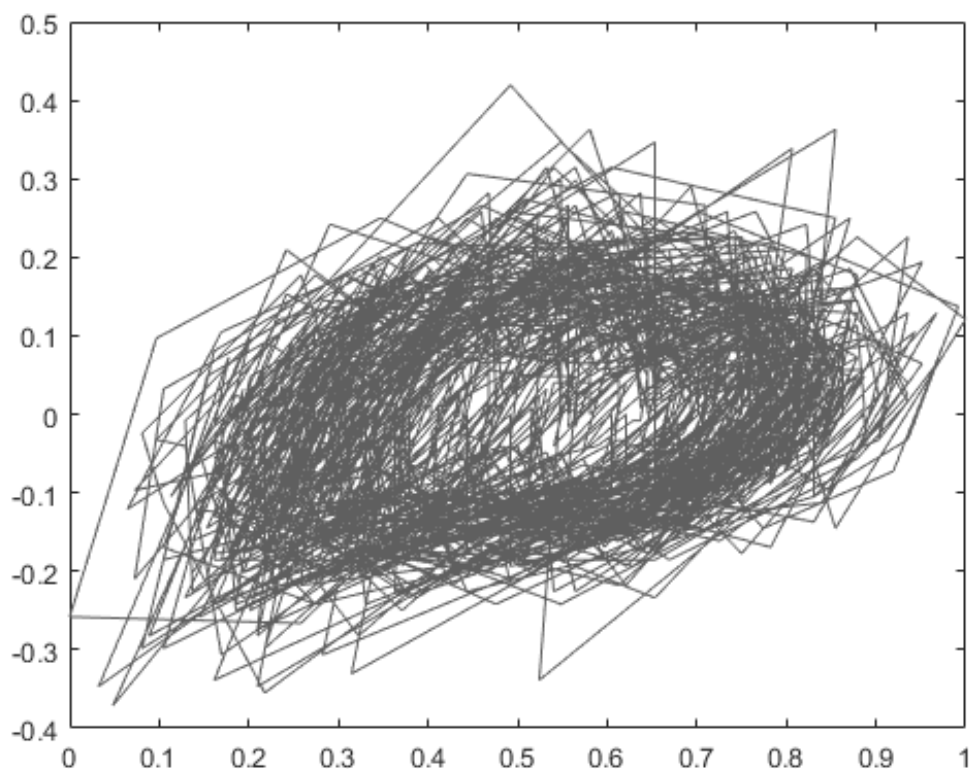


Рисунок 4.4 – Фазовый портрет прогнозируемого ряда

Общий объём выборки составил 1400 наблюдений. Она была разделена на 2 части. Обучающая выборка содержала 1100 наблюдений, тестовая – 300 наблюдений.

Для сравнения эффективности предложенной системы с эффективностью других существующих систем были выбраны многослойный персептрон, радиально-базисная нейронная сети и ANFIS. В качестве критерия качества прогнозирования использовалась среднеквадратичная ошибка MSE (4.5).

Результаты проведенных экспериментов приведены в табл. 4.3.

Таблица 4.3 – Результаты экспериментов

Тип системы	MSE на обучающей выборке	MSE на тестовой выборке	Время обучения, с
Многослойный персептрон	0.0150	0.0168	0.2500
Радиально-базисная нейронная сеть	0.0172	0.0224	0.2391
ANFIS	0.0157	0.0165	0.2031
Эволюционирующая каскадная система с нейро-фаззи узлами	0.0146	0.0156	0.2067
Эволюционирующая каскадная система с нео-фаззи узлами	0.0144	0.0153	0.2060

Как видно из табл. 4.3, предложенные эволюционирующие каскадные системы достаточно неплохо показали себя при решении задачи прогнозирования.



Стоит отметить, что время обучения предложенных систем было одним из лучших. Если сравнивать качество прогнозирования, то эволюционирующая каскадная система с нео-фаззи узлами показала наилучший результат.

Результат прогнозирования для проверочной выборки для эволюционирующей каскадной нейро-фаззи системы приведен на рис. 4.5. Сплошной чёрной линией показан прогнозируемый ряд, пунктирной – прогноз эволюционирующей каскадной системы с нейро-фаззи узлами, сплошной серой – значение ошибки.

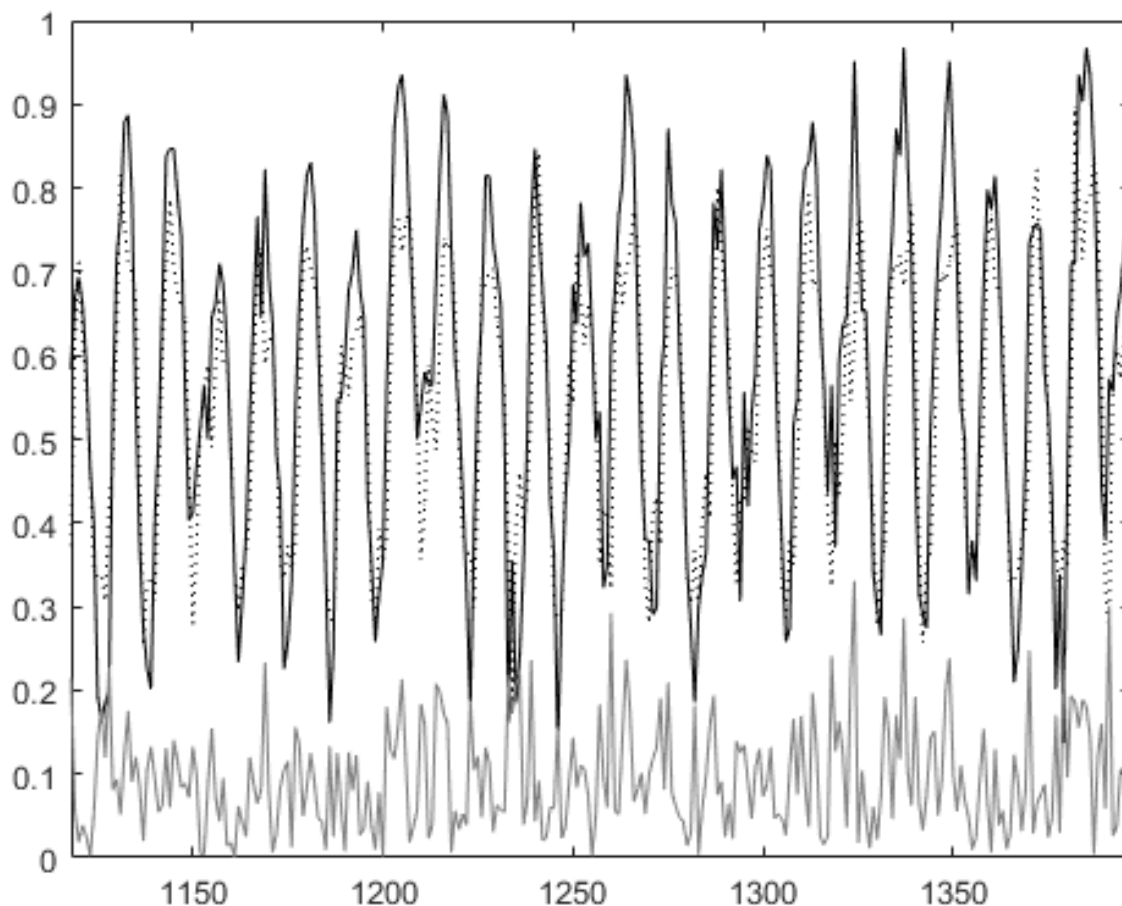


Рисунок 4.5 – Результаты прогнозирования ряда для эволюционирующей каскадной нейро-фаззи системы

Результат прогнозирования для проверочной выборки для эволюционирующей каскадной нео-фаззи системы приведен на рис. 4.6. Сплошной чёрной линией показан прогнозируемый ряд, пунктирной – прогноз эволюционирующей каскадной системы с нео-фаззи узлами, сплошной серой – значение ошибки.

Если сравнивать эволюционирующую каскадную систему с нейро-фаззи узлами и эволюционирующую каскадную систему с нео-фаззи узлами, то можно сказать, что вторая показала лучший результат по сравнению с первой как по среднеквадратичной ошибке на этапе обучения, так и по среднеквадратичной ошибке на этапе тестирования, а также по времени обучения системы.

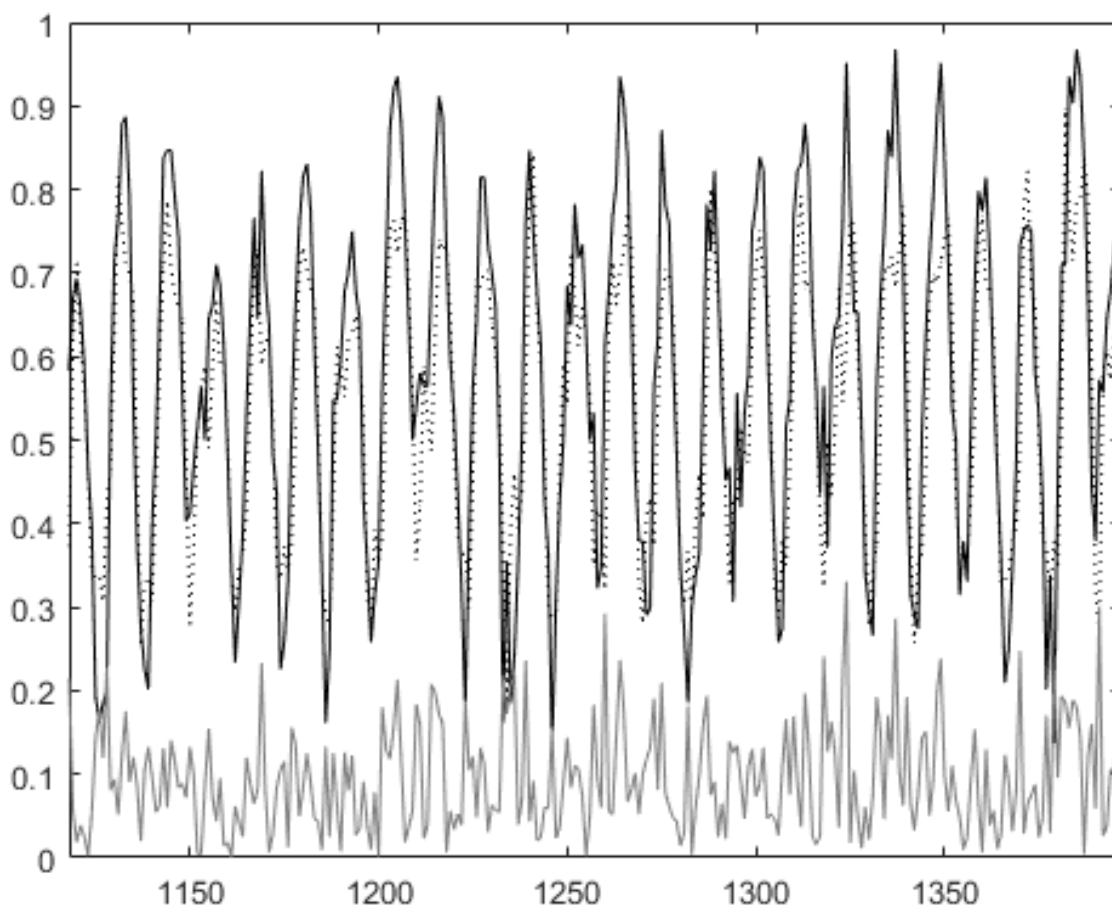


Рисунок 4.6 – Результаты прогнозирования ряда для эволюционирующей каскадной нео-фаззи системы

Также для исследования эволюционирующей каскадной системы с нео-фаззи узлами был проведен ещё один эксперимент. Его задачей была идентификация нелинейной системы, описанной как

$$y_t = \frac{\sum_{i=1}^m y_{t-i}}{1 + \sum_{i=1}^m (y_{t-i})^2} + u_{t-1}, \quad (4.7)$$

$$u_t = \sin(2\pi t / 20), \quad (4.8)$$

при условии, что  $y_j = 0, j = 1, \dots, m, m = 10$ .

На рис. 4.7 представлен фазовый портрет прогнозируемого ряда.

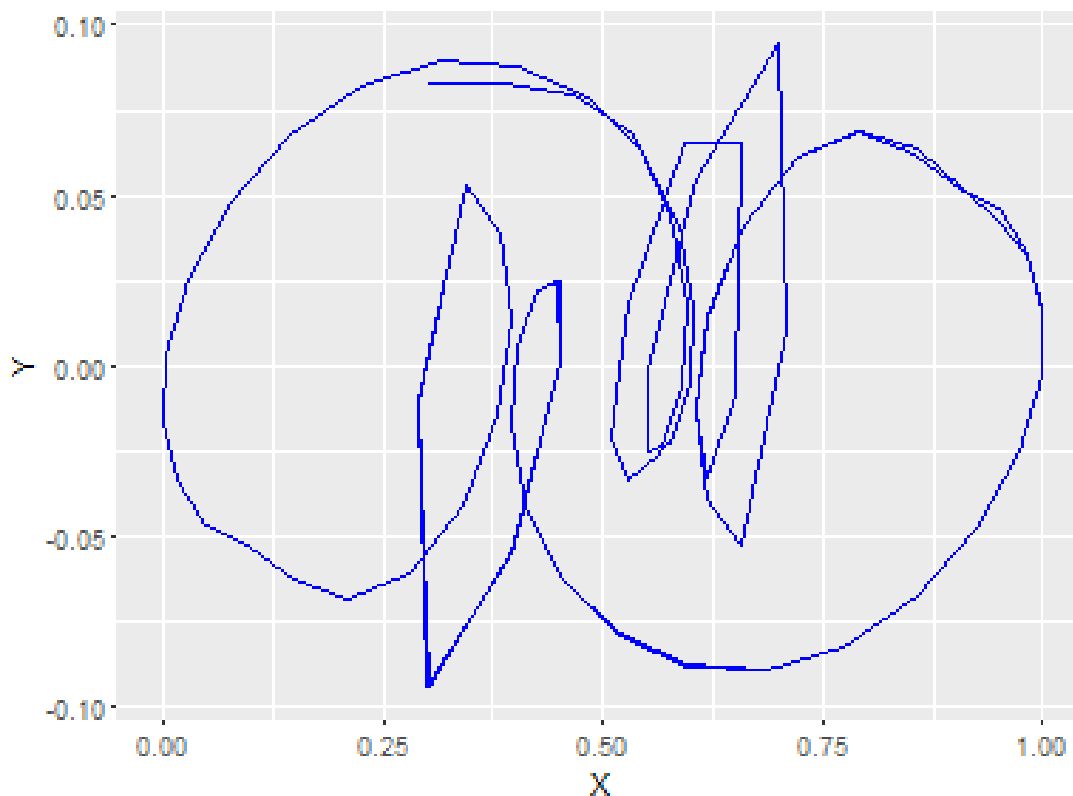


Рисунок 4.7 – Фазовый портрет прогнозируемого ряда

Модель представлена в форме

$$\hat{y}_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-10}, u_{t-1}). \quad (4.9)$$

Общий объём выборки составил 2500 наблюдений. Она была разделена на две части. Обучающая выборка содержала 2000 наблюдений, тестовая – 500 наблюдений.

На рис. 4.8 представлен вариант равномерного распределения треугольных функций принадлежности нео-фаззи узла.

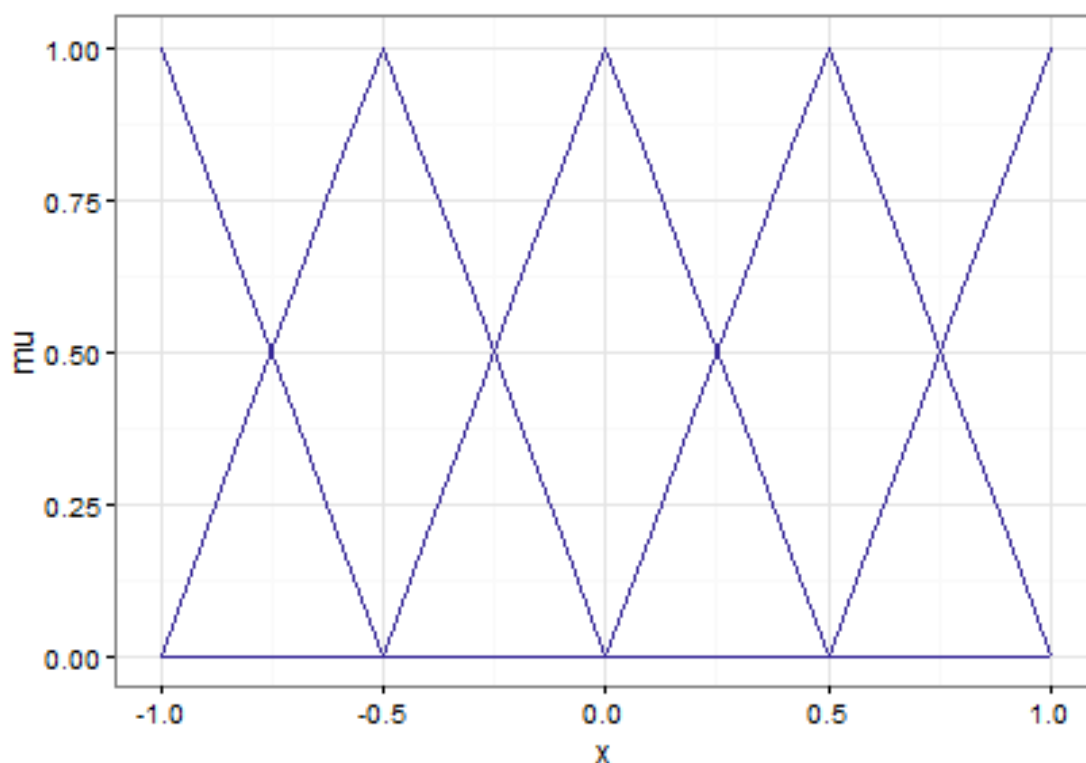


Рисунок 4.8 – Вариант равномерного распределения пяти треугольных функций принадлежности нео-фаззи узла

В данном случае узел содержит 5 равномерно распределённых функций принадлежности.

Сравнение качества прогнозирования проводилось с использованием многослойного персептрона (MLP), радиально-базисной нейронной сети (RBFN) и системы ANFIS.

Результаты прогнозирования представлены в табл. 4.4.

Таблица 4.4 – Результаты прогнозирования временного ряда

Системы	Количество настраиваемых параметров	RMSE (обучающая)	RMSE (тестовая)	Время, с
Многослойный персептрон	51	0.1058	0.1407	0.2813
Радиально-базисная нейронная сеть	21	0.1066	0.2155	0.2219
ANFIS	80	0.0559	0.1965	0.8906
Эволюционирующая каскадная система с нео-фаззи узлами	20	0.0784	0.1081	0.2381

В качестве критерия качества прогноза был использован квадратный корень из среднеквадратичной ошибки (root mean squared error, RMSE) (4.6).

Стоит отметить, что предложенная эволюционирующая каскадная система с нео-фаззи узлами показала лучший результат по точности прогноза.

Результат прогнозирования для проверочной выборки приведен на рис. 4.9. Синей линией показан прогнозируемый ряд, сиреневой – прогноз системы, серой – значение ошибки.

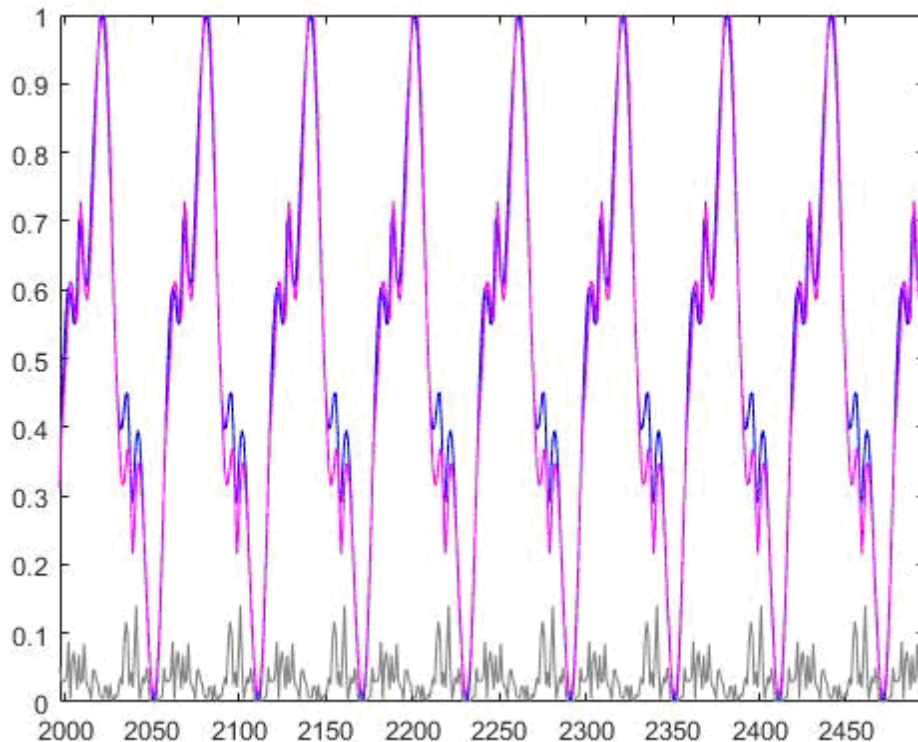


Рисунок 4.9 – Результаты прогнозирования временного ряда для эволюционирующей каскадной нео-фаззи системы

#### 4.4 Моделирование взвешенной ANARX-системы (WANARX-системы)

Для демонстрации эффективности предложенной взвешенной ANARX-системы был проведен прогноз временного ряда.

Для проведения эксперимента использовалась обучающая выборка, содержащая данные о потреблении электричества. Этот набор данных описывает количество потребляемой электроэнергии в 1997 году.

Для экспериментов использовалось 5000 наблюдений из этой выборки. Обучающая выборка содержала 3000 наблюдений, тестовая – 2000 наблюдений.

В качестве критерия качества прогнозирования использовался квадратный корень из среднеквадратичной ошибки RMSE (4.6).

Для сравнения качества прогнозирования использовались многослойный персептрон (MLP), радиально-базисная нейронная сеть (RBFN) и ANFIS.

Результаты прогнозирования для разных систем представлены в табл. 4.4.

Поскольку многослойный персептрон не может работать в online режиме, для сравнения было выбрано 2 варианта этой системы.

Первый многослойный персептрон обучался в течение одной эпохи (что по сути похоже на обработку данных в online режиме).

Второй многослойный персептрон обучался в течение пяти эпох, причём количество настраиваемых параметров в этом случае было приблизительно равным количеству настраиваемых параметров в предлагаемых системах. В обоих случаях многослойные персептроны содержали 4 входа и 7 узлов в скрытом слое. Количество настраиваемых параметров было равным 43.

Для второго многослойного персептрона время работы оказалось почти в 2 раза большим, но, в то же время, качество прогнозирования было тоже почти в 2 раза лучше.

Также было выбрано две радиально-базисные нейронные сети. Количество параметров первой радиально-базисной нейронной сети было практически равным количеству параметров предложенных систем.

Архитектура второй радиально-базисной нейронной сети была выбрана с учётом качества её работы.

В первом случае радиально-базисная нейронная сеть имела 3 входа и 7 ядерных функций. Во втором случае у радиально-базисной нейронной сети было также 3 входа, но 12 ядерных функций.

Таблица 4.5 – Сравнение результатов прогнозирования

Системы	Количество настраиваемых параметров	RMSE (обучающая)	RMSE (тестовая)	Время, с
MLP (вариант 1)	43	0.0600	0.0700	0.4063
MLP (вариант 1)	43	0.0245	0.0381	0.9219
RBFN (вариант 1)	36	0.0681	0.0832	0.6562
RBFN (вариант 2)	61	0.0463	0.0604	1.0250
ANFIS	80	0.0237	0.0396	0.7031
ANARX с нео-фаззи узлами	37	0.0903	0.0922	0.4300
Взвешенная ANARX с нео-фаззи узлами	36	0.0427	0.0573	0.3750

Система ANFIS показала один из лучших результатов прогнозирования в этом эксперименте. Она содержала 4 входа, 55 узлов и обучалась в течение пяти эпох.

Результат прогнозирования для проверочной выборки для ANARX-системы приведен на рис. 4.10. Синей линией показан прогнозируемый ряд, сиреневой – прогноз системы, серой – значение ошибки.



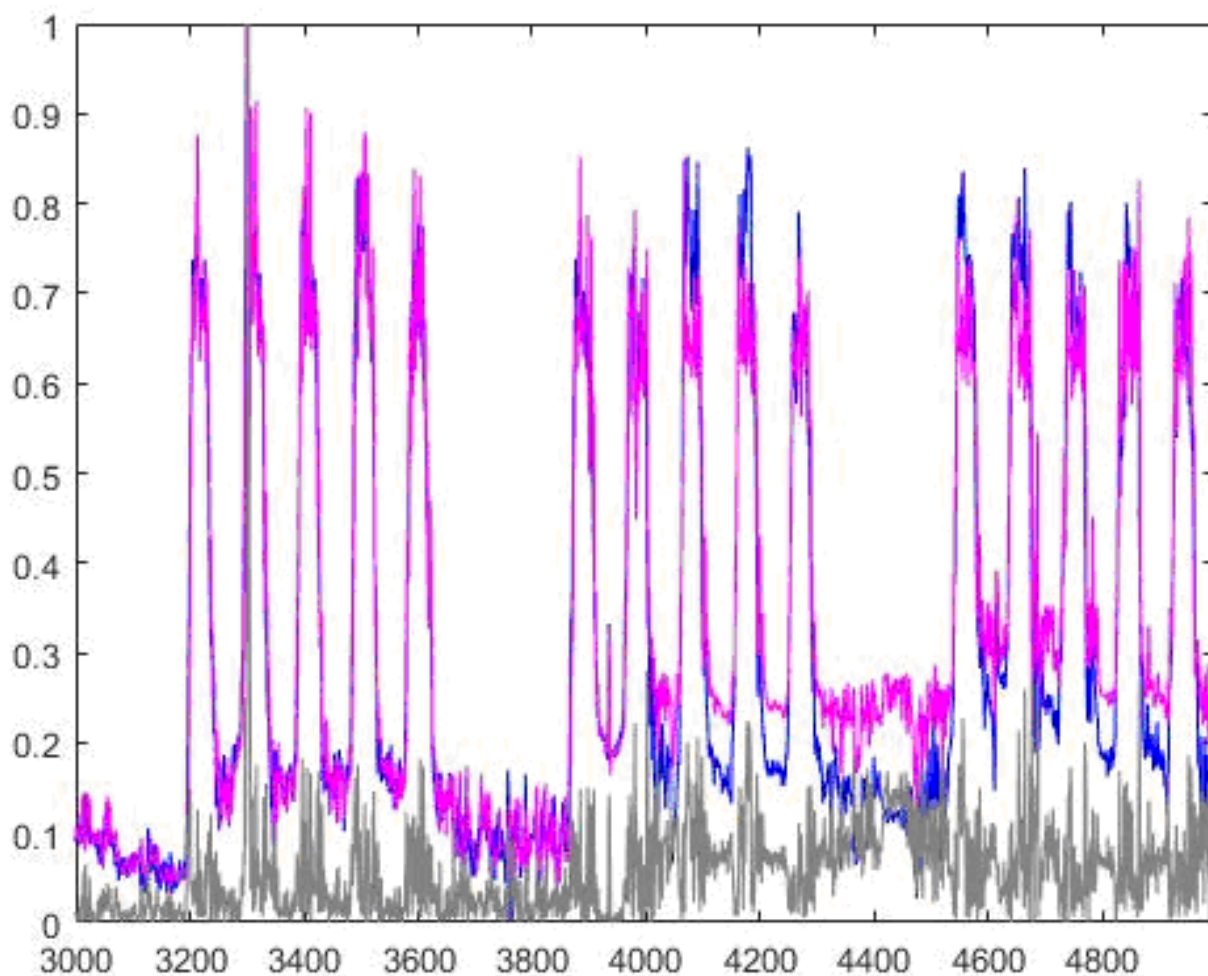


Рисунок 4.10 – Результаты прогнозирования для ANARX-системы с нео-фаззи узлами

Предложенная ANARX-система с нео-фаззи узлами содержала 2 входа, 2 узла, 9 функций принадлежности.

Для обучения параметр  $\alpha$  был задан равным 0.62. Система содержала 37 настраиваемых параметров.

В качестве функций принадлежности использовались  $B$ -сплайны с  $q = 2$  (т. е. треугольные функции принадлежности).

Качество прогноза этой системы было достаточно высоким, а время обучения – самое меньшее.

Результат прогнозирования для проверочной выборки для взвешенной ANARX-системы (WANARX-системы) с нео-фаззи узлами приведен на рис. 4.11. Синей линией показан прогнозируемый ряд, сиреневой – прогноз системы, серой – значение ошибки.

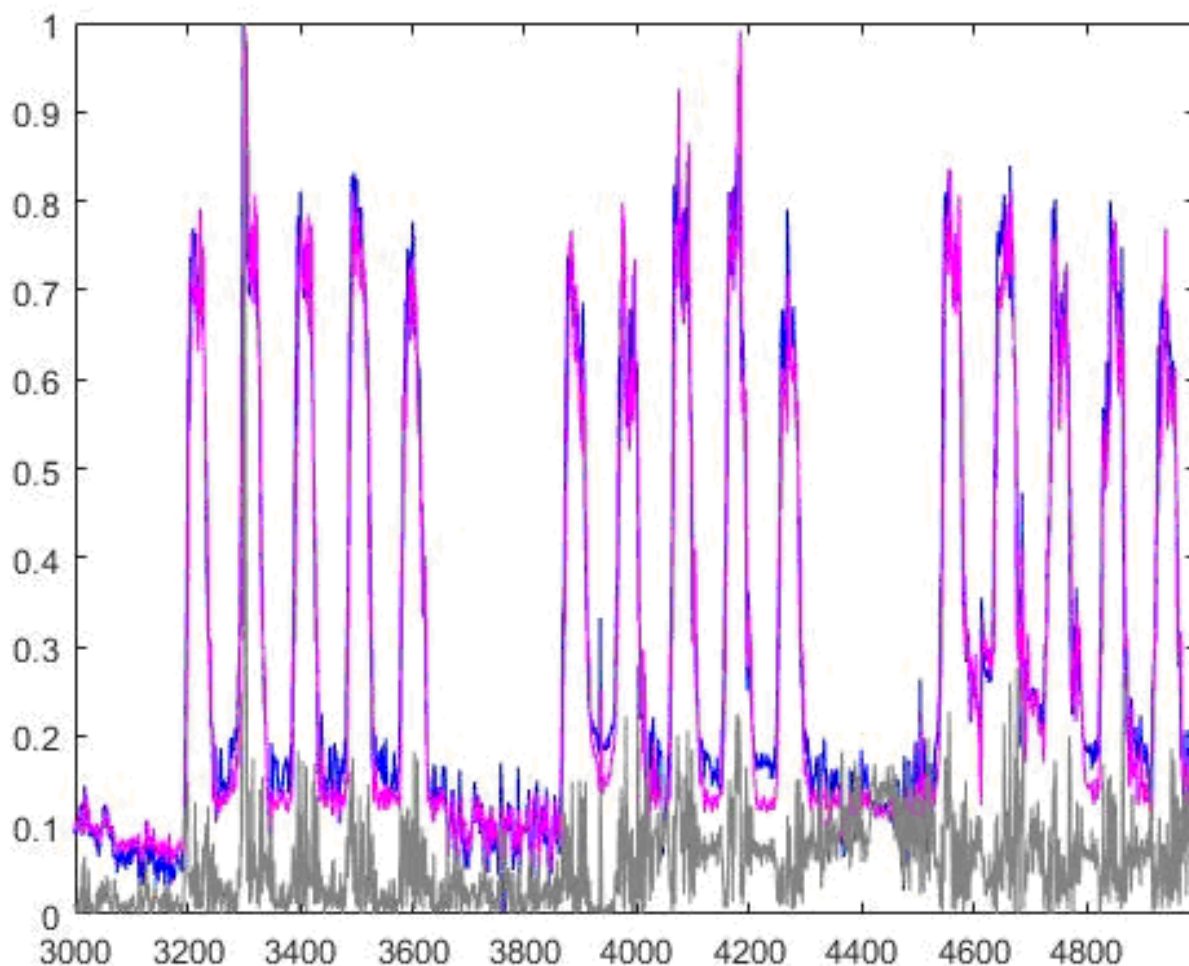


Рисунок 4.11 – Результаты прогнозирования для взвешенной ANARX-системы с нео-фаззи узлами

Предложенная взвешенная WANARX-система с нео-фаззи узлами содержала 2 входа, 2 узла, 8 функций принадлежности.

Параметр  $\alpha$  был задан равным 0.9.

Система содержала 37 настраиваемых параметров.

В качестве функций принадлежности также использовались  $B$ -сплайны с  $q=2$  (т. е. треугольные функции принадлежности). Эта система продемонстрировала лучший результат по скорости и качеству прогноза по сравнению с ANARX-моделью с нео-фаззи узлами.

#### 4.5 Моделирование эволюционирующей адаптивной нечёткой нейронной сети Кохонена

Рассмотрим на примере более подробно настройку центров кластеров для эволюционирующей адаптивной нечёткой нейронной сети Кохонена для кластеризации.

Далее на рисунках красные точки обозначают центры кластеров, синими точками обозначены уже обработанные наблюдения, а зелёной точкой показано новое наблюдение из обучающей выборки, поступившее на обработку.

На первом этапе настройки центров возможны три ситуации:

1. Для расстояний между новым наблюдением и центрами существующих кластеров выполняется условие

$$\exists j \quad \Delta < \|x(k) - c_j\| \leq 2\Delta. \quad (4.10)$$

В этом случае кластеры сдвигают свои центры по направлению к новому наблюдению. На рис. 4.12 и 4.13 представлены примеры такого случая.

Стоит отметить, что перемещение центра кластера зависит от параметра  $\eta(k)$ .

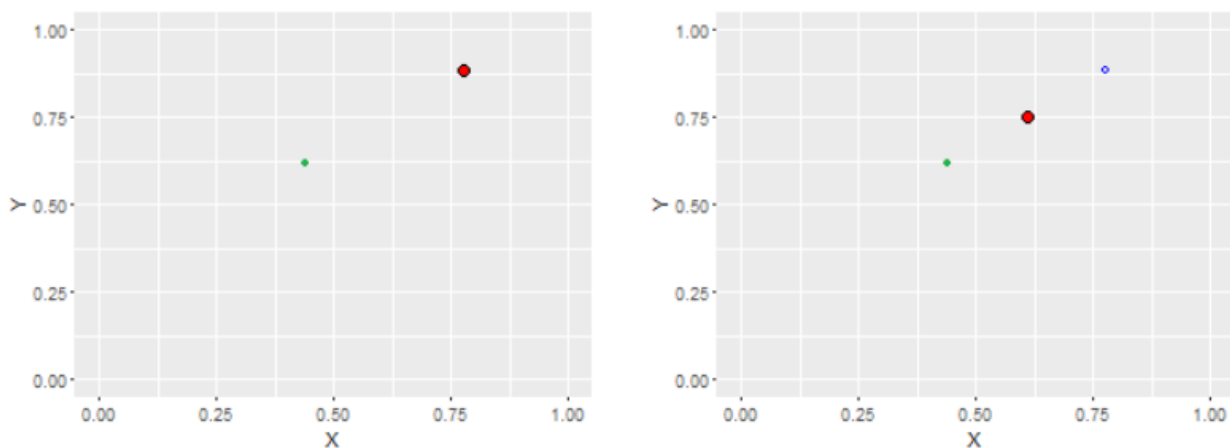


Рисунок 4.12 – Смещение центра кластера к новому наблюдению

При поступлении нового наблюдения, отмеченного зелёным цветом, центр кластера, отмеченный красным цветом, смещается в направлении к этому наблюдению. В данных примерах параметр  $\eta(k) = 0.5$ .

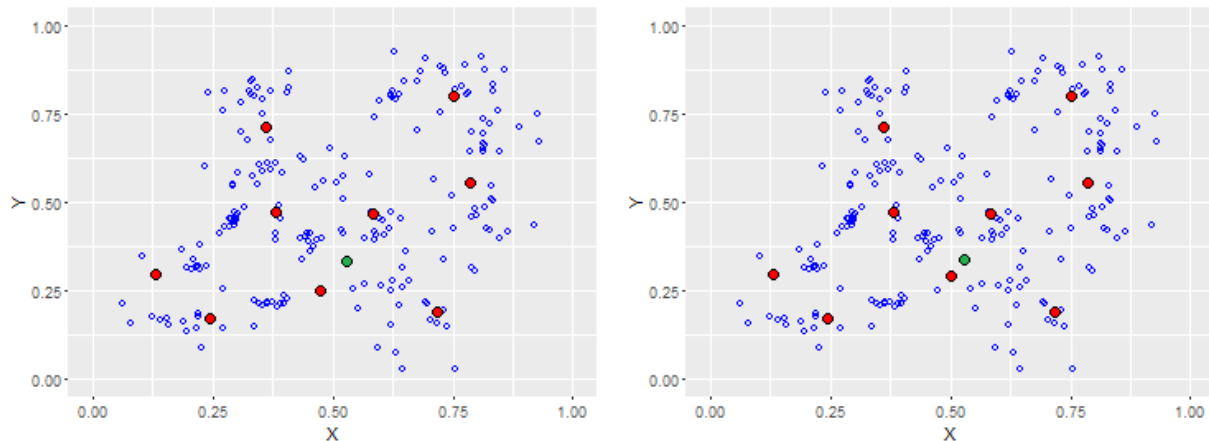


Рисунок 4.13 – Смещение центра кластера к новому наблюдению

2. Для расстояний между новым наблюдением и центрами существующих кластеров выполняется условие

$$\forall j \quad 2\Delta < \|x(k) - c_j\|. \quad (4.11)$$

В этом случае создаётся новый кластер. На рис. 4.14 и 4.15 представлены пример такого случая. Новое наблюдение, отмеченное слева зелёным цветом, становится центром нового кластера.

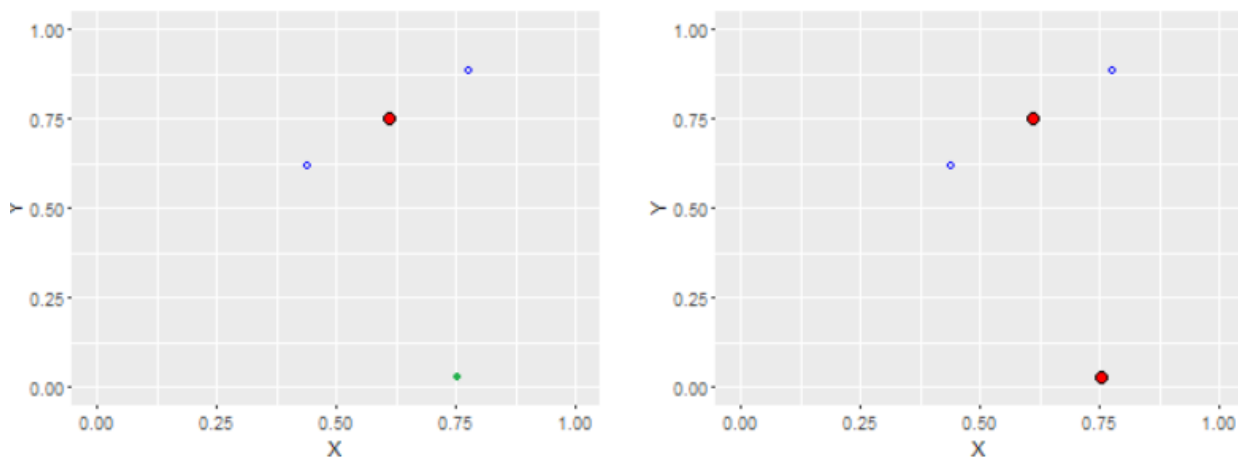


Рисунок 4.14 – Создание нового кластера с приходом нового наблюдения

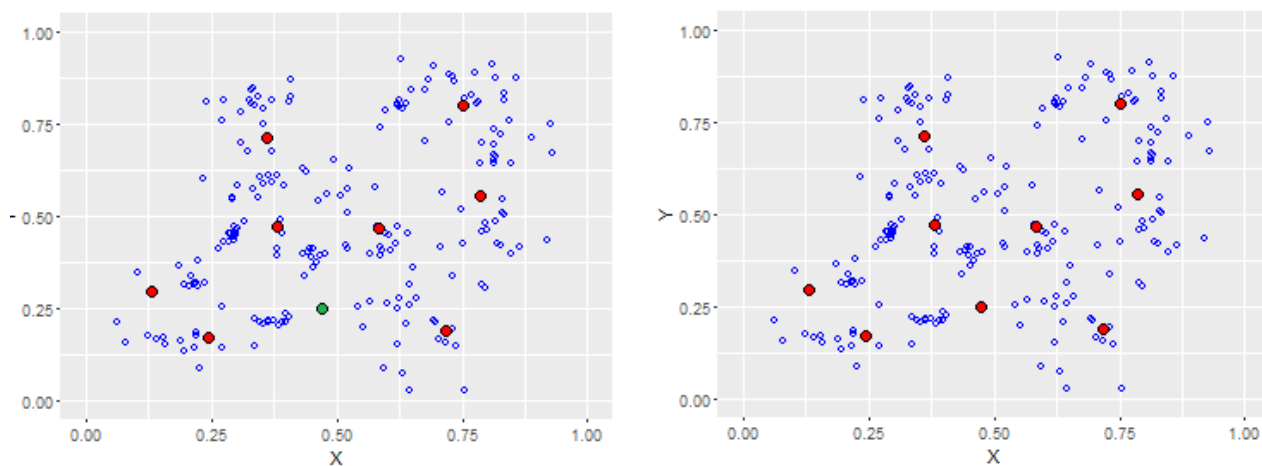


Рисунок 4.15 – Создание нового кластера с приходом нового наблюдения

3. Для расстояний между новым наблюдением и центрами существующих кластеров выполняется условие

$$\forall j \quad \|x(k) - c_j\| \leq \Delta. \quad (4.12)$$

В этом случае это наблюдение принадлежит к существующим кластерам, а центры этих кластеров не изменяются. На рис. 4.16 представлен пример такого случая. С приходом нового наблюдения, отмеченного зелёным цветом, центры существующих кластеров не изменяются.

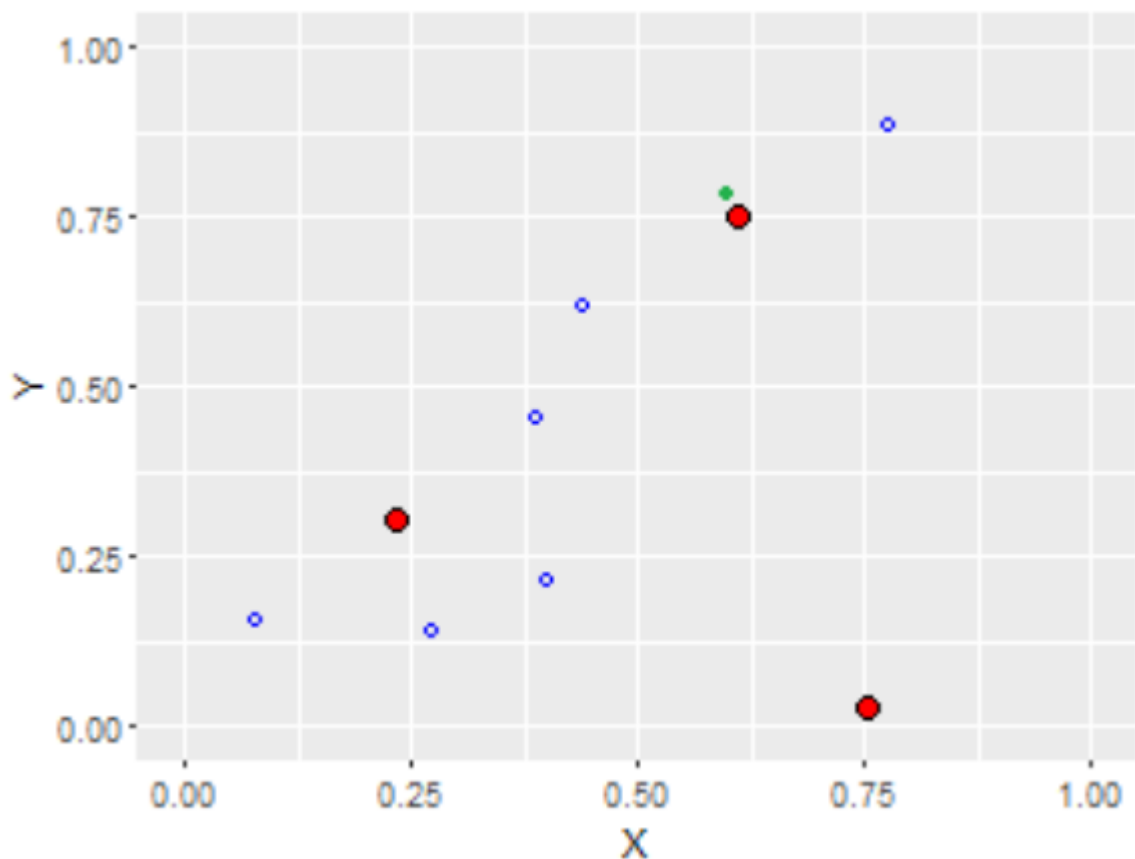


Рисунок 4.16 – Сохранение центров кластеров с приходом нового наблюдения

Для проведения эксперимента использовался двумерный искусственно сгенерированный набор данных. Он содержал 15 кластеров с разным уровнем перекрытия.

Выборка данных содержала 5000 наблюдений. Данные подавались на обработку в последовательном режиме.

Для сравнения качества кластеризации использовались FCM и система, основанная на эволюционирующем методе нечёткой кластеризации (Evolving Fuzzy Clustering Method, EFCM) с разными значениями параметра порога.

В качестве критерия оценки качества кластеризации был использован индекс Кси–Бени.

В табл. 4.6 представлены сравнительные результаты кластеризации.

Таблица 4.6 – Результаты кластеризации

Система	Количество кластеров	Параметры алгоритма	XB	Время кластеризации, с
FCM (Fuzzy C-Means)	15	-	0.0903	2.69
EFCM	9	Dthr = 0.24	0.1136	0.14
EFCM	12	Dthr = 0.19	0.1548	0.19
Предложенная система (пакетный режим)	12	delta = 0.1	0.0978	0.37
Предложенная система (online режим)	12	delta=0.1	0.1127	0.25

На рис. 4.17 представлены промежуточные результаты кластеризации для предложенной системы. На данном этапе было обнаружено 9 нечётких кластеров.

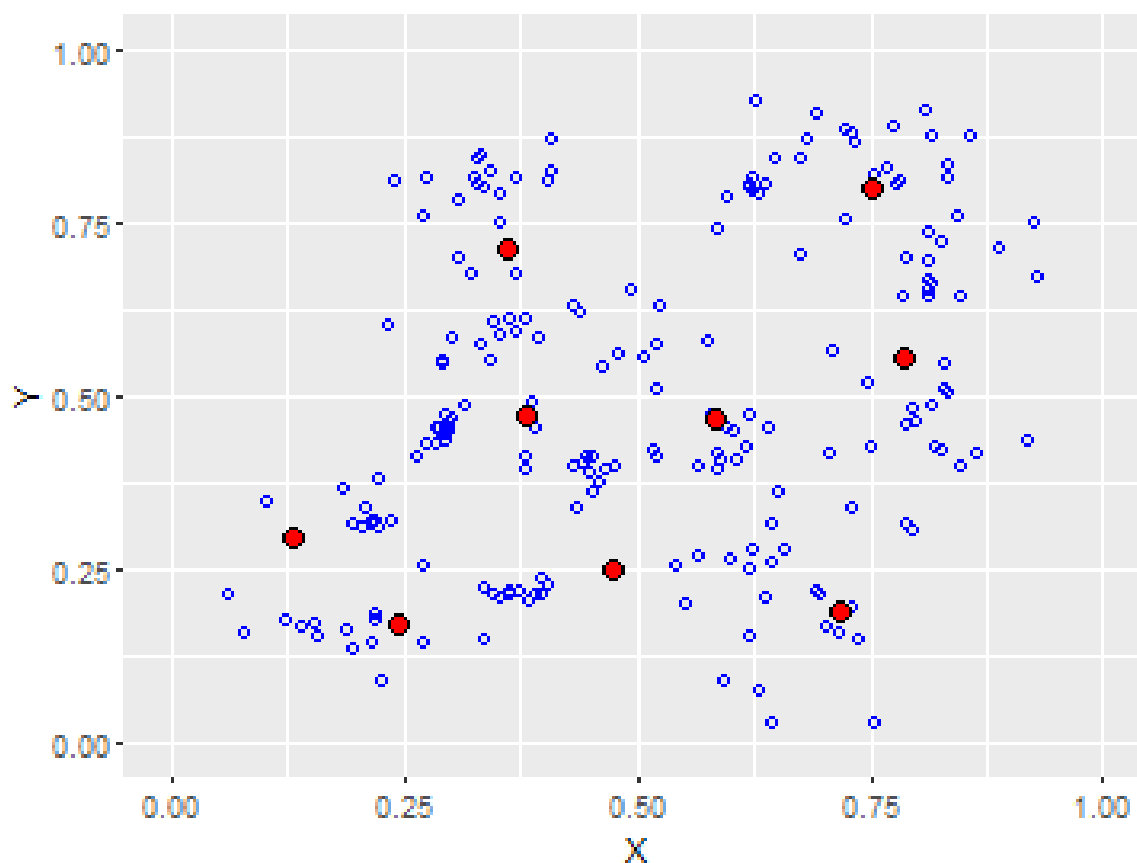


Рисунок 4.17 – Промежуточные результаты кластеризации

Для проведения следующего эксперимента была использована выборка данных, описывающая характеристики зёрен пшеницы.

Выборка данных содержала 210 наблюдений. Каждое наблюдение описывалось семью геометрическими характеристиками зерна пшеницы: площадью, периметром, компактностью, длиной и шириной зерна, коэффициентом асимметрии и длиной борозды зерна.

Перед кластеризацией признаки наблюдений были нормализованы на интервале  $[0,1]$ .

В табл. 4.7 представлены сравнительные результаты кластеризации.



Таблица 4.7 – Результаты кластеризации

Система	Количество кластеров	Параметры алгоритма	ХВ	Время, с
FCM	3	-	0.1963	0.81
EFCM	4	Dthr = 0.6	0.2330	0.04
Предложенная система (пакетный режим)	3	delta = 0.4	0.2078	0.45
Предложенная система (online режим)	3	delta=0.4	0.2200	0.30

Параметры анализируемых систем, а также количество обнаруженных кластеров, также приведены в табл. 4.7.

Для оценки качества работы систем использовался индекс Кси–Бени (ХВ).

Стоит отметить, что предложенная система показала лучший по критерию РС результат по сравнению с EFCM и лучший по времени работы результат по сравнению с FCM.

Как предложенные системы, так и FCM обнаружили три нечётких кластера.

На рис. 4.18 представлено отображения Сэммона для результатов кластеризации.

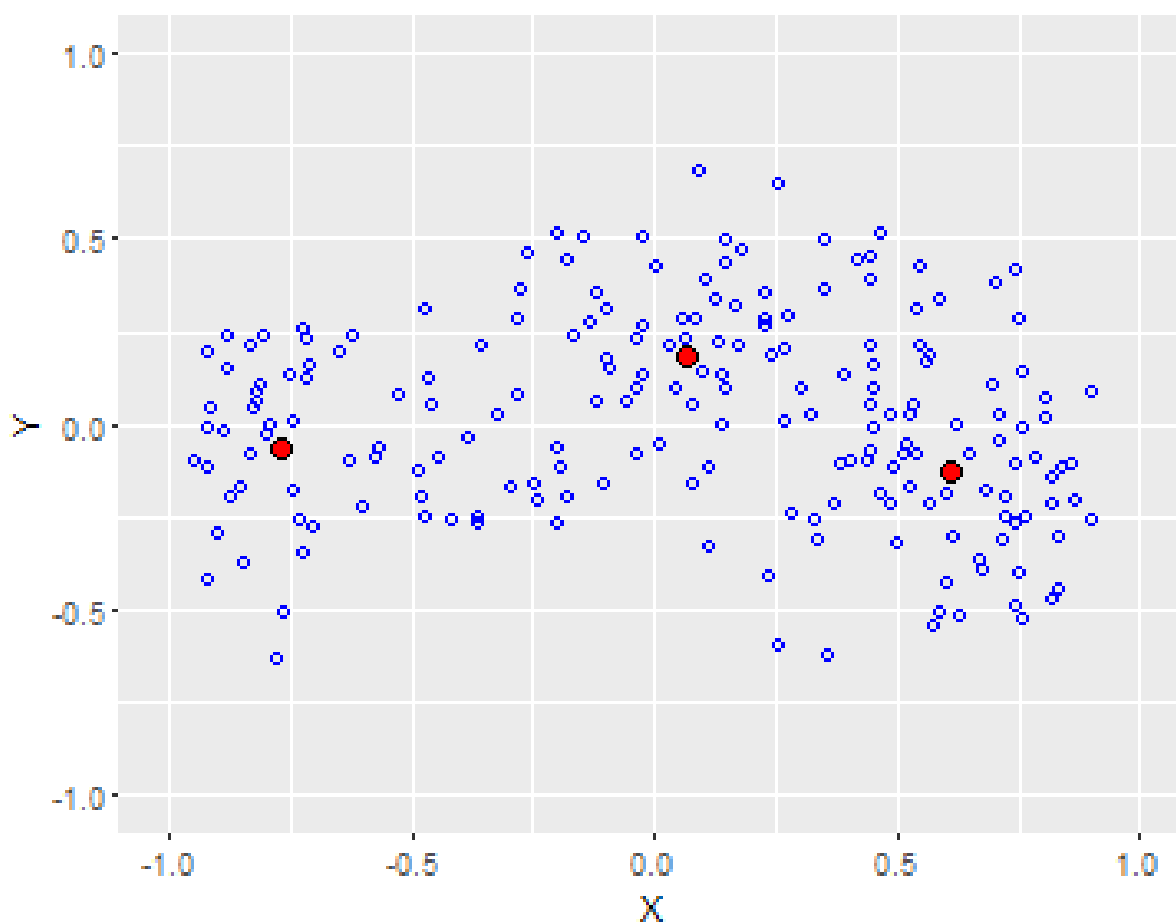


Рисунок 4.18 – Отображение Сэммона для результатов кластеризации

#### 4.6 Решение практической задачи прогнозирования медицинских показателей

Артериальное давление – давление крови в артериях – является одним из основных показателей деятельности сердечно-сосудистой системы. Обычно любой осмотр недомогающего человека сопровождается измерением артериального давления.

Артериальное давление может изменяться при многих заболеваниях, а поддержание его на оптимальном уровне является жизненно важным.

У здоровых людей, как правило, уровень артериального давления относительно устойчив. Но в повседневной жизни он может колебаться. Причиной этому может послужить нервное или физическое перенапряжение, избыточное употребление жидкости, отрицательные эмоции и т. п.

Уровень артериального давления определяет объём крови, выталкиваемый сердцем за минуту, и ширина сосудистого просвета, то есть сопротивление току крови. Артериальное давление (систолическое, диастолическое и пульсовое) измеряется в миллиметрах ртутного столба.

При сокращении желудочков сердца (систоле) кровь выталкивается в крупные артерии под давлением, которое называется систолическим («верхним»). Эту величину определяет сила и частота сокращений сердца и сосудистое сопротивление.

Из желудочков сердца во время систолы выталкивается около 70 мл крови. Такой объём крови не может сразу пройти через мелкие кровеносные сосуды. Поэтому аорта и другие крупные сосуды растягиваются, при этом давление в них повышается. В норме это давление достигает 100 – 130 мм рт. ст.

Давление в артериях в момент сердечного расслабления (диастолы) постепенно падает в норме до 90 мм рт. ст., а в крупных артериях – до 70 мм рт. ст. Это даёт показатель диастолического («нижнего») давления. Это минимальное давление, полностью зависящее от сосудистого сопротивления.

Если из показания систолического артериального давления вычесть показание диастолического артериального давления, то получится пульсовое давление.

На рис. 4.19 представлена волна артериального давления, где максимальное значение соответствует показанию систолического давления, а минимальное значение – показанию диастолического давления.

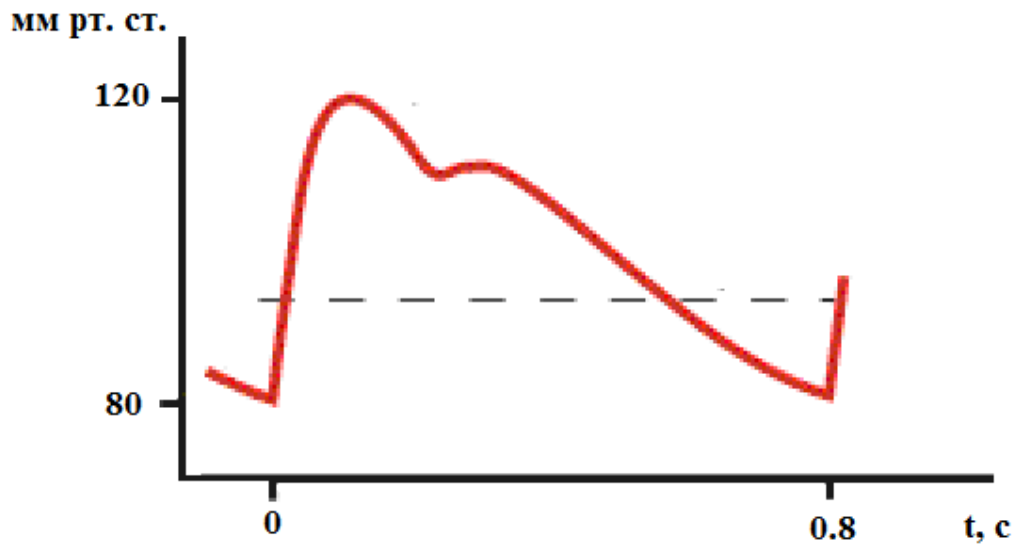


Рисунок 4.19 – Волна артериального давления

Уровень артериального давления колеблется в течение суток. Как правило, наиболее низким он бывает во время сна, а к утру повышается, достигая максимума днём, в часы дневной активности. У здоровых людей разница между самым высоким и самым низким значениями артериального давления в течение суток, как правило не превышает 30 мм рт. ст. для систолического и 10 мм рт. ст. – для диастолического артериального давления.

В целом, артериальное давление зависит от многих факторов: возраста, времени суток, психологического состояния человека (как правило, в стрессовых ситуациях давление повышается), приёма стимулирующих веществ или медикаментов, которые могут повышать или понижать давление. Также величина артериального давления возрастает при непривычных физических нагрузках или эмоциональном напряжении. Очень часто это мешает врачам при обследовании кардиологических больных, которые, как правило, являются людьми впечатлительными. Существует, так называемый, «эффект белого халата»: когда результаты измерения артериального давления в кабинете у врача оказываются на 30–

40 мм рт. ст. выше, чем при самостоятельном измерении его дома. Это объясняется тем, что обстановка медицинского учреждения создаёт стрессовую ситуацию.

В табл. 4.8 представлены нормальные значения артериального давления в зависимости от возраста.

Таблица 4.8 – Нормальные значения артериального давления в зависимости от возраста

Возраст	Артериальное давление, мм рт. ст.	
	Систолическое	Диастолическое
16-20 лет	100-120	70-80
20-40 лет	120-130	70-80
Старше 40 лет	до 140	до 90
Старше 60 лет	до 140	до 90

На рис. 4.20 представлен график суточных колебаний артериального давления.

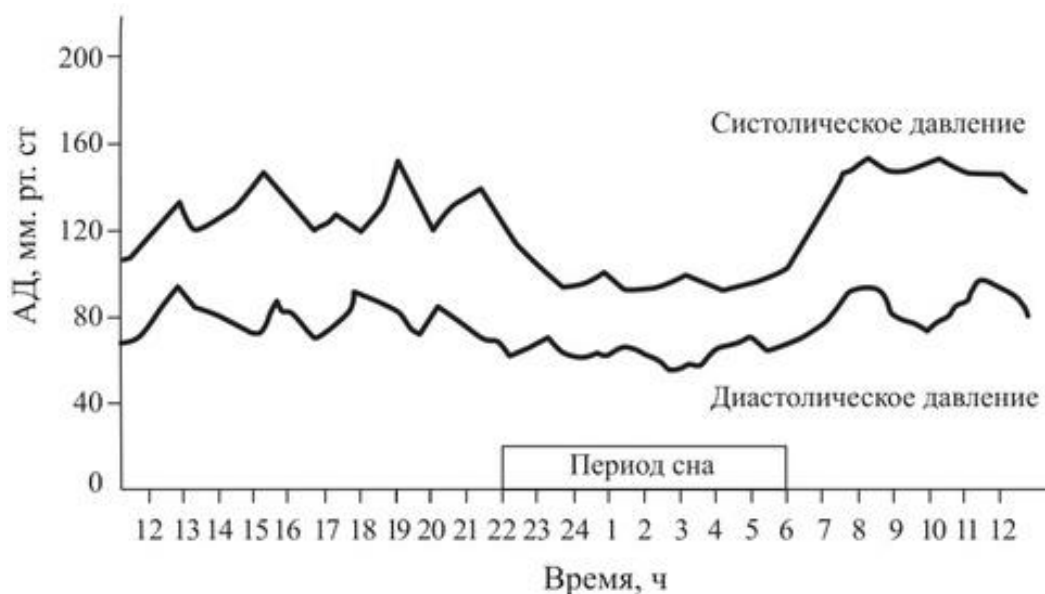


Рисунок 4.20 – График суточных колебаний артериального давления

Кроме рассмотренных выше (систолического, диастолического, пульсового), выделяют также некоторые другие виды давления. В табл. 4.9 представлены виды кровяного давления и их характеристики.

Таблица 4.9 – Виды кровяного давления

Вид давления	Характеристика
Систолическое	Подъем давления до максимума во время систолы
Диастолическое	Снижение давления до минимума во время диастолы
Пульсовое	Амплитуда колебания давления на протяжении сердечного цикла
Среднее динамическое	Давление, усредненное по времени сердечного цикла, т.е. такое давление, которое было бы в сосудистой системе без подъема в систолу и спада в диастолу
Боковое	Сила, с которой кровь действует на стенку сосуда
Конечное	Сумма потенциальной и кинетической энергий, которыми обладает кровь, движущаяся на определенном участке сосудистого русла
Ударное	Разница между конечным и боковым давлениями

Повышение артериального давления наблюдается при гипертонической болезни. Важно отметить, что у больных артериальной гипертонией ночные показатели артериального давления могут оказаться выше дневных. Поэтому для обследования и лечения таких пациентов

важным является проведение суточного мониторинга артериального давления, по результатам которого уточняется время наиболее рационального приёма лекарств и обеспечивается полноценный контроль эффективности лечения.

Артериальное давление, как правило, измеряется в положении сидя. Но иногда эта процедура проводится, когда пациент лежит (например, тяжело больной) или стоит (при функциональных пробах).

Независимо от положения пациента, предплечье его руки, на которой измеряется артериальное давление, и аппарат должны находиться на уровне сердца.

Также артериальное давление часто измеряется с помощью специальных датчиков. Зарегистрированные данные иногда используются для ретроспективного анализа, но чаще всего наиболее ценными и важными они становятся при анализе показателей в реальном масштабе времени [104]. Пожалуй, в этом контексте наиболее существенным примером является использование этих данных в отделениях интенсивной терапии и при дистанционном мониторинге состояния пациентов с различными заболеваниями.

Для решения практической задачи прогнозирования артериального давления была выбрана взвешенная ANARX-система (WANARX-система) с нео-фаззи узлами.

Для сравнения качества прогнозирования были выбраны многослойный персептрон (MLP), радиально-базисная нейронная сеть (RBFN) и ANFIS.

В таблице 4.10 представлены результаты прогнозирования.

Таблица 4.10 – Результаты прогнозирования медицинского ряда

Системы	RMSE (обучение)	RMSE (тестовое)
MLP	0.2011	0.2050
RBFN	0.1890	0.1912
ANFIS	0.1602	0.1610
Предложенная система (взвешенная ANARX-система с нео-фаззи узлами)	0.1578	0.1590

#### 4.7 Решение практической задачи кластеризации земельных участков

Земельный участок – это часть земной поверхности с установленными границами, определённым местом размещения, с определёнными по отношению к ней правами. Земельный участок является основной земельно-кадастровой единицей.

Каждый земельный участок имеет набор определённых свойств, которые вызывают заинтересованность как у владельца, так и у потенциального покупателя. Именно из-за наличия определённых полезных свойств земельного участка возникает потребность определять их в универсальной форме – денежной. Выделяют несколько основных факторов, которые определяют стоимость земельного участка [105]. Приоритет этих факторов зависит от применяемого метода оценки земельного участка. К этим факторам относятся правовые, пространственные, рыночные, количественные, качественные и экономико-технологические.



Пространственные факторы – это факторы местонахождения, т. е. физического нахождения земельного участка на определённой части земной поверхности. Можно выделить два типа описания местонахождения:

- абсолютное местонахождение – это положение земельного участка по отношению к основным физико-географическим и административно-территориальным границам (принадлежность к населённым пунктам, нахождение вне их пределов, в центральной или периферийной части района, на берегу водоёма и т. п.);

- относительное местонахождение – это положение земельного участка по отношению к центрам, которые в значительной степени повышают экономическую оценку территории (например, если участок расположен на окраине города, но рядом с вокзалом).

На рис. 4.21 представлен пример отображения границ земельных участков на карте.



Рисунок 4.21 – Отображение границ земельных участков на карте

Граница земельного участка является его неотъемлемым атрибутом. Граница земельного участка – это условная линия, которая отделяет территорию земельного участка на земной поверхности. Граница чётко фиксируется на местности и в правоустанавливающих документах, технической документации.

После установления границ возникает другой признак земельного участка – место размещения. Координаты установленных границ обеспечивают однозначную определённость места размещения земельного участка.

Для распознавания земельных участков применяют систему кадастровых номеров, которые содержат информацию о географическом местонахождении земельного участка в системе административно-территориального устройства Украины.

Размер земельного участка определяется длиной сторон земельного участка и его площадью.

Площадь земельного участка может быть физической и геодезической. Физическая площадь земельного участка – это площадь земной поверхности в пределах границ земельного участка с учётом неровностей физической поверхности земли: холмов, впадин, склонов и т. п.

В правоустанавливающих документах указывается геодезическая площадь. Геодезической площадью земельного участка называется площадь проекции границ земельного участка на поверхность в системе координат. Геодезическая площадь определяется по координатам углов поворотов границ земельного участка.

Площадь земельного участка измеряется и выражается в разных единицах измерения: квадратных метрах, арах, гектарах. Выбор единицы измерения зависит от разных факторов: от типа использования, установленных требований к измерению, сложившихся на рынке

недвижимости традиций. Например, площадь участков застроенных земель обычно измеряется в квадратных метрах, а площадь участков сельскохозяйственных земель – в гектарах.

Стоит отметить, что единица измерения площади земельного участка часто служит единицей сравнения при анализе цен продаж. Интересно, что с увеличением размера земельного участка цена за единицу его площади уменьшается.

В зависимости от размера земельные участки бывают:

- малыми – площадью до 0,5 га;
- средними – площадью от 0,5 до 5 га;
- большими – площадью более 5 га.

В зависимости от местоположения рассматривают центральную, срединную и окраинную зоны населённых пунктов. За пределами населённых пунктов выделяют пригородные зоны, по отдалённости от крупных населённых пунктов, зоны вдоль транспортных магистралей и природных объектов и т. д.

Земельные участки также характеризуются наличием на них коммуникаций – теплообеспечения, водообеспечения, электрообеспечения, газообеспечения, канализации. Развитая инфраструктура в значительной степени повышает стоимость земельных участков.

Рынок земли имеет ряд отличительных характеристик, что в сравнении с другими рынками делает его особенным. Земельные участки достаточно сложно группировать по характерным признакам из-за уникальности земли и значительного влияния окружения. Поэтому на сегодняшний день задача кластеризации земельных участков является достаточно актуальной.

В качестве фактора кластеризации могут быть определены:

- местоположение объекта оценки относительно районного центра: как правило, чем дальше земельный участок расположен от областного или районного центра, тем его стоимость меньше;

- наличие электричества;
- наличие магистрального газоснабжения;
- наличие водопровода;
- площадь.

Результаты кластеризации земельных участков могут быть использованы в процессе оценки стоимости земельных участков.

Оценка земельных участков производится с использованием таких основных подходов:

- сравнительный;
- доходный;
- затратный.

Для сравнительного подхода к оценке земельных участков характерно то, что методы проведения оценки основываются на результатах анализа цен продажи на подобные земельные участки. Этот подход широко используется в условиях развитого земельного рынка для оценки свободных от застроек земельных участков и для индивидуальной жилой застройки.

В случае применения метода попарного сравнения продаж сравнительного подхода, оценщик анализирует цены продаж свободных от застроек земельных участков, отбирает объекты сравнения продаж на рынке земли и вносит коррекции в их цены на возможные отличия между объектом оценки и объектами сравнения.

Этот метод предусматривает выполнение таких оценочных процедур:

- сбор информации об оцениваемом земельном участке, а также анализ продажи подобных земельных участков и определение рыночных цен объектов сравнения;

- определение отличий оцениваемого земельного участка с объектами сравнения. Затем производится корректировка цен продажи объектов сравнения на отличия;

- согласование полученных результатов расчёта.

Что касается определения отличий земельных участков, то оно, как правило, проводится по таким критериям:

- физические характеристики земельных участков: форма, размер, рельеф, ориентация в плане, подведенные коммуникации, влияние окружающей среды и т. п.;

- условия финансирования договора, давление внешних обстоятельств, льготы и субсидии;

- условия продажи, отклонение от типичных вариантов, платёж эквивалентом денежных средств, рассрочка и т. п.;

- рыночные условия, изменение цен земли во времени;

- изменения в законодательстве, налогообложении;

- местоположение и окружение;

- экономические характеристики, эксплуатационные расходы, условия договора аренды, административные расходы.

Для решения практической задачи кластеризации земельных участков была выбрана эволюционирующая нейро-фаззи система для кластеризации.

Для проведения экспериментов использовались две обучающие выборки. Первая обучающая выборка содержала 230 наблюдений. Вторая выборка данных содержала 500 наблюдений.

Каждое наблюдение имело ряд характеристик, среди которых, в частности, были такие:

- геодезическая площадь земельного участка;

- периметр земельного участка;

- расстояние от областного или районного центра;

- развитость инфраструктуры и наличие коммуникаций;
- тип и качество почвы.

Перед кластеризацией признаки наблюдений были нормализованы на интервале  $[0,1]$ .

В табл. 4.11 приведены результаты кластеризации земельных участков с использованием разных систем для первой выборки обучающих данных.

Таблица 4.11 – Результаты кластеризации земельных участков для первой выборки наблюдений

Система	Количество кластеров	XВ
FCM	4	0.2210
EFCM	6	0.2340
Предложенная система (пакетный режим)	4	0.2015
Предложенная система (online режим)	4	0.2030

По результатам кластеризации можно сделать вывод, что наилучший результат показала предложенная система в пакетном режиме. Стоит отметить, что в online режиме процесс кластеризации потребовал меньше времени по сравнению со всеми остальными системами, рассматривавшимися в ходе эксперимента.

В табл. 4.12 приведены результаты кластеризации земельных участков с использованием разных систем для второй выборки обучающих данных.

Таблица 4.12 – Результаты кластеризации земельных участков для второй выборки наблюдений

Система	Количество кластеров	XВ
FCM	5	0.2215
EFCM	6	0.2330
Предложенная система (пакетный режим)	5	0.2021
Предложенная система (online режим)	5	0.2034

#### Выводы к разделу 4

1. Выполнено имитационное моделирование эволюционирующей многослойной нейро-фаззи системы с гибридным обучением, предложенной в подразделе 2.2. Проведены эксперименты для исследования влияния параметра максимального количества функций принадлежности  $h$  на радиус соседства и качество работы системы.

2. Выполнено имитационное моделирование эволюционирующей системы, основанной на МГУА, с нейро-фаззи узлами, предложенной в подразделе 2.4.

3. Выполнено имитационное моделирование эволюционирующих каскадных систем с нейро-фаззи узлами и с нео-фаззи узлами, предложенных в подразделе 2.5. Проведено сравнение скорости обучения предложенных систем с другими известными системами.

4. Выполнено имитационное моделирование ANARX-модели и взвешенной ANARX-модели (WANARX-модели) с нео-фаззи узлами,

предложенных в подразделе 2.6. Проведено сравнение скорости и качества обучения предложенных систем с другими известными системами.

5. Выполнено имитационное моделирование эволюционирующей адаптивной нечёткой кластеризующей нейронной сети Кохонена, предложенной в подразделе 3.3, в пакетном и в online режиме.

6. Решена практическая задача прогнозирования медицинских показателей с помощью предложенной в пункте 2.6.3 взвешенной ANARX-системы с нео-фаззи узлами.

7. Решена практическая задача кластеризации земельных участков с помощью предложенной в подразделе 3.3 эволюционирующей нейро-фаззи кластеризующей системы.



## ВЫВОДЫ

В диссертационной работе представлены результаты, соответствующие цели исследований, а именно – разработке эволюционирующих нейро-фаззи систем и методов их обучения, которые позволяют в online режиме настраивать не только синаптические веса и параметры функций принадлежности, но и архитектуру системы в целом.

Проведенные исследования позволили сделать такие выводы:

1. Выполнен обзор и анализ известных архитектур нейро-фаззи систем и методов их обучения, а также принципов построения эволюционирующих систем.

2. Предложены архитектура и методы обучения многослойной нейро-фаззи системы, основанной на системе Ванга–Менделя и сочетающей в себе процессы эволюции архитектуры, самообучения функций принадлежности и контролируемого обучения синаптических весов.

3. Разработаны методы обучения для настройки всех параметров нейро-фаззи узлов для улучшения аппроксимирующих свойств эволюционирующих нейро-фаззи систем.

4. Предложены методы обучения эволюционирующих систем на основе метода группового учёта аргументов (МГУА) и каскадных систем с использованием в качестве узлов двухвходовых нейро-фаззи систем Ванга–Менделя и двухвходовых нео-фаззи узлов, что позволяет обрабатывать данные в условиях коротких обучающих выборок.

5. Разработаны архитектура и методы обучения взвешенной ANARX-модели (WANARX-модели) для прогнозирования нестационарных нелинейных временных рядов.

6. Усовершенствована архитектура и методы самообучения эволюционирующей нейро-фаззи системы для кластеризации данных, поступающих на обработку в последовательном online режиме, в условиях неопределённости относительно количества кластеров. Эти методы

обучения основаны на возможностном подходе к нечёткой кластеризации, который решает проблемы, возникающие из-за недостатков вероятностного подхода к нечёткой кластеризации.

7. Проведено имитационное моделирование разработанных методов для решения различных задач интеллектуального анализа данных и выполнено сравнение качества их работы с известными системами.

8. Решена практическая задача прогнозирования медицинских показателей с помощью предложенной взвешенной ANARX-системы с не-фаззи узлами.

9. Решена практическая задача кластеризации земельных участков с помощью предложенной эволюционирующей нейро-фаззи кластеризующей системы.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Jang J.-S. Neuro-Fuzzy And Soft Computing: A Computational Approach To Learning And Machine Intelligence / J.-S. Jang, C.-T. Sun, E. Mizutani. – Upper Saddle River : Prentice Hall, 1997. – 640 p.
2. Rutkowski L. Computational Intelligence. Methods and Techniques. – Berlin-Heidelberg: Springer-Verlag, 2008. – 514 p.
3. Kruse R. Computational Intelligence / R. Kruse, C. Borgelt, F. Klawonn F. – Berlin: Springer, 2013. – 488 p.
4. Du K.-L. Neural Networks and Statistical Learning / K.-L. Du, M. N. S. Swamy. – London: Springer-Verlag, 2014. – 824 p.
5. Takagi T. Fuzzy identification of systems and its applications to modeling and control / T. Takagi, M. Sugeno // IEEE Trans. on Systems, Man, and Cybernetics. – 1985. – Vol. 15. – P. 116-132.
6. Haykin S. Neural Networks: A Comprehensive Foundation. – Upper Saddle River, N.J. : Prentice Hall, Inc., 1999. – 842 p.
7. Ljung L. System Identification: Theory for the User. – Upper Saddle River : Prentice Hall, Inc., 1987. – 432 p.
8. Nelles O. Nonlinear System Identification. – Berlin : Springer, 2001. – 785 p.
9. Kasabov N. Evolving fuzzy neural networks – algorithms, applications and biological motivation // Eds. T. Yamakawa, G. Matsumoto “Methodologies for the Conception, Design and Application of Soft Computing”. – Singapore : World Scientific, 1998. – P. 271-274.
10. Kasabov N. Evolving fuzzy neural networks: theory and applications for on-line adaptive prediction, decision making and control / N. Kasabov // Australian J. of Intelligent Information Processing Systems. – 1998. – 5 (3). – P. 154-160.

11. Kasabov N. Evolving fuzzy neural network for on-line supervised/unsupervised, knowledge-based learning / N. Kasabov // *IEEE Trans. on Man, Machine and Cybernetics*. – 2001. – 31. – №6. – P. 902-918.
12. Kasabov N. *Evolving Connectionist Systems*. – London: Springer-Verlag, 2003. – 307 p.
13. Lughofer E. *Evolving Fuzzy Systems – Methodologies, Advanced Concepts and Applications*. – Berlin : Springer, 2011. – 410 p.
14. Bodyanskiy Ye. V. Adaptive method of hybrid learning for an evolving neuro-fuzzy system / Ye. V. Bodyanskiy, O. O. Boiko, I. P. Pliss // *Cybernetics and Systems Analysis*. – 2015. – V. 51. – 4. – P. 500-505.
15. Бодяньський Є. В. Еволюційна каскадна система на основі нейрофаззи вузлів / Є. В. Бодяньський, О. К. Тищенко, О. О. Бойко // *Радіоелектроніка, інформатика, управління*. – Запоріжжя: ЗНТУ. – 2016. – №2(37). – С. 40-45.
16. Bodyanskiy Ye. V. An ensemble of adaptive neuro-fuzzy Kohonen networks for online data stream fuzzy clustering / Ye. V. Bodyanskiy, O. K. Tyshchenko, O. O. Boiko // *International Journal of Modern Education and Computer Science (IJMECS)*. – 2016. – Vol. 8, No. 5. – P. 12-18.
17. Bodyanskiy Ye. V. An evolving cascade system based on a set of neuro-fuzzy nodes / Zhengbing Hu, Ye. V. Bodyanskiy, O. K. Tyshchenko, O. O. Boiko // *International Journal of Intelligent Systems and Applications (IJISA)*. – 2016. – Vol. 8, No. 9. – P. 1-7.
18. Bodyanskiy Ye. V. Adaptive forecasting of non-stationary nonlinear time series based on the evolving weighted neuro-neo-fuzzy-ANARX-model // Zhengbing Hu, Ye. V. Bodyanskiy, O. K. Tyshchenko, O. O. Boiko // *International Journal of Information Technology and Computer Science (IJITCS)*. – 2016. – Vol. 8, No. 10. – P. 1-10.
19. Bodyanskiy Ye. Evolving multilayer neuro-fuzzy system and its learning / Ye. V. Bodyanskiy, O. Boiko // *Системні технології. Регіональний міжвузівський збірник наукових праць*. – 2015. – №5(100). – С. 161-169.

20. Бодянский Е. В. Многошаговая модификация правила самообучения Д. Хебба – Э. Оя в задачах анализа главных компонент / Е. В. Бодянский, Е. А. Бойко // Интеллектуальные системы принятия решений и проблемы вычислительного интеллекта : материалы международной научной конференции, 28 – 31 мая 2014 р. – Херсон: ХНТУ, 2014. – С. 247-249.

21. Бойко О. О. Багатокрокове адаптивне самонавчання за Д. Хеббом – Е. Оя в аналізі головних компонентів / О. О. Бойко, М. О. Шаламов, Є. В. Бодяньський // Праці VII міжнародної школи-семінару «Теорія прийняття рішень», 29 вересня – 4 жовтня 2014 р. – Ужгород, УжНУ, 2014. – С. 51-52.

22. Бойко Е. А. Эволюционирующая каскадная нейро-фаззи сеть с настройкой центров функций принадлежности // Праці наукової конференції фізико-технічного факультету Донецького національного університету, за період 2014-2015 учбового року : збірник наукових праць, 21–28 квітня 2015 року. – Вінниця, ДонНУ, 2015. – С. 69-70.

23. Бойко Е. А. Многослойная адаптивная нейро-фаззи система и ее обучение в условиях малой выборки / Е. А. Бойко // 19-й Международный молодежный форум «Радиоэлектроника и молодежь в XXI веке» : сб. материалов форума, 20-22 апреля 2015 г. – Т. 6. – Харьков: ХНУРЭ, 2015. – С. 10-11.

24. Бодяньський Є. В. Еволюційна ANARX-модель на основі нео-фаззі вузлів для прогнозування нестационарних нелінійних послідовностей / Є. В. Бодяньський, О. О. Бойко // Праці VIII міжнародної школи-семінару «Теорія прийняття рішень», 26 вересня – 1 жовтня 2016 р.). – Ужгород, УжНУ, 2016. – С. 50-51.

25. Перова І. Г. Діагностуюча нейро-фаззі система зі змінною кількістю входів та виходів / І. Г. Перова, Є. М. Бражникова, О. О. Бойко. // Праці VIII міжнародної школи-семінару «Теорія прийняття рішень», 26 вересня – 1 жовтня 2016 р. – Ужгород, УжНУ, 2016. – С. 209-210.

26. Бодянский Е. В. Искусственные нейронные сети: архитектуры, обучение, применения / Е. В. Бодянский, О. Г. Руденко. – Харьков : ТЕЛЕТЕХ, 2004. – 372 с.
27. Калан Р. Основные концепции нейронных сетей. – М. : Издательский дом «Вильямс», 2001. – 288 с.
28. Kohonen T. Self-Organizing Maps. – Berlin: Springer-Verlag, 1995. – 362 p.
29. Zadeh L. Fuzzy sets / L. Zadeh // Information and Control. – 1965. – 8. – P. 338-353.
30. Zadeh L. Fuzzy logic – a personal perspective / L. Zadeh // Fuzzy Sets and Systems. – 2015. – 281. – P. 4-20.
31. Осовский С. Нейронные сети для обработки информации. – М. : Финансы и статистика, 2002. – 344 с.
32. Cun Y. L. Optimal brain damage / Y. L. Cun, J. S. Denker, S. A. Solla // Advances in Neural Information Processing Systems. – San Mateo, CA : Morgan Kaufman, 1990. – Vol. 2. – P. 598-605.
33. Chauvin Y. A back propagation algorithm with optimal use of hidden units / Y. Chauvin // Advances in Neural Information Processing Systems. – San Mateo, CA : Morgan Kaufman, 1989. – P. 598-605.
34. Fahlman S. E. The cascade-correlation learning architecture / S. E. Fahlman, C. Lebiere // Advances in Neural Information Processing Systems. – 1990. – P. 524-532.
35. Prechelt L. Investigation of the CasCor family of learning algorithms / L. Prechelt // Neural Networks. – 1997. – 10. – P. 885-896.
36. Kasabov N. ECM – a novel on-line, evolving clustering method and its applications / N. Kasabov, Q. Song // Foundations of cognitive science. – 2001. – P. 631-682.
37. Watts M. J. A decade of Kasabov's evolving connectionist systems: a review / M. J. Watts // IEEE Transactions on Systems, Man and Cybernetics. Part C – Applications and Reviews. – 2009. – Vol. 39(3). – P. 253-269.

38. Bodyanskiy Ye. Flexible neo-fuzzy neuron and neuro-fuzzy network for monitoring time series properties / Ye. Bodyanskiy, I. Pliss, O. Vynokurova // *Information Technology and Management Science*. – 2013. – 16. – P. 47-52.
39. Bodyanskiy Ye. Flexible wavelet-neuro-fuzzy neuron in dynamic data mining tasks / Ye. Bodyanskiy, I. Pliss, O. Vynokurova // *Oil and Gas Power Engineering*. – 2013. – 2(20). – P. 158-162.
40. Nelles O. *Nonlinear System Identification*. – Berlin: Springer, 2001. – 785 p.
41. Wang L.-X. Fuzzy basis functions, universal approximation and orthogonal least squares learning / L.-X. Wang, J. M. Mendel // *IEEE Trans. on Neural Networks*. – 1993. – Vol. 3. – P. 807-814.
42. Bodyanskiy Ye. An adaptive learning algorithm for a neuro-fuzzy network // Ye. Bodyanskiy, V. Kolodyazhniy, A. Stephan // *Computational Intelligence. Theory and Applications*. – Berlin-Heidelberg-New York : Springer, 2001. – P. 68-75.
43. Otto P. A new learning algorithm for a forecasting neuro-fuzzy network / P. Otto, Ye. Bodyanskiy, V. Kolodyazhniy // *Integrated Computer-Aided Engineering*. – 2003. – 10. – №4. – P. 399-409.
44. Ivakhnenko A. G. Heuristic self-organization in problems of engineering cybernetics / A. G. Ivakhnenko // *Automatica*. – 1970. – 6. – №2. – P. 207-219.
45. Ivakhnenko A. G. Polynomial theory of complex systems / A. G. Ivakhnenko // *IEEE Trans. on Systems, Man, and Cybernetics*. – 1971. – 1. – №4. – P. 364-378.
46. Ивахненко А. Г. Самообучающиеся системы распознавания и автоматического управления. – Киев: Техніка, 1969. – 392 с.
47. Ивахненко А. Г. Долгосрочное прогнозирование и управление сложными системами. – Киев: Техніка, 1975. – 311 с.
48. Ивахненко А. Г. Помехоустойчивость моделирования / А. Г. Ивахненко, В. С. Степашко. – Киев: Наукова думка, 1985. – 216 с.

49.Ivakhnenko A. G. Self-organization of the neural networks with active neurons / A. G. Ivakhnenko, G. A. Ivakhnenko, J. A. Mueller // Pattern Recognition and Image Analysis. – 1994. – 4. – №2. – P. 177-188.

50.Ivakhnenko G. A. Self-organization of neuronet with active neurons for effects of nuclear test explosions forecasting / G. A. Ivakhnenko // System Analysis Modeling Simulation. – 1995. – 20. – P. 107-116.

51.Ivakhnenko A. G. Inductive sorting-out GMDH algorithms with polynomial complexity for active neurons of neural networks / A. G. Ivakhnenko, D. Wuensch, G. A. Ivakhnenko // Neural Networks. – 1999. – 2. – P. 1169-1173.

52.Zaychenko Yu. The fuzzy Group Method of Data Handling and its application for economical processes forecasting / Yu. Zaychenko // Scientific Inquiry. – 2006. – 7. – №1. – P. 83-96.

53.Зайченко Ю. П. Нечеткие модели и методы в интеллектуальных системах. – К. : Издательский дом «Слово», 2008. – 344 с.

54.Bodyanskiy Ye. The neo-fuzzy neural network structure optimization using the GMDH for the solving forecasting and classification problems / Ye. Bodyanskiy, Yu. Zaychenko, E. Pavlikovskaya, M. Samarina, Ye. Viktorov // Proc. Int. Workshop on Inductive Modeling. – Krynica, Poland, 2009. – P. 77-89.

55.Kondo T. Identification of radial basis function networks by using revised GMDH-type neural networks with a feedback loop / T. Kondo // Proc. of the SICE Annual Conference. – Tokyo, Japan, 2002. – P. 2882-2887.

56.Ohtani T. Automatic variable selection in RBF network and its application to neurofuzzy GMDH / T. Ohtani // Proc. Fourth Int. Conf. on Knowledge-Based Intelligent Engineering Systems and Allied Technologies. – 2000. – V.2. – P. 840-843.

57.Bodyanskiy Ye. Hybrid evolving neural network using kernel activation functions / Ye. Bodyanskiy, N. Teslenko, P. Grimm // Proc. 17th Zittau East-West Fuzzy Colloquium. – Zittau/Goerlitz: HS, 2010. – P. 39-46.



58. Bodyanskiy Ye. Hybrid GMDH-neural network of computational intelligence / Ye. Bodyanskiy, O. Vynokurova, I. Pliss // Proc. 3rd Int. Workshop on Inductive Modeling. – Krynica, Poland, 2009. – P. 100-107.

59. Ohtani T. Structural learning of neurofuzzy GMDH with Minkowski norm / T. Ohtani, H. Ichihashi, T. Miyoushi // Proc. 1998 Second Int. Conf. on Knowledge-Based Intelligent Electronic Systems. – 1998. – V. 2. – P. 100-107.

60. Бодянський Є. В. Адаптивний вейвлон як вузол штучних МГУА-нейронних мереж / Є. В. Бодянський, О. А. Винокурова // Моделювання та керування станом еколого-економічних систем регіонів. – Київ: МННLS ІТС, 2008. – 4. – С. 19-29.

61. Bodyanskiy Ye. Cascade GMDH-wavelet-neuro-fuzzy network / Ye. Bodyanskiy, O. Vynokurova, N. Teslenko // Proc. 4th Int. Workshop on Inductive Modeling «IWIM 2011». – Kyiv, Ukraine, 2011. – P. 22-30.

62. Bodyanskiy Ye. Wavelet-neuro-fuzzy network structure optimization using GMDH for the solving forecasting tasks / Ye. Bodyanskiy, O. Vynokurova, A. Dolotov, O. Kharchenko // Proc. 4th Int. Conf. on Inductive Modelling ICIM 2013. – Kyiv, 2013. – P. 61-67.

63. Bodyanskiy Ye. V. Self-learning cascade spiking neural network for fuzzy clustering based on Group Method of Data Handling / Ye. V. Bodyanskiy, O. A. Vynokurova, A. I. Dolotov // J. of Automation and Information Sciences. – 2013. – 45. – №3. – P. 23-33.

64. Wang L.-X. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning / L.-X. Wang, J. M. Mendel // IEEE Trans. on Neural Networks. – 1992. – 3. – №5. – P. 807-814.

65. Wang L.-X. Adaptive Fuzzy Systems and Control. Design and Statistical Analysis. – Upper Saddle River : Prentice Hall, 1994. – 256 p.

66. Bodyanskiy Ye. The cascade orthogonal neural network / Ye. Bodyanskiy, A. Dolotov, I. Pliss, Ye. Viktorov // Information Science & Computing. – Sofia : FOI ITHEA, 2008. – №2. – P. 13-20.

67. Bodyanskiy Ye. Hybrid cascade neural network based on wavelet-neuron. / Ye. Bodyanskiy, O. Kharchenko, O. Vynokurova // *Int. J. Information Theories and Applications*. – 2011. – 18. – №4. – P. 335-343.

68. Bodyanskiy Ye. Evolving cascade neural network based on multidimensional Epanechnikov's kernels and its learning algorithm. / Ye. Bodyanskiy, P. Grimm, N. Teslenko // *Int. J. Information Technologies and Knowledge*. – 2011. – 5. – №1. – P. 25-30.

69. Bifet A. *Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams*. – IOS Press, 2010. – 224 p.

70. Belikov J. A novel Taylor series based approach for control computation in NN-ANARX structure based control of nonlinear systems / J. Belikov, K. Vassiljeva, E. Petlenkov, S. Nõmm // *Proc. 27th Chinese Control Conference*. – Kunming, China, 2008. – P. 474-478.

71. Vassiljeva K. State-space control of nonlinear systems identified by ANARX and neural network based SANARX models / K. Vassiljeva, E. Petlenkov, J. Belikov // *Proc. WCCI 2010 IEEE World Congress on Computational Intelligence*. – Barcelona, Spain, 2010. – P. 3816-3823.

72. Cybenko G. Approximation by superpositions of a sigmoidal function / G. Cybenko // *Math. Control Signals Systems*. – 1989. – 2. – P. 303-314.

73. Yamakawa T. A neo fuzzy neuron and its applications to system identification and prediction of the system behaviour / T. Yamakawa, E. Uchino, T. Miki, H. Kusanagi // *Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks "IIZUKA-92"*. – Iizuka, Japan, 1992. – P. 477-483.

74. Uchino E. Soft computing based signal prediction, restoration, and filtering / E. Uchino, T. Yamakawa // *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*. – Boston : Kluwer Academic Publishers, 1997. – P. 331-349.

75. Miki T. Analog implementation of neo-fuzzy neuron and its on-board learning / T. Miki, T. Yamakawa // *Computational Intelligence and Applications*. – Piraeus : WSES Press, 1999. – P. 144-149.

76. Takagi T. Fuzzy identification of systems and its applications to modeling and control / T. Takagi, M. Sugeno // IEEE Trans. on Systems, Man, and Cybernetics. – 1985. – 15. – P. 116-132.

77. Sugeno M. Structure identification of fuzzy model / M. Sugeno, G. T. Kang // Fuzzy Sets and Systems. – 1998. – 28. – P. 15-33.

78. Ljung L. System Identification: Theory for the User. – Upper Saddle River: Prentice Hall, Inc., 1987. – 432 p.

79. Polikar R. Learning in nonstationary and evolving environments / R. Polikar, C. Alippi // IEEE Trans. on Neural Networks and Learning Systems. – 2014. – 25. – №1. – P. 9-11.

80. Jin Y. Computational Intelligence in Big Data / Y. Jin, B. Hammer // IEEE Computational Intelligence Magazine. – 2014. – P. 12-13.

81. Friedman J. The Elements of Statistical Learning. Data Mining, Inference and Prediction / J. Friedman, T. Hastie, R. Tibshirani. – Berlin, Springer, 2003. – 552 p.

82. Kolodyazhniy V. Cascaded multiresolution spline-based fuzzy neural network / V. Kolodyazhniy, Ye. Bodyanskiy // Proc. Int. Symp. on Evolving Intelligent Systems. – Leicester, UK: De Montfort University, 2010. – P. 26-29.

83. Bodyanskiy Ye. An adaptive learning algorithm for a neo-fuzzy neuron / Ye. Bodyanskiy, I. Kokshenev, V. Kolodyazhniy // Proc. 3-rd Int. Conf. of European Union Soc. for Fuzzy Logic and Technology (EUSFLAT'03), Zittau, Germany, 2003. – P. 375-379.

84. Bodyanskiy Ye. The cascade neo-fuzzy architecture using cubic-spline activation functions / Ye. Bodyanskiy, Ye. Viktorov // Int. J. Information Theories & Applications. – 2009. – 16. – №3. – P. 245-259.

85. Bodyanskiy Ye. V. A multidimensional cascade neuro-fuzzy system with neuron pool optimization in each cascade / Ye. V. Bodyanskiy, O. K. Tyshchenko, D. S. Kopalani // Int. J. Information Technology and Computer Science. – 2014. – 6. – №8. – P. 11-17.

86. Sharkey A. J. C. On combining artificial neural nets / A. J. C. Sharkey // *Connection Science*. – 1996. – 8. – P. 299-313.
87. Bodyanskiy Ye. An optimal algorithm for combining multivariate forecasts in hybrid systems / Ye. Bodyanskiy, P. Otto, I. Pliss, S. Popov // *LNAI 2774*. – Berlin-Heidelberg-New York: Springer, 2003. – P. 967-973.
88. Bodyanskiy Ye. Adaptive generalization of neuro-fuzzy systems ensemble / Ye. Bodyanskiy, A. Deineko, M. Stolnikova // *Proc. of the Inter. Conf. “Computer Science and Information Technologies”*, November 16-19, 2011. – Lviv, 2011. – P. 13-14.
89. Hoepfner, F. *Fuzzy Clustering Analysis: Methods for Classification, Data Analysis and Image Recognition* / F. Hoepfner, F. Klawonn, R. Kruse, T. Runkler. – Chichester : John Wiley & Sons, 1999 – 289 p.
90. Колчигин Б. В. Адаптивная нечёткая кластеризация с переменным фаззификатором / Б. В. Колчигин, Е. В. Бодянский // *Кибернетика и системный анализ*. – 2013. – №3(49). – С.47-55.
91. Topchy A. Adaptive clustering ensembles / A. Topchy, B. Minaei-Bidgali, A. K. Jain, W.F. Punch // *Proc. 17th Int. Conf. on Pattern Recognition “ICPR 2004”*, Aug. 2004. – P. 272-275.
92. Vega-Pons S. A survey of clustering ensemble algorithms / S. Vega-Pons, J. Ruiz-Shulcloper // *Int. J. Pattern Recognition and Artificial Intelligence*. – Vol. 25, № 337, 2011. – P. 337-372.
93. Bodyanskiy Ye. Adaptive neuro-fuzzy Kohonen network with variable fuzzifier / Ye. Bodyanskiy, B. Kolchygin, I. Pliss // *Int. J. Information Theories & Applications*. – 2011. – 18. – №3. – P. 215-223.
94. Xu R. *Clustering* / R. Xu, D. C. Wunsch. // *IEEE Press Series on Computational Intelligence*. – Hoboken, NJ: John Wiley & Sons, Inc. – 2009. – 370 p.
95. Ravi V. On-line evolving fuzzy clustering / V. Ravi, E. R. Srinivas, N. K. Kasabov // *Proc. Int. Conf. on Computational Intelligence and Multimedia Applications*. – 2007. – 1. – P. 347-351.

96. Bezdek J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. – N. Y.: Plenum Press, 1981. – 256 p.

97. Krishnapuram R. A possibilistic approach to clustering / R. Krishnapuram, J. M. Keller // Fuzzy Systems. – 1993. – 1. – №2. – P. 98-110.

98. Bodyanskiy Ye. An evolving radial basis neural network with adaptive learning of its parameters and architecture / Ye. Bodyanskiy, O. Tyshchenko, A. Deineko // Automatic Control and Computer Sciences. – 2015. – 49. – №5. – P. 255-260.

99. Bodyanskiy Ye. Recursive fuzzy clustering algorithms / Ye. Bodyanskiy, V. Kolodyazhniy, A. Stephan // Proc. 10th East–West Fuzzy Colloquium. – Zittau/Görlitz : HS, 2002. – P. 276-283.

100. Bodyanskiy Ye. Computational intelligence techniques for data analysis / Ye. Bodyanskiy // Lecture Notes in Informatics. – Vol. P-72. – Bonn : GI, 2005. – P. 15-36.

101. Gorshkov Ye. New recursive learning algorithms for fuzzy Kohonen clustering network / Ye. Gorshkov, V. Kolodyazhniy, Ye. Bodyanskiy // Proc. 17th Int. Workshop on Nonlinear Dynamics of Electronic Systems. – Rapperswil, Switzerland, June 21-24, 2009. – P. 58-61.

102. Xie X. L. A validity measure for fuzzy clustering / X. L. Xie, G. Beni // IEEE Trans. on Pattern Analysis and Machine Intelligence. – 1991. – 13. – №8. – P. 841-847.

103. Narendra K. S. Identification and control of dynamical systems using neural networks / K. S Narendra, K. Parthasarathy // IEEE Trans. on Neural Networks. – 1990. – 1. – P. 4-26.

104. Aggarwal C. C. Healthcare Data Analytics / C. C. Aggarwal, C. K. Reddy. – Boca Raton, FL : Chapman & Hall/CRC Press, 2015. – 724 pp.

105. Кірічек Ю. О. Оцінка земель. – Дніпропетровськ : Літограф, 2016. – 454 с.

ПРИЛОЖЕНИЕ А  
АКТЫ О ВНЕДРЕНИИ

Затверджую  
Перший проректор  
Харківського національного  
університету радіоелектроніки  
І. І. Ключник




2016 р.

### А К Т

про впровадження в навчальний процес результатів дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук «Еволюційні нейро-фаззі системи в задачах інтелектуального аналізу даних» аспірантки кафедри штучного інтелекту Харківського національного університету радіоелектроніки Бойко Олени Олександрівни

Комісія у складі декана факультету комп'ютерних наук, д. т. н., проф. Єрохіна А. Л., завідувача кафедри штучного інтелекту, д. т. н., проф. Філатова В. О., проф. каф. штучного інтелекту, к. т. н., доц. Рябової Н. В. підтверджує, що результати дисертаційної роботи Бойко О. О., що пов'язані із розробкою еволюційних нейро-фаззі систем для інтелектуального аналізу даних, впроваджені в навчальний процес на кафедрі штучного інтелекту в курсах «Штучні нейронні мережі: архітектури, навчання та застосування» та «Нейромережеві методи обчислювального інтелекту».


Декан факультету КН, д.т.н., проф.

  
\_\_\_\_\_ А. Л. Єрохін

Зав. каф. ШІ, д.т.н., проф.

  
\_\_\_\_\_ В. О. Філатов

Проф. каф. ШІ, к.т.н., доц.

  
\_\_\_\_\_ Н. В. Рябова

ЗАТВЕРДЖУЮ  
Директор ТОВ «Інфобуд»

А.І. Костіна

« 03 »

жовтня 2016р.



Акт про впровадження результатів дисертаційної роботи на здобуття  
наукового ступеня кандидата технічних наук  
БОЙКО ОЛЕНИ ОЛЕКСАНДРІВНИ

Комісія у складі:

Голова

Гаєвський А.О. – заступник директора з  
розробки програмного забезпечення,

Члени комісії

Кравченко С.Ю. – керівник проектів  
Ковкін О.В. – керівник проектів,

склала даний акт про те, що у ТОВ «Інфобуд» була застосована еволюційна зважена ANARX-система з нео-фаззі вузлами. Запропонована Бойко О.О. система і метод її навчання дозволили підвищити ефективність прогнозування медичних показників, зокрема артеріального тиску.

Результати впровадження довели, що розроблені Бойко О.О. методи, які ґрунтуються на сучасних інтелектуальних технологіях, мають переваги над існуючими підходами.

Акт складений для пред'явлення до спеціалізованої вченої ради із захисту дисертацій і не є підставою для фінансових розрахунків.

Голова комісії:

Гаєвський

Гаєвський А.О.

Члени комісії:

Кравченко

Кравченко С. Ю.

Ковкін

Ковкін О.В.



ЗАТВЕРДЖУЮ

Директор ПП «НВП «Амальгама-Строй»»

Ф. Ф. Сафонов

« 06 »

травня 2016р.



### Акт

про впровадження результатів дисертаційної роботи на здобуття наукового ступеня кандидата технічних наук

**БОЙКО ОЛЕНИ ОЛЕКСАНДРІВНИ**

Комісія у складі

Голова:

Сафонов Ф. Ф. – директор,

Члени комісії:

Дриголя Д. О. – інженер-землевпорядник,

Соловйова О. О. – інженер-землевпорядник,

склала даний акт про те, що у ПП «НВП «Амальгама-Строй»» було застосовано методи еволюційної нечіткої кластеризації. Запропонована Бойко О.О. еволюційна нейро-фаззі система та метод її навчання дозволили ефективно вирішити завдання кластеризації земельних ділянок за умов невизначеності щодо кількості кластерів.

Результати впровадження довели, що розроблені Бойко О.О. засоби ґрунтуються на сучасних інтелектуальних технологіях, виконані на високому науково-технічному рівні, мають переваги над існуючими рішеннями.

Акт складений для пред'явлення до спеціалізованої вченої ради із захисту дисертацій і не є підставою для фінансових розрахунків.

Голова комісії:

Сафонов Ф. Ф.

Члени комісії: •

Дриголя Д. О.

Соловйова О. О.