

Министерство науки и высшего образования Российской Федерации

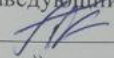
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Кафедра «Прикладная математика»

Направление 01.04.02 – «Прикладная математика и информатика»

Профиль – «Математика и информатика в науке и образовании»

К ЗАЩИТЕ ДОПУСКАЮ
заведующий кафедрой


 А.Л. Григорьева

«__» _____ 2024 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

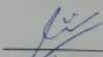
Разработка метода прогнозирования временных рядов
на основе нейрокомпьютерных технологий

Студент группы 2МИМ-1

 13.06.24
подпись, дата


А.Л. Голованов

Научный руководитель
канд. техн. наук, доцент

 13.06.2024
подпись, дата

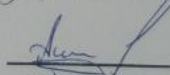
С.А. Гордин

Нормоконтролёр

 13.06.24
подпись, дата

О.В. Козлова

Рецензент

 _____
подпись, дата

А.Н. Анисимов

2024

РЕЦЕНЗИЯ

на магистерскую диссертацию

студента федерального государственного бюджетного
образовательного учреждения высшего образования
«Комсомольский-на-Амуре государственный университет»

Голованова Андрея Леонидовича

(Ф.И.О.)

по направлению 01.04.02 – Прикладная математика и информатика

Тема магистерской диссертации Разработка метода прогнозирования
временных рядов на основе нейрокомпьютерных технологий

Объем работы:

количество листов пояснительной записки 78

количество листов чертежей _____

Заключение о степени соответствия, выполненной работы заданию: работа
полностью соответствует заданию

Характеристика выполнения каждого раздела работы, степень использова-
ния последних достижений науки и техники, передовых методов работы:

В первой главе автор рассмотрел существующие методы прогнозирования
временных рядов, предоставил их классификацию. Также автор выбрал
перспективное направление в решении задачи автоматизированного про-
гнозирования.

Во второй главе рассмотрены базовые методы прогнозирования, основан-
ные на аппарате теории искусственных нейронных сетей. Рассмотрен во-
прос построения эффективного функционала минимизированной ошибки
прогнозирующей нейросети. Предложен метод повышения эффективно-
сти прогнозирующей нейросетевой модели с помощью экспоненциального и
адаптивного распределений вероятности выбора обучающего факта.

В третьей главе предложен генетический алгоритм комплексной оптими-
зации прогнозирующих нейросетей.

описывается процесс разработки аппаратнопрограммного комплекса.

В четвертой главе описан процесс разработки аппаратнопрограммного
комплекса. Рассмотрена разработанная библиотека классов.

Перечень положительных качеств работы:

В работе четко рассмотрены и классифицированы методы прогнозирования временных рядов.

Перечень основных недостатков работы: Отсутствует раздел тестирования подтверждающий практическое применение работы.

Оценка качеств магистерской диссертации: магистерская диссертация выполнена с использованием современного программного обеспечения, ведущих разработок и исследований в области информационных технологий.

Отзыв о работе в целом и предполагаемая оценка: работа выполнена на среднем уровне и заслуживает оценки «удовлетворительно», а Голованов А.Л. присвоения квалификации магистр по направлению 01.04.02 «Прикладная математика и информатика»

Рецензент



(подпись)

кандидат физико-математических наук,
доцент кафедры ИБИС и Ф ФГБОУ ВО
«Амурский гуманитарно-педагогический
государственный университет»

(должность, ученая степень)

Антон Николаевич Анисимов
(Ф.И.О.)

« » 2024 г.

О Т З Ы В
на магистерскую диссертацию

студента федерального государственного бюджетного
образовательного учреждения высшего образования
«Комсомольский-на-Амуре государственный университет»

Голованова Андрея Леонидовича

(Ф.И.О.)

по направлению 01.04.02 – Прикладная математика и информатика

Тема магистерской диссертации Разработка метода прогнозирования временных рядов на основе нейрокомпьютерных технологий

Объем работы:

количество листов пояснительной записки 78

количество листов чертежей нет

Заключение о степени соответствия, выполненной работы заданию _____
работа полностью соответствует заданию

Проявленная диссертантом самостоятельность при выполнении работы. Ритмичность и дисциплинированность в работе. Умение пользоваться литературным материалом, индивидуальные особенности диссертанта _____
При выполнении магистерской диссертации Голованов А.Л. проявил самостоятельность в решении поставленных задач, работал дисциплинированно и ритмично. Голованов А.Л. проявил хорошие навыки работы с литературными источниками и современными средствами получения информации.

Положительные стороны работы В работе четко рассмотрены и классифицированы методы прогнозирования временных рядов.

Недостатки работы отсутствует раздел тестирования, подтверждающий практическое применение работы.

Характеристика общетехнической и специальной подготовки диссертанта
Общетехническая и специальная подготовка соответствует требованиям, предъявляемым к магистрантам направления 01.04.02 – «Прикладная математика и информатика»

Заключение и предлагаемая оценка работы работа выполнена на среднем научном уровне и заслуживает оценки «удовлетворительно»

Научный руководитель, _____ Гордин С.А.
(подпись) (Ф.И.О.)
должность, ученая степень канд. техн. наук, доцент

« 14 » 06 2024 г.



АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

СПРАВКА

о результатах проверки текстового документа
на наличие заимствований

Комсомольский на Амуре государственный
университет

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Гордин Сергей Александрович, *Голованов А.Л.*
Самоцитирование рассчитано для:
Название работы: МД Голованов А.Л.
Тип работы: Магистерская диссертация
Подразделение: кафедра ПМ

РЕЗУЛЬТАТЫ

- **ВНИМАНИЕ. ДОКУМЕНТ ПОДОЗРИТЕЛЬНЫЙ:** обнаружены попытки маскировки заимствований. РЕКОМЕНДУЕТСЯ ПРОВЕРИТЬ ПОЛНЫЙ ОТЧЕТ
- **ОТЧЕТ О ПРОВЕРКЕ КОРРЕКТИРОВАЛСЯ:** ниже представлены результаты проверки до корректировки

СОВПАДЕНИЯ	88.51%	СОВПАДЕНИЯ	83.18%
ОРИГИНАЛЬНОСТЬ	11.49%	ОРИГИНАЛЬНОСТЬ	16.82%
ЦИТИРОВАНИЯ	0%	ЦИТИРОВАНИЯ	0%
САМОЦИТИРОВАНИЯ	0%	САМОЦИТИРОВАНИЯ	0%

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 15.06.2024

ДАТА И ВРЕМЯ КОРРЕКТИРОВКИ: 15.06.2024 05:09

Структура документа: Проверенные разделы: основная часть с.2-3, 6-80

Модули поиска: ИПС Адилет; Переводные заимствования*; Перефразирования по коллекции издательства Wiley; Перефразированные заимствования по коллекции Интернет в русском сегменте; Цитирование; Шаблонные фразы; Библиография; Перефразированные заимствования по коллекции Интернет в английском сегменте; Перефразирования по СПС ГАРАНТ: аналитика; Медицина; Коллекция НБУ; Переводные заимствования по Интернету (EnRu); Перефразирования по Интернету (EN); СПС ГАРАНТ: аналитика; Диссертации НББ; Публикации РГБ; Перефразирования по Интернету; Сводная коллекция ЭБС; Переводные заимствования (RuEn); Публикации eLIBRARY; СПС ГАРАНТ: нормативно-правовая документация; Издательство Wiley; Переводные заимствования по коллекции Гарант: аналитика; Кольцо вузов; Патенты СССР, РФ, СНГ, СМИ России и СНГ;

Работу проверил: Козлова Ольга Викторовна

ФИО проверяющего

Дата подписи: 15.06.2024

[Подпись]
Подпись проверяющего



Чтобы убедиться в подлинности справки, используйте QR-код, который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование корректным, система оставляет на усмотрение проверяющего. Предоставленная информация не подлежит использованию в коммерческих целях.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Кафедра «Прикладная математика»

УТВЕРЖДАЮ

Зав. кафедрой



А.Л. Григорьева

« » 20 г.

ЗАДАНИЕ на магистерскую диссертацию

Выдано студенту Голованову Андрею Леонидовичу

Тема магистерской диссертации Разработка метода прогнозирования
временных рядов на основе нейрокомпьютерных технологий

утверждена приказом по университету №0042-ЛСС-ГИА-ФКТ от 16.02.2024

Срок сдачи студентом законченной работы 13.06.2024

Исходные данные к работе: Разработки и исследования проводились на основе
теории вероятностей, математической статистики, системного анализа,
математического моделирования, теории искусственных нейронных сетей,
технологий объектно-ориентированного программирования.

Перечень подлежащих разработке вопросов: 1. Провести анализ существующих
методов прогнозирования временных рядов и выявить перспективные
направления в области их развития. 2. Исследовать применение искусствен-
ных нейронных сетей для автоматизированного прогнозирования временных
рядов. 3. Разработать методики повышения эффективности прогнозирующих
нейросетевых моделей.

Перечень графического материала нет

Задание принял к исполнению  «16» 02 2024 г.
(подпись)

Научный руководитель  Р.А. Горский
(подпись) (Ф.И.О.)

доцент кафедры ПМ «16» 02 2024 г.
(должность)

Аннотация

Разработка метода прогнозирования временных рядов на основе нейрокомпьютерных технологий

Пояснительная записка 78 с., 31 рис., 6 табл., 34 источника

Данная работа посвящена разработке методов прогнозирования временных рядов с помощью нейрокомпьютерных технологий. В работе проведен анализ уже существующих методов прогнозирования временных рядов, на основе которого выявлены перспективные направления в области их развития. Разработаны эффективные методы прогнозирования временных рядов, позволяющие максимально устранить недостатки существующих методов.

Abstract

Development of a method for predicting time series based on neurocomputer technologies

Explanatory note: 78 p., 31 pic., 6 tabl., 34 sources

This work is devoted to the development of methods for predicting time series using neurocomputer technologies. The paper analyzes the existing methods of forecasting time series, on the basis of which promising directions in the field of their development are identified. Effective methods for predicting time series have been developed to eliminate the shortcomings of existing methods as much as possible.

Содержание

Введение.....	6
1 Методы прогнозирования временных рядов.....	9
1.1 Анализ временных рядов	9
1.2 Классификация методов прогнозирования временных рядов	14
1.3 Статистические методы прогнозирования временных рядов	15
1.3.1 Адаптивные модели.....	15
1.3.2 Регрессионный анализ.....	17
1.3.3 Частотный анализ	18
1.4 Нестатистические методы прогнозирования временных рядов	19
1.4.1 Графические методы прогнозирования временных рядов	19
1.4.2 Аналитические нестатистические методы прогнозирования временных рядов.....	23
1.5 Сравнительный анализ методов прогнозирования временных рядов	25
2 Прогнозирование временных рядов на основе нейрокомпьютерных технологий	26
2.1 Общие положения теории нейронных сетей.....	26
2.2 Особенности использования искусственных нейронных сетей для прогнозирования временных рядов.....	30
2.2.1 Модель	30
2.2.2 Функционал ошибки прогнозирующих нейронных сетей	35
2.3 Нейросетевое прогнозирование временного ряда в режиме скользящего окна	37
2.4 Повышение эффективности прогнозирующей нейросетевой модели с помощью адаптивного распределения плотности вероятности выбора обучающего факта.....	42
2.5 Нейросетевое прогнозирование временных рядов на основе многофакторной модели.....	46

3	Методы повышения эффективности нейросетевого прогнозирования временных рядов.....	49
3.1	Общие замечания	49
3.2	Применение генетических алгоритмов для оптимизации прогнозирующих нейронных сетей	50
3.2.1	Базовые понятия генетического алгоритма	51
3.2.2	Функционирование генетического алгоритма.....	57
4	Программные средства автоматизированного прогнозирования временных рядов на основе нейрокомпьютерных технологий.....	60
4.1	Программное обеспечение «TimeSeries».....	67
	Заключение	75
	Список использованных источников	76

Введение

Появление персональных компьютеров послужило подспорьем широкому внедрению методов анализа и прогнозирования данных. Все шире используются статистические методы прогнозирования в деятельности плановых, аналитических, маркетинговых отделов производственных предприятий и объединений, торговых, страховых компаний, банков, правительственных учреждений. Поскольку условия технологических процессов и экономические условия ведения бизнеса изменяются во времени, необходимо постоянно отслеживать и предсказывать эти изменения для успешной реализации совершения деловых операций.

К настоящему времени разработано большое количество методов прогнозирования, конечной задачей которых является предсказание будущих событий с той или иной степенью надежности с целью использования этого прогноза при принятии решений.

Из-за распространения программных средств принятия решений большое значение представляет разработка методов автоматизированного прогнозирования.

Автоматизированное прогнозирование с использованием вычислительной техники рассчитано на то, чтобы функционирование прогнозирующих моделей и алгоритмов проходило с минимальным участием человека. Выбор модели и параметров модели для прогнозирования конкретных показателей, численно выражаемых с помощью временного ряда также рассчитано на это.

Одним из современных методов, используемых для прогнозирования, являются искусственные нейронные сети, которые способны устранить недостатки классических методов.

Одним из главных преимуществ нейронных сетей является их способность к обучению и самообучению, т.е. адаптации модели к решению конкретной задачи. Что делает искусственные нейронные сети одним из

перспективных направлений в разработке методов автоматизированного прогнозирования.

Объектом исследования являются методы прогнозирования временных рядов.

Предметом исследования является автоматизированная система прогнозирования временных рядов на основе разработанного метода.

Цель дипломной работы состоит в разработке эффективного метода прогнозирования временных рядов на основе искусственных нейронных сетей, позволяющих максимально возможно устранить недостатки существующих методов, обеспечивая при этом высокую достоверность прогноза; создании автоматизированной информационной системы, позволяющей решать задачу прогнозирования временных рядов.

Для достижения цели необходимо решить следующие задачи:

1. Провести анализ существующих методов прогнозирования временных рядов и выявить перспективные направления в области их развития.
2. Исследовать применение искусственных нейронных сетей для автоматизированного прогнозирования временных рядов.
3. Разработать методику повышения эффективности прогнозирующих нейросетевых моделей.

Методы исследования. Разработки и исследования проводились на основе теории вероятностей, математической статистики, системного анализа, математического и имитационного моделирования, теории искусственных нейронных сетей, технологий модульного и объектно-ориентированного программирования.

Научная новизна. В рамках дипломной работы были получены следующие результаты:

1. Показано, что применение искусственных нейронных сетей позволяет повысить эффективность прогнозирования временных рядов, лишено ряда недостатков классических статистических методов анализа и прогнозирования

временных рядов и может служить основой для автоматизированного прогнозирования (с минимальным участием человека).

2. Предложены и исследованы способы повышения эффективности прогнозирующих свойств и скорости обучения искусственной нейронной сети.

3. Усовершенствован генетический алгоритм комплексной оптимизации прогнозирующих нейросетей, позволяющий уменьшить время обучения.

Практическая ценность работы. Разработаны способы повышения эффективности прогнозирующих свойств и скорости обучения искусственной нейронной сети. Усовершенствован генетический алгоритм комплексной оптимизации прогнозирующих нейронных сетей, применение которого может уменьшить время обучения в 2-5 раз для многофакторных прогнозирующих моделей.

Структура и объем диссертации. Дипломная работа состоит из введения, четырех глав и заключения. Содержит 78 страниц, 6 таблиц, 31 рисунков. Список литературы состоит из 34 наименований.

1 Методы прогнозирования временных рядов

1.1 Анализ временных рядов

«Временным рядом (ВР) называют последовательность наблюдений, обычно упорядоченную по времени» [1].

В каждой области существуют явления, которые важно и интересно изучать, а именно их развитие и изменение во времени. В повседневной жизни интерес могут представлять такие явления, как метеорологические условия, изменение цены на тот или иной товар, характеристики состояния здоровья человека и т.д. [1]

Временные ряды можно рассматривать как ряды динамики, у которых в качестве признака упорядочения берется время. Временной ряд, содержащий N наблюдаемых значений Y_1, Y_2, \dots, Y_n , может быть записан в компактной форме $Y_t, t = 1, 2, \dots, n$, где t – порядковый номер наблюдения. Значения временного ряда называют иногда уровнями временного ряда.

Временные ряды бывают мгновенные и интервальные. Мгновенным называется такой ВР, у которого каждое значение соответствует наблюдению в конкретный момент времени. Значения интервального ВР представляют собой обобщенные показатели за некоторый промежуток. Если говорить коротко, то в анализе мгновенных и интервальных временных рядов существенного различия нет.

Статистический анализ временных рядов занимается изучением соотношения между закономерностью и случайностью в формировании значений уровней ряда, а также оценки количественной меры их влияния.

Анализ временных рядов заключается в решении двух задач:

- 1) выявление природы законов, принципов, количественных зависимостей, причинно-следственных связей, определяющих поведение процесса, порождающего временной ряд;

2) прогнозирование – получение оценки значений временного ряда, наблюдаемых в будущем.

Если наблюдаемым временным рядом является Y_1, Y_2, \dots, Y_t , то общей статистической моделью этого ряда служит модель вида:

$$Y_t = f(t) + u_t \quad (1.1)$$

В данном случае временной ряд рассматривается как сумма некоторой полностью детерминированной последовательности $\{f(t)\}$, называемой систематической составляющей, и случайной последовательности $\{u_t\}$, подчиняющейся вероятностному закону.

Иногда для этих двух составляющих, применяют термины «сигнал» и «шум».

Более сложной моделью является аддитивная модель вида:

$$Y_t = f(t) + S(t) + E(t), \quad (1.2)$$

где $f(t)$ – долговременная тенденция развития, тренд;

$S(t)$ – сезонная компонента;

$E(t)$ – остаточная компонента.

Тренд – это устойчивое изменение показателя в течение продолжительного времени. Выражается аналитической функцией, применимой к созданию прогнозных оценок.

Сезонная компонента характеризует устойчивые колебания значений ряда внутри наблюдаемого периода времени. Может проявляться в некоторых показателях, представленных данных за квартал или месяц.

Остаточная компонента – расхождение между значениями фактическими и расчетными.

Так же для временных рядов в экономике может использоваться мультипликативная модель:

$$Y_t = f(t)S(t)E(t) \quad (1.3)$$

или

$$Y_t = f(t)C(t)S(t) + E(t), \quad (1.4)$$

где $C(t)$ – циклическая компонента.

Рассмотрим различия между сезонной компонентой и циклической. Сезонная компонента, характерна регулярными периодическими отклонениями значения ряда в каждом наблюдаемом периоде, циклическая же компонента определяется повторяющимися подъемами и спадами, проходящими в 4 фазы: пик, рецессия, депрессия, подъем.

Исходные данные временного ряда можно охарактеризовать самыми простыми средствами описательной статистики, с помощью которых можно получить представление об особенностях исследуемого показателя, а также определить будет ли использование более глубоких методов анализа многообещающим.

При оценочно-описательном методе статистики, можно получить следующие показатели: среднее значение, дисперсия, моменты различных порядков, мода и медиана, коэффициенты асимметрии и др.

Динамику временных рядов можно охарактеризовать при помощи некоторого количества характеристик, характеристики могут быть базисными или цепными (коэффициент роста, коэффициент прироста, темп прироста и др.).

При малом количестве объектов статистической выборки данные фактических наблюдений могут дополняться, используя численное моделирование методом Бутстрепа [34].

Для определения наличия тренда во временном ряду обычно используют методы Форстера-Стюарта. Подробнее об этом можно узнать в источнике [28].

В методе Форстера-Стюарта гипотеза об отсутствии тренда ($L=0$) проверяется с помощью вспомогательных функций:

$$L = \sum_{t=2}^T I_t, \quad I_t = U_t - V_t, \quad (1.5)$$

$$U_t = \begin{cases} 1, & \text{если } Y_t < Y_{t-1}, \dots, Y_1, \\ 0 & \end{cases} \quad (1.6)$$

$$V_t = \begin{cases} 1, & \text{если } Y_t > Y_{t-1}, \dots, Y_1. \\ 0 & \end{cases} \quad (1.7)$$

Однородность данных обычно проверяется на основе критерия Ирвина, основанном на сравнении соседних значений ряда [23]. Далее в соответствии с ним значения характеристики

$$I_t = \frac{Y_t - Y_{(t-1)}}{\mu Y}, \quad (1.8)$$

сравниваются с табличными значениями.

Для оценки тесноты взаимосвязи уровней ряда используется автокорреляция.

Функция автокорреляции представляет собой совокупность коэффициентов автокорреляции, вычисленных для исследуемого показателя или разностного ряда для различных значений m :

$$k_X \left(\frac{mT}{n} \right) = \frac{1}{n-m} \sum_{i=1}^{n-m} \dot{X}(t_i) \dot{X}(t_{i+m}), \quad (1.9)$$

где n – число наблюдений временного ряда;

$$\dot{X}(t_i) = X(t_i) - \frac{1}{n} \sum_i^n X(t_i). \quad (1.10)$$

Основополагающим, является то, что факторы, влияющие на работу изучаемой системы, действовали, некоторым образом, в прошлом и настоящем, и ожидается, что они будут действовать схожим образом и в ближайшем будущем.

Оптимальный горизонт прогнозирования для каждого временного ряда определяется индивидуально.

Точность и адекватность – два важных определения, которые служат основной мерой качества прогнозирующей модели.

Точность модели характеризует степень близости расчетных данных к фактическим. Точность прогнозов можно оценить построением модели по усеченному объему данных точек с последующим сравнением прогнозных оценок с известными уровнями ряда, используя ретропрогноз.

По результатам сравнения обычно рассчитываются следующие показатели точности:

- среднее значение;
- среднеквадратическое отклонение;

- средний модуль ошибок прогнозирования;
- максимальное и минимальное отклонения.

Далее можно рассчитать точечный и интервальный прогнозы на основе построенной модели. Точечный прогноз формируется путем подстановки в модель соответствующего значения фактора $t=T+1, T+2, \dots, T+k$. Интервальные прогнозы строятся на основе точечных.

Исследуя свойства остаточной компоненты, т.е. расхождений, рассчитанных по модели уровней и фактических наблюдений, можно оценить адекватность моделей. Модели, у которых остаточная компонента имеет свойства независимости, случайности и нормальности распределения, считаются адекватными [1].

Критерий Дарбина-Уотсона является наиболее распространенным критерием для проверки корреляции внутри ряда. Если величина

$$D = \frac{\sum_{i=2}^n (e_i - e_{i-1})^2}{\sum_{i=1}^n e_i^2}, \quad (1.11)$$

где e_i – расхождение между фактическими и расчетными уровнями, имеет значение, близкое к 2, то модель регрессии считается достаточно адекватной.

Свойства нормальности распределения остаточной компоненты, способствуют построению интервального прогноза.

Интервальный прогноз определяется доверительной вероятностью прогноза, которая показывает степень уверенности покрытия построенным интервалом всей прогнозируемой величины. Она изменяется от 0 до 1. Заметим, что с увеличением доверительной вероятности интервальный прогноз увеличивается, и его полезность уменьшается.

К примеру, интервальный прогноз для линейной регрессии:

$$y^* \pm t_{n-2; 0,5(1+\gamma)} \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum (x_i - \bar{x})^2}}, \quad (1.12)$$

где y^* – оценка точечного прогноза;

$t_{n-2; 0,5(1+\gamma)}$ – квантиль распределения Стьюдента с $n-2$ степенями свободы порядка $0,5(1+\gamma)$;

\bar{x} – среднее значение;

x^* – значение, при котором вычисляется прогноз.

1.2 Классификация методов прогнозирования временных рядов

Методы прогнозирования временных рядов можно условно разбить на две категории: статистические и нестатистические. Статистические методы прогнозирования опираются на аппарат математической статистики. Данные методы используют вместе с общим статистическим анализом временных рядов. Совместное применение регрессионного анализа с корреляционным и дисперсионным анализами, является доказательством вышесказанному.

Категория нестатистических методов возникла сравнительно недавно, поэтому она является развивающейся областью. Теорию нейронных сетей, теорию хаоса и графические методы прогнозирования, можно отнести к категории нестатистических методов. Классификация методов представлена на рисунке 1.1.

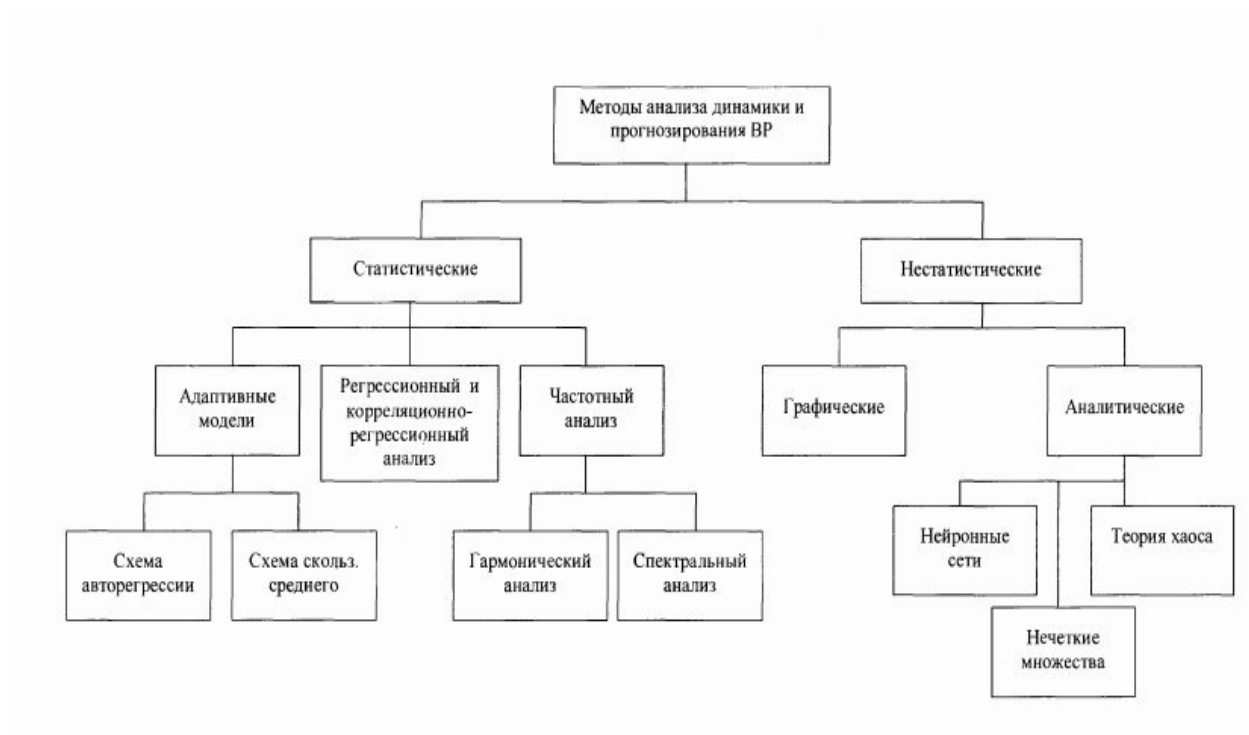


Рисунок 1.1 – Методы прогнозирования временных рядов

1.3 Статистические методы прогнозирования временных рядов

1.3.1 Адаптивные модели

В адаптивных моделях, относящихся к методам статистического прогнозирования, имеются механизмы автоматической настройки при изменении исследуемого показателя. Таким образом имеется возможность учитывать информационную неравнозначность данных. Данный механизм, является преимуществом при краткосрочном прогнозировании, где наблюдается более важны изменения развития исследуемого показателя на конце периода наблюдений, а не на всем пути его изменений.

Исходя из первого этапа наблюдений выводится первоначальная оценка параметров модели. Опираясь на данную оценку, делается прогноз, который в последствии сравнивается с фактическими наблюдениями. Далее модель подлежит корректировки в соответствии с величиной ошибки прогноза и вновь используется для прогнозирования следующего уровня, до исчерпания всех этапов наблюдений.

Таким образом, модель постоянно «впитывает» и обрабатывает новую информацию, приспосабливается к ней, и к концу периода наблюдения отражает тенденцию, сложившуюся на данный момент.

В основе адаптивных моделей положена модель рекурсивного гармонического процесса, выдвинутая Дж. Юлом [23].

Адаптивные модели можно разделить на две подгруппы: схема скользящего среднего и схема авторегрессии. Иногда данные моделей называют моделями авторегрессии – скользящего среднего (АРСС).

Исходя из схемы скользящего среднего, оценкой текущего уровня является взвешенное среднее всех предшествующих уровней, так же веса предшествующих уровней при наблюдениях убывают по мере удаления от последнего уровня, тем самым информационная ценность наблюдений увеличивается, приближаясь к концу периода наблюдений. К схеме скользящего среднего относятся модель Брауна, модель Хольта.

Исходя из схемы авторегрессии, оценкой текущего уровня является взвешенная сумма предшествующих уровней (их количество называется порядком модели). Информационная ценность наблюдений определяется не близостью к моделируемому уровню, а теснотой связи между уровнями.

Опираясь на Модель Брауна. Прогноз в момент времени t можно вычислить следующим образом:

$$\hat{X}_t = \alpha_{1,t} + \alpha_{2,t}, \quad (1.13)$$

где $\alpha_{1,t}$ и $\alpha_{2,t}$ – текущие оценки коэффициентов адаптивного полинома.

Модификацию (адаптацию) коэффициентов линейной модели можно осуществить следующим образом:

$$\alpha_{1,t} = \alpha_{1,t-1} + \alpha_{2,t-1} + (1 - \alpha)^2 e_t; \quad (1.14)$$

$$\alpha_{2,t} = \alpha_{2,t-1} + (1 - \alpha)^2 e_t. \quad (1.15)$$

где α – коэффициент дисконтирования данных;

$e_t = X_t - \hat{X}_t$ – ошибка прогнозирования.

Начальные значения параметров модели можно получить с помощью метода наименьших квадратов (МНК) на основе первых этапов наблюдений. Значение коэффициента дисконтирования определяется методом численной оптимизации, Оптимальное значение коэффициента лежит в пределах от 0 до 1.

Исходя из модели Хольта, коэффициенты линейной модели (1.12) модифицируются следующими соотношениями:

$$\alpha_{1,t} = \alpha_{1,t-1} + \alpha_{2,t-1} + \alpha_1 e_t; \quad (1.16)$$

$$\alpha_{2,t} = \alpha_{2,t-1} + \alpha_2 e_t. \quad (1.15)$$

Первоначальные значения параметров модели можно найти аналогично модели Брауна. Значения коэффициентов α_1 и α_2 устанавливаются в пределах от 0 до 1.

Модификацией модели Хольта для сезонных процессов является модель Хольта-Уинтерса и существует в двух видах: мультипликативной и аддитивной.

Рассмотрим модель авторегрессии. В модели авторегрессии порядка p текущий уровень ряда можно определить в виде взвешенной суммы p предыдущих наблюдений:

$$X(t) = \alpha_1 X(t-1) + \alpha_2 X(t-2) + \dots + \alpha_p X(t-p). \quad (1.18)$$

Параметры модели оцениваются методом наименьших квадратов (МНК).

Рассмотрим метод Бокса-Дженкинса, данный метод отличается тем, что параметры модели оцениваются отличным от МНК методом.

Метод Бокса-Дженкинса, именуемый также авторегрессионной интегрированной моделью скользящего среднего (АРИСС) [8] имеет три сглаживающих параметра $\alpha_1, \alpha_2, \alpha_3$:

$$\alpha_{1,t} = \alpha_1 X_t + \alpha_{2,t-1} + (1 - \alpha_1)(\alpha_{1,t-1} + \alpha_{2,t-1}) + \alpha_3(e_t - e_{t-1}); \quad (1.19)$$

$$\alpha_{2,t} = \alpha_2(\alpha_{1,t} - \alpha_{1,t-1}) + (1 - \alpha_2)\alpha_{2,t-1}; \quad (1.20)$$

где $e_t = X_t - \hat{X}_t$ – ошибка прогнозирования.

1.3.2 Регрессионный анализ

Значительную часть статистических методов представляют методы регрессионного анализа (классическая теория наименьших квадратов), применяющихся при анализе временных рядов.

Множественная линейная регрессия имеет следующий вид уравнения:

$$Y = \alpha_0 + \alpha_1 X_1 + \dots + \alpha_m X_m. \quad (1.21)$$

В каждом виде регрессионного анализа должна быть выбрана зависимая переменная Y (для которой строится уравнение регрессии) и одну или несколько независимых переменных X_i . Уравнение (1.21) позволяет изучаемым показателям установить статистическую взаимосвязь и, в случае ее стабильности, дать не только аналитические оценки, но и прогнозные.

На первом этапе строим уравнение регрессии зависимой переменной. На втором этапе выполняем расчет прогнозных значений зависимой переменной исходя из рассчитанного уравнения регрессии.

Для предварительного выявления и установления форм зависимости значений временного ряда от фактор-переменных при построении

многофакторной регрессии необходимо пользоваться аппаратом корреляционного анализа [11, 12].

1.3.3 Частотный анализ

При долговременных наблюдениях за изменениями во временных рядах могут появляться более или менее четкие регулярные колебания. Значения наблюдаемых изменений могут быть близкими к периодическим или строго периодическими и оцениваться в частотном аспекте.

Частотный анализ используется для выявления наличия колебаний и устойчивости периода колебаний. Частотный анализ включает в себя следующие методы:

- гармонический анализ;
- спектральный анализ;
- частотная фильтрация;
- кросс-спектральный анализ.

Частотный анализ считается эффективным, когда выборка состоит из достаточно большого объема данных, из которых предварительно исключена тенденция.

Гармонический анализ. На основании конечного преобразования Фурье, временной ряд может быть представлен с помощью линейных комбинаций функций времени – синусов и косинусов.

Пусть $Y(t)$ – временной ряд. Его значения будут выглядеть следующим образом:

$$Y_t = \bar{Y} + \sum_j (\alpha_j \cos(\frac{2j}{T} t)) + \sum_j (b_j \sin(\frac{2j}{T} t)), \quad (1.22)$$

где t меняются от 1 до T ;

\bar{Y} – оценка математического ожидания временного ряда.

Гармонический анализ дает возможность выявить существенные гармоники для восстановления по ним вида функции, аппроксимирующей временной ряд.

Спектральный анализ. Периодограмма временного ряда $X(t)$, $t = 0, 1, \dots, T$, принимает следующий вид, формула 1.23, в результате чего рассчитывается оценка спектральной плотности:

$$I_t(\lambda) = \frac{1}{T} |\sum_{t=0}^{T-1} X(t) \exp(i\lambda t)|, \quad (1.23)$$

где λ – значение частоты.

Кросс-спектральный анализ. Дает оценку связи между частотными составляющими двух временных рядов при помощи параметров когерентности, фазового сдвига и коэффициента усиления.

Частотная фильтрация. Задействует высокочастотный и низкочастотный фильтр. Каждому фильтру рассчитываются соответствующие силовая и фазовая характеристики. Низкочастотный фильтр выполняет функции устранения тренда. Высокочастотный фильтр, выполняет функции выделения тренда из исходных данных.

1.4 Нестатистические методы прогнозирования временных рядов

Методы прогнозирования временных рядов, не базирующиеся на математической статистике, можно отнести к категории «нестатистические».

Такими методами являются:

- методы прогнозирования на основе графиков (графические);
- методы прогнозирования на основе теории хаоса;
- методы прогнозирования на основе использования нечеткой логики и нечетких множеств;
- методы прогнозирования использующие положения теории искусственных нейронных сетей (теория искусственного интеллекта).

1.4.1 Графические методы прогнозирования временных рядов

Методы анализа и прогнозирования временных рядов, опирающиеся на исследование изображений графиков, называются графическими. Графический метод позволяет получить наглядно, динамику изменения временного ряда [33].

Выводы, характеризующие природу и поведение ряда, могут опираться на объективные характеристики и функции временного, а также на фигуры, образуемые графиком, носящими исключительно субъективный характер.

Методы графического прогнозирования пользуются спросом при анализе временных рядов, связанных с финансовой деятельностью. В качестве примера служит динамика изменения биржевых котировок валют и ценных бумаг [33].

Наглядность является основной чертой графического метода. Недостатком метода является не очень точная и не всегда объективная интерпретация моделей авторегрессии временного ряда.

Благодаря графическим методам можно увидеть тип тенденции, а именно направление, в котором движется временной ряд.

Рассмотрим преимущества графических методов:

- специалист, не обладающий математическим образованием, может в короткий срок освоить данный метод. В сравнении с математическими моделями графический метод получает преимущество, а именно простоту понимания использования.

- анализ и построение прогноза временного ряда можно проводить вручную, а также прибегать к использованию специального программного обеспечения и специальных технических средств. Как следствие метод имеет хорошую скорость расчета.

Но у графических методов имеются и свои недостатки, основным из которых является субъективность прогнозирования, исходя из этого относительно низкая достоверность.

Так же, при анализе временных рядов, связанных с финансами, графическое изображение обладает своей спецификой динамики временного ряда. Поясним. Используя различные виды графиков: гистограммы, «японские свечи» и тому подобные, имеющих различие во внешнем отображении информации следующего рода изображенных на рисунке 1.2:

- начальное значение временного ряда за период (например, цена открытия);
- конечное значение ряда (цена закрытия);
- максимальное и минимальное значение ряда за период.



Рисунок 1.2 – Биржевой график типа «японские свечи»

Наблюдая за рисунком 1.2 и подводя итоги о вышесказанном, видно, что график отражает четыре различных значения интервального временного ряда, а не точечные значения временного ряда в текущий момент времени.

Используя условные направляющие линии, их еще называют «линии поддержки и сопротивления», специалист сужает диапазон изменения значений временного ряда, отображаемый на графике. Линии поддержки и сопротивления возлагают ограничение на график, а выход графика за пределы ограничения сигнализируется как смены тенденции.

Так же графические методы ориентируются на определении некоторых фигур, наблюдающихся при анализе временных рядов. Статистически установлено, что фигуры, которые можно разглядеть на графике, являются, как правило сигналом перелома тенденции или же сигналом, что тренд временного ряда не изменит своего направления.

Среди наиболее распространенных моделей, сигнализирующих перелом тенденции, можно назвать фигуру «голова и плечи» изображенную на рисунке 1.3, «тройная вершина» изображенную на рисунке 1.4а, «бриллиант», изображенный на рисунке 1.4б.

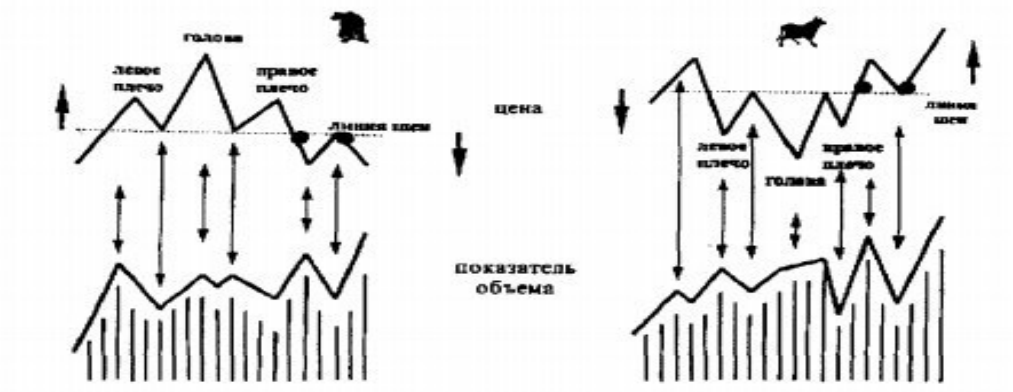


Рисунок 1.3 – Модель перелома тенденции «голова и плечи»

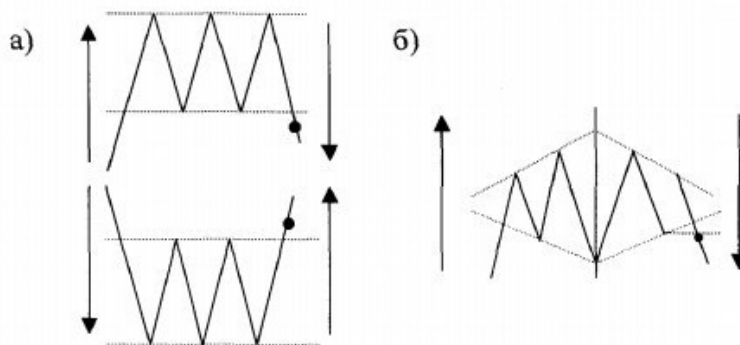


Рисунок 1.4 – Модели перелома тенденции
а - тройная вершина; б - бриллиант

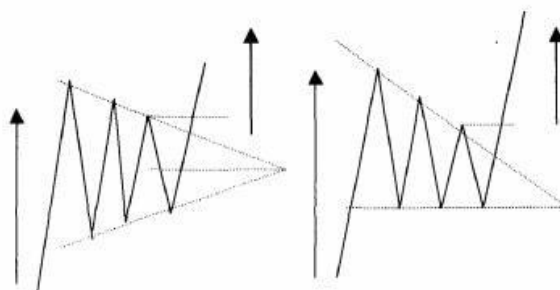


Рисунок 1.5 – Модели продолжения тенденции «треугольник»

Наиболее часто используемыми фигурами продолжения тенденции являются «треугольник», изображенный на рисунке 1.5, «флаг», изображенный на рисунке 1.6а, «вымпел», изображенный на рисунке 1.6б.

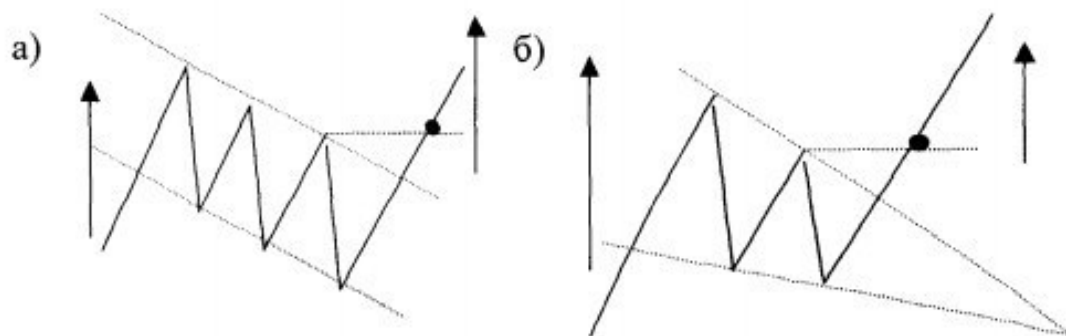


Рисунок 1.6 – Модели продолжения тенденции «флаг» и «вымпел»
а - флаг; б - вымпел

1.4.2 Аналитические нестатистические методы прогнозирования временных рядов

К этой группе методов прогнозирования можно отнести множество разнообразных моделей и методик, таких как теория детерминированного хаоса, теория нечетких множеств и нечеткой логики, волновая теория.

Теория детерминированного хаоса [18, 32] связана с пониманием незамкнутых систем как самоорганизующихся, то есть систем с уменьшающейся с течением времени энтропией. Необходимая предпосылка эффектов самоорганизации заключается в наличии потока энергии, поступающего в систему извне и диссипируемого ею. Благодаря этому потоку система становится активной, способной к автономному образованию структур, что не противоречит законам термодинамики.

Возникающие в системе новообразования по отношению к малым возмущениям являются критерием отличия системы хаотической от беспорядочной.

Поведение хаотической системы описывается аттракторами: устойчивыми циклами и образованиями более сложной структуры.

Временной ряд вполне может быть рассмотрен как последовательность, порожденная хаотической системой. Отделить шум от детерминированного поведения и восстановить систему конечномерных дифференциальных уравнений, описывающих аттрактор, позволит теория динамического хаоса, далее можно спрогнозировать поведение временного ряда в будущем.

Нечеткие множества и нечеткая логика основаны на концепции, предложенной в работе. Теория нечетких множеств благополучно решает задачи распознавания образов и обработки сигналов. Прогнозирование временных рядов применяется как перспективное направление теории нечетких множеств.

Волновая теория, так же называемая «Волновая теория Эллиотта», опирается на некоторые свойства последовательности чисел Фибоначчи и непосредственно связана с графическими методами прогнозирования [33]. В волновой теории временной ряд описывают последовательностью волн:

- представляющих основную тенденцию;
- корректирующих волн.

Волны делятся на наборы более мелких волн аналогичной структуры. Предположение того, что структура волн будет однотипна, служит предпосылкой для прогнозирования временных рядов. Наиболее успешно волновая теория применяется при прогнозировании временных рядов, связанных с экономикой.

1.5 Сравнительный анализ методов прогнозирования временных рядов

Методы прогнозирования временных рядов, выбранные для исследования, изображены на рисунке 1.1, имеют свои положительные стороны и отрицательные стороны, но все эти методы являются дополнением друг друга.

Основным преимуществом методов относящихся к категории статистических методов выступает высокая точность прогноза, к сожалению, за точность прогноза приходится платить постоянной подстройкой параметров модели, а иногда и изменением типа модели, что в свою очередь требует больше затрат по времени и сил. Подстройка параметров и изменением типа модели требует высокой математической подготовки специалиста, выполняющего расчеты. К примерам таких методов и моделей относят методы частотного анализа и регрессионные модели.

Такой недостаток как подстройка параметров и изменением типа модели частично отсутствует у адаптивного метода обработки данных, но данный метод содержит некоторые показатели модели, которые приходится подбирать вручную, пусть и в меньшем количестве.

Рассмотренный графический метод обработки данных, относящийся к нестатистическим методам, зарекомендовал себя как достаточно простой метод для понимания, но данный метод не может дать высокой точности прогноза относительно статистических методов.

Недостатки классических методов обработки и анализа временных рядов, выявленные выше, составляют некоторые трудности для применения их при решении задач прогнозирования. Наиболее подходящим и многообещающим подходом при решении задач прогнозирования данных являются исследования, выполняемые при помощи нестатистических методов прогнозирования, таких как: нечеткие множества, теория хаоса, искусственные нейронные сети и генетические алгоритмы.

2 Прогнозирование временных рядов на основе нейрокомпьютерных технологий

2.1 Общие положения теории нейронных сетей

В попытках создать модель, которая могла бы повторять функции головного мозга, была создана теория искусственных нейронных сетей. В данной теории за основу брали функции такие как обучаемость, обобщение и абстрагирование.

В основу теории нейронных сетей легла модель искусственного нейрона, изображенного на рисунке 2.1, данная модель прототип простейшей клетки нервной системы человека [14]. Нейроны могут образуют из своих соединений различной сложности структуры, данные структуры имеют название искусственные нейронные сети.

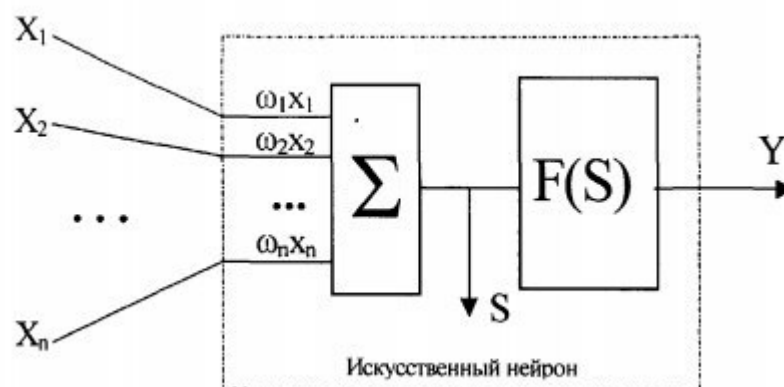


Рисунок 2.1 – Модель искусственного нейрона

На вход искусственного нейрона поступает некоторое количество сигналов $X = \{x_1, x_2, \dots, x_n\}$ – вектор входов. Первый блок получает взвешенное суммирование значений входов, формула 2.1:

$$S = \sum_i \omega_i x_i, \quad (2.1)$$

где ω_i – веса, с которыми суммируются входы x_i .

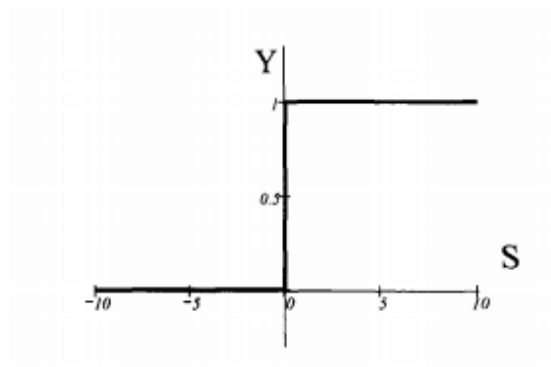


Рисунок 2.2 – Пороговая функция

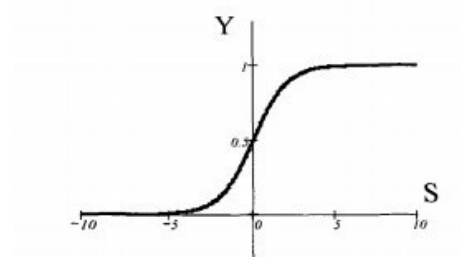


Рисунок 2.3 – Сигмоидальная функция

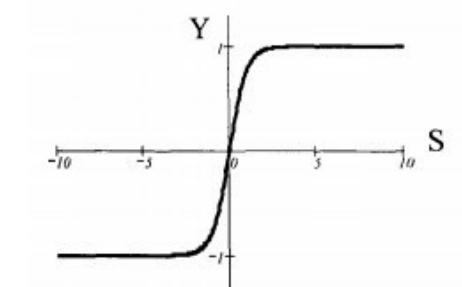


Рисунок 2.4 – Гиперболический тангенс

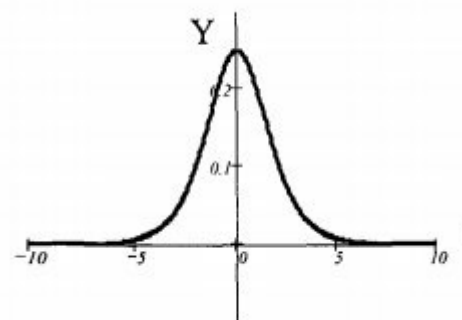


Рисунок 2.5 – Гауссова функция

Сигнал S преобразовывается с помощью активационной функции $Y = F(S)$, в качестве которой чаще всего используются [14, 25]:

- 1) линейная функция $Y = KS$;

$$2) \text{ пороговая функция (рисунок 2.2): } Y = \begin{cases} 1, & \text{если } S > T \\ 0, & \text{если } S \leq T \end{cases}$$

где T – значение порога;

$$3) \text{ сигмоидальная функция (рисунок 2.3): } Y(S) = \frac{1}{1 + \exp(-\alpha S)},$$

где α – параметр сигмоиды;

$$4) \text{ гиперболический тангенс (рисунок 2.4): } Y(S) = th(S) = \frac{e^S - e^{-S}}{e^S + e^{-S}};$$

$$5) \text{ Гауссова функция (рисунок 2.5): } Y(S) = \exp\left(-\frac{(x-c)^2}{r^2}\right),$$

где c – центральное значение;

r – параметр функции.

Основополагающее свойство искусственной нейронной сети, наличие алгоритмов обучения. Способность адаптироваться и решать текущую задачу путем перебора весов между нейронами в ответ на эталонные предъявляемые образцы способствует решению большого числа задач.

Применяются два подхода обучения нейросети: «с учителем» и «без учителя».

Алгоритм обучения с учителем. Факторы обучения - пара входного и выходного векторов. Другими словами, изначально известно, что должно быть на выходе при имеющемся входе. Примером такого обучения будет являться непосредственно алгоритм обучения персептрона, процедуру обратного распространения ошибки, некоторые стохастические алгоритмы обучения и др.

Алгоритм обучения без учителя или самообучение нейросети. Факторы обучения – только выходные векторы, входные же векторы неизвестны. При помощи данного подхода искусственная нейронная сеть самостоятельно учится распознавать и классифицировать предъявляемые ей факты. Примером такого обучения, представляют самоорганизующиеся карты Кохонена.

Задачи, решаемые искусственной нейронной сетью связаны с обучением. Области, в которых применяются искусственные нейронные

сети являются: распознавание образов, прогнозирование, анализ данных, принятие решений, оптимизация.

Выделим ряд областей, где искусственные нейронные сети успешно зарекомендовали себя. Существует три категории задач:

- прогнозирование;
- распознавание;
- классификация.

Такие свойства искусственных нейронных сетей как обобщение и абстрагирование увеличивают эффективность применения при решении задач прогнозирования [14, 25].

Способность извлекать данные из входных сигналов и отдавать на выход то, что сеть никогда «не видела», является сутью свойства абстрагирования некоторых нейронных сетей.

Способность нейронных сетей к обобщению состоит в том, что отклик сети после обучения может быть до некоторой степени нечувствителен к небольшим изменениям входных сигналов. Данное свойство имеет особенное значение при прогнозировании временных рядов. В качестве модели временного ряда в самом общем случае принимается сумма двух составляющих (1.1), которые еще можно назвать: систематическая и случайная составляющая, «сигнал» и «шум».

Для сильно зашумленных временных рядов (для которых значительно влияние трудно учитываемых или не учитываемых факторов, представленных в классической модели случайной составляющей) свойство способности нейронных сетей к обобщению играет важную роль.

2.2 Особенности использования искусственных нейронных сетей для прогнозирования временных рядов

2.2.1 Модель

Разберем как функционирует обученная нейронная сеть для целевого назначения. Первым шагом подаются значения входного сигнала (входной вектор). На втором шаге происходит возбуждение сети, и она третьем – выдается выходной вектор.

Нейронная сеть сама по себе строит некую внутреннюю модель в процессе ее обучения или самообучения. По итогу эта модель может быть точной или приближительной. Что же значит точная модель? В точной модели входным сигналам соответствуют в точности те же выходные сигналы, что и выходные векторы обучающих пар. А что касается приближительной модели, то она образуется при помощи прогнозирующей нейросети.

Перечислим этапы подготовки нейросети к ее использованию. После того как мы определили целевое назначение нейронной сети, переходим к этапу определения входных и выходных векторов.

В общем случае для анализа временных рядов при построении нейросетевой модели прогнозирования возможны два различных подхода.

В случае «однофакторной» прогнозирующей модели, непосредственно, на входы нейросети подается только предыстория временного ряда $x_{i-1}, x_{i-2}, \dots, x_{i-d}$. В качестве единственного выхода нейросети имеет место оценка значения временного ряда x_i в прогнозируемый момент или интервал времени t_i (рисунок 2.6).

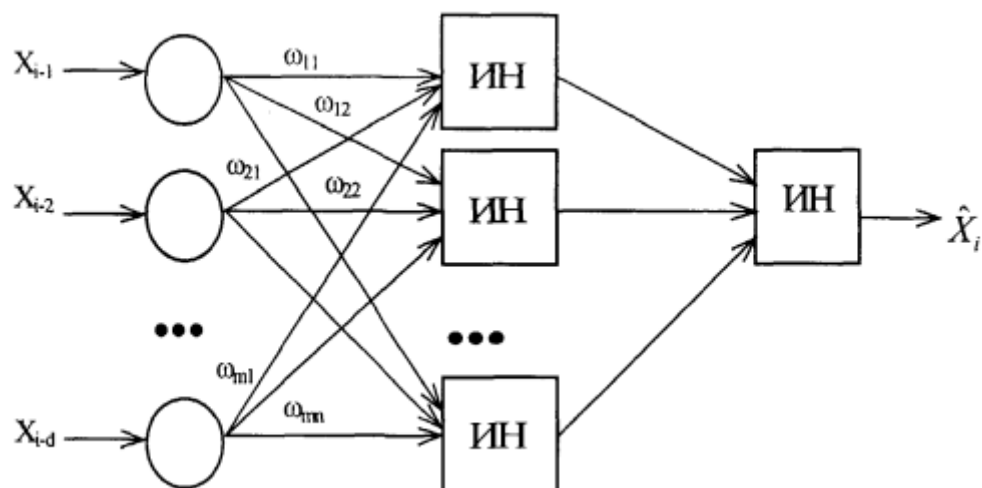


Рисунок 2.6 – Простейшая нейросетевая прогнозирующая модель

Исходным пунктом для такого подхода являются различные теоретические исследования, собственно, в том числе и теорема Такенса. В соответствии с данной теоремой, если временной ряд порождается динамической системой, т.е. значения x_i есть произвольная функция состояния такой системы, то существует такая глубина погружения d (примерно равная эффективному числу степеней свободы данной динамической системы), которая обеспечивает однозначное предсказание следующего значения временного ряда.

Исходя из вышесказанного можно сделать вывод, о том, что прогнозирование временного ряда сводится к задаче интерполяции функции многих переменных. С помощью обучающей выборке нейросеть используется для восстановления этой неизвестной функции, а эта функция в свою очередь представляет собой историю исследуемого временного ряда.

Отметим, что для некоторых временных рядов, в общем-то, более корректным является использование в качестве выхода нейросети приращения временного ряда $\Delta x_i = x_i - x_{i-1}$, а не оценки значения x_i . Но данный способ не всегда имеет смысл, он уместен в тех случаях, когда эти приращения незначительны в сравнении с самими значениями ряда, или когда более важным является определение направления и абсолютного значения изменения ряда, а не его значение.

Вообще говоря, в первую очередь интерес представляет знак изменения ряда для многих временных рядов (таких как финансовые ряды, и ряды в некоторых технических процессах), а не его абсолютное значение. Но нельзя забывать, что необходимо иметь некоторый критерий оценки степени уверенности нейросети в предсказании.

В таком случае разумно строить нейронную сеть с двумя выходами, один из которых принимает бинарные значения, отражающие направление предполагаемого изменения ряда (увеличение, уменьшение), другой принимает значения из диапазона $[0;1]$ и представляет собой вероятность, отражающую уверенность нейронной сети в сделанном предсказании.

Обычно, значения активационной функции искусственных нейронов, составляющих прогнозирующую нейросеть, лежат в диапазоне $[0;1]$, а значения входных и выходных векторов, лежат в более широком диапазоне.

Поэтому перед обучением и непосредственным использованием прогнозирующей нейросети необходима нормализация входных и выходных векторов.

Линейная нормализация является простейшим видом нормализации:

$$x_i^* = \frac{x_i - x_{min}}{x_{max} - x_{min}}, \quad (2.3)$$

учитывающая минимальное x_{min} и максимальное x_{max} значения признака.

Тем не менее, линейная нормализация приемлема только для узкого круга задач. К сожалению, невозможно заранее определить минимальное и максимальное значения ВР, часто так же неизвестен весь диапазон значений признака. Кроме того, распределение значений часто не равномерное, а носит более сложный характер. Такие не равномерные распределения, чаще всего, аппроксимируемы нормальным распределением или распределением Стьюдента.

В последнем случае для значений, расположенных ближе к математическому ожиданию величины, необходима большая точность отображения, чем для лежащих по краям распределения.

Тогда разумно использовать экспоненциальную нормализацию, вид которой представлен следующей формулой:

$$x_i^* = \frac{1}{1 + \exp(\mu - \alpha x_i)} \quad (2.4)$$

На рисунке 2.6 в качестве иллюстрации внутреннего представления нейронной сети выбрана архитектура сети прямого распространения. Общим алгоритмом обучения таких сетей, непосредственно, является процедура обратного распространения ошибки. На самом деле можно использовать так же и другие алгоритмы обучения, такие как стохастические. Нейронная сеть прямого распространения наиболее успешно применима в качестве нейросетевой парадигмы для решения общих задач прогнозирования.

Когда используется многофакторная модель, то кроме предыстории прогнозируемого временного ряда на входы нейросети подаются значения различных внешних факторов. Здесь непосредственно речь идет об анализе многомерных временных рядов [1, 23, 28].

Принципиального отличия прогнозирования многофакторной и однофакторной модели нет. Процесс подготовки исходных данных, их нормализации, обучения нейронной сети и ее функционирования у них почти схож.

Чем все-таки различие моделей, так в сложности выбора внешних факторов при использовании многофакторной нейросетевой модели.

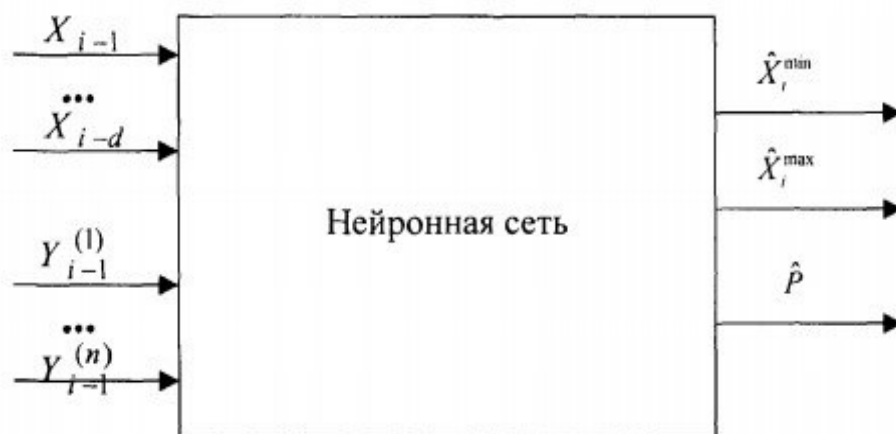


Рисунок 2.7 – Многофакторная модель для построения интервального прогноза

При описании прогнозирующих моделей решается задача построения точечного прогноза.

Все же, иногда требуется получить интервальный прогноз, в который попадает реальное значение с заданной доверительной вероятностью.

Для построения интервального прогноза можно воспользоваться как традиционными методами получения интервального прогноза на основе точечного, который выдает нейросеть, (см. формулу (1.12)), так и организовать саму нейросеть несколько иным образом.

Что бы организовать нейросеть так сказать иным образом, выходной вектор ИНС должен состоять из трех компонент: начального значения интервала, конечного значения (или ширины интервала) и доверительной вероятности.

На рисунке 2.7 проиллюстрирована схема построения входных и выходных векторов для многофакторной прогнозирующей нейросетевой модели, выдающей интервальный прогноз. Значения внешних факторов обозначены через $Y^{(1)} \dots Y^{(n)}$.

Рассмотрим из-за чего может уменьшаться точность прогнозирования. Ответ прост: из-за роста глубины прогноза. Это происходит если требуется глубина прогноза более чем на один шаг вперед, то прогнозирование повторяется необходимое количество раз, при этом в качестве значений входов нейросети используется уже полученный ранее прогноз.

Отметим, что любой алгоритм настройки нейросети является множественно-вероятностным. Поэтому точность решения может оказаться выше при использовании достаточно сложных методов классического анализа.

Тем не менее, время, затраченное на «классическое» решение, может оказаться слишком большим для конкретной задачи. Применимость такого решения для широкого диапазона реальных значений будет под вопросом, а время корректировки модели займет большое количество времени. Для

решения задач классическими методами требуются достаточно квалифицированные специалисты с хорошей математической подготовкой.

Нейросетевое решение, собственно, свободно от указанных недостатков.

Все вышесказанное подталкивает отдать предпочтение использованию нейросетевых технологий (минимальное участие человека или вовсе без него), а не классическими методами автоматизированного прогнозирования.

Особенно привлекательно применение многофакторных прогнозирующих нейросетевых моделей. Все же, в этом случае время обучения нейросети может оказаться неприемлемо большим из-за растущего нелинейно числа факторов, не говоря об открытом вопросе о сходимости таких сетей.

2.2.2 Функционал ошибки прогнозирующих нейронных сетей

Одной из задач многофакторной оптимизации является обучение искусственных нейронных сетей. При этом ищется глобальный минимум параметра – в теории нейронных сетей это функционал ошибки.

Суммарная квадратичная ошибка при обучении большинства нейронных сетей выступает в качестве функционала ошибки, она служит мерой обученности нейронной сети и готовности ее к основному предназначению.

$$E = \frac{1}{2} \sum_p \sum_j (y_j^{(p)} - \hat{y}_j^{(p)}), \quad (2.5)$$

где $y_j^{(p)}$ – действительное;

$\hat{y}_j^{(p)}$ – прогнозные значения.

Суммирование ведется по всем обучающим парам p , по всем нейронам выходного слоя j .

При помощи метода градиентного спуска (2.6) ведется обучение нейронной сети, которое сводится к минимизации функционала ошибки:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}, \quad (2.6)$$

где η – коэффициент скорости обучения;

∂w_{ij} – изменение веса от i -го к j -му нейрону.

Одним из критериев прекращения дальнейшего обучения НС служит достижение заданного уровня средней или минимальной квадратичной ошибки нейросети.

Все же, для многих временных рядов среднеквадратичная ошибка не имеет большого практического смысла, а гораздо важнее правильность угадывания знака изменения значения ряда.

Для модели, где выход нейросети представляет собой степень уверенности сети в знаке изменения временного ряда $y_t \in [-1;1]$ и нелинейностью вида $y = th(S)$ можно предложить следующий функционал ошибки:

$$E = -\sum_p \ln [1 + r_p \operatorname{sgn}(y_p)], \quad (2.7)$$

где $r_i = \frac{\Delta X_i}{X_i}$ – относительное изменение значения ряда,

p – номер обучающего факта.

Для того чтобы сравнивать качество прогнозов различных нейросетей с классическими методами прогнозирования необходим отдельный критерий, отражающий статистические свойства обученной нейросети для множества прогнозов. Так же отдельный критерий необходим для оценки прогнозирующих свойств обученной нейронной сети

Статистические свойства качества прогнозирующей модели могут быть изучены при использовании прогноза для нескольких уже известных отсчетов временного ряда и построении на их основе оценок параметров, характеризующих прогнозирующую модель.

В качестве одной из наиболее простых оценок может выступать среднее модулей отклонений прогнозов от истинных значений:

$$M = \frac{1}{m} \sum_j |X_j - \hat{X}_j|. \quad (2.8)$$

Среднеквадратичная ошибка прогнозов (СКО) служит более сложной оценкой, которая отвечает характеру нейросетевых моделей:

$$СКО = \frac{1}{m} \sum_j (X_j - \hat{X}_j)^2. \quad (2.9)$$

Непосредственно, этой оценке соответствует среднеквадратичное отклонение или момент второго порядка для обычных вариационных рядов. Недостатком СКО является различная размерность для разных рядов и, как следствие, трудность их сопоставления.

Оценка, характеризующая правильность угадывания знака изменения временного ряда, выражается следующим образом:

$$P = \frac{1}{m} \sum_j \text{sign} \left((X_j - X_{j-1})(\hat{X}_j - X_{j-1}) \right). \quad (2.10)$$

Эти характеристики позволяют получить численное выражение качества сделанных прогнозов, сравнить их между собой и с прогнозами, полученными с помощью других методов.

В источнике [28] также встречаются ссылки на сравнение дисперсий различных прогнозов, позволяющих оценить их вариабельность.

2.3 Нейросетевое прогнозирование временного ряда в режиме скользящего окна

Адаптируемость во времени без существенного пересмотра прогнозирующей модели – вот важное преимущество использования ИНС для прогнозирования временных рядов. Это их свойство чрезвычайно важно и играет ключевую роль в использовании НС для автоматизированного прогнозирования.

Наибольший интерес представляет решение задачи прогнозирования временного ряда, динамично меняющегося во времени, где требуется постоянное переобучение (до обучения) нейронной сети. Примерами, где необходимо подобное прогнозирование, являются: прогнозирование состояния дискретного канала связи, прогнозирование загрузки модемного пула интернет-провайдера [13, 5–7], прогнозирование биржевых котировок ценных бумаг в реальном времени [10], некоторые задачи военно-прикладного характера.

Для реализации прогнозирования в реальном времени используется модель скользящего окна. Рассмотрим из чего она состоит на примере одномерного ВР [3]. Для прогнозирования в момент времени используются значения ряда в моменты t_{i-1}, \dots, t_{i-s} (s – размер окна), которые рассматриваются как набор из $(s - d + 1)$ фактов для обучения нейронной сети (d – размерность входов сети). После обучения сети и получения прогнозного значения скользящее окно смещается на один шаг, и нейронная сеть дообучается с учетом вновь полученных данных.

С помощью программных средств, описанных в источнике [3], прогнозировался временной ряд чисел солнечной активности (чисел Вольфа). Использовалась нейронная сеть прямого распространения с алгоритмом обучения по методу обратного распространения ошибки с одним нейроном в выходном слое и дополнительным «скрытым» слоем нейронов с сигмоидальной активационной функцией.

Для оценки качества прогноза и его сравнительного анализа проводился прогноз того же ряда на том же временном интервале с помощью одной из самых распространенных моделей авторегрессии – модели скользящего среднего. Глубина скользящего среднего была выбрана равной 3 шагам.

Модель авторегрессии была выбрана как наиболее простая и характерная для ряда чисел Вольфа. Ввиду его периодичности линейная и полиномиальная регрессия не являются адекватными моделями этого ряда, а простая тригонометрическая регрессия не обладает достаточной точностью.

Сравнение прогнозов, полученных с помощью нейронной сети и модели авторегрессии, показано на рисунке 2.9. Кроме меньшей среднеквадратической ошибки и более высокого процента угаданных знаков (таблица 2.1) из рисунка 2.9 видно меньшее запаздывание прогноза нейронной сети относительно исходных данных по сравнению с классическим методом.

Таблица 2.1 – Сравнение прогнозов, полученных с помощью нейронной сети и модели авторегрессии

Характеристика	Нейронная сеть	Модель авторегрессии
Число прогнозируемых отсчетов	56	56
Число угаданных знаков приращения ряда	40	16
Процент угаданных знаков	71%	29%
Среднеквадратическая ошибка прогноза	276,58	904,93

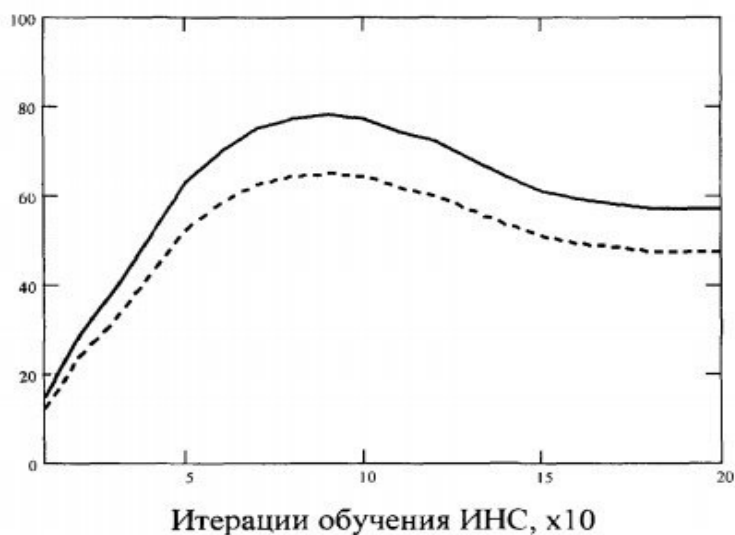


Рисунок 2.8 – Изменение процентной точности угадывания знака изменения ВР со временем

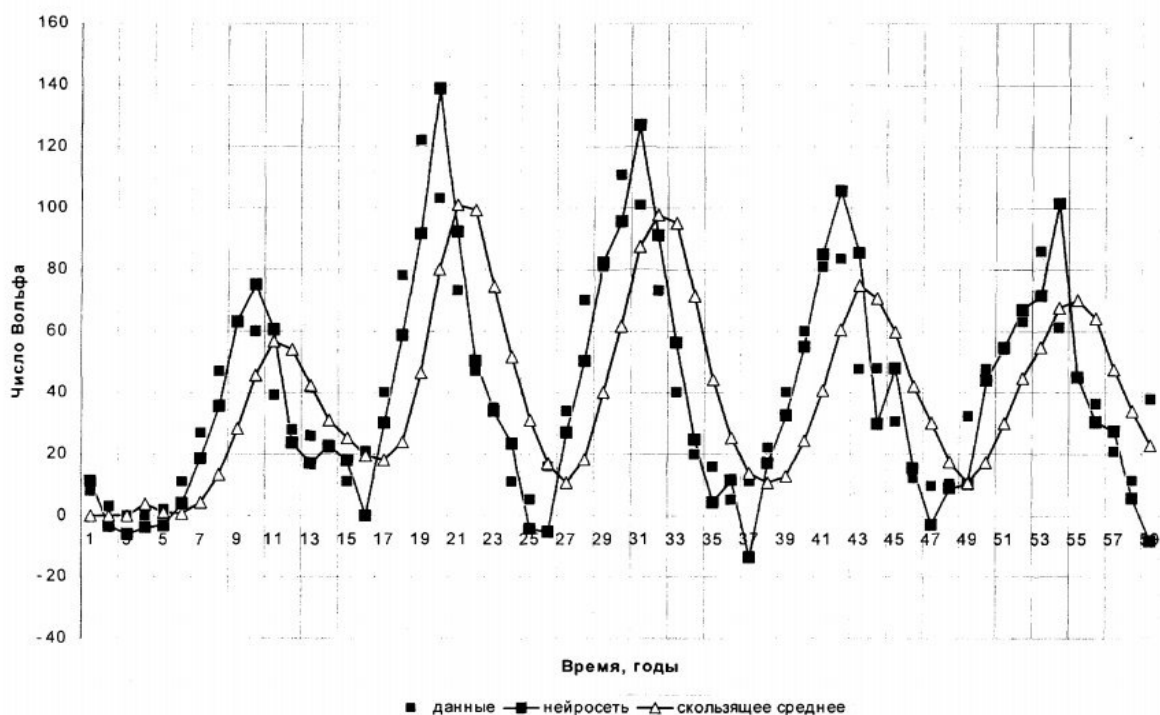


Рисунок 2.9 – Прогноз временного ряда чисел солнечной активности

Изменение точности угаданных знаков от времени (номера прогнозируемого отсчета) при перемещении скользящего окна обучающей выборки имеет приблизительный характер, изображенный на рисунке 2.8. Вначале идет резкий рост точности прогнозирования из-за обучения и дообучения нейронной сети, а затем некоторый ее спад и асимптотическое приближение к некоторому стабильному значению. Для наиболее «трудных» временных рядов точность стабилизируется на уровне чуть более 50% угаданных знаков, что соответствует случайному совпадению.

Такое поведение НС можно объяснить следующим образом. Со временем дополнительное обучение нейросети и адаптация модели перестает приносить желаемый эффект, так как происходит чрезмерное обучение или «переобучении» нейронной сети. Как только дополнительное обучение начинает приводить к снижению точности прогнозирования, борьба с этим эффектом проводится с помощью периодической рандомизации некоторых или всех весов нейронной сети.

Исследование зависимости времени обучения нейронной сети от ее структуры описаны в источниках [5, 21]. Характер этой зависимости приведен на рисунках 2.10, 2.11.

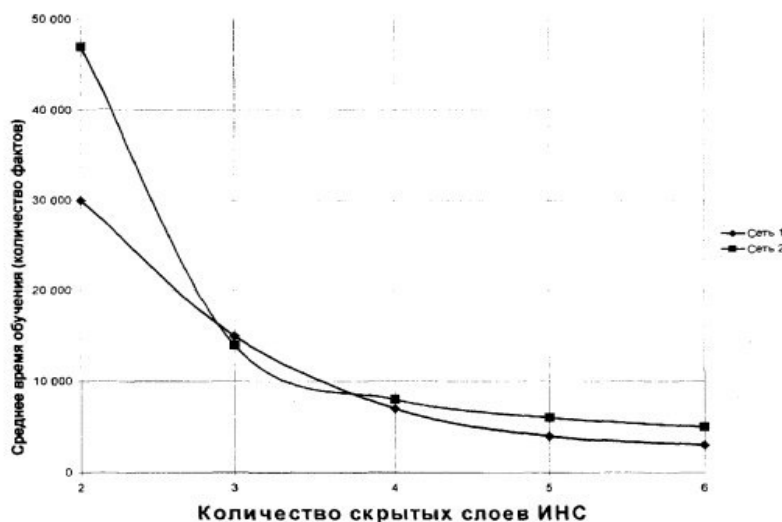


Рисунок 2.10 – Зависимость времени обучения нейросети от количества скрытых слоев

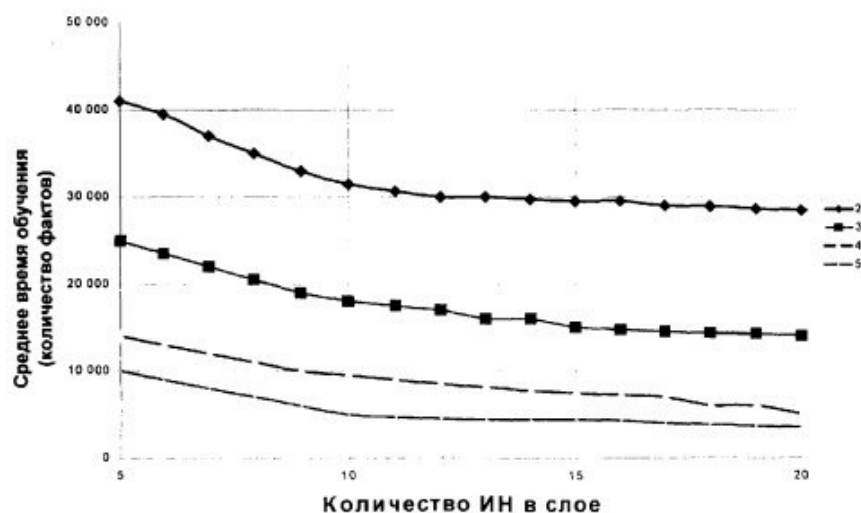


Рисунок 2.11 – Зависимость времени обучения нейросети от количества нейронов в скрытом слое

Зависимость адекватности модели и точности обученной нейронной сети от ее структуры носит значительно более сложный характер и существенно зависит от конкретной задачи [3]. В таблице 2.2 показана зависимость доли угаданных знаков временного ряда чисел Вольфа от структуры ИНС с одним выходным и одним скрытым слоем по результатам имитационного моделирования.

Таблица 2.2 – Зависимость доли угаданных знаков временного ряда чисел Вольфа от структуры ИНС

Размерность входов ИНС		4	6	8	10
Число нейронов в скрытом слое	3	57,3%	58,2%	62,7%	57,6%
	5	59,5%	61,0%	58,8%	63,3%
	7	57,4%	62,0%	59,2%	65,0%
	9	60,6%	61,7%	64,6%	65,0%

2.4 Повышение эффективности прогнозирующей нейросетевой модели с помощью адаптивного распределения плотности вероятности выбора обучающего факта

При прогнозировании различных временных рядов, непосредственно, можно отметить, что точность прогнозов обученной нейросети и, в особенности, время обучения в значительной степени зависят от объема обучающей выборки.

Для описанной в источнике [3] модели прогнозирования в режиме скользящего окна, объемом обучающей выборки является размерности входов нейросети (разность размера скользящего окна и глубины погружения).

Перечислим последствия использования большего объема выборки. Выбор большого объема помогает увеличить адекватность модели, но от увеличения объема растет и время обучения нейросети. Вытекающие последствия из всего этого заключаются в том, что прогнозирующие свойства модели выборки так же ухудшаются.

Подытожив, можно сказать с уверенностью, что явной функциональной или близкой к функциональной зависимости прогнозирующих свойств и времени обучения нейросети от объема выборки не выявлено.

Плотность вероятности выбора обучающего факта является плотностью вероятности равномерного распределения, проще говоря, в классических вариантах обучающих алгоритмов нейросетей выбор отдельных фактов обучающей выборки осуществляется с равными вероятностями.

Однако, для модели скользящего окна [3], где каждый факт обучающей выборки соответствует тому или иному дискретному моменту времени наблюдения ряда, актуальность данных в качестве образцов для обучения нейросети различна. Для моментов времени, более близких к тому отсчету, для которого обученная нейросеть делает прогноз, соответствующие этим моментам обучающие факты более ценны, чем соответствующие более отдаленным отсчетам. Поскольку данные факты несут больше информации о процессах, порождающих изменения временного ряда. Так как природа и

характеристики изменений ВР постоянно меняются, то соответственно им должна меняться прогнозирующая модель. К параметрам прогнозирующей модели, в случае нейросетевого прогнозирования, относятся весовые коэффициенты нейросети, а также ее архитектура.

Делая вывод из вышеперечисленного, необходимо делать выбор в сторону более близких к текущему моменту времени фактов с большей вероятностью, чем более отдаленные.

В качестве простой и естественной модели следует принять экспоненциальное распределение номера обучающего факта:

$$f_i(x) = \lambda e^{-\lambda x} \quad (2.11)$$

Для получения номера обучающего факта следует перейти от непрерывно распределенной случайной величины к дискретной.

Кроме того, как для равномерного, так и для экспоненциального распределений, верно, следующее.

В классическом анализе временные ряды представляются как сумма двух составляющих: сигнал и шум (формула 1.1). В различных моделях эти составляющие выделяются по-разному.

Непосредственно рассмотрим, что представляют понятия: сигнал и шум. В общем случае сигнал характеризует некую «систематическую» составляющую временного ряда, которая порождается объективными причинами, которые в свою очередь зависят, в общем-то, от поведения временного ряда. Шум – это случайная составляющая, демонстрирующая влияние таких факторов, как трудно учитываемых или несущественных.

Соотношение сигнал/шум $f(T) / u_T$ различно для разных временных рядов. Можно привести следующий пример для рассмотрения, для ряда чисел солнечной активности, о котором говорилось чуть выше, зашумленность ряда значительно меньше, чем для большинства финансовых ВР. С помощью данных из таблиц 2.1, 2.3 можно увидеть данную разницу соотношений и более высокую точность прогнозирования первого ряда (ряд чисел солнечной активности).

Таблица 2.3 – Сравнение прогнозов

Характеристика	Модель авторегрессии	Нейронная сеть, равномерный выбор фактов	Нейронная сеть, экспоненциальный выбор фактов
Число прогнозируемых отсчетов	47	47	64
Число угаданных знаков приращения ряда	24	26	37
Процент угаданных знаков	51,1%	55,3%	57,8%
Среднеквадратическая ошибка прогноза	2693,25	1878,74	1465,82

Соотношение сигнал/шум различается для отдельных моментов отсчета одного и того же ряда. Обучающие факты в моменты времени с наибольшим соотношением $f(T) / u_T$ несут больше информации о природе временного ряда и порождающих его процессах.

Применим следующий алгоритм для реализации всего вышеизложенного.

В качестве простейшей модели временного ряда дана наиболее распространенная модель авторегрессии: модель скользящего среднего [1]. Внесем конкретику: использованную методику можно применять и для других, более сложных моделей.

Для отдельного обучающего факта $\langle X_i, Y_i \rangle$, где i – номер факта в обучающей выборке, вероятность его выбора при каждом цикле обучения нейросети равна P_i .

Для равномерно распределенной вероятности выбора факта

$$P_i = \frac{1}{n} \forall i. \quad (2.12)$$

Для экспоненциально распределенной вероятности выбора

$$P_i = \int_{i-1}^i \lambda e^{-\lambda x} dx = e^{-\lambda i} (e^\lambda - 1). \quad (2.13)$$

Функционирование алгоритма для одной итерации цикла обучения нейронной сети выглядит следующим образом.

1) Для всех обучающих фактов вычисляется величина k_i , характеризующая отношение сигнал/шум для данного обучающего факта:

$$k_i = \frac{f(T)}{|u_T|}, f(T) = \frac{1}{d} \sum_{j=0}^{d-1} x_{T-j}. \quad (2.14)$$

2) По всем обучающим фактам находятся максимальное и минимальное значения величины k_i . Обозначим их как max и min соответственно.

Для всех i вычисляется коэффициент:

$$r_i = \frac{max - k_i}{max - min}. \quad (2.15)$$

3) Вычисляется величина:

$$Q_i = r_i P_i. \quad (2.16)$$

4) Вычисляется адаптивная вероятность выбора обучающих фактов:

$$P'_i = \frac{Q_i}{\sum_{j=1}^n Q_j}. \quad (2.17)$$

5) Осуществляется случайный выбор факта из обучающей выборки (скользящего окна) с вероятностью P'_i . Подобная задача давно известна в практике программирования и решается с помощью разбиения интервала $[0,1]$, в котором лежат значения псевдослучайного числа, полученного от генератора случайных чисел, на неравные интервалы, длина которых соответствует вероятности соответствующих им событий [20].

На рисунке 2.13 показан прогноз ряда котировок фьючерсов РАО Газпром на фондовой бирже РТС (контракт GAZR.-3.03) за период с 16 сентября 2017 г. по 14 февраля 2018 г., полученный с помощью описанной выше методики.

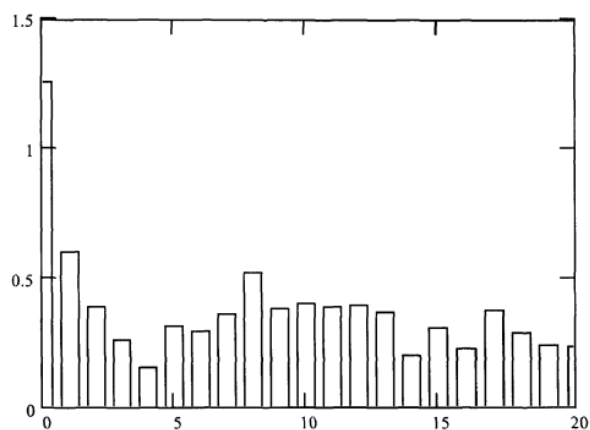


Рисунок 2.12 – Пример формы распределения плотности вероятности выбора номера обучающего факта (адаптивная модель)

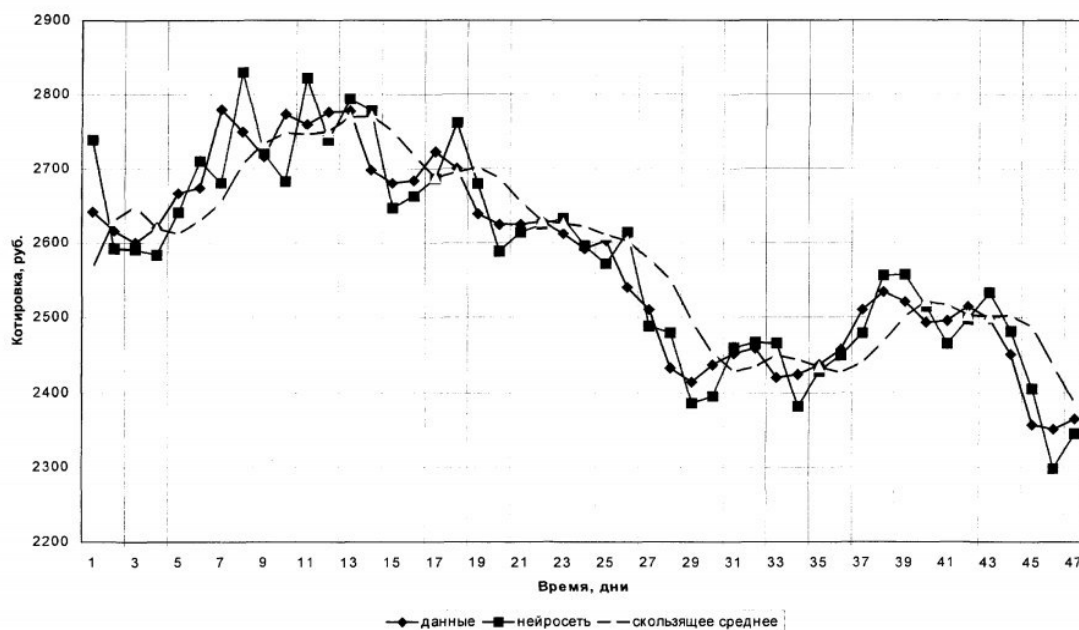


Рисунок 2.13. – Прогноз временного ряда котировок фьючерса на акции РАО Газпром

Результаты проведенного прогнозирования показали, что возрастает как эффективность прогнозирования (точность угаданных знаков), так и время обучения сети.

2.5 Нейросетевое прогнозирование временных рядов на основе многофакторной модели

Выделяют два основных недостатка нейронных сетей. Первое над чем ученые ломают голову – это как повысить потенциально низкую скорость обучения при большом количестве факторов. И второй недостаток – трудность выбора пространства входов и наиболее эффективной для решаемой задачи структуры сети.

Перечисленные факторы взаимно дополняют друг друга и определяют трудность или невозможность выбора оптимального вектора входов методом полного перебора.

В общем-то говоря, задача, где необходимо найти оптимальные вектора входов ИНС относится к классу NP-полных проблем. Лучшим решением для этого класса задач, собственно говоря, является полный перебор всех возможных вариантов [25]. Однако, такой выход решения является не целесообразным, с тем учетом, что каждый шаг перебора связан с полным обучением нейронной сети соответствующей структуры.

Таким образом, вопрос достижения максимально возможной точности прогнозирования ИНС для многофакторной модели фактически сводится к правильному выбору пространства входов нейросети, а вопрос скорости обучения – к понижению размерности пространства входов.

Перейдем к особенности выбора вектора входов нейронной сети для того, чтобы построить оптимально прогнозирующую модель.

Перечислим методы, которые широко используются для понижения размерности входов нейронных сетей:

- 1) использование в качестве возможных входов различных функций авторегрессии и индикаторов технического анализа (для финансовых временных рядов);
- 2) вейвлетное представление;
- 3) метод box-counting, реализованный в пакете ExcelNeuralPackage;
- 4) понижение размерности входов на основе корреляционного и дисперсионного анализа;
- 5) метод искусственных примеров;
- 6) использование генетических алгоритмов.

Одной из проблем при обучении нейросети на имеющихся статистических данных является их значительная зашумленность. Применение различных функций авторегрессии в качестве входов сети является одним из способов повышения эффективности нейросетевого прогнозирования. Чаще всего, как более универсальную функцию авторегрессии, используют скользящее среднее [1].

В качестве функционала ошибки используется правильность угадывания знака изменения временного ряда, что для финансовых временных рядов имеет больший смысл, чем классическая среднеквадратичная ошибка [3]. Иными словами: для финансовых временных рядов среднеквадратичная ошибка не имеет большого практического смысла. А вот оценивание точности угаданных знаков изменения ряда считается более корректным показателем. При использовании в качестве входов сети только предыстории временного ряда котировок не удаётся достигать точности угадывания знака, такой результат сравним со случайным совпадением угаданного и действительного знака изменения ряда.

Использование в качестве входов функций авторегрессии скользящее среднее или, в более сложном случае, экспоненциальное скользящее среднее является наиболее простым методом снижения влияния зашумленности исходных данных.

3 Методы повышения эффективности нейросетевого прогнозирования временных рядов

3.1 Общие замечания

Основанные на искусственных нейронных сетях, прогнозирующие модели, обладают большой эффективностью, но несмотря на это есть ряд мало значительных недостатков. Самый важный из которых – это трудность выбора для многофакторной модели пространства входов нейронной сети.

Так же нельзя обойти стороной недостаток, который заключается во времени обучения нейронной сети, возрастающим геометрически с ростом сложности модели.

Адаптация ИНС для конкретного применения, непосредственно, включает в себя следующие основные подзадачи:

- 1) выбор входного пространства признаков ИНС;
- 2) выбор структуры ИНС;
- 3) обучение ИНС (настройка весов).

Настройка весов решается с помощью какого-либо из классических алгоритмов обучения (например, процедуры обратного распространения ошибки) путем предъявления сети обучающей выборки.

Выбор оптимального пространства входов и структуры ИНС, обычно предшествующие обучению нейросети, являются достаточно сложными задачами, не имеющими однозначного решения. Тем не менее, в настоящее время большинство нейросетевых пакетов включают аналитический блок, позволяющий частично автоматизировать их решение [9].

В частности, при выборе входного пространства признаков (вектора входов ИНС) требуется максимальное уменьшение его размерности без потери информативности. Эта задача решается в различных пакетах с помощью таких методов, как вейвлетное представление, метод box-counting и др. [31]

3.2 Применение генетических алгоритмов для оптимизации прогнозирующих нейронных сетей

Впервые генетические, или эволюционные, алгоритмы были предложены в 1976 г. в работе американского ученого Дж. Холланда. В теоретической основе алгоритмов лежит модель эволюции живой природы.

Отличие генетического алгоритма (ГА), от других задач многофакторной оптимизации, заключается не в отдельном решении задачи, а набором решений, называемым популяцией. В процессе функционирования ГА популяция переживает некоторое количество поколений (итераций ГА), и, подобно биологическому прототипу (живой природе), средняя приспособленность популяции, возрастает в каждом последующем поколении.

Существует множество различных проведений в жизнь генетических алгоритмов. Термин ГА определяет целый класс методов, которые различаются между собой. Для функционирования любого ГА должны быть определены, во-первых, функция приспособленности особи и всей популяции в целом, во-вторых, две операции: кроссинговер и мутация.

Генетические алгоритмы используются для решения некоторых из указанных выше подзадач: обучения нейронных сетей и выбора пространства входов. Сегодня ГА находят широкое применение для решения большинства задач оптимизации.

С помощью набора входных параметров в виде битовой карты и их отсеивание с помощью генетического алгоритма можно уменьшить количество входов. В пакете Statistica Neural Networks [9] используется подобный подход.

Для удачного применения нейронной сети в работе предлагается генетический алгоритм, который содержит в себе решение трех указанных выше подзадач. Как одна из самых распространенных и универсальных архитектур была выбрана многослойная сеть без обратных связей, в качестве структуры ИНС, обрабатываемых с помощью алгоритма. Многослойная сеть

без обратных связей наиболее успешно используемая как базовая архитектура прогнозирующих нейронных сетей.

Предлагаемый алгоритм решает задачу оптимизации. Управляемыми параметрами, которого являются пространство входов ИНС, структура (количество слоев и нейронов в слоях) ИНС и веса нейронов, а целью – минимизация суммарной функции ошибки популяции сетей.

В отличие от традиционных генетических алгоритмов обучения нейронных сетей, предлагаемый алгоритм оптимизации управляет всей совокупностью параметров ИНС (входы, структура, веса нейронов). В данном алгоритме управляемыми параметрами являются только веса нейронов, или алгоритмов выбора входов сети, которые, напротив, не осуществляют непосредственно обучение сети.

Преимуществом такого подхода является комплексное, а не поэтапное решение всех задач. Эти задачи необходимы для конкретной адаптации ИНС, что позволяет учесть их взаимное влияние и предоставляет большее удобство применения.

3.2.1 Базовые понятия генетического алгоритма

Для функционирования ГА необходимо определить способ представления управляемых параметров в хромосомах и, как минимум, две операции: кроссинговер и мутация.

Разберем как происходит кодирование признаков особи в генотипе. При использовании генетических алгоритмов для обучения нейронных сетей с варьирующейся структурой одной из проблем является эффективное представление структуры ИНС в генотипе.

В предлагаемой модели каждому слою нейросети соответствует одна хромосома, соответствующая матрице весов нейронов этого слоя:

$$W_{m_L \times m_{L-1}}^{(L)} = \begin{pmatrix} w_{1,1}^{(L)} & \cdots & w_{1,n_L}^{(L)} \\ \vdots & \ddots & \vdots \\ w_{m_L,1}^{(L)} & \cdots & w_{m_L,n_L}^{(L)} \end{pmatrix} = \begin{pmatrix} w_{1,1}^{(L)} & \cdots & w_{1,n_{L-1}}^{(L)} \\ \vdots & \ddots & \vdots \\ w_{m_L,1}^{(L)} & \cdots & w_{m_L,n_{L-1}}^{(L)} \end{pmatrix}, \quad (3.1)$$

где L – номер слоя;

m_L – число нейронов в L -м слое;

n_L – число входов L -го слоя; для скрытых слоев (кроме входного) $n_L = m_{L-1}$;

$w_{i,j}$ – двоичное представление веса ω_{ij} , формируется путем дискретизации вещественного значения ω_{ij} с заданной точностью и последующего его представления с помощью кода Грея.

Чем же характерен код Грея? В отличие от обычного двоичного представления десятичного числа, расстояние Хемминга для любых двух чисел, представленных данным кодом, равно математической разнице этих чисел.

Следовательно, в отличие от традиционных ГА, в предлагаемой модели генотип состоит не из одной хромосомы, а из хромосомного набора. Что очень отвечает биологическому прототипу).

Если алгоритм используется также для подбора оптимального вектора входных параметров, то в хромосомный набор включается еще хромосома $X=W^{(0)}$, представляющая собой битовую карту, в которой 1 означает присутствие признака в пространстве входов ИНС, 0 – его отсутствие.

Можно представить хромосому X в виде матрицы размерности $p \times 1$, где p – общее число возможных входных признаков, которые предположительно могут влиять на значения выходов. Элементами матрицы являются двоичные разряды.

Общий хромосомный набор особи:

$$\{W^{(L)}\} = W^{(0)}, W^{(1)}, W^{(2)}, \dots, W^{(S)}, \quad (3.2)$$

где S – число слоев ИНС.

ИНС с различной структурой имеют различное количество и размер хромосом.

Для каждой особи в популяции вычисляется суммарная ошибка E нейронной сети, характеризующая степень приспособленности особи. Способ вычисления E зависит от конкретного применения ИНС.

Кроссинговер в общем случае для хромосом произвольной длины и разного количества хромосом в двух наборах определяется следующим образом:

$$\begin{aligned}
& cross(\{W_{m_L, m_{L-1}}^{(L)}\}, \{Q_{n_L, n_{L-1}}^{(L)}\}) = \\
& = cross\left(\left\{\left(\begin{array}{ccc} w_{1,1}^{(L)} & \cdots & w_{1, m_{L-1}}^{(L)} \\ \vdots & \ddots & \vdots \\ w_{m_L, 1}^{(L)} & \cdots & w_{m_L, m_{L-1}}^{(L)} \end{array}\right)\right\}, \left\{\left(\begin{array}{ccc} q_{1,1}^{(L)} & \cdots & q_{1, n_{L-1}}^{(L)} \\ \vdots & \ddots & \vdots \\ q_{n_L, 1}^{(L)} & \cdots & q_{n_L, n_{L-1}}^{(L)} \end{array}\right)\right\}\right) = \\
& = \left\langle \left\{\left(\begin{array}{cccc} w_{1,1}^{(L)} & \cdots & w_{1,j}^{(L)} & q_{1,j+1}^{(L)} & \cdots & q_{1, n_{L-1}}^{(L)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ w_{i,1}^{(L)} & \cdots & w_{i,j}^{(L)} & q_{i,j+1}^{(L)} & \cdots & q_{i, n_{L-1}}^{(L)} \\ q_{i+1,1}^{(L)} & \cdots & q_{i+1,j}^{(L)} & q_{i+1,j+1}^{(L)} & \cdots & q_{i+1, n_{L-1}}^{(L)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ q_{n_L, 1}^{(L)} & \cdots & q_{n_L, j}^{(L)} & q_{n_L, j+1}^{(L)} & \cdots & q_{n_L, n_{L-1}}^{(L)} \end{array}\right)\right\}, \left\{\left(\begin{array}{cccc} \cdots & \cdots & \cdots & \cdots \\ \cdots & q_{i,j}^{(L)} & w_{i,j+1}^{(L)} & \cdots \\ \cdots & w_{i+1,j}^{(L)} & w_{i+1,j+1}^{(L)} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{array}\right)\right\} \right\rangle = \quad (3.3) \\
& = \langle \{W_{n_L, n_{L-1}}^{(L)}\}', \{Q_{m_L, m_{L-1}}^{(L)}\}' \rangle
\end{aligned}$$

Значения i и j определяют точку разрыва и выбираются случайно. В частном случае, когда структура всех ИНС в популяции одинакова и неизменна ($m_L = n_L \forall L$), можно принять $j^{(L)} = m_{L-1} = n_{L-1}$.

В отдельных случаях (прежде всего для моделей с переменным пространством входов) также можно принять $j^{(L)} = i^{(L-1)}$.

Особь $\{W\}$ и $\{Q\}$ выбираются с помощью одного из традиционных методов: рулетка, турнирный отбор, стратегия элитизма.

После выполнения кроссинговера в популяцию добавляются особи $\{W\}'$ и $\{Q\}'$.

Теперь рассмотрим три типа мутаций в предлагаемой модели:

1) «Обычная» мутация. Затрагивает только один ген или группу генов, не меняя количество хромосом и состав генов. Совершается для каждой особи популяции с вероятностью $P_{ом}$. Состоит в инвертировании одного или нескольких битов хромосомы, выбираемых случайным образом (равномерное распределение). Мутации в хромосоме можно отнести к «обычным», т.к. состоят только в инвертировании битов. Всё же такая мутация должна обязательно сопровождаться изменением размерности столбцов матрицы первого слоя.

2) Мутация в количестве генов в хромосоме (изменение длины хромосомы). Отвечает изменению числа нейронов в слое без изменения количества слоев. Совершается для каждой особи с вероятностью $P_{\text{ГМ}}$. Состоит в изменении числа строк хромосомной матрицы. Изменение числа нейронов в i -м слое, непосредственно, влечет за собой соответствующее изменение числа столбцов матрицы весов $(i+1)$ -го слоя. Следовательно, изменение длины затрагивает обязательно две хромосомы. Если число нейронов увеличивается, новые строки приписываются к матрице снизу, веса вновь добавляемых нейронов рандомизируются. Если число нейронов уменьшается, то номера удаляемых нейронов выбираются случайно (равномерное распределение).

3) Мутация в количестве хромосом. Соответствует изменению структуры слоев сети. Совершается для каждой особи с вероятностью $P_{\text{ХМ}}$. Эту мутацию можно представить как дублирование одной из хромосом + мутация в количестве генов в этой хромосоме.

Очевидно, что необходимо принять $P_{\text{ХМ}} \ll P_{\text{ГМ}} \ll P_{\text{ОМ}}$.

Назовем мутации вида 2 и 3 структурными. Структурная мутация вызывает увеличение или уменьшение числа генов или даже хромосом.

Обозначим изменение числа генов (хромосом) как случайную величину Δ . Естественным будет принять предположение о нормальном распределении Δ с математическим ожиданием, равным 0. Однако, поскольку Δ принимает дискретные значения, рассмотрим распределение непрерывной случайной величины δ .

$$f_{\delta}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}, \quad (3.4)$$

значения которой соответствуют значениям Δ следующим образом:

$$N = \{\Delta = n\} = \left\{ n - \frac{1}{2} < \delta \leq n + \frac{1}{2} \right\}, \quad (3.5)$$

где N – событие, состоящее в том, что величина Δ принимает конкретное значение n из множества целых чисел;

Z – множество целых чисел.

Заметим, что в более поздних поколениях, вероятность выживания особи со структурной мутацией крайне мала, когда общая приспособленность популяции достаточно высока. Из-за того, что веса нейронов требуют корректировки под новую структуру сети и суммарная ошибка E достаточно велика. В то время как в начальных поколениях, когда веса нейронов еще недостаточно обучены, варьирование структуры сети должно быть очень эффективным.

Ввиду вышесказанного, $P_{\text{хм}}$ и $P_{\text{гм}}$ должны постепенно убывать с увеличением приспособленности популяции.

В данной модели предлагается принять вероятности мутаций, изменяющиеся по закону:

$$P_M(t) = \frac{1}{1 + e^{b(t-\tau)}}, \quad (3.6)$$

где t – номер поколения;

b – характеризует плавность убывания $P(t)$;

τ – номер поколения, при котором $P(\tau)=0.5$.

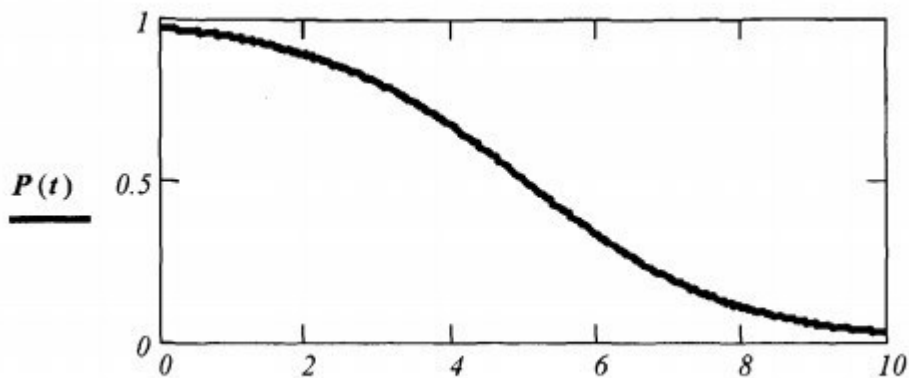


Рисунок 3.1 – Зависимость вероятности мутации особи от числа поколений

Можно заметить, что данная функция является зеркальным отображением сигмоидальной функции, часто применяемой в качестве активационной функции нейронов [25].

Вероятность мутации хотя бы в одной особи популяции выглядит так:

$$P_{\text{Мобщ}} = 1 - (1 - P_M)^N = 1 - (1 + e^{b(\tau-t)})^{-N}. \quad (3.7)$$

Событие «мутация» можно определить следующим образом:

$$M = \{\Delta \neq 0\}. \quad (3.8)$$

Тогда

$$\begin{aligned} P_M(t) &= P\{\Delta \neq 0\} = 1 - P\{\Delta = 0\} = 1 - P\left\{-\frac{1}{2} < \delta \leq \frac{1}{2}\right\} = \\ &= 1 - \frac{1}{\sigma\sqrt{2\pi}} \int_{-0.5}^{0.5} e^{-\frac{x^2}{2\sigma^2}} dx = 1 - 2\Phi\left(\frac{1}{2\sigma}\right). \end{aligned} \quad (3.9)$$

В результате получаем уравнение:

$$\frac{1}{1+e^{b(t-\tau)}} = 1 - 2\Phi\left(\frac{1}{2\sigma}\right), \quad (3.10)$$

из которого можно найти зависимость σ от t .

После разложения $f_\delta(x)$ в ряд Тейлора до второго члена и интегрирования получаем:

$$1 - \frac{1}{\sigma\sqrt{2\pi}} = \frac{1}{1+e^{b(t-\tau)}}, \quad (3.11)$$

откуда

$$\sigma(t) = \frac{1+e^{b(t-\tau)}}{\sqrt{2\pi}e^{b(t-\tau)}} = \frac{1}{\sqrt{2\pi}}(1 + e^{b(t-\tau)}). \quad (3.12)$$

На рисунке 3.2 показано, как меняется плотность распределения Δ при уменьшении со временем вероятности мутации $P(t)$ (штриховая линия).

Можно применять также модификацию алгоритма с вероятностью «обычной» мутации, также меняющейся в соответствии с законом (3.6). В этом случае Δ будет соответствовать количеству инвертируемых битов.

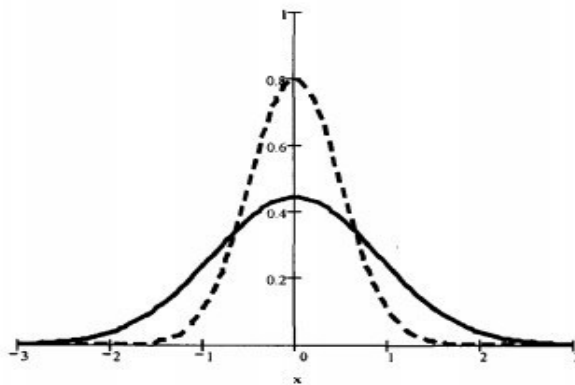


Рисунок 3.2 – Изменение формы плотности распределения Δ в зависимости от числа поколений

Зависимость $P(t)$, $\sigma(t)$ и вероятности мутации хотя бы в одной особи популяции $Q(t)$ показана на рисунке 3.3.

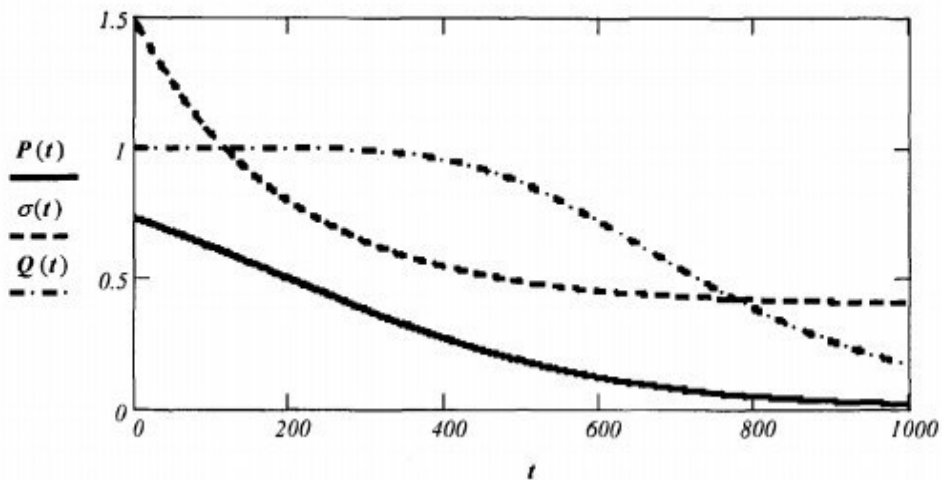


Рисунок 3.3 – Зависимость параметров распределения A от числа поколений

Примеры выбора вероятностей мутаций $P_{ом}$, $P_{гм}$, $P_{хм}$ для различных реализаций ГА показаны на рисунке 3.4.

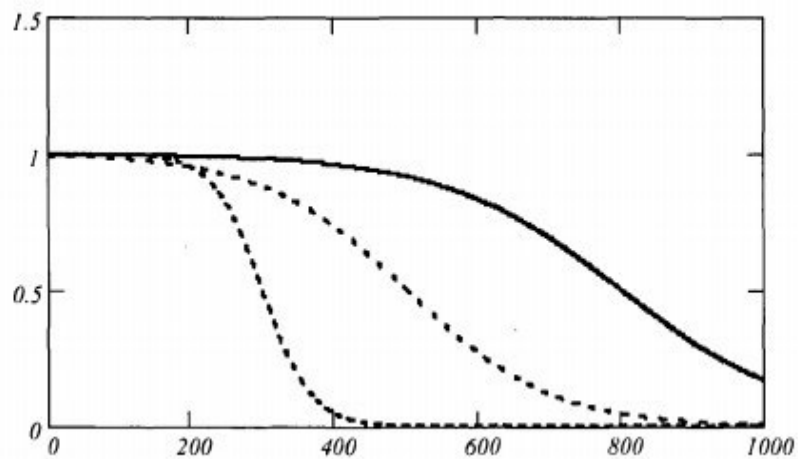


Рисунок 3.4 – Примеры зависимости вероятности мутации от числа поколений

3.2.2 Функционирование генетического алгоритма

Начальная популяция состоит из N нейронных сетей. Структура и набор входов для каждой нейросети в начальной популяции определяются случайно. Число скрытых слоев сети и количество нейронов в каждом слое являются

нормально распределенными случайными величинами, математическое ожидание и дисперсия которых являются параметрами алгоритма. Веса всех нейросетей популяции рандомизируются.

На каждом шаге алгоритма выполняются следующие действия:

1) Осуществляются мутации с вероятностями $P_{ом}$, $P_{гм}$, $P_{хм}$ для каждой особи. Мутация состоит в изменении генотипа, что отражается как варьирование структуры сети, набора входов или весов нейронов.

2) Осуществляется кроссинговер с вероятностью $P_{кр}$.

3) Проводится несколько циклов обучения нейросетей, входящих в популяцию, с помощью какого-либо из известных алгоритмов обучения. Назовем этот алгоритм вспомогательным. Использование вспомогательного алгоритма необходимо потому, что после структурной мутации необходима некоторая адаптация видоизмененной особи (жизненный цикл). В первую очередь это необходимо в начальных поколениях популяции, когда велика вероятность мутации. Когда вероятность мутаций становится чрезвычайно мала, в более поздних поколениях, оптимизация происходит в значительной степени за счет вспомогательного алгоритма. Количество проводимых между сменами поколений эпох обучения является параметром алгоритма.

4) Из популяции удаляются наименее приспособленные особи. Если применяется модификация алгоритма с постоянным размером популяции, удаляется две наименее приспособленные особи (ИНС с наибольшим значением суммарной ошибки E) только после выполнения кроссинговера.

Критерием остановки алгоритма служит достижение заданной суммарной ошибки популяции $\sum_{k=1}^N E_k \leq E_{min}$, либо достижение максимального заданного числа поколений t_{max} .

Так, для начала функционирования алгоритма необходимо задать следующие параметры: N - размер популяции; E_{min} - максимально допустимая суммарная ошибка; t_{max} - максимальное число поколений; $P_{кр}$ - вероятность кроссинговера; b , τ - параметры законов изменения вероятностей трех возможных типов мутаций; множество предполагаемых входных признаков;

4 Программные средства автоматизированного прогнозирования временных рядов на основе нейрокомпьютерных технологий

Существует множество различных нейросетевых библиотек, но для выполнения данного исследования была предложена библиотека, которая ориентирована на оптимизацию объектного кода и более высокую скорость функционирования. Перечисленные факторы, собственно говоря, являются чрезвычайно важными при обучении искусственных нейронных сетей.

Библиотека классов, которая описана ниже, состоит, непосредственно, из трех основных частей (каждой части соответствует своя иерархия классов):

1) классы, реализующие обучение и функционирование различных нейросетевых парадигм;

2) классы, реализующие различные приложения ИНС, такие как прогнозирование, распознавание или классификация;

3) классы, отражающие специфику применения прогнозирующих ИНС.

В основе каждого класса лежит базовый абстрактный класс, который с помощью общих интерфейсных элементов реализует свой.

В первую часть библиотеки классов входят классы, которые, так сказать, осуществляют алгоритмы функционирования различных нейросетевых парадигм, не учитывая их область применения.

Базовым классом этой подыерархии классов является абстрактный класс CNeuroNet. Он реализует понятие «нейронная сеть» и базовые функции, обеспечивающие, непосредственно, функционирование нейронной сети, а именно: возбуждение ИНС в режиме нормального функционирования, обучение («с учителем» и «без учителя») и работу вспомогательных функций.

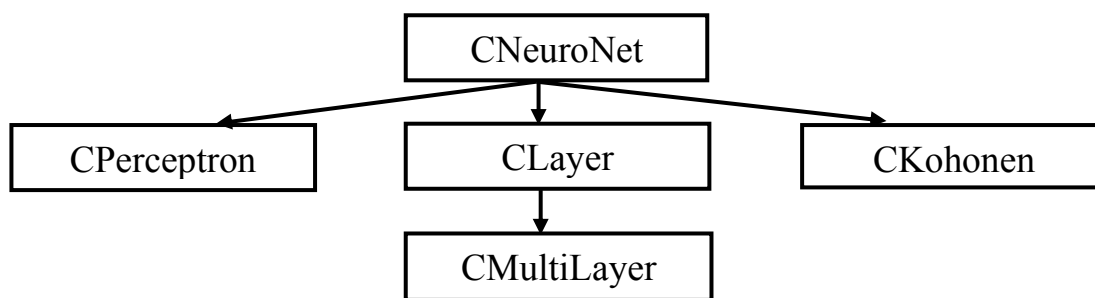


Рисунок 4.1 – Базовая иерархия классов, реализующих понятие «нейронная сеть» и различные нейросетевые парадигмы

Некоторые функции класса являются общими для всех нейросетевых парадигм (например, «установить/возвратить число входов нейросети») и реализованы непосредственно в базовом классе CNeuroNet. Многие другие функции, являющиеся общими для многих парадигм, сильно отличаются, когда их необходимо реализовать для различных архитектур и алгоритмов обучения. Данные функции в базовом классе являются чисто виртуальными, и их реализация возложена на разработчика производных классов.

Функции, реализующие интерфейс класса CNeuroNet и объявленные как общие, кратко описаны в таблице 4.1.

Таблица 4.1 – Функции, реализующие интерфейс класса CNeuroNet

Функция	Описание функции
virtual const double* Excite(double* input) = 0;	Нормальное функционирование нейронной сети, input – входной вектор, возвращает выходной вектор..
virtual void Train(double* fact) = 0;	Возбуждение нейронной сети с обучением (подстройкой весов), fact – обучающий факт, может состоять из пары входного и выходного векторов (обучение «с учителем») или одного входного вектора (обучение «без учителя»). Перегружаемая функция.
virtual void Train(double* input, double* desired) {Train(input);};	Возбуждение нейронной сети с обучением, input, desired – входной и желаемый выходной векторы (обучение «с учителем»). Перегружаемая функция.

Продолжение таблицы 4.1

Функция	Описание функции
<code>virtual double GetRes(UINT i=0) {return res[i];};</code>	Возвращает выходной вектор нейронной сети. Необходимо использовать после <code>Excite()</code> или <code>Train()</code>
<code>UINT GetNI() {return nInp;};</code>	Возвращает число входов нейронной сети.
<code>virtual void SetNI(UINT nInput) {nInp = nInput;};</code>	Устанавливает число входов нейронной сети.
<code>virtual UINT GetNO() = 0;</code>	Возвращает число выходов нейронной сети.
<code>virtual double Activate(double sum) = 0;</code>	Активационная функция нейронной сети. Чисто виртуальная функция. Переопределяется в производных классах.
<code>virtual double Derivee(double y) = 0;</code>	Производная активационной функции по выходу. Чисто виртуальная функция. Переопределяется в производных классах.
<code>virtual void Randomize(double a=-0.5, double b=0.5) = 0;</code>	Рандомизирует (присваивает начальные значения) все веса нейронной сети. Чисто виртуальная функция.

От класса `CNeuroNet` происходят производные классы. Эти в свою очередь реализуют различные нейросетевые парадигмы: перцептронную сеть (`CPerceptron`), сеть прямого распространения ошибки (`CLayer`, `CMultiLayer`), самоорганизующиеся карты Кохонена (`CKohonen`).

За счет добавления новых классов библиотека может расширяться. Эти классы производные от `CNeuroNet`, реализуют иные нейросетевые положения, например, такие как машина Больцмана, звезды Гроссберга, сеть Хопфилда [25] и др.

При работе с ИНС необходимо иметь вспомогательные классы, так как одна и та же нейросетевая парадигма может использоваться для решения различных классов задач, позволяющие отвлечься от внутренней архитектуры и алгоритма обучения ИНС, заостря внимание на целевом применении ИНС. Эта часть спроектированной иерархии изображена на рисунке 4.2.

Базовым классом этой подиерархии служит виртуальный класс `CNeuroTask`, функции которого описаны в таблице 4.2.

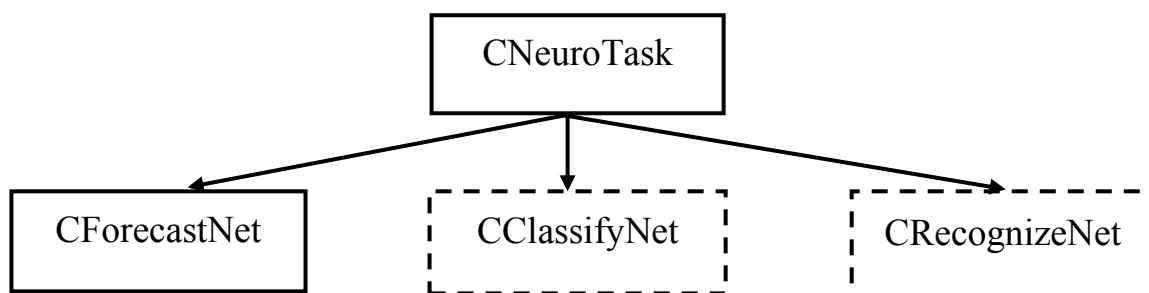


Рисунок 4.2 – Иерархия классов C++, реализующая приложения нейронных сетей для различных классов задач

Таблица 4.2 – Функции, реализующие интерфейс класса CNeuroTask

Функция	Описание функции
CNeuroTask();	Устанавливает начальные значения защищенным членам класса.
virtual ~CNeuroTask();	Деструктор по умолчанию. Виртуальная функция.
int StartTrain();	Запуск обучения нейронной сети в отдельном потоке Windows.
void StopTrain();	Остановка обучения нейронной сети.
void PauseTrain();	Приостановка обучения нейронной сети на некоторое время (до очередного вызова StartTrain()).
virtual void OperateFact(ULONG nFact) = 0;	Предъявление сети отдельного обучающего факта с номером nFact из выборки. Чисто виртуальная функция.
virtual void TrainRandomFact() = 0;	Предъявление сети случайного обучающего факта из выборки.
virtual void ValidateNet(double &max, double &avg, double &good) = 0;	Оценка степени обученности ИНС (max – макс. квадратичная ошибка по выборке, avg – средняя квадратичная ошибка, good – относительное количество распознанных фактов). Чисто виртуальная.
Virtual double GetRes(UINT i=0);	Получение результата функционирования сети, (нейроны обладают кратковременной памятью и хранят значения выходов до следующего возбуждения)
CNeuroNet* GetNet();	Возвращает указатель на объект класса, производного от CNeuroNet, используемый для решения задачи.
Virtual void SetNet(CNeuroNet* thenet);	Устанавливает внутренний указатель на объект класса CNeuroNet, который предполагается использовать для решения задачи.

Класс CNeuroTask содержит защищенный член CNeuroNet*net, являющийся указателем на конкретный объект одного из производных от CNeuroNet классов, представляющий необходимую для решения задачи CNeuroTask нейросетевую парадигму с заданными параметрами.

Функции StartTrain() – запускают, StopTrain() – останавливают, PauseTrain() – приостанавливают процесс обучения ИНС, выполняемый в отдельном потоке операционной системы Windows. Для создания данного потока используется вспомогательная глобальная функция UINT TrainFunction(LPVOID pParam), которая реализует для всех парадигм итерационный алгоритм обучения.

Способы выбора обучающих фактов из выборки реализуются путем переопределения виртуальных функций. и определения готовности сети к целевому использованию. Для каждой задачи это сугубо индивидуально. Данные функции описаны в таблице 4.2.

С помощью класса CForecastNet, порожденного от CNeuroTask строятся нейронные сети, предназначенные для прогнозирования временных рядов. В нем реализованы функции для обучения и функционирования сети, объявленные в базовом классе. Порожденные классы от CSlide, предоставляют исходные данные для реализации скользящего окна по временному ряду.

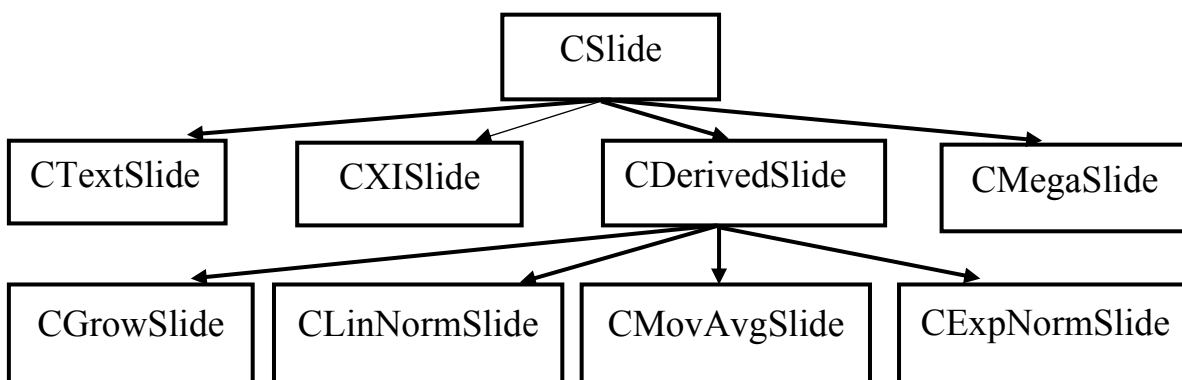


Рисунок 4.3 – Иерархия классов C++, обеспечивающая представление исходных данных временного ряда для прогнозирующих нейронных сетей

Класс CSlide (рисунок 4.3), являющийся базовым классом подиерархии, представляет концепцию скользящего окна. Объекты класса CSlide служат для

получения значений временного ряда по индексу кадра и индексу элемента в кадре. Кадр – это один обучающий факт для прогнозирующей нейронной сети, состоящий из значений временного ряда $x_{i-k}, x_{i-k-1}, \dots, x_{i-k-d}$, где i – положение скользящего окна, k – номер кадра в окне, d – размер кадра. Номера кадров и элементов в кадре отсчитываются от более поздних моментов времени.

От класса CSlide, непосредственно, порождаются классы, которые считывают значения временных рядов из различных источников данных: дисковой памяти ЭВМ, компьютерной сети, баз данных и т.п. Здесь рассматриваются классы CTextSlide, предназначенный для чтения данных ВР, записанных в текстовый файл через разделитель, и класс CXISlide, считывающий данные ВР из формата Microsoft Excel.

Временные ряды, строящиеся на основе других временных рядов моделируются с помощью класса CDerivedSlide, производного от CSlide. Класс CDerivedSlide является базовым для ряда классов:

- 1) CGrowSlide – ряд приращений $\Delta x_i = x_i - x_{i-1}$;
- 2) CMovAvgSlide – скользящее среднее базового ВР;
- 3) CLinNormSlide – линейно нормализованные значения базового ВР;
- 4) CExpNormSlide – экспоненциально нормализованные значения базового ВР.

Перечисленные классы имеют защищенный член-указатель на базовый слайд. Член-указатель представляет значения временного ряда, которые используются для выполнения соответствующих преобразований.

Классы, производные от CDerivedSlide, могут выстраиваться в цепочки любой сложности для выполнения необходимых преобразований с временным рядом.

Для моделирования многомерных временных рядов служит класс CMegaSlide, он также применим при многофакторной прогнозирующей модели.

Основные функции класса CSlide отображены в таблице 4.3.

Таблица 4.3 – Основные функции класса CSlide

Функция	Описание функции
CSlideQ;	Конструктор по умолчанию. Устанавливает начальные значения защищенным членам класса.
virtual ~CSlide();	Деструктор по умолчанию. Виртуальная функция.
virtual int Set(ULONG start, UINT length=0);	Устанавливает позицию скользящего окна и (возможно) размер окна. Виртуальная функция.
virtual int Move(ULONG step=1) = 0;	Перемещает позицию скользящего окна на step шагов. По умолчанию step=1. Чисто виртуальная функция.
virtual double GetValue(UINT frame, UINT i=0) {return data[frame+i];};	Возвращает значение, содержащееся в i-м элементе кадра с номером frame. Виртуальная функция.
virtual double GetValue(double x) {return x;}	Осуществляет преобразование (денормализацию) значения x к диапазону значений окна базового уровня.
virtual double* GetFrame(UINT i=0);	Возвращает кадр с номером i целиком в виде массива. Виртуальная функция.
virtual CSlide* GetBase() {return NULL;}	Возвращает указатель на базовое окно (если есть), в противном случае -- нулевой указатель. Виртуальная функция.
virtual UINT GetDisp() {return 0;}	Возвращает смещение относительно базового кадра. Виртуальная функция.
UINT GetFrameSize();	Возвращает размер кадра.
virtual int SetFrameSize(UINT framesize);	Устанавливает размер кадра равным framesize. Число кадров в окне приводится в соответствие с новым размером. Виртуальная функция.
virtual UINT GetNFrames();	Возвращает число кадров, содержащихся в окне. Виртуальная функция.
UINT GetSize() {return size;}	Возвращает размер окна (число значений временного ряда, содержащихся в нем).
virtual int SetSize(UINT length);	Устанавливает размер окна равным length. Виртуальная функция.
virtual int SetSize(UINT nframes, UINT framesize);	Устанавливает размер окна с заданием числа кадров nframes и размера кадра framesize. Виртуальная функция.

4.1 Программное обеспечение «TimeSeries»

Уже разработано множество специализированных, универсальных нейросетевых пакетов. Среди последних отметим такие пакеты как BrainMaker, NeuralBench, Statistica Neural Networks и др. Большинство из них представляют собой зарубежные разработки, стоимость которых достаточно высока, либо которые не широко доступны для простого обывателя. Но именно они имеют наибольшую эффективность для прогнозирования временных рядов, так как относятся к более специализированным нейросетевым пакетам.

Программное обеспечение «TimeSeries» предназначено для построения нейросетевых прогнозирующих моделей с использованием описанных в дипломной работе методик.


Для работы программного обеспечения требуется:

- IBM-совместимый компьютер на базе процессора Pentium с тактовой частотой не менее 200 МГц;
- объем оперативной памяти не менее 32 Мбайт;
- графический адаптер и монитор SVGA;
- мышь или другое указывающее устройство;
- объем памяти на жестком диске 1 Мбайт;
- операционная система Windows 7/8/10.

Основным принципом системного подхода, использованного при создании программного обеспечения «TimeSeries», является принцип модульного построения, при котором программная система рассматривается и проектируется в виде взаимосвязанной совокупности модулей, каждый из которых реализует решение соответствующей ему подзадачи.

Программное обеспечение «TimeSeries» разработано в среде Microsoft Visual C++ с использованием библиотеки Microsoft Foundation Classes.

При разработке пользовательского интерфейса использовались стандартные элементы графического интерфейса операционных систем Windows: диалоговые окна и средства управления.



Работа программы начинается после запуска на выполнение файла TimeSeries.exe. На экране открывается главное окно программы, изображенное на рисунке 4.4. В этом окне расположены основные интерфейсные элементы пакета программ. Для окончания работы с программой можно щелкнуть мышью на кнопке  в правом верхнем углу окна, выбрать пункт «Выход» меню «Файл».

Интерфейс основной части программы состоит из строки меню и рабочей области приложения. На рабочей области приложения расположены несколько закладок, каждая из которых открывает элементы интерфейса для ввода и контроля соответствующих ей параметров.

В верхней части основного окна расположена строка меню, содержащая универсальные команды для работы с программой. Активация меню выполняется нажатием курсора мыши на требуемый пункт. Отдельные слова в строке главного меню представляют собой пункты меню, которые разворачиваются при выборе их с помощью мыши. При открытии выпадающего меню можно выбрать любой его пунктов. Когда нажимается клавиша <Enter> или левая кнопка мыши, выбирается подсвеченный элемент.

При запуске программы по умолчанию открывается закладка «Временной ряд», (рисунок 4.4). Данная закладка служит для задания параметров скользящего окна, используемого в качестве источника данных, а также контроля за процессом обучения и функционирования прогнозирующей нейронной сети. На этой закладке можно задать параметры скользящего окна: размер окна в соответствующем поле ввода, способ выбора фактов (равномерный, экспоненциальный), параметр λ для экспоненциального распределения, тип модели (обычная, адаптивная).

Перемещение на определенный момент времени осуществляется либо вводом значения в поле «Шаг», либо последовательным нажатием кнопок

«Перемещение вперед»  и «Перемещение назад» . Программой осуществляется контроль правильности ввода номера шага (не меньше 1 и не больше общего числа шагов). В случае несоответствия этому условию выводится диалоговое окно с соответствующим сообщением.

Обучение нейронной сети и прогнозирование осуществляется с помощью кнопки «Выполнение», либо кнопки «Авто».

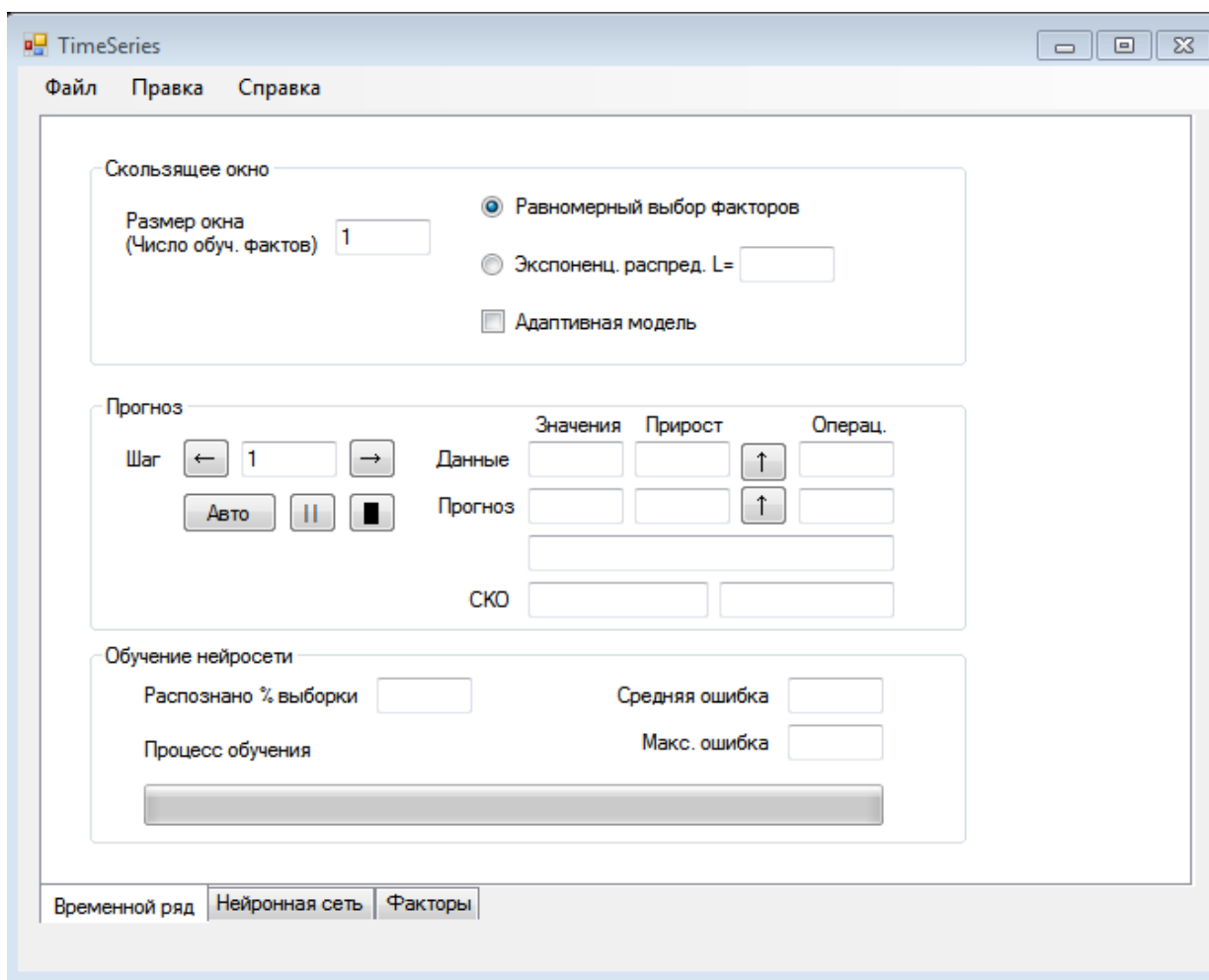




Рисунок 4.4 – Главное окно программы

В первом случае осуществляется обучение нейронной сети и прогнозирование на один шаг вперед. В случае автоматического прогнозирования после получения прогноза скользящее окно смещается на один шаг вперед и осуществляется дальнейшее обучение ИНС и прогнозирование, пока не будет исчерпан объем статистической выборки.

Процесс функционирования программы в режиме обучения нейросети показан на рисунке 4.5. Полоса прогресса «Процесс обучения» помогает

осуществлять визуальный контроль обучения. Степень заполненности полосы свидетельствует о готовности сети к непосредственному использованию.

С помощью кнопки «Остановка»  пользователь может досрочно остановить обучение нейронной сети или прекратить его на некоторое время с помощью кнопки «Пауза» .

Результат функционирования нейросети отображается в окне программы (рисунок 4.5) по завершению каждого шага обучения. В полях «Данные» и «Прогноз» выводится реальное (если есть) и спрогнозированное значения ВР (колонка «Значения»). Колонка «Прирост» отражает реальное и ожидаемое приращения временного ряда. Крайняя правая колонка цифр отражает «операционные» значения, т.е. те, которые были выданы нейронной сетью.

Выводится также среднеквадратическая ошибка, средняя ошибка в процентах и процент угаданных знаков в среднем по проведенному количеству прогнозов.

Поля «Распознано % выборки», «Средняя ошибка», «Макс, ошибка» отражают соответствующие показатели обученной нейронной сети (степень ее готовности к полному использованию).

Перед началом обучения ИНС необходимо задать ее параметры и параметры входного и выходного векторов. Это можно сделать с помощью интерфейсных элементов, расположенных на закладках «Нейронная сеть», «Факторы».

Закладка «Нейронная сеть» (рисунок 4.6) позволяет пользователю задавать параметры архитектуры нейронной сети и алгоритма ее обучения.

Программный пакет реализует многослойную нейронную сеть прямого распространения, обучаемую с помощью модифицированного алгоритма обратного распространения ошибки.

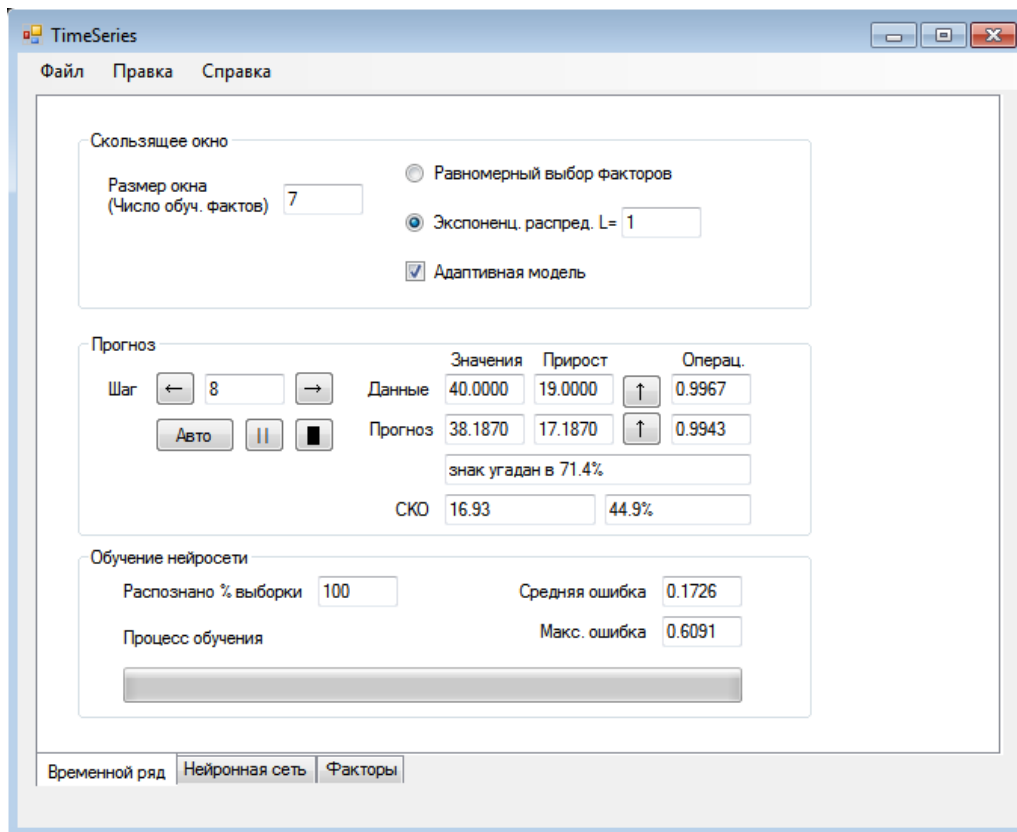


Рисунок 4.5 – Процесс функционирования программы

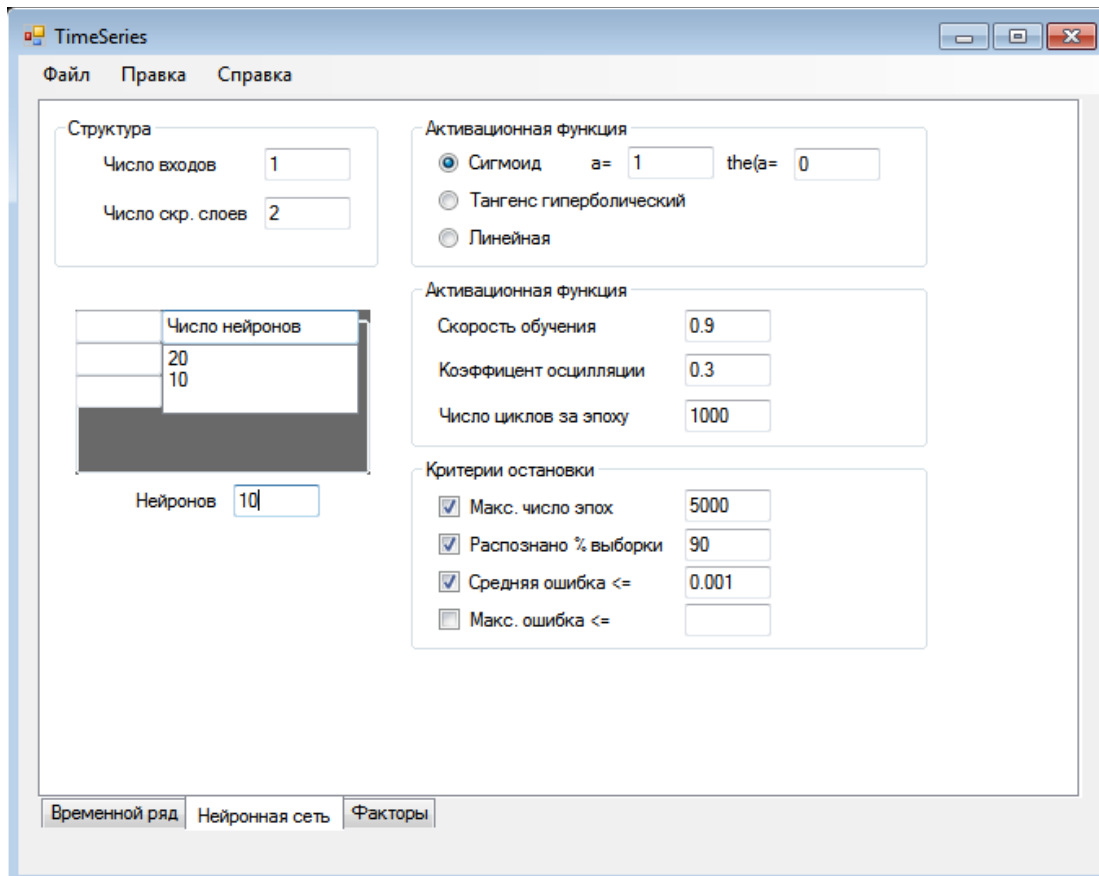


Рисунок 4.6 – Параметры парадигмы нейронной сети

Поле ввода «Число скр. слоев» позволяет задавать количество скрытых слоев нейронной сети. Скрытые слои ИНС – понимаются все слои, кроме входного и выходного.

Программой осуществляется контроль правильности ввода числа скрытых слоев (не меньше одного и не больше 100) и выводится соответствующее сообщение в случае неправильно введенных данных.

После изменения пользователем числа скрытых слоев ИНС соответствующим образом меняется размерность списка «Число нейронов», в котором выводится число нейронов по слоям, начиная с первого. Изменить число нейронов можно в поле ввода «Нейронов».

Задать форму активационной функции и ее параметры (для сигмоиды) можно в поле «Активационная функция». Для ввода параметра наклона сигмоиды используется поле ввода «а», поле ввода «theta» для ввода параметра аналогичного порогу для пороговой функции.

Для ввода коэффициента обучения нейронной сети η служит поле ввода «Скорость обучения», поле ввода «Коэффициент осцилляции» позволяет задать коэффициент осцилляции α (коэффициент импульса).

Для ввода числа циклов обучения ИНС предназначено поле «Число циклов за эпоху», после чего происходит контроль степени обученности нейросети согласно критериям остановки, которые были заданы. Если достигается критерий остановки, то обучение нейросети прекращается и происходит прогноз на один шаг вперед со смещением скользящего окна.

В качестве критериев остановки могут выступать:

- 1) достижение максимально допустимого числа эпох;
- 2) распознавание нейронной сетью заданной части обучающей выборки;
- 3) достижение заданной средней квадратичной ошибки;
- 4) достижение заданной максимальной квадратичной ошибки.

Перечисленные критерии можно использовать совместно, для этого необходимо использовать логическую операцию «ИЛИ». С помощью

соответствующего переключателя можно задать критерии, после чего становится активным поле ввода для параметра этого критерия.

Закладка «Факторы» (рисунок 4.7). Данная закладка служит для задания входного и выходного векторов нейронной сети. Программа позволяет осуществлять анализ многомерных временных рядов (входной и выходной векторы могут определяться несколькими ВР).

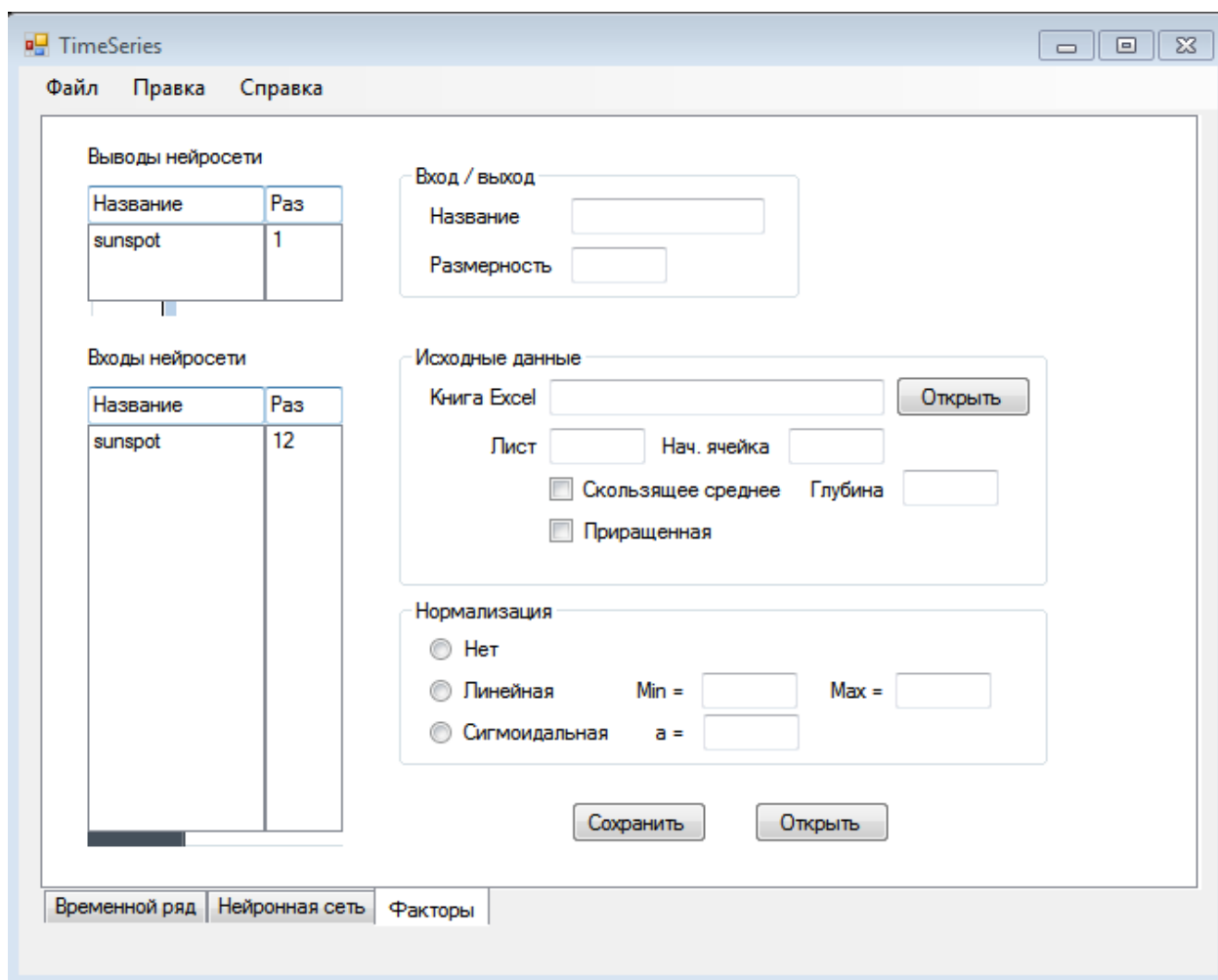


Рисунок 4.7 – Входы и выходы нейронной сети

Выходной вектор определяется в списке «Выводы нейросети», входной вектор – «Входы нейросети». Работа с этими списками осуществляется аналогично.

При нажатии правой кнопки мыши в одном из этих списков появляется всплывающее меню, состоящее из трех пунктов:

- 1) «Создать...» – добавить новый вход/выход к вектору;
- 2) «Изменить...» – изменить параметры существующего входа/выхода;

3) «Удалить...» – удалить входа/выхода.

При создании/изменении входа/выхода в правой части окна активизируются элементы управления, позволяющие задавать характеристики входа/выхода.

Каждый вход и выход характеризуется идентификатором, который задается в поле ввода «Название» и отображается в списке слева.

Поле ввода «Размерность» используется для задания глубины погружения ВР (размер его предыстории), используемой для обучения ИНС.

Группа элементов интерфейса «Исходные данные» позволяет задать источник данных ВР (файл книги MS Excel, лист MS Excel, начальную ячейку с данными), и некоторые преобразования, производимые с временным рядом: скользящее среднее и приращения ряда.

Поле «Нормализация» используется для задания типа нормализации значений ВР и параметров нормализации. После завершения ввода всех значений необходимо нажать кнопку «Сохранить» либо кнопку «Отменить». В последнем случае все сделанные изменения отменяются и не имеют никакого эффекта. При выборе во всплывающем меню команды «Удалить...» программа выводит диалоговое окно с требованием подтверждения удаления входа/выхода нейросети. В случае положительного ответа вход/выход удаляется из вектора.

Главное меню программы состоит из следующих пунктов. Меню «Файл» содержит следующие команды: «Новая модель» – создание новой прогнозирующей модели (значение всех параметров устанавливаются по умолчанию). «Открыть модель...» – открыть сохраненную ранее модель из файла. «Сохранить модель...» – сохранить выполняемую программой модель в файл. «Выход» – завершение работы. Меню «Правка» содержит стандартные команды «Вырезать», «Копировать», «Вставить». Меню «Справка» содержит команду «О программе...» с краткой информацией о программном продукте.

Заключение

После изучения данной проблемы, проанализировав все данные, имеющиеся по данной тематике, и мы предложили свои решения проблемы и наконец завершив работу, перейдем к некоторым выводам. Задача прогнозирования временных рядов сегодня, как оказалось, имеет высокую актуальность для многих предметных областей. Непосредственно, установлено, что к настоящему времени разработано множество моделей для решения задачи прогнозирования временного ряда. Но не все они находятся в общем пользовании из-за своей малодоступности для обычного обывателя. Для работы с качественным бесплатным программным продуктом требуются профессиональные навыки и высокая квалификация пользователя. А более профессиональные приложения требуют не малых затрат.

В данной работе выявлены достоинства и недостатки рассмотренных моделей. Установлено, что существенным недостатком большинства моделей является большое число свободных параметров, требующих идентификации. При этом мы пришли к выводу, что наиболее перспективным направлением развития прогнозирования является создание комбинированных моделей, которые могли бы повысить точность прогноза.

Перечислим основные результаты, полученные в дипломной работе. Применение ИНС позволяет повысить эффективность прогнозирования временных рядов, так как лишены ряда недостатков классических методов и может служить основой для автоматизированного прогнозирования (с минимальным участием человека). В работе предложены и исследованы способы повышения эффективности прогнозирующих свойств и скорости обучения ИНС. Усовершенствован генетический алгоритм комплексной оптимизации прогнозирующих нейросетей, позволяющий уменьшить время обучения.

Список использованных источников

- 1 Андерсон, Т. Статистический анализ временных рядов / Т. Андерсон. – М. : Мир, 1976. – 760 с.
- 2 Анил, К. Д. Введение в искусственные нейронные сети / К. Д. Анил, Мао Жианчанг, К.М. Моиуддин. – М. : Открытые системы, 1997. – С. 16-24.
- 3 Артемкин, Д.Е. Прогнозирование временного ряда с помощью искусственной нейронной сети в режиме скользящего окна / Д.Е. Артемкин. – СПб. : Санкт-Петербургский военный университет связи, 2002. – С. 659-660.
- 4 Артемкин, Д.Е. Использование различных архитектур искусственных нейронных сетей для прогнозирования временных рядов / Д.Е. Артемкин, А.Н. Пылькин. – СПб. : Санкт-Петербургский военный университет связи, 2002. – С. 660-661.
- 5 Благодаров, А.В. Идентификация состояний дискретного канала передачи данных с использованием нейросетевых технологий / А.В. Благодаров, Г.А. Новиков. – Рязань : РИРО, 1998. – С. 131-132.
- 6 Благодаров, А.В. Моделирование интеллектуальных процедур идентификации состояний дискретных каналов связи / А.В. Благодаров, Г.А. Новиков, А.Н. Пылькин. – М. : МГАПИ, 1998. – С. 12-17.
- 7 Благодаров, А.В. Пакет программ нейросетевого моделирования процедур идентификации состояний дискретного канала передачи данных / А.В. Благодаров, Г.А. Новиков, А.Н. Пылькин. – Рязань : РИРО, 1998. – С. 146-148.
- 8 Бокс, Дж. Анализ временных рядов, прогноз и управление / Дж. Бокс, Г. Дженкинс. – М. : Мир, 1974. – 210 с.
- 9 Боровиков, В.П. Нейронные сети. STATISTICA Neural Networks / В.П. Боровиков. – М. : Горячая линия – Телеком, 2008. – 392 с.
- 10 Бэстенс, Д.Э. Нейронные сети и финансовые рынки. Принятие решений в торговых операциях / Д.Э. Бэстенс, Ден Берг В.-М. Ван, Д. Вуд. – М. : ТВП Научное издательство, 1997. – 320 с.

- 11 Вентцель, Е.С. Теория вероятностей / Е.С. Вентцель. – М. : Наука, 1964. – 576 с.
- 12 Вентцель, Е.С. Прикладные задачи теории вероятностей / Е.С. Вентцель, Л.А. Овчаров. – М. : Радио и связь, 1983. – 416 с.
- 13 Галушкин, А.И. Нейроматематика / А.И. Галушкина. – М. : ИПРЖР, 2002. – 448 с.
- 14 Галушкин, А.И. Теория нейронных сетей / А.И. Галушкин. – М. : ИПРЖР, 2000. – 175 с.
- 15 Кильдишев, Г.С. Анализ временных рядов и прогнозирование / Г.С. Кильдишев, А.А. Френкель. – М. : Статистика, 1973. – 528 с.
- 16 Князевский, В.С. Анализ временных рядов и прогнозирование / В.С. Князевский, И.В. Житников. – Ростов-на-Дону : Рост. гос. экон. акад., 1998. – 161 с.
- 17 Коршунов, Ю.М. Математические основы кибернетики / Ю.М. Коршунов. – М. : Энергия, 1972. – 347 с.
- 18 Мун, Ф. Хаотические колебания / Ф. Мун. – М. : Мир, 1990. – 312 с.
- 19 Назаров, А.В. Прогнозная модель параметров технического состояния космического аппарата на основе нейросетевой структуры «нейронная сеть прямого распространения – карта Кохонена» / А.В. Назаров, Г.И. Козырев, С.В. Шкляр – М. : Нейрокомпьютеры, 2002. – С. 57-61.
- 20 Петросяна, Л.А. Введение в системный анализ. / Л.А. Петросяна. – СПб. : Изд-во Ленинградского университета, 1988. – 232 с.
- 21 Сарычев, Н.А. Исследование влияния выбора параметров обучения на скорость обучения ИНС / Н.А. Сарычев, А.Е. Сметанников, Д.А. Филиппов. – М. : НИЦПрИС, 1998. – С. 93-95.
- 22 Сарычев, Н.А. Принципы функционирования генетических алгоритмов / Н.А. Сарычев, А.Е. Сметанников, Д.А. Филиппов. – Рязань : Вычислительные машины, комплексы и сети, 1996. – 432 с.
- 23 Скляр, В.А. Язык С++ и объектно-ориентированное программирование / В.А. Скляр. – М. : Выш. шк., 1997. – 478 с.

- 24 Танеев, Р.М. Математические модели в задачах обработки сигналов / Р.М. Танеев. – М. : Горячая линия-Телеком, 2002. – 346 с.
- 25 Уоссермен, Ф. Нейрокомпьютерная техника: Теория и практика / Ф. Уоссермен. – М.: Мир, 1992. – 240 с.
- 26 Феллер, В. Введение в теорию вероятностей и ее приложения / В. Феллер. – М. : Мир, 1967. – 498 с.
- 27 Фернбах, С. СуперЭВМ. Аппаратная и программная организация / С. Фернбах. – М. : Радио и связь, 1991. – 320 с.
- 28 Четыркин, Е.М. Статистические методы прогнозирования / Е.М. Четыркин. – М. : Статистика, 1977. – 564 с.
- 29 Шилдт, Г. Теория и практика С++ / Г. Шилдт. – СПб. : ВHV – Санкт-Петербург, 1996. – 416 с.
- 30 Шукович, Г. Применение генетических алгоритмов и систем генерирующих графов для создания модулярных нейросетей / Г. Шукович. – М. : Программирование, 2002. – С. 13-20.
- 31 Шумский, С.А. Нейросетевое предсказание финансовых временных рядов / С.А. Шумский, А.В. Яровой, А.А. Лукьяница. – Снежинск : ВНИИТФ, 1998г. – 167 с.
- 32 Шустер, Г. Детерминированный хаос / Г. Шустер. – М. : Мир, 1988. – 240 с.
- 33 Элдер, А. Основы биржевой игры. Учебное пособие для участников торгов на мировых биржах. Психология. Тактика торгов. Денежный менеджмент / А. Элдер. – М. : Издательство «Светоч», 1995. – 327с.
- 34 Эфрон, Б. Нетрадиционные методы многомерного статистического анализа / Б. Эфран. – М. : Финансы и статистика, 1988. – 263 с.