

Федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВ-  
СКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им.  
Н.П.ОГАРЕВА»

Факультет математики и информационных технологий  
Кафедра прикладной математики, дифференциальных уравнений и тео-  
ретической механики

ВАСИЛЬКИН НИКОЛАЙ ВИКТОРОВИЧ

**Научно-квалификационная работа (диссертация)**  
**«РАЗРАБОТКА НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ДЛЯ АНАЛИЗА**  
**УПРАВЛЯЕМЫХ ПРОЦЕССОВ»**

Направление подготовки 01.06.01 — Математика и механика,  
профиль подготовки — Дифференциальные уравнения, динамические  
системы и оптимальное управление.

Научный руководитель работы  
канд. физ.-мат. наук, доцент

Т.Ф. Мамедова

Саранск  
2018

## РЕФЕРАТ

Научно-квалификационная работа (диссертация) содержит 79 страниц, 4 рисунка, 24 использованных источников, 3 приложения.

ВРЕМЕННОЙ РЯД, МАТЕМАТИЧЕСКАЯ МОДЕЛЬ, НЕЙРОННАЯ СЕТЬ, АППРОКСИМАЦИЯ ВЫБОРКИ, ПОДОБИЕ ВЫБОРОК, ОПИСАНИЕ МОДЕЛИ

Объектом исследования являются временные ряды. Цель работы – разработка новой модели анализа временных рядов и соответствующего ей метода прогнозирования, относящейся к классу авторегрессионных моделей.

В процессе работы были использованы модель экстраполяции временных рядов по выборке максимального подобия, среда разработки приложений IntelliJ IDEA 13.1 на Java, программа пакет прикладных программ для решения задач технических вычислений MATLAB и программа для работы с электронными таблицами Microsoft Office Excel 2013.

В диссертационной работе проведен обзор и сравнение моделей прогнозирования временных рядов. На основании этого анализа выбрана и модифицирована одна из рассмотренных моделей. Разработана программа, реализующая полученный алгоритм.

Область применения – прогнозирование временных рядов.

Повышение эффективности моделирования прогнозов временных рядов достигнуто за счет применения нейронных сетей для вычисления коэффициентов в уравнении линейной корреляции Пирсона. В конечном итоге это повышает эффективность и точность получаемых прогнозов значений в исследуемых временных рядах. Разработана программа позволяющая выполнить вычисление значений ряда по входящим данным.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Постановка задачи прогнозирования временных рядов	8
1.1 Содержательная постановка задачи	8
1.2 Формальная постановка задачи	13
1.3 Обзор и сравнение моделей прогнозирования	14
1.4 Достоинства и недостатки моделей.	16
1.5 Комбинированные модели	20
1.6 Выводы	24
2 Модели экстраполяции временных рядов по выборке максимального подобия	25
2.1 Модель с учетом внешних факторов	25
2.1.1 Выборки временных рядов	25
2.1.2 Аппроксимация выборки	26
2.1.3 Подобие выборок	28
2.2 Применение нейронных сетей для нахождения максимального значения линейной корреляции Пирсона	32
2.3 Описание модели	32
2.4 Варианты моделей по выборке максимального подобия	35
2.5 Выводы	38
3 Многослойный персептрон	39
3.1 Структура сети	39
3.2 Архитектура сети	42
3.2.1 Алгоритм обучения с обратным распространением ошибки	43
4 Расчет будущих значений временного ряда и программная реализация	50
4.1 Алгоритм экстраполяции временного ряда с учетом внешних факторов	50
4.2 Программная реализация	55
ЗАКЛЮЧЕНИЕ	58
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	59

ПРИЛОЖЕНИЕ А (ОЯЗАТЕЛЬНОЕ) ЛИСТИНГ ПРОГРАММЫ НЕЙРОННОЙ СЕТИ	61
ПРИЛОЖЕНИЕ Б (ОБЯЗАТЕЛЬНОЕ) ДИСК С ВРЕМЕННЫМ РЯДОМ	77
ПРИЛОЖЕНИЕ В (НЕОБЯЗАТЕЛЬНОЕ) ДИСК С ПРОГРАММОЙ	78

## ВВЕДЕНИЕ

Задача анализа временных рядов, является в настоящее время актуальной проблемой во многих областях науки и техники. Прогнозирование будущих значений временного ряда на основе его исторических значений является основой для финансового планирования в экономике и торговле, планирования, управления и оптимизации объемов производства, складского контроля. В настоящее время компаниями осуществляется накопление исторических значений экономических и физических показателей в базах данных, что существенно увеличивает объемы входной информации для задачи прогнозирования. Вместе с тем, развитие аппаратных и программных средств, предоставляет все более мощные вычислительные платформы, на которых возможна реализация сложных алгоритмов прогнозирования. Кроме того, современные подходы к экономическому и техническому управлению предъявляют все более жесткие требования к точности прогнозирования. Таким образом, задача прогнозирования временных рядов усложняется одновременно с развитием информационных технологий. В настоящее время задача прогнозирования различных временных рядов актуальна и является неотъемлемой частью ежедневной работы многих компаний. Задача прогнозирования временного ряда решается на основе создания модели прогнозирования, адекватно описывающей исследуемый процесс. На сегодняшний день существует множество моделей прогнозирования временных рядов: регрессионные и авторегрессионные модели, нейросетевые модели, модели экспоненциального сглаживания, модели на базе цепей Маркова, классификационные модели и др. Наиболее популярными и широко используемыми являются классы авторегрессионных и нейросетевых моделей. Существенным недостатком авторегрессионного класса является большое число свободных параметров, идентификация которых неоднозначна и ресурсоемка. Существенным недостатком класса нейросетевых моделей является недоступность промежуточных вычислений, выполняющихся в «черном ящике», и, как следствие, сложность

интерпретации результатов моделирования. Кроме того, еще одним недостатком данного класса моделей является сложность выбора алгоритма обучения нейронной сети.

Научно-квалификационная работа (диссертация) посвящена разработке новой авторегрессионной модели прогнозирования, которая имеет сравнимую с другими моделями эффективность прогнозирования различных временных рядов и при этом устраняет основной и наиболее существенный недостаток авторегрессионного класса моделей – большое число свободных параметров.

В качестве примера можно назвать задачу прогнозирования энергопотребления. Задача планирования потребления электроэнергии всегда была важна для правильного расчета процессов ее приобретения и расходования, а сейчас стала весьма актуальной в связи с преобразованиями в российской энергетике.

Целью работы является разработка новой модели и соответствующего ей метода прогнозирования, относящейся к классу авторегрессионных моделей и устраняющей основной недостаток данного класса моделей – большое число свободных параметров. Новая модель и соответствующий ей метод должны иметь высокую скорость вычисления прогнозных значений и сравнимую с другими моделями точность прогнозирования различных временных рядов.

Для достижения этой цели были поставлены и решены следующие задачи.

1. Осуществить обзор моделей и методов прогнозирования временных рядов, выявить достоинства и недостатки каждого класса моделей. Выявить наиболее используемые классы моделей прогнозирования и их основные недостатки, определить перспективные подходы, позволяющие устранить недостатки авторегрессионного класса моделей.

2. Разработать новую модель прогнозирования временных рядов, устраняющую указанный недостаток авторегрессионного класса моделей.

3. Разработать новый метод прогнозирования на основании предложенной модели и выполнить программную реализацию алгоритмов.

4. Оценить эффективность предложенной модели прогнозирования при решении задачи прогнозирования различных временных рядов.

Методы исследования. При решении поставленных задач в работе использованы методы математического моделирования, анализ временных рядов, регрессионный анализ, методы объектно-ориентированного программирования.

Научная новизна. В работе получены следующие основные результаты, которые выносятся на защиту.

1. Модель экстраполяции временных рядов по выборке максимального подобия, относящаяся к классу авторегрессионных моделей и имеющая единственный параметр.

2. Метод прогнозирования временных рядов на основании разработанной модели, содержащий набор алгоритмов для экстраполяции временных рядов, идентификации модели и построения доверительного интервала прогнозных значений.

3. Результаты прогнозирования временных рядов показателей урожайности, а также временных рядов из других предметных областей, подтверждающие эффективность разработанной модели.

Практическая ценность работы. Разработанная модель и метод прогнозирования по выборке максимального подобия могут применяться для прогнозирования временных рядов различных предметных областей.

Разработанные алгоритмы экстраполяции временных рядов с учетом внешних факторов наглядны для программной реализации. Скорость вычисления прогнозных значений при использовании модели высока. Задача идентификации модели упрощена в сравнении с другими моделями авторегрессионного класса.

# **1 ПОСТАНОВКА ЗАДАЧИ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ**

## **1.1 Содержательная постановка задачи**

Слово прогноз возникло от греческого  $\text{προϋννοσις}$ , что означает предвидение, предсказание. Под прогнозированием понимают предсказание будущего с помощью научных методов. Процессом прогнозирования называется специальное научное исследование конкретных перспектив развития какого-либо процесса. Процессы, перспективы которых необходимо предсказывать, чаще всего описываются временными рядами, то есть последовательностью значений некоторых величин, полученных в определенные моменты времени. Временной ряд включает в себя два обязательных элемента – отметку времени и значение показателя ряда, полученное тем или иным способом и соответствующее указанной отметке времени. Каждый временной ряд рассматривается как выборочная реализация из бесконечной популяции, генерируемой стохастическим процессом, на который оказывают влияние множество факторов.

Согласно [1] временные ряды различаются способом определения значения, временным шагом, памятью и стационарностью. В зависимости от способа определения значений временного ряда они делятся на

- интервальные временные ряды,
- моментные временные ряды.

Интервальный временной ряд представляет собой последовательность, в которой уровень явления (значение временного ряда) относят к результату, накопленному или вновь произведенному за определенный интервал времени. Интервальным, например, является временной ряд показателя выпуска продукции предприятием за неделю, месяц или год; объем воды, сброшенной



гидроэлектростанцией за час, день, месяц; объем электроэнергии, произведенной за час, день, месяц и другие.

Если же значение временного ряда характеризует изучаемое явление в конкретный момент времени, то совокупность таких значений образует моментный временной ряд. Примерами моментных рядов являются последовательности финансовых индексов, рыночных цен; физические показатели, такие как температура окружающего воздуха, влажность, давление, измеренные в конкретные моменты времени, и другие.

Согласно [1] в зависимости от частоты определения значений временного ряда, они делятся на

- равноотстоящие временные ряды,
- неравноотстоящие временные ряды.

Равноотстоящие временные ряды формируются при исследовании и фиксации значений процесса в следующие друг за другом равные интервалы времени. Большинство физических процессов описываются при помощи равноотстоящих временных рядов. Неравноотстоящими временными рядами называются те ряды, для которых принцип равенства интервалов фиксации значений не выполняется. К таким рядам относятся, например, все биржевые индексы в связи с тем, что их значения определяются лишь в рабочие дни недели. В зависимости от характера описываемого процесса временные ряды разделяются на

- временные ряды длинной памяти,
- временные ряды короткой памяти.

В целом, говоря о временных рядах с длинной памятью, подразумеваются временные ряды, для которых автокорреляционная функция, убывает медленно. К временным рядам с короткой памятью относят временные ряды, автокорреляционная функция которых убывает быстро. Скорость потока транспорта по дорогам, а также многие физические процессы, такие как потребление электроэнергии, температура воздуха, относятся к временным ря-

дам с длинной памятью. К временным рядам с короткой памятью относятся, например, временные ряды биржевых индексов.

Дополнительно временные ряды принято разделять на

- стационарные временные ряды,
- нестационарные временные ряды.

Стационарным временным рядом называется такой ряд, который остается в равновесии относительно постоянного среднего уровня. Остальные временные ряды являются нестационарными. В промышленности, и в торговле, и в экономике, где прогнозирование имеет важное значение, многие временные ряды являются нестационарными, то есть не имеющими естественного среднего значения. В работе нестационарные временные ряды для решения задачи прогнозирования часто приводятся к стационарным при помощи разностного оператора. Горизонт времени, на который необходимо определить значения временного ряда, называется временем упреждения. Например, в работе [2], в зависимости от времени упреждения задачи прогнозирования, как правило, делятся следующие категории срочности:

- долгосрочное прогнозирование;
- среднесрочное прогнозирование;
- краткосрочное прогнозирование.

Важно отметить, что для каждого временного ряда приведенная классификация имеет собственные диапазоны. Для временного ряда уровня сахара крови классификация срочности задачи прогнозирования обуславливается типами инсулина:

- ультракраткосрочное прогнозирование: до 3 - 4 часа;
- краткосрочное прогнозирование: до 5 - 8 часов;
- среднесрочное прогнозирование: до 16 - 24 часов.

Для задачи прогнозирования урожайности классификация задач:

- ультракраткосрочное прогнозирование: до одного дня;
- краткосрочное прогнозирование: от одного дня до недели;
- среднесрочное прогнозирование: от одной недели до года;

– долгосрочное прогнозирование: более чем на год вперед.

То есть для различных временных рядов, с различным временным разрешением классификация срочности задач прогнозирования индивидуальна.

Говоря о прогнозировании временных рядов, необходимо различить два взаимосвязанных понятия – метод прогнозирования и модель прогнозирования. Метод прогнозирования представляет собой последовательность действий, которые нужно совершить для получения модели прогнозирования временного ряда.

Модель прогнозирования есть функциональное представление, адекватно описывающее временной ряд и являющееся основой для получения будущих значений процесса. Часто, говоря о моделях прогнозирования, используется термин модель экстраполяции [3].

Метод прогнозирования содержит последовательность действий, в результате выполнения которой определяется модель прогнозирования конкретного временного ряда. Кроме того, метод прогнозирования содержит действия по оценке качества прогнозных значений. Общий итеративный подход к построению модели прогнозирования состоит из следующих шагов.

Шаг 1. На первом шаге, на основании предыдущего собственного или стороннего опыта выбирается общий класс моделей для прогнозирования временного ряда на заданный горизонт.

Шаг 2. Определенный общий класс моделей обширен. Для непосредственной подгонки к исходному временному ряду, развиваются грубые методы идентификации подклассов моделей. Такие методы идентификации используют качественные оценки временного ряда.

Шаг 3. После определения подкласса модели, необходимо оценить ее параметры, если модель содержит параметры, или структуру, если модель относится к категории структурных моделей. На данном этапе обычно используются итеративные способы, когда производится оценка участка (или всего) временного ряда при различных значениях изменяемых величин. Как правило, данный шаг является наиболее трудоемким в связи с тем, что часто

в расчет принимаются все доступные исторические значения временного ряда.

Шаг 4. Далее производится диагностическая проверка полученной модели прогнозирования. Чаще всего выбирается участок или несколько участков временного ряда, достаточных по длине для проверочного прогнозирования и последующей оценки точности прогноза. Выбранные для диагностики модели прогнозирования участки временного ряда называются контрольными участками (периодами).

Шаг 5. В случае если точность диагностического прогнозирования оказалась приемлемой для задач, в которых используются прогнозные значения, то модель готова к использованию. В случае если точность прогнозирования оказалось недостаточной для последующего использования прогнозных значений, то возможно итеративное повторение всех описанных выше шагов, начиная с первого.

Моделью прогнозирования временного ряда является функциональное представление, адекватно описывающее временной ряд. При прогнозировании временных рядов возможны два варианта постановки задачи. В первом варианте для получения будущих значений исследуемого временного ряда используются доступные значения только этого ряда. Во втором варианте для получения прогнозных значений возможно использование не только фактических значений искомого ряда, но и значений набора внешних факторов, представленных в виде временных рядов. В общем случае временные ряды внешних факторов могут иметь разрешение по времени отличное от разрешения искомого временного ряда. Например, в работе [4] подробно обсуждаются внешние факторы, оказывающие влияние на временной ряд урожайности картофеля. К таким внешним факторам относят температуру окружающей среды, влажность воздуха, а также сезонность, т. е. час суток, день недели, месяц года. Лишь некоторые модели прогнозирования позволяют учитывать категориальные внешние факторы, большинство моделей позволяют учитывать только дискретных.

При прогнозировании временного ряда требуется определить функциональную зависимость, адекватно описывающую временной ряд, которая называется моделью прогнозирования. Цель создания модели прогнозирования состоит в получении такой модели, для которой среднее абсолютное отклонение истинного значения от прогнозируемого стремится к минимальному для заданного горизонта, который называется временем упреждения [5]. После того, как модель прогнозирования временного ряда определена, требуется вычислить будущие значения временного ряда, а также их доверительный интервал.

## 1.2 Формальная постановка задачи

Прогнозирование с учетом внешних факторов. Пусть значения исходного временного ряда  $Z(t)$  доступны в дискретные моменты времени  $t=1, 2, \dots, T$ . Предполагается, что на значения  $Z(t)$  оказывает влияние набор внешних факторов. Пусть первый внешний фактор  $X_1(t_1)$  доступен в дискретные моменты времени  $t_1=1, 2, \dots, T_1$ , второй внешний фактор  $X_2(t_2)$  доступен в моменты времени  $t_2=1, 2, \dots, T_2$  и т.д. В случае если дискретность исходного временного ряда и внешних факторов, а так же значения  $T, T_1, \dots, T_s$  различны, то временные ряды внешних факторов  $X_1(t_1), \dots, X_s(t_s)$  необходимо привести к единой шкале времени  $t$  [4].

В момент прогноза  $T$  необходимо определить будущие значения исходного процесса  $Z(t)$  в моменты времени  $T+1, \dots, T+P$ , учитывая влияние внешних факторов  $X_1(t), \dots, X_s(t)$ . При этом считаем, что значения внешних факторов в моменты времени  $X_1(T+1), \dots, X_1(T+P), \dots, X_s(T+1), \dots, X_s(T+P)$  являются доступными.

Для вычисления будущих значений процесса  $Z(t)$  в указанные моменты времени требуется определить функциональную зависимость, отражающую

связь между прошлыми значениями  $Z(t)$  и будущими, а также принимающую во внимание влияние внешних факторов  $X_1(t), \dots, X_s(t)$  на исходный временной ряд

$$Z(t) = F(Z(t-1), Z(t-2), \dots, X_1(t), X_1(t-1), \dots, X_s(t), X_s(t-1), \dots) + \varepsilon_t. \quad (1.2.1)$$

Зависимость (1.2.1) называется моделью прогнозирования с учетом внешних факторов  $X_1(t), \dots, X_s(t)$ . Требуется создать такую модель прогнозирования, для которой среднее абсолютное отклонение истинного значения от прогнозируемого стремится к минимальному для заданного  $P$ .

Кроме получения будущих значений  $\hat{Z}(T+1), \dots, \hat{Z}(T+P)$  требуется определить доверительный интервал возможных отклонений этих значений. Задача прогнозирования временного ряда с учетом одного внешнего фактора представлена на рисунке 1.

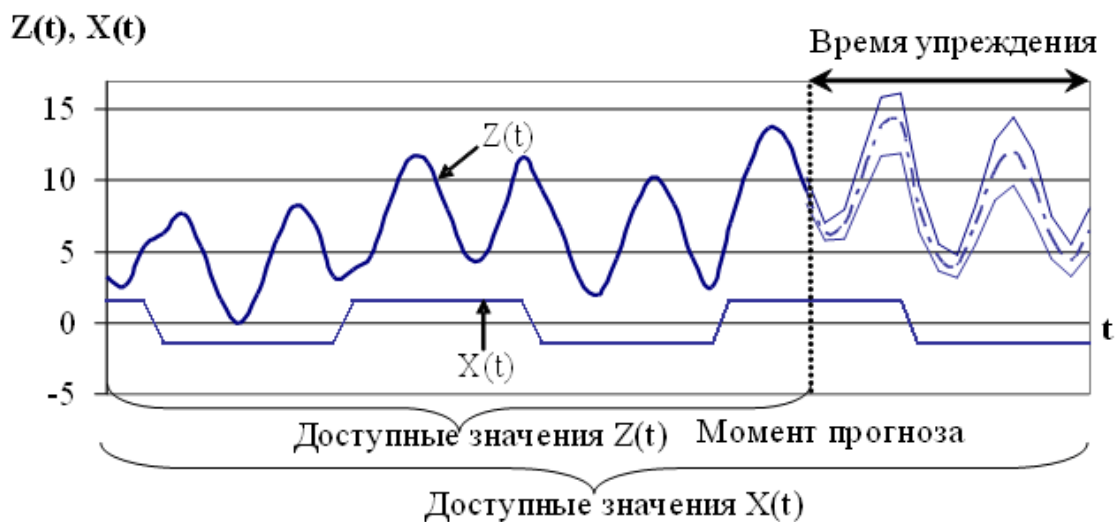


Рисунок 1 – Иллюстрация задачи прогнозирования временного ряда с учетом внешнего фактора

### 1.3 Обзор и сравнение моделей прогнозирования

Перед тем как перейти к обзору моделей, необходимо отметить, что названия моделей и соответствующих методов, как правило, совпадают.

Например работы [1],[5],[7],[8] посвящены одной из самых распространенных моделей прогнозирования авторегрессии проинтегрированного скользящего среднего с учетом внешнего фактора (autoregression moving average external, ARIMAX). Эту модель и соответствующий ей метод обычно называют ARIMAX. В настоящее время принято использовать английские аббревиатуры названий, как моделей, так и методов.

В настоящее время насчитывается свыше 100 классов моделей [19]. Число общих классов моделей, которые в тех или иных вариациях повторяются в других, гораздо меньше. Часть моделей и соответствующих методов относится к отдельным процедурам прогнозирования. Часть методов представляет набор отдельных приемов, отличающихся от базовых или друг от друга количеством частных приемов и последовательностью их применения. Все методы прогнозирования делятся на две группы: интуитивные и формализованные.

Интуитивное прогнозирование применяется тогда, когда объект прогнозирования либо слишком прост, либо, напротив, настолько сложен, что аналитически учесть влияние внешних факторов невозможно. Интуитивные методы прогнозирования не предполагают разработку моделей прогнозирования и отражают индивидуальные суждения специалистов (экспертов) относительно перспектив развития процесса. Интуитивные методы основаны на мобилизации профессионального опыта и интуиции. Такие методы используются для анализа процессов, развитие которых либо полностью, либо частично не поддается математической формализации, то есть для которых трудно разработать адекватную модель. К таким методам относятся методы экспертных оценок, исторических аналогий, предвидения по образцу. Кроме того, в настоящее время широко распространено применение экспертных систем, в том числе с использованием нечеткой логики.

Формализованные методы рассматривают модели прогнозирования. Модели прогнозирования разделяются на статистические модели и структурные модели.

В статистических моделях функциональная зависимость между будущими и фактическими значениями временного ряда, а также внешними факторами задана аналитически. К статистическим моделям относятся следующие группы [7]:

- регрессионные модели;
- авторегрессионные модели;
- модели экспоненциального сглаживания.

В структурных моделях функциональная зависимость между будущими и фактическими значениями временного ряда, а также внешними факторами задана структурно. К структурным моделям относятся следующие группы [8]:

- нейросетевые модели;
- модели на базе цепей Маркова;
- модели на базе классификационно-регрессионных деревьев.

Кроме того, необходимо отметить, что для узкоспециализированных задач иногда применяются особые модели прогнозирования. Так, например, для задачи прогнозирования энергопотребления применяются модели на основе дифференциальных уравнений [9]. Для задачи прогнозирования транспортного потока, которая в последние несколько лет актуальна для мегаполисов, применяются гидродинамические модели. Для прогнозирования природных явлений, таких как землетрясения, применяется, например, модель, в основу которой положены нелинейные клетки (или соты), находящиеся под воздействием внешнего поля, и у которых есть внутреннее состояние, изменяющееся во времени под воздействием этого поля. Аналогичные модели разрабатываются и применяются для специальных процессов и систем. В рамках настоящей работы данный класс формализованных моделей не рассматривается.



## 1.4 Достоинства и недостатки моделей

Регрессионные модели и методы. К достоинствам данных моделей прогнозирования относят простоту, гибкость, а также единообразие их анализа и проектирования. При использовании линейных регрессионных моделей результат прогнозирования может быть получен быстрее, чем при использовании остальных моделей. Кроме того, достоинством является прозрачность моделирования, т. е. доступность для анализа всех промежуточных вычислений.

Основным недостатком нелинейных регрессионных моделей является сложность определения вида функциональной зависимости, а также трудоемкость определение параметров модели. Недостатками линейных регрессионных моделей являются низкая адаптивность и отсутствие способности моделирования нелинейных процессов [11].

Авторегрессионные модели и методы. Важными достоинствами данного класса моделей являются их простота и прозрачность моделирования. Еще одним достоинством является единообразие анализа и проектирования. На сегодняшний день данный класс моделей является одним из наиболее популярных, а потому в открытом доступе легко найти примеры применения авторегрессионных моделей для решения задач прогнозирования временных рядов различных предметных областей.

Недостатками данного класса моделей являются: большое число параметров модели, идентификация которых неоднозначна и ресурсоемка; низкая адаптивность моделей, а также линейность и, как следствие, отсутствие способности моделирования нелинейных процессов, часто встречающихся на практике.

Модели и методы экспоненциального сглаживания. Достоинствами данного класса моделей являются простота и единообразие их анализа и проектирования. Данный класс моделей чаще других используется для дол-

госрочного прогнозирования. Недостатком данного класса моделей прогнозирования является отсутствие гибкости.

Нейросетевые модели и методы. Согласно [11], основным достоинством нейросетевых моделей является нелинейность, т.е. способность устанавливать нелинейные зависимости между будущими и фактическими значениями процессов. Другими важными достоинствами являются: адаптивность, масштабируемость (параллельная структура ANN ускоряет вычисления) и единообразие их анализа и проектирования.

При этом недостатками ANN являются отсутствие прозрачности моделирования; сложность выбора архитектуры, высокие требования к непротиворечивости обучающей выборки; сложность выбора алгоритма обучения и ресурсоемкость процесса их обучения.

Модели и методы на базе цепей Маркова [12]. Простота и единообразие анализа и проектирования являются достоинствами моделей на базе цепей Маркова. Недостатком данных моделей является отсутствие возможности моделирования процессов с длинной памятью.

Модели на базе классификационно – регрессионных деревьев. Достоинствами данного класса моделей являются: масштабируемость, за счет которой возможна быстрая обработка сверхбольших объемов данных; быстрота и однозначность процесса обучения дерева (в отличие от ANN), а также возможность использовать категориальные внешние факторы. Недостатками данных моделей являются неоднозначность алгоритма построения структуры дерева; сложность вопроса останова, т.е. вопроса о том, когда стоит прекратить дальнейшие ветвления; отсутствие единообразия их анализа и проектирования [13].

Достоинства и недостатки моделей и методов систематизированы в таблице 1– Сравнение моделей и методов прогнозирования.

Таблица 1– Сравнение моделей и методов прогнозирования

Модель и метод	Достоинства	Недостатки
1	2	3
Регрессионные модели и методы	простота, гибкость, прозрачность моделирования; единообразие анализа и проектирования	сложность определения функциональной зависимости; трудоемкость нахождения коэффициентов зависимости; отсутствие возможности моделирования нелинейных процессов (для нелинейной регрессии)
Авторегрессионные модели и методы	простота, прозрачность моделирования; единообразие анализа и проектирования; множество примеров применения	трудоемкость и ресурсоемкость идентификации моделей; невозможность моделирования нелинейностей; низкая адаптивность
Модели и методы экспоненциального сглаживания	простота моделирования; единообразие анализа и проектирования	недостаточная гибкость; узкая применимость моделей
Нейросетевые модели и методы	нелинейность моделей; масштабируемость, высокая адаптивность; единообразие анализа и проектирования; множество примеров применения	отсутствие прозрачности; сложность выбора архитектуры; жесткие требования к обучающей выборке; сложность выбора алгоритма обучения; ресурсоемкость процесса обучения

Продолжение таблицы 1		
1	2	3
Модели и методы на базе цепей Маркова	простота моделирования; единообразие анализа и проектирования	невозможность моделирования процессов с длинной памятью; узкая применимость моделей
Модели и методы на базе классификационно-регрессионных деревьев	масштабируемость; быстрота и простота процесса обучения; возможность учитывать категориальные переменные	неоднозначность алгоритма построения дерева; сложность вопроса останова

Нужно дополнительно отметить, что ни для одной из рассмотренных групп моделей (и методов) в достоинствах не указана точность прогнозирования. Согласно [14], это сделано в связи с тем, что точность прогнозирования того или иного процесса зависит не только от модели, но и от опыта исследователя, от доступности данных, от располагаемой аппаратной мощности и многих других факторов. Точность прогнозирования будет оцениваться для конкретных задач, решаемых в рамках данной работы.

На сегодняшний день наиболее распространенными моделями прогнозирования являются авторегрессионные модели (ARIMAX), а также нейросетевые модели (ANN).

## 1.5 Комбинированные модели

Одной из популярных современных тенденций в области создания моделей прогнозирования является создание комбинированных моделей и методов. Подобный подход дает возможность компенсировать недостатки

одних моделей при помощи других, и направлен на повышение точности прогнозирования, как одного из главных критериев эффективности модели.

Подход прогнозирования временного ряда осуществляется в два этапа. На первом этапе, на основании моделей распознавания образов (pattern recognition) выделяются гомогенные группы (patterns) временного ряда. На следующем этапе для каждой группы строится отдельная модель прогнозирования. При помощи вейвлет преобразования (wavelet transform) доступные значения временного ряда разделяются на несколько последовательностей, для каждой из которых строится отдельная модель ARIMA [14],[15].

В обзоре моделей прогнозирования энергопотребления рассматриваются следующие типы комбинаций [16]:

- *нейронные сети + нечеткая логика;*
- *нейронные сети + ARIMA;*
- *нейронные сети + регрессия;*
- *нейронные сети + GA + нечеткая логика;*
- *регрессия + нечеткая логика.*

В большинстве комбинаций, модели на основе нейронных сетей применяются для решения задачи кластеризации, а далее для каждого кластера строится отдельная модель прогнозирования на основе ARIMA, GA, нечеткой логики и др. В работе утверждается, что применение комбинированных моделей, выполняющих предварительную кластеризацию и последующее прогнозирование внутри определенного кластера, является наиболее перспективным направлением развития моделей прогнозирования.

Согласно работе [17], для кластеризации предлагается два метода: метод К-средних (K-mean) и метод нечетких С-средних (fuzzy C-mean). Целью обоих алгоритмов кластеризации является извлечение полезной информации из временного ряда для последующего прогнозирования. Авторы утвержда-

ют, что применение кластеризации дает возможность повысить точность прогнозирования.

Применение комбинированных моделей является направлением, которое при корректном подходе позволяет повысить точность прогнозирования. Главным недостатком комбинированных моделей является сложность и ресурсоемкость их разработки: нужно разработать модели таким образом, чтобы компенсировать недостатки каждой из них, не потеряв достоинств.

Ряд исследователей пошли по альтернативному пути [18] и разработали авторегрессионные модели, в основе которых лежит предположение о том, что временной ряд есть последовательность повторяющихся кластеров (patterns). Однако при этом разработчики не создавали комбинированных моделей, а определяли кластеры и выполняли прогноз на основании одной модели. Рассмотрим эти модели подробнее.

Рассмотрим модель прогнозирования направления движения индексов рынка(index movement) [19], учитывающая кластеры временного ряда. Пусть временной ряд содержит три значения -1, 0 и 1, которые характеризуют спад, стабильное состояние и подъем рынка соответственно. Кластером (pattern) называется последовательность  $Z_i^M = Z(i), Z(i+1), \dots, Z(i+M)$  для  $i=1, 2, \dots, N-M$ , где  $N$  – число доступных отчетов временного ряда  $Z(t)$ . Для определения прогнозного значения рассмотрена последняя доступная информация, а именно последовательность  $Z(N, M) = Z(N-M+1), Z(N-M+2), \dots, Z(N)$ , для которой определена ближайшая похожая (closet match)  $Z(Q, M) = Z(Q+1), Z(Q+2), \dots, Z(Q+M)$ . При этом функция, определяющая близость, имеет вид

$$F(N-M, Q) = \sum_{j=1}^M |Z(N-M+1) - Z(Q+1)|, \quad (1.5.1)$$

т.е. близость кластеров определяется простым сравнением. Далее вычисляется прогнозное значение

$$Z(N + 1) = Z(Q + M + 1). \quad (1.5.2)$$

Таким образом, в данной модели предполагается, что если в некоторый момент времени в прошлом рынок вел себя определенным образом, то в будущем его поведение повторится в связи с тем, что временной ряд является последовательностью кластеров.

Согласно работам [21],[22], существуют модели прогнозирования, основанная на модели авторегрессии, но принимающая во внимание кусочки временного ряда. Здесь прогнозное значение временного ряда определено выражением

$$Z(t) = \alpha_0 + \alpha_1 Z(t - 1) + \alpha_2 Z(t - 2) + \dots + \alpha_M Z(t - M), \quad (1.5.3)$$

которое является линейной авторегрессией порядка  $M$ . При этом коэффициенты авторегрессии  $\alpha_0, \alpha_1, \dots, \alpha_M$  определяются следующим образом. Предполагается, что существует  $K$  кусочков (векторов) длины  $M$  временного ряда, для которых выполняется выражение

$$\begin{bmatrix} Z(i_1) \\ Z(i_2) \\ \dots \\ Z(i_K) \end{bmatrix} = \alpha_0 \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} + \alpha_1 \begin{bmatrix} Z(i_1 - 1) \\ Z(i_2 - 1) \\ \dots \\ Z(i_K - 1) \end{bmatrix} + \dots + \alpha_M \begin{bmatrix} Z(i_1 - M) \\ Z(i_2 - M) \\ \dots \\ Z(i_K - M) \end{bmatrix}. \quad (1.5.4)$$

При определении ближайших векторов (closest vectors)  $Z(i_1-1), Z(i_1-2), \dots, Z(i_1-M), \dots, Z(i_K-1), Z(i_K-2), \dots, Z(i_K-M)$  используется значение линейной корреляции Пирсона между всеми возможными векторами и новейшим вектором (last available vector)  $Z(t-1), Z(t-2), \dots, Z(t-M)$ .

Разработчики рассмотренных выше моделей утверждают, что предложенные модели просты, прозрачны и эффективны для исследованных временных рядов [37]. При этом очевидно, что главными недостатками данных моделей являются [39],[40],[41]:

*–невозможность учитывать внешние факторы;*

- неоднозначность критерия определения похожей выборки;*
- сложность определения эффективной комбинации двух параметров  $M$  (длина векторов) и  $K$  (число векторов, принимаемых в расчет).*

В рамках диссертации установлено, что подход, предложенный авторами работ, является перспективным в области создания моделей прогнозирования временных рядов. Предложенная в диссертации модель прогнозирования развивает модели и устраняет все перечисленные выше недостатки: модель позволяет учитывать влияния внешних факторов; формулируется критерий определения похожей выборки для двух видов постановок задачи прогнозирования; количество параметров модели сокращается до одного, что существенно упрощает идентификацию модели [18].

## **1.6 Выводы**

1) Задача прогнозирования временных рядов имеет высокую актуальность для многих предметных областей и является неотъемлемой частью повседневной работы многих компаний.

2) Установлено, что к настоящему времени разработано множество моделей для решения задачи прогнозирования временного ряда, среди которых наибольшую применимость имеют авторегрессионные и нейросетевые модели.

3) Выявлены достоинства и недостатки рассмотренных моделей. Установлено, что существенным недостатком авторегрессионных моделей является большое число свободных параметров, требующих идентификации; недостатками нейросетевых моделей является ее непрозрачность моделирования и сложность обучения сети.

4) Определено, что наиболее перспективным направлением развития моделей прогнозирования с целью повышения точности является создание комбинированных моделей, выполняющих на первом этапе кластеризацию, а затем прогнозирование временного ряда внутри установленного кластера.



## **2 Модели экстраполяции временных рядов по выборке максимального подобия**

### **2.1 Модель с учетом внешних факторов**

#### **2.1.1 Выборки временных рядов**

Пусть дан временной ряд  $Z(t)$  и внешние факторы, представленные в виде временных рядов  $X_1(t), \dots, X_S(t)$ , соответствующие исходному ряду по отметкам времени. Требуется построить модель прогнозирования исходного временного ряда  $Z(t)$ , которая будет учитывать влияние значений внешних факторов  $X_1(t), \dots, X_S(t)$ .

Если внешние факторы  $X_1(t), \dots, X_S(t)$  имеют временное разрешение, отличное от разрешения исходного ряда  $Z(t)$ , то необходимо произвести дополнительные преобразования и привести внешние факторы в соответствие с исходным временным рядом  $Z(t)$  по отметкам времени.

В основу модели экстраполяции с учетом внешнего фактора положено предположение о повторяемости выборок временного ряда. Кроме того, известно, что для учета внешних факторов в авторегрессионных моделях вводятся дополнительные регрессоры.

В настоящем разделе сначала рассматривается задача аппроксимации одной выборки при помощи другой с учетом выборок внешних факторов, а затем результаты аппроксимации применяются для построения модели прогнозирования временного ряда.

#### **2.1.2 Аппроксимация выборки**

Пусть задана выборка исходного временного ряда  $Z_t^M$  и выборки  $X_{(1)t, \dots, X_{(S)t}^M$ , являются выборками внешних факторов, соответствующих  $Z_t^M$  по

отметкам времени. Аппроксимируем выборку  $Z_t^M$  с учетом выборок  $X_{(1)t, \dots, (S)t}^M$  по формуле

$$Z_t^M = \alpha_{S+1} Z_{t-k}^M + \alpha_S X_{(S)t}^M + \dots \alpha_1 X_{(1)t}^M + \alpha_0 I^M + E^M. \quad (2.1.2.1)$$

Здесь  $\alpha_{S+1}, \alpha_S, \dots, \alpha_0$  – коэффициенты, определяемые по методу наименьших квадратов. Вектор  $E^M$  – вектор значений ошибок аппроксимации,  $I^M$  – единичный вектор. Выражение (2.1.2.1) можно переписать в виде

$$\hat{Z}_t^M = \alpha_{S+1} Z_{t-k}^M + \alpha_S X_{(S)t}^M + \dots \alpha_1 X_{(1)t}^M + \alpha_0 I_M. \quad (2.1.2.2)$$

Постановка задачи аппроксимации выборки. Пусть дана функциональная зависимость (2.1.2.2). Необходимо определить такие значения коэффициентов  $\alpha_{S+1}, \alpha_S, \dots, \alpha_0$ , чтобы квадрат отклонений модельных значений выборки  $\hat{Z}_t^M$  от фактических  $Z_t^M$  был минимален

$$\sigma^2 = \sum_{i=1}^{M-1} (Z(t+i) - \hat{Z}(t+i))^2 \rightarrow \min. \quad (2.1.2.3)$$

После нахождения коэффициентов  $\alpha_{S+1}, \alpha_S, \dots, \alpha_0$ , необходимо оценить вектор ошибок  $E^M$ .

Выполним решение данной задачи. Пусть дана функциональная зависимость (2.1), тогда функция суммы квадратов имеет вид

$$S_k^M(\alpha_{S+1}, \dots, \alpha_0) = \sum_{i=0}^{M-1} (Z(t+i) - (\alpha_{S+1} Z(t-k+i) + \alpha_S X_{(S)}(t+i) + \dots \alpha_1 X_{(1)}(t+i) + \alpha_0))^2. \quad (2.1.2)$$

Функция называется функцией суммы квадратов (sum of squares function). Задача состоит в том, чтобы подобрать такие значения  $\alpha_1$  и  $\alpha_0$ , чтобы при подстановке их в (2.4) было получено минимальное возможное значение  $(\alpha_1, \alpha_0)$ .

Приведем задачу к матричному уравнению  $Z_X A = Z_Y$ , где

$$A = \begin{bmatrix} \alpha_{s+1} \\ \alpha_s \\ \dots \\ \alpha_1 \\ \alpha_0 \end{bmatrix}. \quad (2.1.2.5)$$

Решая матричное уравнение (2.1.2.5), определяем значения коэффициентов  $\alpha_{s+1}, \alpha_s, \dots, \alpha_0$ , соответствующие минимуму функции  $S_k^M$ . Решение будет найдено при помощи обратной матрицы

$$Z_X^{-1} Z_X A = Z_X^{-1} Z_Y, \quad (2.1.2.6)$$

$$A = Z_X^{-1} Z_Y. \quad (2.1.2.7)$$

Оценка ошибки аппроксимации. Ошибка определяется по формуле

$$E^M = Z_t^M - \hat{Z}_t^M = Z_t^M - (\alpha_1 Z_{t-k}^M + \alpha_0 I^M). \quad (2.1.2.8)$$

В настоящее время точность моделирования временных рядов  $E^M$  принято оценивать при помощи двух показателей:

–средняя абсолютная ошибка (mean absolute error, MAE)

$$MAE = \frac{1}{M} \sum_{i=t}^{t+M-1} |Z(i) - \hat{Z}(i)|; \quad (2.1.2.9)$$

–средняя абсолютная ошибка в процентах, (mean absolute percentage error, MAPE)

$$MAPE = \frac{1}{M} \sum_{i=t}^{t+M-1} \frac{|Z(i) - \hat{Z}(i)|}{Z(i)} 100\%. \quad (2.1.2.10)$$

Здесь и далее, говоря о точности моделирования временных рядов (аппроксимации, прогнозирования) будем применять значения показателей MAE (2.1.2.9) и MAPE (2.1.2.10).

В настоящем разделе была рассмотрена аппроксимация одной выборки временного ряда при помощи другой, лежащей на оси времени отчетов левее, то есть раньше. Данное свойство представления новых выборок временного ряда при помощи старых будет использовано для определения модели экстраполяции.

### 2.1.3 Подобие выборок

Свойство двух выборок, заключенное в том, что одна выборка может быть выражена через другую с помощью линейной зависимости (2.2.1.1), назовем подобием двух выборок.

Покажем, что для общего случая линейной регрессии минимум ошибки регрессии соответствует максимуму линейной корреляции Пирсона. Пусть дана модель

$$\hat{Y} = \alpha_1 X + \alpha_0 I. \quad (2.1.3.1)$$

Тогда функция суммы квадратов определяется как разность модельных  $\hat{Y}(i)$  и фактических значений  $Y(i)$  некоторых наблюдений

$$S_{reg} = \sum_{i=1}^M (\hat{Y}(i) - Y(i))^2. \quad (2.1.3.2)$$

Обозначим  $\bar{Y}$  – среднее значение модельных и фактических наблюдений, которые по свойству линейной регрессии равны

$$\bar{Y} = \frac{1}{M} \sum_{i=1}^M Y(i) = \frac{1}{M} \sum_{i=1}^M \hat{Y}(i). \quad (2.1.3.3)$$

Известно, что сумма квадратов отклонений исследуемых наблюдений  $Y(i)$  от среднего значения  $\bar{Y}$  складывается из суммы квадратов отклонений модельных значений  $\hat{Y}(i)$  от  $\bar{Y}$  и суммы квадратов ошибок регрессии [15],

определенной в выражении (2.1.3.2). Таким образом имеет место соотношение

$$\sum_{i=1}^M (Y(i) - \bar{Y})^2 = \sum_{i=1}^M (\hat{Y}(i) - \bar{Y})^2 + \sum_{i=1}^M (Y(i) - \hat{Y}(i))^2. \quad (2.1.3.4)$$

Выразим ошибку регрессии  $\sum_{i=1}^M (Y(i) - \hat{Y}(i))^2$  и получим

$$\begin{aligned} \sum_{i=1}^M (Y(i) - \hat{Y}(i))^2 &= \sum_{i=1}^M Y(i) - \bar{Y})^2 - \sum_{i=1}^M (\hat{Y}(i) - \bar{Y})^2 \Rightarrow \\ S_{reg} &= \sum_{i=1}^M (Y(i) - \hat{Y}(i))^2 = \sum_{i=1}^M (Y(i) - \bar{Y})^2 \left( 1 - \frac{\sum_{i=1}^M (\hat{Y}(i) - \bar{Y})^2}{\sum_{i=1}^M (Y(i) - \bar{Y})^2} \right). \end{aligned} \quad (2.1.3.5)$$

Величина

$$R^2 = \frac{\sum_{i=1}^M (\hat{Y}(i) - \bar{Y})^2}{\sum_{i=1}^M (Y(i) - \bar{Y})^2} \in [0,1], \quad (2.1.3.6)$$

называется квадратом множественного коэффициента корреляции. Иногда данный коэффициент называется коэффициентом детерминации. Стоит обратить внимание на то, что при  $Y(i)=const$  для  $i=\{1,2,3,\dots\}$  знаменатель  $R^2$  обращается в ноль. Однако по свойству регрессии, для такого случая модельные значения также будут постоянными  $\hat{Y}(i), (i) = const$  для  $i=\{1,2,3,\dots\}$  знаменатель  $R^2=1$ .

Известно, что на практике такой случай невозможен, в связи с тем, что значения  $Y(i)$ , как правило, являются результатами измерений.

Преобразовав (2.15) получим выражение

$$S_{reg} = \sum_{i=1}^M (Y(i) - \bar{Y})^2 (1 - R^2). \quad (2.1.3.7)$$

При этом сумма квадратов отклонений исследуемых наблюдений  $Y(i)$  от среднего значения  $\hat{Y}(i)$  является величиной неизменной и характеризует свойство наблюдаемого процесса. Таким образом, из равенства (2.1.3.7) очевидно, что при  $R^2 \rightarrow \max, S_{reg} \rightarrow \min$ .

Далее рассмотрим коэффициент линейной корреляции Пирсона  $\rho$ , определяемый выражением

$$\rho(X, Y) = \frac{\sum_{i=1}^M (X(i) - \bar{X})(Y(i) - \bar{Y})}{\sqrt{\sum_{i=1}^M (X(i) - \bar{X})^2 \sum_{i=1}^M (Y(i) - \bar{Y})^2}} \in [-1, 1] \quad (2.1.3.8)$$

Связь двух рассматриваемых коэффициентов имеет следующий вид  $\rho^2(\hat{Y}, Y) = R^2$

$$\rho(\hat{Y}, Y) = \text{sign}(\alpha_1) \sqrt{R^2}. \quad (2.1.3.9)$$

Таким образом, известно, что модуль величины  $\rho(\hat{Y}(i), Y)$  равен модулю величины  $R$ , а, следовательно, можно сформулировать следующее свойство

$$R^2 \rightarrow \max, S_{reg} \rightarrow \min; \quad (2.1.3.10)$$

$$|\rho(\hat{Y}, Y)| \rightarrow \max, S_{reg} \rightarrow \min.$$

Вернемся к исследуемым выборкам временного ряда. Пусть дан временной ряд  $Z_1^T$ , для некоторой выборки  $Z_t^M$ , принадлежащей данному временному ряду, определим все значения  $S_k^M(\alpha_1, \alpha_0)$  для  $k = \{1, 2, \dots, t-1\}$ ,  $M = \text{const}$ . Далее, во множестве значений найдем минимальное

$$S_{k \min}^M = \min(S_1^M, S_2^M, \dots, S_{t-1}^M). \quad (2.1.3.11)$$

Согласно свойству (2.1.3.10) минимум ошибки регрессии  $S_{k \min}^M$  соответствует максимуму модуля коэффициента линейной корреляции (2.1.3.8). То

есть если для  $k = \{1, 2, \dots, t-1\}$  и  $M = \text{const}$  определить множество значений модуля корреляции

$$\rho_k^M = \left| \rho(\hat{Z}_t^M, Z_t^M) = \frac{\sum_{i=1}^M \hat{Z}(t+i) - \bar{Z})(Z(t+i) - \bar{Z})}{\sqrt{\sum_{i=1}^M (\hat{Z}(t+i) - \bar{Z})^2 \sum_{i=1}^M (Z(t+i) - \bar{Z})^2}} \right| \in [0, 1], \quad (2.1.3.12)$$

а после определить максимальное значение полученного множества

$$\rho_{k \max}^M = \max(\rho_1^M, \rho_2^M, \dots, \rho_{t-1}^M), \quad (2.1.3.13)$$

где задержка  $k_{\min}$  из выражения (2.1.3.11) и задержка  $k_{\max}$  из выражения (2.1.3.13) будут равны между собой, т. е.  $k_{\min} = k_{\max}$ . Проведенные расчеты подтверждают данное утверждение.

Определенную в (2.1.3.11) и (2.1.3.13) задержку, соответствующую минимуму ошибки регрессии  $S_{k \min}^M$  и максимуму модуля корреляции  $\rho_{k \max}^M$  обозначим, а выборку  $Z_{t-k \max}^M$  назовем выборкой максимального подобия (most similar pattern). Выборка максимального подобия  $Z_{t-k \max}^M$  является выборкой, которая при подстановке в уравнение (2.1.3.10), дает в результате значения выборки  $\hat{Z}_t^M$ , которая максимально точно описывает исходную выборку  $Z_t^M$ .

При реализации вычислений для определения выборки максимального подобия  $Z_{t-k \max}^M$  можно использовать как значения ошибки  $S_k^M$ , так и значения модуля корреляции  $\rho_k^M$ .

## 2.2 Применение нейронных сетей для нахождения максимального значения линейной корреляции Пирсона

В разделе 2.1.3 было сформулировано следующее свойство (2.1.3.10), из чего следует, что для того чтобы ошибка регрессии была минимальной, нам необходимо получить максимальное значение коэффициента линейной регрессии Пирсона. Анализируя коэффициент линейной корреляции Пирсона  $\rho$ , определяемый выражением (2.1.3.8).

Мы можем сделать вывод, что для нахождения максимального значения коэффициента  $\rho=1$ , нам необходимо найти такие значения  $X(i)$ ,  $\bar{X}$  и  $Y(i)$ ,  $\bar{Y}$  из временного ряда, чтобы выполнялось равенство

$$\sum_{i=1}^M (X(i) - \bar{X})(Y(i) - \bar{Y}) = \sqrt{\sum_{i=1}^M (X(i) - \bar{X})^2 \sum_{i=1}^M (Y(i) - \bar{Y})^2}. \quad (2.2.1)$$

На основании полученных результатов в главе 1, делаем вывод, что для нахождения значений коэффициентов  $X(i)$ ,  $\bar{X}$  и  $Y(i)$ ,  $\bar{Y}$  из временного ряда, лучше использовать нейросетевые модели.

В работе [5] описана модель многослойной нейронной сети, обучение которой проводилось методом обратного распространения ошибки, которую можно применить для нахождения значений  $X(i)$ ,  $\bar{X}$  и  $Y(i)$ ,  $\bar{Y}$ .

## 2.3 Описание модели

Пусть дан исходный временной ряд  $Z_1^T$  и внешние факторы  $X_{(1)1}^{T+P}, \dots, X_{(S)1}^{T+P}$ . Для исходного временного ряда требуется определить значения  $Z_{T+1}^P$ , учитывая доступные значения  $Z_1^T$  и  $X_{(1)1}^{T+P}, \dots, X_{(S)1}^{T+P}$ . Используя свойство выборок, сформулированное в разделе 2.1.3, выразим выборку  $Z_{T+1}^P$  через некоторую выборку  $Z_T^P$ , лежащую внутри исходного временного ряда  $Z_1^T$  и выборки  $X_{(1)1}^{T+P}, \dots, X_{(S)1}^{T+P}$  следующим образом



$$\hat{Z}_{T+1}^P = \alpha_{S+1} Z_T^P + \alpha_S X_{(S)T+1}^P + \dots + \alpha_1 X_{(1)T+1}^P + \alpha_0 I^P. \quad (2.3.1)$$

Алгоритм определения выборки  $Z_T^P$  состоит из двух шагов.

1. Определение выборки  $Z_{k \max}^M$ .
2. Вычисление  $Z_T^P$ .

Определение выборки  $Z_{k \max}^M$ . На данном шаге, как в случае экстраполяции без учета внешних факторов, для выборки  $Z_{T-M+1}^M$ , содержащей значения процесса непосредственно перед моментом прогноза, находим выборку максимального подобия  $Z_{k \max}^M$ . Поиск выборки максимального подобия осуществляем перебором всех возможных значений задержек  $k = \{1, 2, \dots, T-M-1\}$ . Для каждого значения  $k$  из указанного диапазона решаем задачу аппроксимации, в результате которой определяем коэффициенты  $\alpha_{S+1}, \alpha_S, \dots, \alpha_0$ , соответствующие  $k$ . Далее для вычисленных коэффициентов определяем значение модельной выборки, на основании которых вычисляем значение ошибки регрессии  $S_k^M$  (2.1.3.7). После того, как множество значений  $S_k^M$  для  $k = \{1, 2, \dots, T-M-1\}$  получено, определяем значение  $S_{k \max}^M$  (2.1.3.11) и соответствующую выборку  $Z_{T-M+1}^M$ . Как и ранее обозначим задержку  $k \max^* = T-M+1-k \max$ , а выборку максимального подобия  $Z_{k \max}^M$  (SimilarHistory).

Вычисление  $Z_T^P$ . Используем выборку  $Z_{k \max^*+M}^P$ , то есть выборку, расположенную на оси времени сразу за выборкой максимального подобия. Для случая учета одного внешнего фактора положение выборок показано на рисунке 2 - Положение выборок на оси времени.

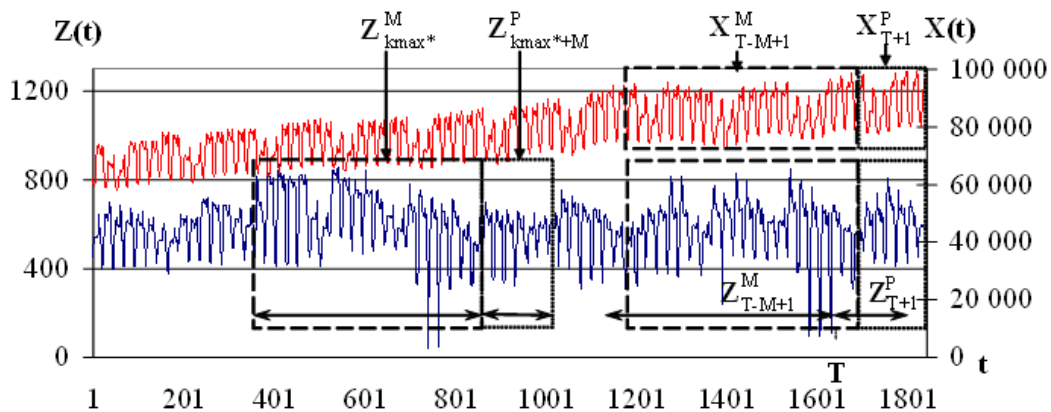


Рисунок 2 – Положение выборок на оси времени

Для удобства назовем выборки следующим образом:

- выборка новой истории  $Z_{T-M+1}^M$  (NewHistory);
- выборка максимального подобия  $Z_{kmax*}^M$  (SimilarHistory);
- выборка базовой истории  $Z_T^P$  (BaseHistory);
- выборка истории внешнего фактора  $X_{T-M+1}^M$  (FactorHistory);
- выборка прогноза внешнего фактора  $X_{T+1}^P$  (FactorForecast).

Экстраполированные значения выборки  $\hat{Z}_{T+1p}$  (Forecast) определяем по формуле

$$\hat{Z}_{T+1}^P = \alpha_{S+1} Z_{kmax*+M}^P + \alpha_S X_{(S)T+1}^P + \dots + \alpha_1 X_{(1)T+1}^P + \alpha_o I^P = EMMSPX(M), \quad (2.3.2)$$

которая представляет собой расширенную модель экстраполяции временных рядов по выборке максимального подобия (extrapolation model on most similar pattern extended, далее EMMSPX).

Таким образом, по свойству процессов учет внешних факторов необходим в соответствующий момент времени. В случае, если отсутствуют значения внешних факторов на будущий период, а именно, значения выборок  $X_{(1)T+1}^P, \dots, X_{(S)T+1}^P$ , то необходимо или спрогнозировать прежде внешние факторы, а потом исследуемый основной процесс, либо удалить внешние факторы

из модели. В случае прогноза урожайности часто в качестве внешнего фактора используется прогноз температуры окружающей среды, формируемый Гидрометцентром России. Таким образом, по свойству данных процессов учет температуры при прогнозе энергопотребления должен производиться в соответствующие моменты времени.

В завершении данного раздела необходимо отметить особенности EMMSPX:

- модель относится к классу авторегрессионных моделей прогнозирования, учитывающих дискретные внешние факторы;
- модель может учитывать несколько дискретных внешних факторов;
- модель эффективно работает с неравноотстоящими временными рядами;
- модель работает со стационарными и нестационарными временными рядами;
- модель имеет один параметр  $M$ , определение которого подробно рассмотрено в третьей главе диссертации;
- экстраполяция  $P$  значений временного ряда производится за одну итерацию.

## 2.4 Варианты моделей по выборке максимального подобия

Модель (2.3.2) можно разделить на две части – модель авторегрессии и модель внешнего фактора

$$\hat{Z}_{T+1}^P = \underbrace{\alpha_2 Z_{k \max^* + M}^P + \alpha_0 I^P}_{\text{авторегрессия}} + \underbrace{\alpha_1 X_{T+1}^P}_{\text{внешний фактор}}. \quad (2.4.1)$$

Согласно работе [46],[48], модель авторегрессии и модель внешнего фактора могут быть модифицированы с целью повышения точности прогнозирования.

Авторегрессионная EMMSP со множеством выборок. Прогнозные значения  $\hat{Z}_{T+1}^P$  вычисляются как линейная комбинация нескольких выборок базовой истории с различными задержками

$$\hat{Z}_{T+1}^P = \alpha_q Z_{T_q}^P + \alpha_{q-1} X_{T_{q-1}}^P + \dots + \alpha_1 X_{T_1}^P + \alpha_o I^P. \quad (2.4.2)$$

На практике количество используемых выборок исходного временного ряда не превышает двух, т. е. прогнозные значения  $\hat{Z}_{T+1}^P$  вычисляются как линейная комбинация двух выборок базовой истории с различными задержками

$$\hat{Z}_{T+1}^P = \alpha_2 Z_{T_2}^P + \alpha_1 X_{T_1}^P + \alpha_o I^P. \quad (2.4.3)$$

Однако не исключено, то в ряде задач полезным будет использование трех-четырех выборок. Данная модель (2.4.3) применялась для прогнозирования временных рядов урожайности картофеля.

Авторегрессионная EMMSP с использованием q-той степени значений выборок. Модель прогнозирования представляет собой линейную комбинацию степеней выборки максимального подобию

$$\hat{Z}_{T+1}^P = \alpha_q (Z_T^P)^q + \alpha_{q-1} (Z_T^P)^{(q-1)} + \dots + \alpha_1 Z_T^P + \alpha_o I^P. \quad (2.4.4)$$

Здесь  $(Z_T^P)^n$  – выборка, значениями которой являются n-ные степени значений временного ряда

$$(Z_T^P)^n = \begin{bmatrix} Z^n(T) \\ \dots \\ Z^n(T + P) \end{bmatrix}. \quad (2.4.5)$$

В рамках диссертационной работы проводились исследования эффективности повышения степени выборки максимального подобия, которые показали, что точность прогнозирования для некоторых временных рядов повышается при использовании второй степени, то есть модели

$$\hat{Z}_{T+1}^P = \alpha_2 (Z_T^P)^2 + \alpha_1 Z_T^P + \alpha_o I^P. \quad (2.4.6)$$

Дальнейшее увеличение степени выборки максимального подобия эффективности модели не повышает.

EMMSPX с использованием  $q$ -той степени значений внешних факторов. В случае доступности данных лишь по одному внешнему фактору возможна модификация модели аналогично (2.4.4)

$$\hat{Z}_{T+1}^P = \alpha_1 Z_T^P + \alpha_o I^P + \beta_q (X_T^P)^q + \beta_{q-1} (X_T^P)^{(q-1)} + \dots + \beta_1 X_T^P. \quad (2.4.7)$$

При этом значения выборки определяются как  $n$ -ные степени значений внешнего фактора (2.4.5).

Важно отметить, что при построении моделей (2.4.2) – (2.4.7) применяется единообразный подход к анализу и проектированию моделей, описанный в разделе 2.2. настоящей работы.

Ограничения применимости модели экстраполяции временных рядов по выборке максимального подобия. Предложенные в настоящей главе модели EMMSP и EMMSPX могут применяться для прогнозирования временных рядов на значений вперед при выполнении набора условий.

- 1) Длина временного ряда составляет не менее  $500 P - 700 P$ . Временной ряд равноотстоящий; в случае неравноотстоящего временного ряда применение модели возможно только при условии, что его равноотстоящие отрезки должны быть содержать не менее, чем  $100 P - 150 P$  отсчетов.
- 2) Временной ряд относится к классу временных рядов с длинной памятью.

Задача прогнозирования на  $P$  значений вперед относится к классу краткосрочного или среднесрочного прогнозирования данного типа временного ряда. Не рекомендуется использовать разработанную модель для долгосрочного прогнозирования.

3) В случае учета набора дискретных внешних факторов, их временное разрешение должно быть приведено к разрешению исходного временного ряда. Длина исходного временного ряда и временных рядов внешних факторов может не совпадать [50].

## 2.5. Выводы

- 1) В настоящей главе предложены модели экстраполяции временных рядов по выборке максимального подобия с учетом и без учета внешних факторов.
- 2) Предложенные модели относятся к классу авторегрессионных моделей прогнозирования и обладают всеми достоинствами, характерными для данного класса.
- 3) Предложенные модели устраняют существенный недостаток указанного класса – большое число свободных параметров, требующих идентификации. Обе модели экстраполяции по выборке максимального подобия имеют единственный параметр.
- 4) Разработаны варианты моделей по выборке максимального подобия, использование которых может приводить к повышению точности прогнозирования временного ряда.

### **3. Многослойный персептрон**

#### **3.1 Структура**

Обычно сеть состоит из множества сенсорных элементов (входных узлов или узлов источника), которые образуют входной слой, одного или нескольких скрытых слоев вычислительных нейронов и одного выходного слоя нейронов. Входной сигнал распространяется по сети в прямом направлении, от слоя к слою. Такие сети обычно называют многослойными персептронами. Они представляют собой обобщение однослойного персептрона рассмотренного, в работе.

Многослойные персептроны успешно применяются для решения разнообразных сложных задач. При этом обучение с учителем выполняется с помощью такого популярного алгоритма, как алгоритм обратного распространения ошибки. Этот алгоритм основывается на коррекции ошибок. Его можно рассматривать как обобщение столь же популярного алгоритма адаптивной фильтрации-алгоритма минимизации среднеквадратичной ошибки [2].

Обучение методом обратного распространения ошибки предполагает два прохода по всем слоям сети: прямого и обратного. При прямом проходе образ (входной вектор) подается на сенсорные узлы сети, после чего распространяется по сети от слоя к слою. В результате генерируется набор выход-

ных сигналов, который и является фактической реакцией сети на данный входной образ. Во время прямого прохода все синоптические веса фиксированы. Во время обратного прохода все синоптические веса настраиваются в соответствии с правилом коррекции ошибок, а именно фактический выход сети вычитается из желаемого(целевого) отклика, в результате чего формируется сигнал ошибки. Этот сигнал впоследствии распространяется по сети в направлении, обратном направлению синоптических связей. Отсюда и название – алгоритм обратного распространения ошибки. Синоптические веса настраиваются с целью максимального приближения выходного сигнала сети к желаемому в статистическом смысле. Алгоритм обратного распространения ошибки в литературе иногда называют упрощённо – алгоритмом обратного распространения. Процесс обучения, реализуемый этим алгоритмом, называется обучением на основе обратного распространения.

Многослойные персептроны имеют три отличительных признаки

1) Каждый нейрон сети имеет нелинейную функцию активации. Важно подчеркнуть, что данная нелинейная функция является гладкой (т.е. всюду дифференцируемой) [8]. В отличие от жесткой пороговой функции, и используемой в персептроне Розенблатта. Самой популярной формой функции. Удовлетворяющей этому требованию, является сигмоидальная, определяемая логистической функцией

$$y_i = \frac{1}{1 + \exp(-v_j)}, \quad (3.1.1)$$

где  $v_j$  - индуцированное локальное поле (т.е. взвешенная сумма всех синоптических входов плюс пороговое значение) нейрона  $j$ ;  $y_i$  - выход нейрона. Наличие нелинейности играет очень важную роль, так как в противном случае отображение «вход-выход» сети можно свести к обычному однослойному персептрону. Более того, использование логистической функции мотивировано биологически, так как в ней учитывается восстановительная фаза реального нейрона.



2) Сеть содержит один или несколько слоев скрытых нейронов, не являющийся частью входы или выхода сети. Эти нейроны позволяют сети обучаться решению сложных задач, последовательно извлекая наиболее важные признаки из входного образа (вектора).

3) Сеть владеет высокой степенью связности, реализуемой посредством синоптических соединений. Изменение уровня связности требует изменения множества синоптических соединений или их весовых коэффициентов.

Комбинация всех этих свойств наряду с способностью к обучению на собственном опыте обеспечивает вычислительную мощь многослойного персептрона. Однако эти же качества являются причиной неполноты современных знаний о поведении такого рода сетей. Во-первых, распределённая форма нелинейности и высокая связность сети существенно осложняют теоретический анализ многослойного персептрона. Во-вторых, наличие скрытых нейронов делает процесс обучения более трудным для визуализации. Именно в процессе обучения необходимо определить, какие признаки входного сигнала следует представлять скрытыми нейронами. Тогда процесс обучения становится еще более сложными, поскольку поиск должен выполняться в очень широкой области возможных функций, а выбор должен производиться среди альтернативных представлений входных образов.

Стандартная  $L$  - слойная сеть прямого распространения информации состоит из слоя входных элементов (будем считать, что он не включается в сеть в качестве самостоятельного слоя),  $(L-1)$  скрытых слоев и выходного слоя, соединенных последовательно в прямом направлении и не содержащих связей между элементами внутри слоя и обратных связей между слоями. На рисунке 3 – Типовая архитектура трехслойного персептрона, в качестве примера приведена структура трехслойной сети (трехслойного персептрона).

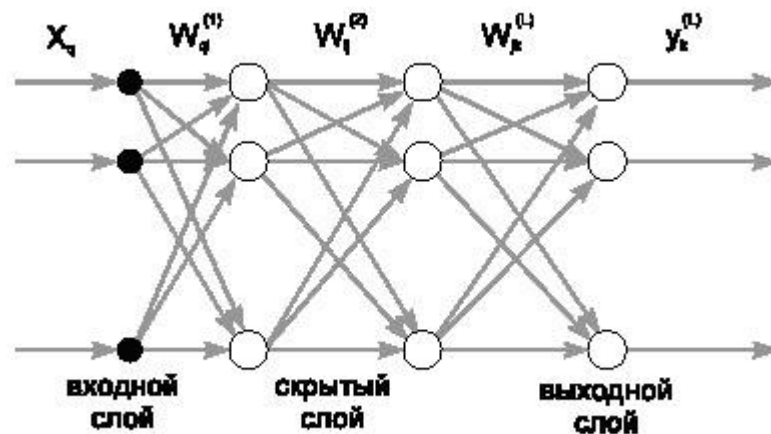


Рисунок 3 — Типовая архитектура трехслойного персептрона

В многослойных персептронах нейроны располагаются слоями. Обратные связи в них отсутствуют и информация передается лишь от входа к выходу. Они состоят из входного слоя, одного или более скрытых слоев (названных так, потому что они не имеют непосредственных связей с "внешним миром") и выходного слоя. Для обучения многослойных персептронов часто применяется алгоритм обучения с обратным распространением ошибки. Многослойные персептроны были предложены и исследованы в 1960-х годах Розенблаттом, Минским, Пейпертом и др. Они оказались весьма эффективными для решения разнообразных задач распознавания, управления и прогнозирования.

### 3.2 Архитектура сети

Тип входных сигналов: целые или действительные.

Тип выходных сигналов: действительные из интервала, заданного функцией активации или выхода нейронов.

Функция активации или выхода нейронов: сигмоидальная (функция Ферми, гиперболический тангенс и другие).

Рассмотрим функцию  $y = f(x)$ , которая ставит в соответствие  $n$ -мерному вектору  $x$  некоторый  $m$ -мерный вектор  $y$ . Например, в задаче классификации  $x$  - это вектор значений  $n$  признаков распознаваемого объекта, вектор  $y$ , состоящий из одной единицы и остальных нулей, является индикатором класса, к которому принадлежит образ  $x$  (позиция единицы в записи вектора  $y$  означает номер класса). Функция  $f$  ставит в соответствие каждому объекту тот класс, к которому он принадлежит.

Согласно [2],[8], обучение сети состоит в таком выборе весов связей между нейронами, чтобы суммарная среднеквадратичная ошибка нейросети для образов обучающей последовательности была минимальной. После обучения перцептрона проводится проверка эффективности его работы. Для этого множество образов, классифицированных учителем, разделяется на две части: обучающую и контролирующую последовательности, причем первая используется для обучения, а вторая – для определения эффективности работы обученного перцептрона. Один из таких критериев эффективности - доля правильно классифицированных образов тестирующей последовательности.

### **3.2.1 Алгоритм обучения с обратным распространением ошибки**

Алгоритм обратного распространения ошибки предложенный Румельхартом Мак-Клеllandом рассмотрим применительно к сетям прямого распространения сигналов. Основная его идея очень проста. Средняя квадратическая ошибка ИНС естественно зависит от весов всех связей ИНС и может быть представлена в виде поверхности в пространстве этих весов с многими локальными и глобальным минимумами. Цель обучения ИНС состоит в определении таких значений весов ее связей, которые обеспечивают в идеале глобальный минимум средней квадратической ошибки ИНС.

Ошибку нейросети представим в виде поверхности в зависимости от весов ее связей

$$E(w) = E(w_1, w_2, \dots, w_n), \quad (3.2.1.1)$$

где  $w = (w_1, w_2, \dots, w_n)'$  - матрица весов нейронов. При помощи метода наискорейшего спуска путем изменения весов стремятся по возможности быстрее найти глобальный минимум функции ошибки  $E(w)$ , т.е. такую комбинацию весов, при которой суммарная ошибка минимальна по всем образам обучающей последовательности. Изменение веса связи  $w_{ij}$  от  $i$ -го нейрона слоя  $n-1$  к  $j$ -му нейрону следующего слоя  $n$  выбирается пропорционально частной производной от функции ошибки  $E(w)$  по  $w_{ij}$

$$\Delta w_{ij}^n = \frac{-\eta \partial E(w)}{\partial w_{ij}}, \quad (3.2.1.2)$$

где  $\eta$  - коэффициент коррекции ( $0 < \eta < 1$ ).

Ошибка при распознавании  $p$ -го образа обучающей последовательности составляет

$$E_p = 0,5 \sum_{j=1}^m (t_{pj} - o_{pj}^{(N)})^2 = 0,5 \sum_{j=1}^m (y_{pj} - o_{pj}^{(N)})^2, \quad (3.2.1.3)$$

где  $m$  - число выходных нейронов сети,  $o_{pj}^{(N)}$  и  $y_{pj} = t_{pj}$  - действительный и требуемый выход нейрона  $j$  выходного слоя  $N$  сети для  $p$ -го образа обучающей последовательности. Для обозначения действительного и требуемого выхода нейросети используем соответственно обозначения  $o$  от output = выход и  $t$  от teacher = учитель.

Суммарная ошибка по всем образам обучающей последовательности составит

$$E = \sum E_p, \quad (3.2.1.4)$$

или

$$E(w) = \frac{1}{2} \sum_{j,p} (t_{pj} - o_{pj}^{(N)})^2 = 0,5 \sum_{j,p} (y_{pj} - o_{j,p}^{(N)})^2. \quad (3.2.1.5)$$

Частную производную от функции ошибки  $E_p(w)$  по  $w_{ij}$  представим в виде

$$\frac{\partial E_p}{\partial w_{ij}} = \frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial z_{pj}} \frac{\partial z_{pj}}{\partial w_{ij}}. \quad (3.2.1.6)$$

Здесь под  $o_{pj}$ , как и ранее, подразумевается действительный выход нейрона  $j$ , а под  $z_{pj}$  – взвешенная сумма его входных сигналов для  $p$ -го образа обучающей последовательности, то есть аргумент функции активации или выхода. Так как множитель  $do_{pj}/dz_{pj}$  – это производная этой функции по ее аргументу, то производная функции активации должна быть определена на всей области определения активности  $z$ . В связи с этим функция активации в виде единичного скачка не применима в сетях, использующих алгоритм обучения с обратным распространением ошибки. Согласно работе [56], в них применяются непрерывные функции: функция Ферми, гиперболический тангенс и другие. В случае гиперболического тангенса имеем:  $do/dz = 1 - z^2$ . А для сигмоидальной функции (1.11) при  $k = 1$

$$do/dz = o(1-o). \quad (3.2.1.7)$$

Третий множитель  $\partial z_{pj}/\partial w_{ij}$ , очевидно, равен выходу нейрона предыдущего слоя  $o_{pi}^{(n-1)}$ .

Первый множитель в (3.2.1.7) представим следующим образом

$$\frac{\partial E_p}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial o_{pk}} \frac{\partial o_{pk}}{\partial z_{pk}} \frac{\partial z_{pk}}{\partial o_{pj}} = \sum_k \frac{\partial E_p}{\partial o_{pk}} \frac{\partial o_{pk}}{\partial z_{pk}} w_{jk}^{(n+1)}. \quad (3.2.1.8)$$

Здесь суммирование по  $k$  выполняется по всем нейронам следующего слоя  $n+1$ . Введем новую переменную

$$\delta_{pj}^{(n)} = - \frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial z_{pj}}, \quad (3.2.1.9)$$

где  $do_{pj}/dz_{pj} = S'(z_{pj})$  - значение первой производной функции активации или выхода нейрона  $j$  для  $p$ -го образа обучающей последовательности. Тогда можно получить следующую рекуррентную формулу для расчета величины ошибки  $\delta_{pj}(n)$   $j$ -нейрона слоя  $n$  на основе ошибок  $\delta_{pk}^{(n+1)}$  нейронов следующего слоя  $n+1$  для  $p$ -го образа обучающей последовательности

$$\partial_{pj}^{(n)} = \left[ \sum_k \delta_{pk}^{(n+1)} w_{jk}^{(n+1)} \right] \frac{do_{pj}}{dz_{pj}} = S(z_{pj}) \sum_k \delta_{pk}^{(n+1)} w_{jk}. \quad (3.2.1.10)$$

Для выходного же слоя

$$\delta_{pj}^{(N)} = (t_{pj}^{(N)} - o_{pj}^{(N)}) S(z_{pj}) = (y_{pj}^{(N)} - o_{pj}^{(N)}) S(z_{pj}). \quad (3.2.1.11)$$

Соотношение (3.2.1.10) для изменения веса  $w_{ij}$  после предъявления сети  $p$ -го образа обучающей последовательности можно представить теперь в виде

$$\Delta w_{ij}^{(n)} = \eta \delta_{pj}^{(n)} o_{pj}^{(n-1)}, \quad (3.2.1.12)$$

причем ошибка  $\delta_{pj}$  для выходного нейрона определяется на основе (3.2.1.11). Изменение веса связи  $i$ -го нейрона предшествующего слоя  $n-1$  с  $j$ -м нейроном следующего слоя  $n$  равно произведению коэффициента коррекции  $\eta$ , выхода  $i$ -го нейрона  $o_{pi}$  предшествующего слоя  $n-1$  и ошибки  $\delta_{pj}$   $j$ -го нейрона следующего слоя  $n$  при классификации  $p$ -го образа обучающей последовательности. Приведем итоговые соотношения для расчета ошибки  $\delta_{pj}$  для нейронов с сигмоидальной функцией активации или выхода (при  $k = 1$ )

$$\begin{aligned} \delta_{pj} &= o_{pj} (1 - o_{pj}) (t_{pj} - o_{pj}), \\ \delta_{pj} &= o_{pj} (1 - o_{pj}) \sum_k \delta_{pk} w_{jk}. \end{aligned} \quad (3.2.1.13)$$

Иногда для придания процессу коррекции весов некоторой инерционности, сглаживающей резкие скачки при перемещении по поверхности целевой функции, соотношение (3.2.1.13) дополняется изменением веса на предыдущем шаге процесса обучения сети

$$\Delta w_{ij}^{(n)}(r) = \eta (\mu \Delta w_{ij}^{(n)}(r-1) + (1 - \mu) \delta_{pj}^{(n)} o_{pj}^{(n-1)}), \quad (3.2.1.14)$$

где  $\mu$  – коэффициент инерционности,  $r$  – номер текущего шага процесса обучения сети.

Таким образом, алгоритм обучения ИНС с обратным распространением ошибки формируется следующим образом:

1. Множество образов, классифицированных учителем, разделяется на обучающую и контролирующую последовательности. Обычно обучающая последовательность содержит больше образов, чем проверочная. Производится инициализация всех весов, включая пороговые, небольшими случайными величинами (обычно в диапазоне  $[-1; +1]$ ). Это определяет начальную точку на поверхности ошибок для метода градиентов [11].

2. Производится прямой проход сети для первого образа обучающей последовательности от входного слоя через скрытые слои к выходному слою: каждый нейрон суммирует произведения входов на веса и выдает результат функции активации, примененной к этой сумме, на входы нейронов следующего слоя. Рассчитываются значения выходов нейросети. Напомним, что

$$z_j^{(n)} = \sum_{i=0}^M o_i^{(n-1)} w_{ij}^{(n)}, \quad (3.2.1.15)$$

где  $M$  – число нейронов в слое  $n-1$  с учетом нейрона с постоянным выходом  $+1$  (BIAS), задающего смещение;  $o_i^{(n-1)} = y_i^{(n-1)} = x_{ij}^{(n)}$  –  $i$ -й вход нейрона  $j$  слоя  $n$ .

$$y_j^{(n)} = S(z_j^{(n)}), \quad (3.2.1.16)$$

где  $S(\bullet)$  – сигмоид

$$y_q^{(0)} = x_q. \quad (3.2.1.17)$$

3.  $x_q$  –  $q$ -ая составляющая вектора входного образа.

4. Вычисляются разность между действительным и требуемым выходом каждого нейрона  $j$  выходного слоя, по ней ошибка  $\delta_{pj}^{(N)}$  этого нейрона для  $p$ -го образа обучающей последовательности по формуле, а по  $\delta_{pj}^{(N)}$  по формуле (3.2.1.15) определяются изменения весов  $\Delta_p w^{(N)}$  нейронов выходного слоя  $N$  при распознавании  $p$ -го образа обучающей последовательности.

5. При несовпадении действительного и требуемого выходов нейросети имеет место ошибка в распознавании (классификации) образов. Эта ошибка распространяется в обратном направлении от выходного слоя нейронов к входному. По известным ошибкам нейронов выходного слоя сначала определяются ошибки на выходах нейронов предпоследнего слоя, а по ним осуществляется коррекция весов для входов этих нейронов. На основе ошибок выходов нейронов предпоследнего слоя определяются ошибки нейронов предпоследнего слоя и т.д. Эта процедура повторяется до нейронов первого слоя.

6. То есть по формулам (3.2.1.14) и (3.2.1.15) (или (3.2.1.16) и (3.2.1.17)) рассчитываются соответственно  $\delta^{(n)}$  и  $\Delta w^{(n)}$  для всех остальных слоев  $n = N-1, \dots 1$ .

7. Скорректировать все веса нейросети на основе

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t). \quad (3.2.1.18)$$

8. Если ошибка сети существенна, перейти к шагу 2. В противном случае – конец. На шаге 2 сети попеременно в случайном порядке предъявляются все образы обучающей последовательности. Данная процедура повторяется и с остальными образами обучающей последовательности. Однократное предъявление всех образов обучающей последовательности образует цикл или итерацию (Epoch). Процесс обучения многослойного персептрона обычно занимает несколько сотен или тысяч циклов. Шаги 2-6 повторяются до достижения некоторого критерия, например, достижения ошибкой установленного предела.

По окончании фазы обучения осуществляется проверка эффективности работы обученной нейросети при помощи контролирующей последовательности. При этом коррекция весов естественно не производится, а лишь вычисляется ошибка. Если эффективность обученной нейросети удовлетворительна, сеть считается готовой к работе. В противном случае она может быть модифицирована, а затем обучена при измененных параметрах того же алгоритма обучения или с использованием другого алгоритма обучения. Из (2.13)



следует, что когда выход  $o_{ri}^{(n-1)}$  стремится к нулю, эффективность обучения заметно снижается. При двоичных входных векторах в среднем половина весовых коэффициентов не будет корректироваться, поэтому область возможных значений выходов нейронов  $[0, 1]$  желательно преобразовать в интервал  $[-0.5, +0.5]$ , например, с помощью функции активации

$$f(x) = -0.5 + \frac{1}{1 + e^{-\alpha x}}. \quad (3.2.1.20)$$

Согласно работе [58] проблема выбора числа нейронов выходного слоя сети, выполняющего окончательную классификацию образов. Для разделения множества образов, например, на два класса достаточно всего одного нейрона. При этом каждый логический уровень "1" и "0" будет обозначать отдельный класс. При двух нейронах можно закодировать уже 4 класса и так далее. Однако результаты работы сети, организованной таким образом, не очень надежны. Для повышения достоверности классификации желательно ввести избыточность путем выделения каждому классу одного нейрона в выходном слое или даже нескольких, каждый из которых обучается определять принадлежность образа к классу со своей степенью достоверности, например: высокой, средней и низкой. Такие сети позволяют производить классификацию образов, объединенных в нечеткие (размытые или пересекающиеся) множества.

Алгоритм обучения с обратным распространением ошибки оказался весьма эффективным при обучении многослойных сетей для решения широкого класса задач [59]. Однако он имеет ряд недостатков. В частности применение метода наискорейшего спуска не гарантирует нахождения глобального минимума целевой функции. Эта проблема связана еще с одной, а именно с выбором величины коэффициента коррекции  $\eta$ . В качестве  $\eta$  обычно выбирается число меньше 1, но не очень малое, например, 0.1, и оно постепенно уменьшается в процессе обучения. Кроме того, для исключения случайных попаданий в локальные минимумы иногда, после того как значения весовых

коэффициентов стабилизируются,  $\eta$  кратковременно сильно увеличивают, чтобы начать градиентный спуск из новой точки. Если повторение этой процедуры несколько раз приведет алгоритм в одно и то же состояние нейросети, то можно более или менее уверенно считать, что определен глобальный, а не локальный минимум.

#### **4. Расчет будущих значений временного ряда и программная реализация**

##### **4.1 Алгоритм экстраполяции временного ряда с учетом внешних факторов**

Модель экстраполяции временного ряда с учетом внешних факторов сформулирована в разделе 2.2 диссертационной работы. Алгоритм экстраполяции состоит из следующих шагов.

1. Определить выборку новой истории.
2. Определить выборку максимального подобия.
3. Определить выборку базовой истории.
4. Вычислить прогнозные значения.

Далее приведем описание каждого указанного шага, иллюстрируя расчеты решением следующей задачи. Пусть даны значения временного ряда энергопотребления на территории с 01.01.2016 до 23.04.2016, а также значения температуры в тех же временных рамках. Длина временного ряда равна 2737. Обозначим временной ряд энергопотребления  $Z(t)$ , временной ряд температур –  $X(t)$ . Требуется определить 24 значения временного ряда на

24.04.2016 с учетом влияния временного ряда температур  $X(t)$ . Считаем параметр модели  $M = 216$  заданным.

В связи с тем, что значения ряда  $X(t)$  доступны до той же отметки времени, что и значения временного ряда  $Z(t)$ , необходимо на первом этапе определить значения временного ряда  $\hat{X}(t)$  по алгоритму:

1) Определить выборку новой истории.

Выборкой новой истории является выборка временного ряда [60], значения которой предшествуют моменту прогноза. В текущей постановке задачи выборка новой истории равна  $X_{T-M+1}^M = X_{2522}^{216}$ .

2) Определить выборку максимального подобия

Для определения выборки максимального подобия необходимо определить значения модуля линейной корреляции  $\rho_k^M$  для выборки  $X_{T-M+1}^M$  и всех выборок с задержкой  $k = \{1, 2, \dots, T-M-1\}$ .  $X_{T-M+1}^M$ . Обозначим момент времени  $T-M+1 = t$  и решим данную задачу.

Вычислим аппроксимированные значения выборки

$$\hat{X}_{T-M+1}^M = \alpha_1 X_{T-M+1-k}^M + \alpha_0 I^M, \quad (4.1.1)$$

а с учетом обозначения

$$\hat{X}_t^M = \alpha_1 X_{t-k}^M + \alpha_0 I^M. \quad (4.1.2)$$

Согласно методу наименьших квадратов, коэффициенты аппроксимации определим, исходя из уравнения

$$Z_X A = Z_Y, \quad (4.1.3)$$

где значение элементов матрицы  $Z_X$  и  $Z_Y$  определяются следующим образом:

$$Z_X = \begin{bmatrix} \sum_{i=0}^{M-1} Z^2(k+i) & \sum_{i=0}^{M-1} Z(k+i) \\ \sum_{i=0}^{M-1} Z(k+i) & M \end{bmatrix}; \quad (4.1.4)$$

$$Z_Y = \begin{bmatrix} \sum_{i=0}^{M-1} Z(k+i)Z(T-M+1+i) \\ \sum_{i=0}^{M-1} Z(T-M+1+i) \end{bmatrix}. \quad (4.1.5)$$

Найденные коэффициенты аппроксимации подставим в (4.1.1), а далее определим значение модуля корреляции  $\rho_k^M$  с помощью нейронной сети. Повторяя вычисления для каждого значения  $k$  из указанного диапазона, определим множество значений  $\rho_1^M, \rho_2^M, \rho_3^M, \dots, \rho_{T-M-1}^M$ .

Значения  $\rho_k^M$  для  $k=\{1,2,\dots,1000\}$  приведены на рисунке.

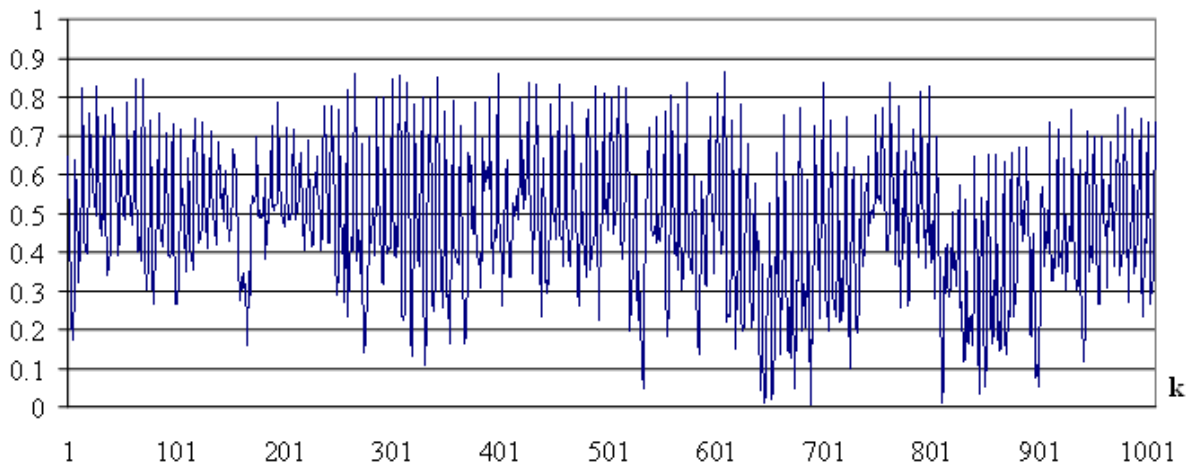


Рисунок 4 – Значения меры максимального подоби́я  $\rho_k^M$  для  $k = \{1,2,\dots,1000\}$

Далее на основании (2.1.3.13) определим значение максимума корреляции  $\rho_{k_{\max}}^M$  и соответствующую задержку  $k_{\max}$ . Для решаемой задачи значение максимума корреляции  $\rho_{k_{\max}}^M = 0,862$  и соответствует задержке  $k_{\max} = 2736$ . Для задержки уравнение аппроксимации имеет вид

$$\hat{X}_{2522}^{24} = 1.0789X_{1434}^{24} - 4689.78I^{24}. \quad (4.1.6)$$

Подставим полученные значения в (2.1.2.9) и (2.1.2.10), найдем среднюю абсолютную ошибку и среднюю абсолютную ошибку в процентах.

Получим,  $MAPE = 1.07\%$ ;  $MAE = 1282 \text{ МВт} \cdot \text{ч}$ . Считаем задачу определения  $\hat{X}(t)$  решенной.

Далее необходимо:

1) Определить выборку новой истории.

Выборкой новой истории временного ряда  $Z(t)$  является выборка временного ряда, значения которой предшествуют моменту прогноза  $T$ . В текущей постановке задачи выборка новой истории равна  $Z_{T-M+1}^M = Z_{2522}^{216}$ . Выборка новой истории соответствует значениям цены на электроэнергию за период на 24.04.2013.

2) Определить выборку максимального подобия.

Для определения выборки максимального подобия необходимо определить значения ошибки регрессии  $s_k^M$  (2.1.3.11) для выборки  $Z_{T-M+1}^M$  и всех выборок с задержкой  $k = \{1, 2, \dots, T-M-1\}$ . При этом для каждого значения  $k$  из указанного диапазона требуется решить задачу аппроксимации выборки  $Z_{T-M+1}^M$  при помощи выборок  $Z_{T-M+1-k}^M$  и  $X_{T-M+1}^M$ . Обозначим момент времени  $T-M+1 = t$  и решим данную задачу.

Вычислим аппроксимированные значения выборки

$$\hat{Z}_{T-M+1}^M = \alpha_2 Z_{T-M+1-k}^M + \alpha_1 X_{T-M+1}^M + \alpha_0 I^M, \quad (4.1.7)$$

а с учетом обозначения

$$\hat{Z}_t^M = \alpha_2 Z_{t-k}^M + \alpha_1 X_t^M + \alpha_0 I^M. \quad (4.1.8)$$

Согласно методу наименьших квадратов, коэффициенты аппроксимации определим исходя из уравнения

$$Z_X A = Z_Y, \quad (4.1.9)$$

где матрицы  $Z_X$  и  $Z_Y$  определяются следующим образом

$$Z_X = \begin{bmatrix} \sum_{i=0}^{M-1} Z^2(t-k+i) & \sum_{i=0}^{M-1} Z(t-k+i)X(t+i) & \sum_{i=0}^{M-1} Z(t-k+i) \\ \sum_{i=0}^{M-1} Z(t-k+i) & \sum_{i=0}^{M-1} X^2(t+i) & \sum_{i=0}^{M-1} X(t+i) \\ \sum_{i=0}^{M-1} Z(t-k+i) & \sum_{i=0}^{M-1} X(t+i) & M \end{bmatrix}; \quad (4.1.10)$$

$$Z_Y = \begin{bmatrix} \sum_{i=0}^{M-1} Z(t-k+i)Z(t+i) \\ \sum_{i=0}^{M-1} Z(t+i)X(t+i) \\ \sum_{i=0}^{M-1} Z(t+i) \end{bmatrix}. \quad (4.1.11)$$

Например, для задержки  $k=11689$  коэффициенты аппроксимации равны

$$A = \begin{bmatrix} \alpha_2 \\ \alpha_1 \\ \alpha_0 \end{bmatrix} = Z_X^{-1} Z_Y = \begin{bmatrix} -0.1544 \\ 0.0178 \\ -476.7678 \end{bmatrix}. \quad (4.1.12)$$

Найденные коэффициенты аппроксимации подставим в выражение (4.1.7), а далее определим значение ошибки регрессии  $S_k^M$  (2.1.3.7). Повторяя вычисления для каждого значения  $k$  из указанного диапазона, определим множество значений  $S_1^M, S_2^M, S_3^M, \dots, S_{T-M-1}^M$ .

Далее на основании (2.21) определим значение минимума ошибки регрессии  $S_{k \max}^M$  и соответствующую задержку  $k_{\max}$ .

Для решаемой задачи с помощью нейронной сети описанной в главе 2 находим минимальную ошибку  $S_{k \max}^M = 9987.65$  соответствует задержке  $k_{\max} = 11689$ . Для задержки  $k_{\max}$  уравнение аппроксимации имеет вид

$$\hat{Z}_t^{216} = -0.1544 Z_{t-1689}^{216} + 0.0178 X_t^{216} - 476.7678 I^{216}. \quad (4.1.13)$$

3) Определить выборку базовой истории.

Согласно гипотезе подобия [8] в качестве выборки базовой истории  $Z_T^P$  берем выборку, следующую за выборкой максимального подобия  $Z_{t-11689}^{216} = Z_{12720}^{216}$ , то есть выборка базовой истории равна  $Z_T^P = Z_{12936}^{24}$ .

4) Вычислить прогнозные значения.

Вычислим значения выборки  $\hat{Z}_{2462}^{24}$  согласно зависимости

$$\hat{Z}_{2462}^{24} = -0.1544Z_{1293}^{24} + 0.02\hat{X}_{2462}^{24} - 476.77I^{24}. \quad (4.1.14)$$

Оценки ошибки аппроксимации: значение MAPE аппроксимации равно 2.11%, значение MAE = 12.24 МВт·ч. Оценки точности для модели экстраполяции: MAPE = 4.42%; MAE = 28.29 МВт·ч. Результаты показывают, что ошибка аппроксимации близка, но не равна ошибке экстраполяции.

## 4.2 Программная реализация

Целью прогнозирования энергопотребления является поддержание надежной работы единой энергосистемы РФ. Прогноз энергопотребления необходим, в первую очередь, системному оператору для балансирования энергосистемы РФ. С введением рынка электроэнергии и мощности взамен общему планированию каждая компания-потребитель самостоятельно прогнозирует собственное энергопотребление. Система финансовых расчетов на энергорынках устроена таким образом, чтобы мотивировать потребителей как можно точнее планировать собственное потребление: чем точнее прогноз энергопотребления, тем выше финансовый результат. В связи с этим каждая компания-потребитель заинтересована в предельно точном прогнозе собственного потребления.

Созданная система прогнозирования не имеет графического интерфейса, работает автономно, но требует вмешательства эксперта для определения входящих данных.

Алгоритмы идентификации модели и оценки доверительных интервалов (3.3. и 3.4.) реализованы в программной среде MATLAB.

Нейронная сеть реализована на Java.

Прогнозирование энергопотребления производится на один день вперед и выполняется на модели ARIMA.

Полученные для краткосрочного прогнозирования значения MAPE лежат в диапазоне от 0.91% до 1.83% для прогнозирования на сутки вперед. Листинг программы Приложение 1.

Результаты обработки приведены в таблице 2.

Таблица 2 – Исследуемый временной ряд

№	Временной ряд	Длина ряда	Среднее значение	Стандартное отклонение	Мин. значение	Макс. значение
1	Энергопотребление	2762	1621	113	990	2224

Результаты прогнозирования приведены в таблице 3.

Таблица 3 – Результаты прогнозирования временных рядов энергопотребления

№	Временной ряд	Контрольный период	Время упреждения	Параметр модели $M$	MAE (MAPE)
1	Энергопотребление	24.04.2013 (24 значения)	24	216	1282(1.07%)

В таблице 4 отражен почасовой прогноз на 24.04.2013.



Таблица 4 – Прогнозные значения на 24.04.2013 год

№	Дата (24.03.2014), ч.	Значения ря- да, МВт/ч.	Прогнозные значения ряда МВт·ч.	MAE (MAPE) МВт·ч., (%)
1	2	3	4	5
1	1:00	1,013	1,023	(0,94)
2	2:00	1,075	1,086	(1,03)
3	3:00	1,192	1,203	(0,98)
4	4:00	1,308	1,325	(1,32)
5	5:00	1,416	1,436	(1,45)
6	6:00	1,478	1,494	(1,14)
7	7:00	1,489	1,512	(1,57)
8	8:00	1,512	1,539	(1,83)
9	9:00	1,460	1,484	(1,67)
10	10:00	1,473	1,486	(0,95)
11	11:00	1,436	1,459	(1,66)
12	12:00	1,437	1,456	(1,37)
Продолжение таблицы 4				
1	2	3	4	5
14	14:00	1,389	1,410	(1,55)
15	15:00	1,382	1,398	(1,22)
16	16:00	1,307	1,324	(1,34)
17	17:00	1,338	1,358	(1,56)
18	18:00	1,472	1,498	(1,77)
19	19:00	1,391	1,405	(1,05)
20	20:00	1,241	1,258	(1,43)
21	21:00	1,093	1,108	(1,45)
22	22:00	1,024	1,036	(1,22)
23	23:00	1,000	1,012	(1,27)
24	0:00	1,013	1,031	(1,83)

Полученные для краткосрочного прогнозирования энергопотребления значения MAPE лежат в диапазоне от 0.91% до 1.83% для прогнозирования на сутки вперед.



## ЗАКЛЮЧЕНИЕ

1) Задача прогнозирования временных рядов актуальна и решается на основании модели прогнозирования. Одним из наиболее используемых классов моделей прогнозирования является класс авторегрессионных моделей. Установлено, что основным недостатком данного класса является большое число свободных параметров, требующих определения. Определено перспективное направление развития моделей прогнозирования, позволяющее устранить указанный недостаток.

2) Разработана новая модель прогнозирования временных рядов по выборке максимального подобия для двух видов постановки задачи прогнозирования временного ряда – с учетом внешних факторов. Применена нейронная сеть для определения коэффициентов линейной корреляции Пирсона. Новая модель относится к авторегрессионному классу моделей и имеет единственный параметр, что упрощает задачу идентификации модели, устраняя основной недостаток моделей данного класса.

3) Разработан новый метод прогнозирования на основе предложенной модели, содержащий набор алгоритмов для экстраполяции временных рядов, идентификации модели и построения доверительного интервала прогнозных значений. Применена многослойная нейронная сеть с алгоритмом обучения обратного распространения ошибки.

4) Выполнена программная реализация разработанных алгоритмов средствами математического пакета MATLAB и Java.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Бокс Дж. Анализ временных рядов, прогноз и управление/Дж. Бокс, Г. М. Дженкинс– М. : Мир, 1974.–406 с.
- 2 I. A. Gheyas A Neural Network Approach to Time Series Forecasting // Proceedings of the World Congress on Engineering. –London, 2009.–1292 p.
- 3 M. Norizan Short Term Load Forecasting Using Double Seasonal ARIMA Model // Regional Conference on Statistical Sciences.–Malaysia, Kelantan, 2010. –73 p.
- 4 D. H. Mazengia Forecasting Spot Electricity Market Prices Using Time Series Models: Thesis for the degree of Master of Science in Electric Power Engineering.–Gothenburg, Chalmers University of Technology, 2008.–89 p.
- 5 N. Morariu A neural network model for time series forecasting// Romanian Journal of Economic Forecasting. –USA, 2009.–223 p
- 6 N. Draper Applied regression analysis.– New York: Wiley, In press, 1981.–693 p.
- 7 S. K. Prajakta, Time series Forecasting using Holt-Winters Exponential Smoothing // Kanwal Rekhi School of Information Technology Journal.– Boston, 2004.–13p.
- 8 Хайкин С. Нейронные сети: полный курс/С. Хайкин.– М. : ООО «И. Д. Вильямс», 2006.–1104 с.
- 9 R. P. Pradhan Forecasting Exchange Rate in India: An Application of Artificial Neural Network Model // Journal of Mathematics Research.–USA, 2010.–117 p.
- 10 B. Yildiz Forecasting the Istanbul Stock Exchange National 100 Index Using an Artificial Neural Network // An International Journal of Science, Engineering and Technology.–London, 2008.–39 p.
- 11 J. Catalao An Artificial Neural Network Approach for Day-Ahead Electricity Prices Forecasting// 6th WSEAS international conference on Neural networks.– USA, Stevens Point, 2005.–83 p.

- 12 Семенов В. В. Математическое моделирование динамики транспортных потоков/В. В. Семенов.–М. : ИПМ им. М. В. Келдыша, 2004.–44 с.
- 13 W. Huanga Forecasting stock market movement direction support vector machine// Elsevier: computers and operation research.–USA, New-York, 2005.–2522 p.
- 14 S. Mahfoud Financial Forecasting Using Genetic Algorithms // Applied Artificial Intelligence.–USA, 1996.–565 p.
- 15 F. J. Nogales Electricity price forecasting through transferfunction models // Journal of the Operational Research Society.–London, 2006.–356 p.
- 16 H. K. Alfares Electric load forecasting: literature survey and classification of methods // International Journal of Systems Science. –Gloster, 2002.–34 p.
- 17 H. R. Fogler A pattern recognition model for forecasting //Management science. 1974.–1189 p.
18. H. Zareipour Forecasting the Hourly Ontario Energy Price byMultivariate Adaptive \_Regression Splines // IEEE Power Engineering Society General Meeting.–Canada, Montreal, 2006.–7 p.
19. Мамедова Т.Ф., Василькин Н.В. М. Модель Солоу-Свона. Актуальные вопросы прикладной математики, 26.12.2012 г., Саранск, СВМО, стр. 12-17
20. Василькин Н.В. Задачи оптимального управления в экономике. Материалы конференции молодых ученых. Саранск, 2013 г.
21. Василькин Н.В. Задачи оптимального управления в экономике. Материалы Всероссийской конференции: Современные тенденции в развитии научной мысли. Саранск, 2013 г.
22. Мамедова Т.Ф., Василькин Н.В. Огарев-online. – 2015. – №23. – Режим доступа: <http://journal.mrsu.ru/arts/o-prognozirovanie-vremennogo-ryada-s-pomoshhyu-nejronnykh-setej>., 6с.
23. ВасилькинН.В., Катенин А.В., Новиков О.В. Проблемы разработки и создания территориально распределенных автоматизированных специальных систем управления связью. Научно-технический сборник АО «Кон-

церн «Системпром», 2017, № 1(8). – Под ред. кандидата технических наук А.П. Мурзина. – М.: Издательство: «АО «Концерн «Системпром», 2017. – XXXXXXXX. – 567 с. – Инв. № Д-2030с.

24. Василькин Н.В., Егорова Д.К., Мамедова Т.Ф., Метод покомпонентной асимптотической эквивалентности в задаче об устойчивости электрической цепи. Научно-технический вестник Поволжья. – 2018. - №5.

## ПРИЛОЖЕНИЕ А (ОБЯЗАТЕЛЬНОЕ)

### ЛИСТИНГ ПРОГРАММЫ НЕЙРОННОЙ СЕТИ

Листинг.

Layer.java

```
packagennet;

importjava.io.Serializable;

/**Интерфейснейронногослоя*/

public interface Layer extends Serializable {

    /**

        * Получает размер входного вектора

        * @return Размер входного вектора

        */

    intgetInputSize();

    /**

        * Получает размер слоя

        * @return Размер слоя

        */

    intgetSize();

    /**

        * Вычисляет отклик слоя

        * @paraminput Входной вектор

        * @return Выходнойвектор

        */

    float[] computeOutput(float[] input);

}
```

## Продолжение приложения А

Network.java

Реализация базовой функциональности нейронной сети, состоящей из нескольких слоев.

```
packagennet;

import java.io.*;

/**
 * Базоваяреализациянейроннойсети
 */

public class Network implements Serializable {

    /**
     * Конструирует нейронную сеть с заданными слоями
     * @param layers Нейронныйслой
     */

    public Network(Layer[] layers) {

        // проверки

        if (layers == null || layers.length == 0) throw new IllegalArgumentException();

        // проверимдетально

        finalint size = layers.length;

        for (inti = 0; i< size; i++)

            if (layers[i] == null || (i> 1 && layers[i].getInputSize() != layers[i - 1].getSize()))

                throw new IllegalArgumentException();

        // запомнимслои

        this.layers = layers;

    }

    /**
```



## Продолжение приложения А

```
* Получает размер входного вектора

* @return Размер входного вектора

*/

public final intgetInputSize() {

return layers[0].getInputSize();

}

/**

* Получает размер выходного вектора

* @return Размер выходного вектора

*/

public final intgetOutputSize() {

return layers[layers.length - 1].getSize();

}

/**

* Получает размер сети

* @return Размер сети

*/

public final intgetSize() {

returnlayers.length;

}

/**

* Получает нейронный слой по индексу

* @paramindex Индекс слоя

* @return Нейронныйслой

*/

public final Layer getLayer(int index) {
```

## Продолжение приложения А

```
return layers[index];

    }

    /**
     * Вычисляетоткликсети
     *
     * @paraminput Входной вектор
     *
     * @return Выходной вектор
     */

    public float[] computeOutput(float[] input) {

        // проверки

        if (input == null || input.length != getInputSize())

            throw new IllegalArgumentException();

        // вычислимвыходнойоткликсети

        float[] output = input;

        Продолжение приложения А

        finalint size = layers.length;

        for (inti = 0; i< size; i++)

            output = layers[i].computeOutput(output);

        // вернем выход

        returnoutput;

    }

    /**
     * Сохраняет нейронну сеть в файл
     *
     * @paramfileNameИмяфайла
```

## Продолжение приложения А

```
*/

public void saveToFile(String fileName) {

    // проверки

    if (fileName == null) throw new IllegalArgumentException();


    // сохраняем

    try {

        ObjectOutputStream outputStream = new ObjectOutputStream(new FileOutputStream(fileName));

        outputStream.writeObject(this);

        outputStream.close();

    }

    catch (Exception e) {

        throw new IllegalArgumentException(e);

    }

}

/**

 * Загружает нейронную сеть из файла

 * @param fileName Имя файла

 * @return Нейронную сеть

 */

public static Network loadFromFile(String fileName) {

    // проверки

    if (fileName == null) throw new IllegalArgumentException();


    // загружаем

    Object network = null;
```

## Продолжение приложения А

```
try {

ObjectInputStream inputStream = new ObjectInputStream(new FileInputStream(fileName));

network = inputStream.readObject();

inputStream.close();

    }

catch (Exception e) {

throw new IllegalArgumentException(e);

    }


    // отдадим сеть

return (Network)network;

}

/**

* Слой

*/

private Layer[] layers;

}
```

### BackpropLayer.java

computeBackwardError принимает входной вектор, который подавался на вход и вектор ошибки для этого слоя. Метод же adjust подгоняет веса нейронов в сторону уменьшения ошибки.

```
packagennet;

/**

* Интерфейс слоя обучаемого по алгоритму обратного распространения ошибки

*/

public interface BackpropLayer extends Layer {

/**

* Придает случайные значения весам нейронов
```

## Продолжение приложения А

```
* @parammin Минимальное значение

* @param max Максимальное значение

*/

void randomize(float min,float max);

/**

* Вычисляет следующий вектор ошибки в обратном направлении

* @paraminput Входной вектор

* @paramerror Вектор ошибки

* @return Следующий вектор ошибки в обратном направлении

*/

float[] computeBackwardError(float[] input,float[] error);

/**

* Подгоняет веса нейронов в сторону уменьшения ошибки

* @paraminput Входной вектор

* @paramerror Вектор ошибки

* @paramrate Скорость обучения

* @parammomentumМоментум

*/

void adjust(float[] input,float[] error,floatrate,float momentum);

}

BackpropNetwork.java

Реализация нейронной сети, обучаемой по алгоритму обратного распространения ошибки.

packagenet;

/**
```

## Продолжение приложения А

\* Нейронная сеть обучаемая по алгоритму обратного распространения ошибки

\*/

```
public final class BackpropNetwork extends Network {
```

```
/**
```

\* Конструирует нейронную сеть с заданными слоями

\* @param layers

\*/

```
public BackpropNetwork(Layer[] layers) {
```

```
// передадим родакам
```

```
super(layers);
```

```
    // рандомизируем веса
```

```
    randomize(0,0.3f);
```

```
}
```

```
/**
```

\* Придает случайные значения весам нейронов в сети

\* @param min

\* @param max

\*/

```
public void randomize(float min,float max) {
```

```
// придаем случайные значения весам в сети
```

```
    final int size = getSize();
```

```
    for (int i = 0; i < size; i++) {
```

```
        Layer layer = getLayer(i);
```

```
        if (layer instanceof BackpropLayer) ((BackpropLayer)layer).randomize(min,max);
```

```
    }
```

```
}
```

## Продолжение приложения А

```
/**
 * Обучает сеть паттерну
 * @paraminput Входной вектор
 * @paramgoal Заданный выходной вектор
 * @paramrate Скорость обучения
 * @parammomentumМоментум
 * @return Текущую ошибку обучения
 */

public float learnPattern(float[] input,float[] goal,floatrate,float momentum) {

    // проверки

    if (input == null || input.length != getInputSize() ||
        goal == null || goal.length != getOutputSize()) throw new IllegalArgumentException();

    // делаемпроходвперед

    finalint size = getSize();

    float[][] outputs = new float[size][];

    outputs[0] = getLayer(0).computeOutput(input);

    for (inti = 1; i< size; i++)

        outputs[i] = getLayer(i).computeOutput(outputs[i - 1]);

    // вычислимошибкувыходногослоя

    Layer layer = getLayer(size - 1);

    finalintlayerSize = layer.getSize();

    float[] error = new float[layerSize];

    floattotalError = 0;

    for (inti = 0; i<layerSize; i++) {
```

## Продолжение приложения А

```
error[i] = goal[i] - outputs[size - 1][i];

totalError += Math.abs(error[i]);

    }

    // обновим выходной слой

    if (layer instanceof BackpropLayer)

        ((BackpropLayer)layer).adjust(size == 1 ? input : outputs[size - 2], error, rate, momentum);

    // идем по скрытым слоям

    float[] prevError = error;

    Layer prevLayer = layer;

    for (int i = size - 2; i >= 0; i--, prevError = error, prevLayer = layer) {

        // получим очередной слой

        layer = getLayer(i);

        // вычислим для него ошибку

        if (prevLayer instanceof BackpropLayer)

            error = ((BackpropLayer)prevLayer).computeBackwardError(outputs[i], prevError);

        else

            error = prevError;

        // обновим слой

        if (layer instanceof BackpropLayer)

            ((BackpropLayer)layer).adjust(i == 0 ? input : outputs[i - 1], error, rate, momentum);

    }

    // вернем суммарную ошибку

    return totalError;
```



## Продолжение приложения А

```
    }  
}  
  
SigmoidLayer.java  
  
packagenet;  
  
/**  
 * Сигмоидальный слой  
 */  
  
public final class SigmoidLayer implements BackpropLayer {  
  
    /**  
     * Вес  
     */  
  
    private final int WEIGHT = 0;  
  
    /**  
     * Дельта  
     */  
  
    private final int DELTA = 1;  
  
    /**  
     * Конструирует сигмоидальный слой  
     */  
  
    /**  
     * @param inputSize Размер входного вектора  
     * @param size Размер слоя  
     * @param bipolar Флаг биполярного слоя  
     */  
  
    public SigmoidLayer(int inputSize, int size, boolean bipolar) {
```

## Продолжение приложения А

```
// проверки
if (inputSize < 1 || size < 1) throw new IllegalArgumentException();

// создаем слой
matrix = new float[size][inputSize + 1][2];

// запомним параметры
this.inputSize = inputSize;

this.bipolar = bipolar;

}

/**
 * Конструирует биполярный слой
 * @param inputSize Размер входного вектора
 * @param size Размер слоя
 */
public SigmoidLayer(int inputSize, int size) {
    this(inputSize, size, true);
}

public int getInputSize() {
    return inputSize;
}

public int getSize() {
    return matrix.length;
}
```

## Продолжение приложения А

```
public float[] computeOutput(float[] input) {  
  
    // проверки  
  
    if (input == null || input.length != inputSize)  
  
        throw new IllegalArgumentException();  
  
  
    // ВЫЧИСЛИМ ВЫХОД  
  
    final int size = matrix.length;  
  
    float[] output = new float[size];  
  
    for (int i = 0; i < size; i++) {  
  
        output[i] = matrix[i][0][WEIGHT];  
  
        for (int j = 0; j < inputSize; j++)  
  
            output[i] += input[j] * matrix[i][j + 1][WEIGHT];  
  
        if (bipolar)  
  
            output[i] = (float) Math.tanh(output[i]);  
  
        else  
  
            output[i] = 1 / (1 + (float) Math.exp(-output[i]));  
  
    }  
}
```

## Продолжение приложения А

```
    // вернем оклик  
  
    return output;  
  
}
```

```
public void randomize(float min, float max) {  
  
    final int size = matrix.length;  
  
    for (int i = 0; i < size; i++) {  
  
        for (int j = 0; j < inputSize + 1; j++) {
```

## Продолжение приложения А

```
matrix[i][j][WEIGHT] = min + (max - min) * (float)Math.random();

matrix[i][j][DELTA] = 0;

    }

}

}

public float[] computeBackwardError(float[] input, float[] error) {

    // проверки

    if (input == null || input.length != inputSize ||

        error == null || error.length != matrix.length) throw new IllegalArgumentException();

    // вычислим входящую ошибку

    float[] output = computeOutput(input);

    final int size = matrix.length;

    float[] backwardError = new float[inputSize];

    for (int i = 0; i < inputSize; i++) {

        backwardError[i] = 0;

        for (int j = 0; j < size; j++)

            backwardError[i] += error[j] * matrix[j][i + 1][WEIGHT] *

                (bipolar ? 1 - output[j] * output[j] : output[j] * (1 - output[j]));

    }

    // вернем ошибку

    return backwardError;

}
```

## Продолжение приложения А

```
public void adjust(float[] input, float[] error, float rate, float momentum) {

    // проверки

    if (input == null || input.length != inputSize ||

        error == null || error.length != matrix.length) throw new IllegalArgumentException();

    // обновляем веса

    float[] output = computeOutput(input);

    final int size = matrix.length;

    for (int i = 0; i < size; i++) {

        final float grad = error[i] * (bipolar ? 1 - output[i] * output[i] : output[i] * (1 - output[i]));

        // обновляем нулевой вес

        matrix[i][0][DELTA] = rate * grad + momentum * matrix[i][0][DELTA];

        matrix[i][0][WEIGHT] += matrix[i][0][DELTA];

        // обновим остальные веса
```

## Продолжение приложения А

```
        for (int j = 0; j < inputSize; j++) {

            matrix[i][j + 1][DELTA] = rate * input[j] * grad + momentum * matrix[i][j + 1][DELTA];

            matrix[i][j + 1][WEIGHT] += matrix[i][j + 1][DELTA];

        }

    }

}

/**

 * Размер входного вектора

 */

private final int inputSize;
```

## Окончание приложения А

```
/**  
  
 * Флаг биполярного слоя  
  
 */  
  
private final boolean bipolar;  
  
/* Матрица слоя */  
  
private float[][][] matrix;  
  
}
```

## **ПРИЛОЖЕНИЕ Б (ОБЯЗАТЕЛЬНОЕ)**

### **ДИСК С ВРЕМЕННЫМ РЯДОМ**

**ПРИЛОЖЕНИЕ В (НЕОБЯЗАТЕЛЬНОЕ)**

**ДИСК С ПРОГРАММОЙ**