

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

**( Н И У « Б е л Г У » )**

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
ИНФОРМАЦИОННЫХ СИСТЕМ

**Разработка инструментального средства прогнозирования инвестиций**

Выпускная квалификационная работа  
обучающегося по направлению подготовки  
02.04.01. «Математика и компьютерные науки»  
очной формы обучения,  
группы 07001631

Кофановой Татьяны Владимировны

Научный руководитель

Доцент, кандидат технических  
наук, Румбешт В.В.

Рецензент

Доцент, кандидат технических  
наук, Зайцева Т.В.

Введение.....	5
1 Теоретические сведения и постановка задачи .....	7
1.1 Модели прогнозирования .....	12
1.2 Обзор моделей прогнозирования .....	15
1.2.1 Регрессионные модели .....	17
1.2.2 Авторегрессионные модели.....	19
1.2.3 Модели экспоненциального сглаживания.....	22
1.2.4 Нейросетевые модели.....	23
1.2.5 Модели на базе цепей Маркова.....	25
1.2.6 Модели на базе классификационно-регрессионных деревьев.....	26
1.2.7 Комбинированные модели.....	27
1.3 Обзор искусственных нейронных сетей.....	29
1.3.1 Классификация нейронных сетей .....	32
1.3.2 Алгоритмы обучения нейронных сетей.....	37
2 Описание средств реализации.....	42
2.1 Описание методики прогнозирования .....	44
2.1.1 Поиск и обработка показателей.....	44
2.1.2 Анализ взаимного влияния финансовых показателей .....	47
2.1.3 Формирование структуры нейронной сети.....	50
2.1.4 Апробация полученного инструментального средства .....	52
2.2 Описание используемых программных продуктов.....	53
3 Разработка и тестирование программного средства прогнозирования инвестиций.....	55
3.1 Описание разработанного приложения .....	55
3.2 Тестирование программного средства.....	60
Заключение .....	65
Список использованной литературы.....	67
Приложение 1 .....	71

Приложение 2 .....	76
--------------------	----

## ВВЕДЕНИЕ

Почти в каждой области встречаются явления, которые интересно и важно изучать в их развитии и изменении во времени. В повседневной жизни могут представлять интерес, например, метеорологические условия, цены на тот или иной товар, те или иные характеристики состояния здоровья индивидуума и т. д. Все они изменяются во времени. С течением времени изменяются деловая активность, режим протекания того или иного производственного процесса, режим жизни человека, восприятие видео и аудио информации.

Однако, несмотря на такой широкий разброс по сферам применения и наблюдения, большинство подобных явлений можно представить в виде временного ряда. Совокупность измерений какой-либо одной характеристики подобного рода в течение некоторого периода времени представляют собой временной ряд, а совокупность существующих методов анализа таких рядов наблюдений называется анализом временных рядов.

Основной чертой, выделяющей анализ временных рядов среди других видов статистического анализа, является существенность порядка, в котором производятся наблюдения. Если во многих задачах наблюдения статистически независимы, то во временных рядах они, как правило, зависимы, и характер этой зависимости может определяться положением наблюдений в последовательности. Природа ряда и структура порождающего ряд процесса могут предопределять порядок образования последовательности.

Цель данной работы заключается в исследовании зависимости между экономическими и социальными показателями, такими как курсы валют, цены на нефть, курса Доу-Джонса, ВВП, индекса человеческого развития и прочих, анализ и обработка полученных данных, структурирование данных в виде временного ряда. Итогом данной работы является построение на

основании данного исследования методики прогнозирования финансовых показателей при использовании нейронной сети.

Если подавляющее большинство методов прогнозирования стремится к уменьшению влияния факторов, отражающихся на модели, то для нейронных сетей такие факторы способствуют улучшению результатов. Главное условие, чтобы данное влияние было возможно измерить и оценить.

Проектирование методики прогнозирования включает в себя: проведение анализа влияния финансовых показателей друг на друга, установление значений влияния между финансовыми показателями, анализ наиболее подходящих методов для решения данной задачи, построение модели прогнозирования, и разработка инструментального средства инвестиционного прогнозирования.

В ходе достижения общей цели работы решаются следующие задачи:

- изучение методов прогнозирования;
- выбор подходящих для программного решения методов и среды;
- анализ и оценка взаимного влияния выбранных финансовых показателей.

Работа состоит из трех глав, посвященных отдельным вопросам. В первой главе освещены основные методы прогнозирования временных рядов, подробно описана история нейронных сетей. Во второй главе представлено обоснование выбранных для разработки программных продуктов, описаны их достоинства и недостатки, описана методика прогнозирования, разработанная в рамках данной работы. Третья глава содержит описание функционала программного продукта, а также описание процесса тестирования и его результаты. Работа содержит 69 страниц, 38 рисунков, 3 таблицы и 2 приложения.

## 1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ И ПОСТАНОВКА ЗАДАЧИ

Ни для кого не секрет, что предсказание курса валют довольно сложная задача, т.к. на ее курс влияет множество как экономических, так и социальных и политических факторов. Также валюты могут иметь довольно тесную корреляцию между собой, что так же обусловлено внешними факторами. Однако влияние их не равнозначно, поэтому в задачу работы входит именно определение наиболее влияющих компонент. К счастью, данное взаимное влияние показателей можно достоверно определить благодаря математическому аппарату. И потому что данное влияние можно измерить, его же можно и учесть при построении модели прогнозирования.

В рамках данной работы определена основная цель работы – построить модель прогнозирования, учитывающую взаимное влияние экономических показателей. Для достижения указанной цели необходимо решить следующие задачи:

- проведение анализа взаимного влияния показателей;
- построение модели на основе результатов анализа;
- апробация полученной модели;
- подведение итогов о целесообразности полученной модели.

Данные задачи состоят в свою очередь из множества подзадач, что во многом усложняет первостепенную цель. Однако, при системном подходе к вопросу, время на решение подобной задачи сокращается. Для решения вопроса трудоемкости задачи, её решение необходимо формализовать, т.е. разработать методику прогнозирования для конкретного вида задачи, использование которой в типовых задачах в дальнейшем способно сократить время на получение результата.

Однако, прежде чем приступить к реализации поставленных задач необходимо столкнуться с теоретической стороной работы, а именно обзором методов прогнозирования.

Слово прогноз происходит из греческого языка и означает предвидение, предсказание. Прогнозирование – предсказание будущего с использованием научных методов. Под процессом прогнозирования понимается специфическое научное исследование определенных перспектив развития процессов. Процессы, перспективы которых нуждаются в предсказании, как правило, описываются посредством временных рядов, иными словами последовательностью значений определенных величин, которые получаются в конкретных моментах времени. Временной ряд определяется двумя обязательными элементами, а именно временной отметкой и значением элемента ряда, который получен каким-либо способом и соответствует данной отметке времени. Любой временной ряд должен рассматриваться как выборочная реализация из бесконечного множества популяций, полученными в результате стохастического процесса, на который влияют большое количество факторов.

Различают временные ряды по способу определения значения, временным шагом, памятью и стационарностью.

- интервальные временные ряды,
- моментные временные ряды.

Под интервальным временным рядом подразумевают такую последовательность, где уровень явления относится к результату, который был накоплен или был вновь произведен за конкретный отрезок времени. К примеру, интервальным можно назвать временной ряд значений выпуска продукции предприятием за определенный период времени; либо объем воды, выпущенной гидроэлектростанцией за какой-либо период времени; либо объем электроэнергии, выработанной за час, день, месяц и другие.

В случае, если значение временного ряда определяет рассматриваемое явление в определенный момент времени, то общность данных значений представляет собой моментный временной ряд. Например, моментными рядами можно назвать значения динамики финансовых индексов, рыночных цен и индексов; физические показатели, к примеру, температура

окружающей среды, уровень влажности, давление, измеренные в различные моменты времени, и прочие.

- По частоте определения значений временные ряды подразделяются на
- равноотстоящие временные ряды,
  - неравноотстоящие временные ряды.

Равноотстоящие временные ряды образуются путем исследования и фиксации значений процесса за равные, последовательные отрезки времени. Большую часть физических процессов принято описывать с помощью равноотстоящих временных рядов. Под неравноотстоящими временными рядами подразумеваются такие ряды, для которых не применим принцип равенства интервалов фиксации значений. Такими рядами можно считать все биржевые индексы так как данные значения указываются только за рабочие дни недели.

От специфики рассматриваемого процесса временные ряды подразделяются на

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

Под временными рядами с длинной памятью, имеются в виду временные ряды, автокорреляционная функция которых убывает медленно. А под временными рядами с короткой памятью понимают те временные ряды, в которых быстро убывает автокорреляционная функция. Например, скорость движения транспортного потока, и другие физические процессы, как, например, расход электроэнергии, температура окружающей среды, можно отнести к категории временных рядов с длинной памятью. Временные ряды с короткой памятью представляют собой, например, значения биржевых индексов.

Также временные ряды подразделяются на:

- стационарные временные ряды,
- нестационарные временные ряды.



Стационарным временным рядом является ряд, остающийся в состоянии равновесия по отношению к постоянному среднему значению. Прочие временные ряды относят к категории нестационарных. Как правило, решая задачи прогнозирования, нестационарные временные ряды приводят к стационарным с использованием разностного оператора.

Временем упреждения можно назвать горизонт времени, на период которого нужно указать значения временного ряда. По времени упреждения задачи прогнозирования обычно подразделяются на такие категории срочности как:

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

Отметим, что любой временной ряд имеет собственные диапазоны. Например, для временного ряда цен на акции и котировки классификация срочности задачи прогнозирования определяется конкретным случаем:

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

При рассмотрении прогноза временных рядов, следует различать такие понятия как: модель прогнозирования и метод прогнозирования.

Модель прогнозирования является функциональным отображением, которое достоверно описывает поведение временного ряда и представляет собой основу для получения будущих значений процесса. Термин модель экстраполяции тесно связан с моделями прогнозирования.

Метод прогнозирования включает в себя череду действий, при выполнении которых выявляется модель прогнозирования для данного временного ряда. Помимо этого, метод прогнозирования включает в себе действия направленные на оценку адекватности значений прогноза. Построение модели прогнозирования представляет итеративный подход, который состоит из данных шагов.

Шаг 1. Первый шаг заключается в выборе общего класса моделей для осуществления прогноза временного ряда на указанный горизонт.

Шаг 2. Далее необходимо подогнать к исходному временному ряду, используя методы определения подклассов моделей. Данные методы идентификации основаны на качественных оценках временного ряда.

Шаг 3. Затем нужна оценка параметров модели. В том случае, если модель имеет структуру, или параметры, эту модель следует отнести к типу структурных моделей. В таком случае, как правило, прибегают к помощи итеративных способов, т.е. тогда, когда оценивается временной ряд при наличии различных значений исследуемых величин. Обычно, этот этап считают наиболее сложным из-за того, что часто при расчетах используют все доступные значения временного ряда.

Шаг 4. Затем необходимо провести диагностику полученной модели прогнозирования. Для этого выбирают один или несколько участков временного ряда, длины которых достаточно проверки прогнозирования и произведения оценивания точности прогноза. Такие отрезки временного ряда, отобранные для диагностики называются контрольными участками.

Шаг 5. Модель готова к применению, когда точность прогнозирования приемлема для задач, где встречаются прогнозные значения. Точность прогнозирования модели может оказаться недостаточной для дальнейшего применения значений прогноза, в таком случае требуется итеративное повторение предыдущих шагов.

Прогнозирование временных рядов подразумевает два возможных варианта постановки задачи. Согласно первому варианту: для получения будущих значений исследуемого временного ряда используются доступные значения только этого ряда. Второй же вариант гласит: для получения прогнозных значений возможно использование не только фактических значений искомого ряда, но и значений набора внешних факторов, представленных в виде временных рядов.

Прогнозируя временной ряд важно найти функциональную зависимость, которая достоверно описывает временной ряд, данная зависимость именуется моделью прогнозирования. Целью создания такой модели является получение модели, где среднее значение абсолютного отклонения реального значения от прогнозируемого минимизируется для указанного горизонта (время упреждения). Затем после определения модели прогнозирования необходимо просчитать будущие значения временного ряда, и их доверительный интервал.

## 1.1 Модели прогнозирования

Одной из задач, поставленных в рамках работы – является построение модели прогнозирования, в связи с чем, необходимо подробное рассмотрение подходящих существующих моделей.

В рамках задачи рассматривается прогнозирование без учета внешних факторов.

Пусть значения временного ряда отмечены в дискретные моменты времени  $t = 1, 2, \dots, T$ . Обозначим временной ряд как  $Z(t) = Z(1), Z(2), \dots, Z(T)$ . В момент времени  $T$  необходимо определить значения процесса  $Z(t)$  в моменты времени  $T+1, \dots, T+P$ . Момент времени  $T$  называется моментом прогноза, а величина  $P$  — временем упреждения.

1) В процессе вычисления значений временного ряда в будущие моменты времени необходимо определить функциональную зависимость, содержащую связь между прошлыми и будущими значениями указанного ряда

$$Z(t) = F(Z(t-1), Z(t-2), (Z-3), \dots) + \varepsilon_t, (1.1)$$

Зависимость (1.1) называется моделью прогнозирования. Согласно условиям прогноза, необходимо создать такую модель прогнозирования, для

которой среднее абсолютное отклонение истинного значения от прогнозируемого стремится к минимальному значению для заданного P:

$$\bar{E} = \frac{1}{P} \sum_{t=T+1}^{T+P} |\varepsilon_t| \rightarrow \min, (1.2)$$

Выражение (1.1) можно также представить в виде:

$$\hat{Z}(t) = F(Z(t-1), Z(t-2), Z(t-3), \dots), (1.3)$$

где  $\hat{Z}(t)$  – прогнозные значения временного ряда  $Z(t)$ . Здесь и далее использование символа «^» означает вычисляемые значения временного ряда.

2) Помимо получения будущих значений  $\hat{Z}(T), \dots, \hat{Z}(T+P)$  требуется определить размер доверительного интервала возможных отклонений данных значений.

Задача прогнозирования временного ряда проиллюстрирована на рис. 1.1.



Рис. 1.1. Иллюстрация задачи прогнозирования временного ряда без учета внешних факторов

Концепция же прогнозирования с учетом внешних факторов заключается в следующем:

Пусть значения исходного временного ряда  $Z(t)$  будут доступны в дискретные моменты времени  $t = 1, 2, \dots, T$ . Предполагается, что значения  $Z(t)$  находятся в ощутимой зависимости от набора 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 также 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  $t$ .

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.

1) В рамках вычисления будущих значений процесса  $Z(t)$  в указанные моменты времени необходимо рассчитать функциональную зависимость, содержащую связь между прошлыми значениями  $Z(t)$  и будущими, а также отражающую оказанное влияние внешних факторов  $X_1(t), \dots, X_S(t)$  на исходный временной ряд

$$Z(t) = F(Z(t-1), Z(t-2), \dots, X_1(t), X_1(t-1), \dots, X_S(t), X_S(t-1), \dots), (1.4.)$$

Зависимость (1.4.) называется моделью прогнозирования с учетом внешних факторов  $X_1(t), \dots, X_S(t)$ . Необходимо разработать такую модель прогнозирования, в которой среднее абсолютное отклонение истинного значения от прогнозируемого стремится к минимальному для заданного  $P$  (2).

2) Кроме получения будущих значений  $\hat{Z}(T), \dots, \hat{Z}(T+P)$  необходимо рассчитать значения доверительного интервала возможных отклонений этих значений.

Задача прогнозирования временного ряда с учетом одного внешнего фактора представлена на рис. 1.2.

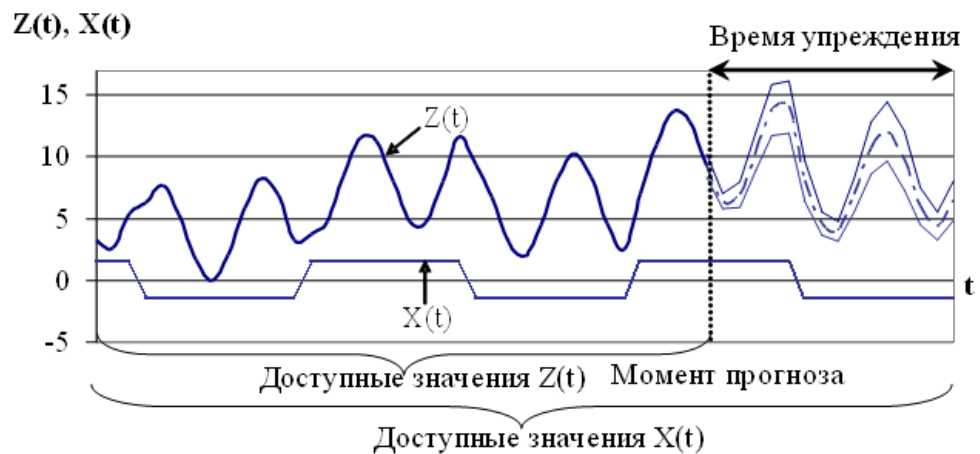


Рис. 1.2. Иллюстрация задачи прогнозирования временного ряда с учетом внешнего фактора

## 1.2 Обзор моделей прогнозирования

На сегодняшний день создано более сотни различных классов моделей прогнозирования. К тому же, часть из этих классов, моделей и соответствующих им методов выделяют в качестве самостоятельных процедур прогнозирования. В то время как, другая часть методов представляет собой набор отдельных способов, отличающихся от базовых или от друг друга количеством своеобразных приемов и спецификой их применения.

В общем случае, методы прогнозирования часто делят на группы: интуитивные и формализованные.

Использование интуитивного метода прогнозирования правомерно в случаях, когда в качестве объекта прогнозирования выступает самый обычный предмет, или же, наоборот, столь неординарный и сложный объект, что не представляется возможным учесть влияние всех факторов внешней среды. При использовании интуитивных методов нет необходимости в построении моделей прогнозирования, так как сами методы основаны на субъективных экспертных суждениях по поводу хода развития процесса. В

качестве фундамента для интуитивных методов выступают профессиональный опыт и наблюдательность. Использование данных методов справедливо при осуществлении анализа таких процессов, развитие которых не представляется возможным формализовать с использованием существующих математических инструментов, проще говоря, для таких процессов сложно подобрать достоверную модель. К подобным методам относят: методы экспертных оценок, предвидения по образцу и исторических аналогий. К счастью, на сегодняшний день широкое распространено применение экспертных систем, а также экспертных систем с использованием нечеткой логики.

В рамках формализованных методов прогнозирования, модели также можно классифицировать как статистические и структурные модели.

Характерным признаком статистических моделей является аналитически определенная функциональная зависимость между текущими и будущими значениями временного ряда, внешними факторами. Таким образом, к группам статистических моделей относятся:

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

В то же время, для структурных моделей характерна функциональная зависимость между текущими и будущими значениями временного ряда, и внешними факторами определенная структурно, принадлежат следующие группы:

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

Однако, для решения узкоспециализированных задач иногда необходимо использование определенных моделей прогнозирования. Так, например, для задач, связанных с прогнозированием уровня сахара в крови человека применяются модели, основанные на дифференциальных

уравнениях. А в задачах прогнозирования потока транспортных средств обосновано применение гидродинамических моделей. Подобные модели разрабатываются и применяются для прогнозирования определенных процессов.

### 1.2.1 Регрессионные модели

Существует множество задач, требующих изучения взаимного влияния между двумя и более переменными. Чаще всего для решения подобных задач применяют регрессионный анализ. На настоящий момент регрессия имеет широкий охват сфер использования, включающий задачи прогнозирования и управления. Основной задачей регрессионного анализа является установление количественного выражения зависимости между переменной и внешними факторами. Сами же коэффициенты регрессии могут рассчитываться с использованием метода наименьших квадратов или метода максимального правдоподобия.

Линейная регрессионная модель. Самым распространенным видом регрессионной модели считается линейная регрессия. За основу модели взято предположение, что существует некий дискретный внешний фактор  $X(t)$ , влияющий на заданный процесс  $Z(t)$ , при этом проявляется линейная зависимость между процессом и внешним фактором. Модель прогнозирования описываемая линейной регрессией выражается уравнением

$$Z(t) = \alpha_0 + \alpha_1 X(t) + \varepsilon_t, (1.5.)$$

где  $\alpha_0$  и  $\alpha_1$  — коэффициенты регрессии;  $\varepsilon_t$  — ошибка модели. Для получения прогнозных значений  $Z(t)$  в момент времени  $t$  необходимо иметь значение  $X(t)$  в тот же момент времени  $t$ , однако, данное требование сложно осуществить на практике.



Множественная регрессионная модель. На практике на процесс  $Z(t)$  оказывают влияние целый ряд дискретных внешних факторов  $X_1(t), \dots, X_S(t)$ . Тогда модель прогнозирования имеет вид

$$Z(t) = \alpha_0 + \alpha_1 X_1(t) + \alpha_2 X_2(t) + \dots + \alpha_S X_S(t) + \varepsilon_t, \quad (1.6.)$$

Главным недостатком указанной модели является то, что для определения значения процесса  $Z(t)$  необходимо точно задать значения для всех факторов  $X_1(t), \dots, X_S(t)$  в будущем, что сложно осуществить на практике.

В основу нелинейной регрессионной модели положено предположение о том, что существует известная функция, описывающая зависимость между исходным процессом  $Z(t)$  и внешним фактором  $X(t)$

$$Z(t) = F(X(t), A), \quad (1.7.)$$

В рамках построения модели прогнозирования необходимо определить параметры функции  $A$ . Например, можно предположить, что:

$$Z(t) = \alpha_1 \cos(X(T)) + \alpha_0, \quad (1.8.)$$

В рамках построения модели достаточно задать параметры  $A = [\alpha_1, \alpha_0]$ . Недостатком такой модели является то, что на практике, процессы, для которых заранее известен вид функционального влияния между процессом  $Z(t)$  и внешним фактором  $X(t)$  практически не встречаются. В связи с чем, применение нелинейных регрессионных моделей крайне ограничено.

Модель группового учета аргументов (МГУА) была разработана Ивахтенко А.Г. Модель имеет вид:

$$Z(t) = \alpha_0 \sum_{i=1}^S \alpha_i X_i(t) + \sum_{i=1}^S \sum_{j=1}^S \alpha_{i,j} X_i(t) X_j(t) + \sum_{i=1}^S \sum_{j=1}^S \sum_{k=1}^S \alpha_{i,j,k} X_i(t) X_j(t) X_k(t) + \dots, \quad (1.9.)$$

Уравнение (1.9) называется опорной функцией. С помощью опорной функции, задаются разные варианты моделей для некоторых или всех аргументов. Так, например, строятся полиномы с одной переменной, полиномы со всевозможными парами переменных, полиномы со

всевозможными тройками переменных и т.д. Для каждой модели рассчитываются её линейные коэффициенты  $\alpha_{i,j,k,\dots}$  с помощью метода регрессионного анализа. Из полученных моделей выбираются несколько (от 2 до 10) наиболее эффективных. Так, качество моделей вычисляется с помощью, например, среднеквадратичного отклонения или прочих критериев. Если во множестве выбранных существует модель, результаты которой достаточны при использовании полученных прогнозных значений, то процесс подбора очередных моделей оканчивается. В противном случае выбранные модели выполняют роль аргументов  $X_1(t), \dots, X_S(t)$  для опорных функций на следующем шаге итераций. Таким образом, получается, что полученные на предыдущих итерациях модели используются в процессе создания более сложных [42].

### 1.2.2 Авторегрессионные модели

Самая популярная модель данного класса – ARIMAX.

В основу авторегрессионных моделей легло предположение о том, что значение процесса  $Z(t)$  линейно зависит от некоторого множества предыдущих значений того же процесса  $Z(t-1), \dots, Z(t-p)$ .

Авторегрессионная модель скользящего среднего. В области анализа временных рядов модель авторегрессии (autoregressive, AR) и модель скользящего среднего (moving average, MA) получили наиболее широкое применение.

На практике модель авторегрессии проявляет себя крайне полезной для описания некоторых процессов, встречающихся на практике временных рядов. В этой модели текущее значение процесса выражается как конечная линейная совокупность предыдущих значений процесса и импульса, который называется «белым шумом»,

$$Z(t) = C + \phi_1 Z(t-1) + \phi_2 Z(t-2) + \dots + \phi_p Z(t-p) + \varepsilon_t, (1.10.)$$

С помощью формулы (1.10.) отражен процесс авторегрессии порядка  $p$ , который в литературных источниках можно встретить как  $AR(p)$ , здесь  $C$  — вещественная константа,  $\phi_1, \dots, \phi_p$  — коэффициенты,  $\varepsilon_t$  — ошибка модели. Для определения  $\phi_i$  и  $C$  используют метод наименьших квадратов или метод максимального правдоподобия/

Модель скользящего среднего порядка  $q$ , также является наиболее распространенным типом, к тому же имеет большое значение в описании временных рядов, часто применяется вкупе с авторегрессией и описывается уравнением:

$$Z(t) = \frac{1}{q} + Z(t-1) + Z(t-2) + \dots + Z(t-q) + \varepsilon_t, (1.11.)$$

В источниках процесс (1.11) зачастую обозначается как  $MA(q)$ ; здесь  $q$  — порядок скользящего среднего,  $\varepsilon_t$  — ошибка прогнозирования. Модель скользящего среднего в общем понимании представляет собой фильтр низких частот. Важно понимать, что имеют место простые, взвешенные, кумулятивные, экспоненциальные модели скользящего среднего.

Для получения наибольшей гибкости в процессе подбора модели часто необходимо объединить в рамках одной модели авторегрессию и скользящее среднее. Такая общая модель обозначается  $ARMA(p,q)$  и объединяет в себе фильтр в виде скользящего среднего порядка  $q$  и авторегрессию фильтрованных значений процесса порядка  $p$ .

Если вместо самих значений входных данных временного ряда используется их разность  $d$ -того порядка (на практике  $d$  необходимо заранее вычислять, однако в большинстве случаев  $d \leq 2$ ), то полученная модель именуется авторегрессией проинтегрированного скользящего среднего. В источниках же данную модель называют  $ARIMA(p,d,q)$  (autoregression integrated moving average).

Эта модель  $ARIMA(p,d,q)$  получила развитие в виде модели  $ARIMAX(p,d,q)$ , которая описывается уравнением:

$$Z(t) = AR(p) + \alpha_1 X_1(t) + \dots + \alpha_S X_S(t), (1.12.)$$

Здесь  $\alpha_1, \dots, \alpha_S$  — коэффициенты внешних факторов  $X_1(t), \dots, X_S(t)$ . В указанной модели чаще всего процесс  $Z(t)$  является результатом, полученным в результате модели  $MA(q)$ , то есть отфильтрованными значениями исходного процесса. Далее для прогнозирования  $Z(t)$  применяется модель авторегрессии, в которой обозначены дополнительные значения внешних факторов  $X_1(t), \dots, X_S(t)$ .

Авторегрессионная модель с условной гетероскедастичностью (autoregressive conditional heteroskedasticity, GARCH) была разработана в 1986 году Тимом Петером Борреслевым и представляет собой модель остатков для модели  $AR(p)$ . На первом этапе для исходного временного ряда определяется модель  $AR(p)$ . Далее выдвигается предположение о том, что ошибка модели  $\varepsilon_t$  делится на две составляющие

$$\varepsilon_t = \sigma_t \zeta_t, (1.13)$$

где  $\sigma_t$  — зависимое от времени стандартное отклонение;  $\zeta_t$  — случайная величина, имеющая нормальное распределение, среднее значение, равное 0, и стандартное отклонение, равное 1. При этом имеющее зависимость от времени стандартное отклонение характеризуется уравнением

$$\sigma_t^2 = \beta_0 + \beta_1 \varepsilon_{t-1}^2 + \dots + \beta_q \varepsilon_{t-q}^2 + \gamma_1 \sigma_{t-1}^2 + \dots + \gamma_p \sigma_{t-p}^2, (1.14)$$

Здесь  $\beta_0, \dots, \beta_q$  и  $\gamma_0, \dots, \gamma_p$  — коэффициенты. Уравнение (1.14) называется моделью GARCH(p,q) и имеет два параметра:  $p$  — описывает порядок авторегрессии квадратов остатков; под  $q$  понимается количество предшествующих оценок остатков [4].

Наиболее обоснованное применение данная модель нашла в сфере финансов, где с использованием модели получают значения показателя волатильности. На сегодняшний день развит целый ряд дополнений модели, носящих названия NGARCH, IGARCH, EGARCH, GARCH-M и другие.

Авторегрессионной модели с распределенным лагом (autoregressive distributed lag models, ARDLM) не уделено достаточного внимания в

литературных источниках. Особое внимание данной модели предоставляется в книгах по эконометрике[11].

Часто при моделировании процессов на изучаемую переменную влияют не только текущие значения процесса, но и его лаги, то есть значения временного ряда, предшествующие изучаемому моменту времени. Модель авторегрессии распределенного лага описывается уравнением:

$$Z(t) = \phi_0 + \phi_1 Z(t-l-1) + \dots + \phi_p Z(t-l-p) + \varepsilon_t, (1.15)$$

Здесь  $\phi_0, \dots, \phi_p$  — коэффициенты,  $l$  — величина лага. Модель (1.15) называется ARDLM(p,l) и чаще всего применяется для моделирования экономических процессов.

### 1.2.3 Модели экспоненциального сглаживания

Модели экспоненциального сглаживания были разработаны в середине XX века и до сегодняшнего дня не потеряли своей актуальности, имея широкое распространение за счет их простоты и наглядности.

Модель экспоненциального сглаживания (exponential smoothing, ES) используется для задач моделирования финансовых и экономических процессов. Основной идеей экспоненциального сглаживания является постоянная переоценка существующих прогнозных значений по мере пребывания новых. В рамках модели ES задаются экспоненциально убывающие веса значениям наблюдений по мере их старения. Согласно такому подходу, последние из проведенных наблюдений оказывают большее влияние на прогнозируемые значения, нежели более ранние.

Функция модели ES имеет вид

$$Z(t) = S(t) + \varepsilon_t,$$
$$S(t) = \alpha \cdot Z(t-1) + (1-\alpha) \cdot S(t-1), (1.16)$$

где  $\alpha$  — коэффициент сглаживания,  $0 < \alpha < 1$ ; начальные условия определяются как  $S(1) = Z(0)$ . В данной модели каждое последующее сглаженное значение  $S(t)$  является взвешенным средним между предыдущим значением временного ряда  $Z(t)$  и предыдущего сглаженного значения  $S(t-1)$  [7].

Модель Хольта или двойное экспоненциальное сглаживание применяется для моделирования процессов, имеющих тренд. В этом случае в модели необходимо рассматривать две составляющие: уровень и тренд. Уровень и тренд сглаживаются отдельно.

Модель Хольта-Винтерса или тройное экспоненциальное сглаживание применяется для процессов, которые имеют тренд и сезонную составляющую.

Модели экспоненциального сглаживания наиболее популярны для долгосрочного прогнозирования.

### 1.2.4 Нейросетевые модели

В настоящее время самой популярной среди структурных моделей является модель на основе искусственных нейронных сетей (artificial neural network, ANN). Нейронные сети состоят из нейронов (рис. 1.3.).

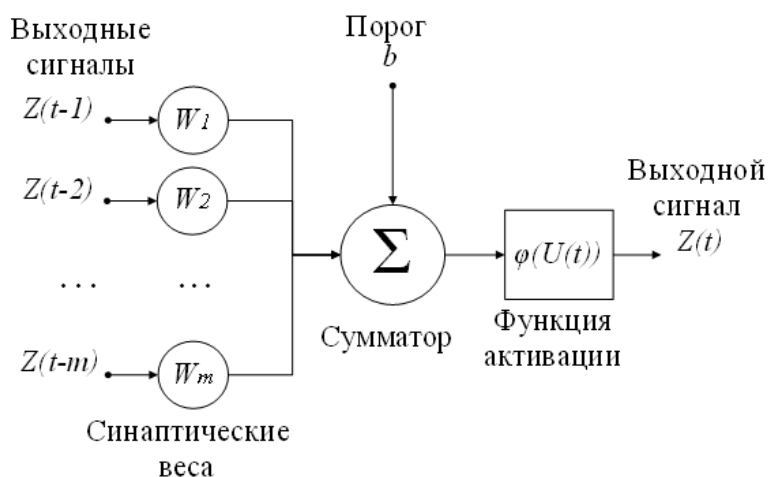


Рис. 1.3. Схема работы нейронной сети

Модель нейрона можно описать парой уравнений:

$$U(t) = \sum_{i=1}^m \omega_i \cdot Z(t-i) + b,$$

$$Z(t) = \varphi(U(t)), \quad (1.17)$$

где  $Z(t-1), \dots, Z(t-m)$  — входные сигналы;  $\omega_1, \dots, \omega_m$  — синаптические веса нейрона;  $p$  — порог;  $\varphi(U(t))$  — функция активации.

Функция активации бывают трех основных типов:

- функция единичного скачка;
- кусочно-линейная функция;
- сигмоидальная функция.

Способ связи нейронов определяет архитектуру нейронной сети. В зависимости от способа связи нейронов сети делятся на

- однослойные нейронные сети прямого распространения,
- многослойные нейронные сети прямого распространения,
- рекуррентные нейронные сети.

На рис. 1.4. представлена структура трехслойной нейронной сети прямого распространения.

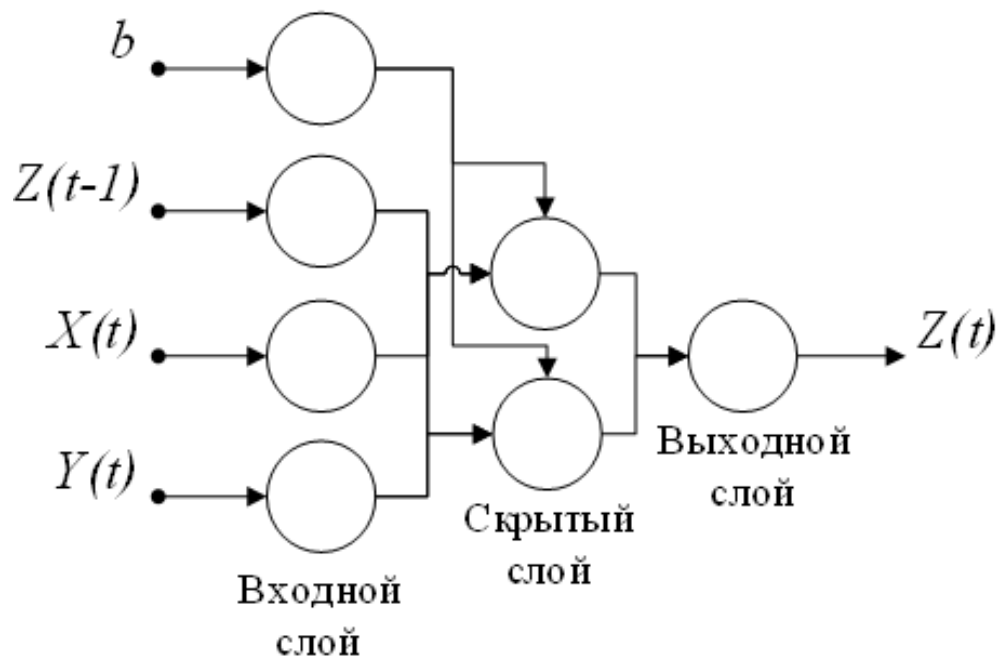


Рис. 1.4. Трехслойная нейронная сеть прямого распространения

Таким образом, при помощи нейронных сетей возможно моделирование нелинейной зависимости будущего значения временного ряда от его фактических значений и от значений внешних факторов. Нелинейная зависимость определяется структурой сети и функцией активации.

### 1.2.5 Модели на базе цепей Маркова

Модели прогнозирования на основе цепей Маркова (Markov chain model) предполагают, что будущее состояние процесса зависит только от его текущего состояния и не зависит от предыдущих. В связи с этим процессы, моделируемые цепями Маркова, должны относиться к процессам с короткой памятью.

Пример цепи Маркова для процесса, имеющего три состояния, представлен на рис 1.5.

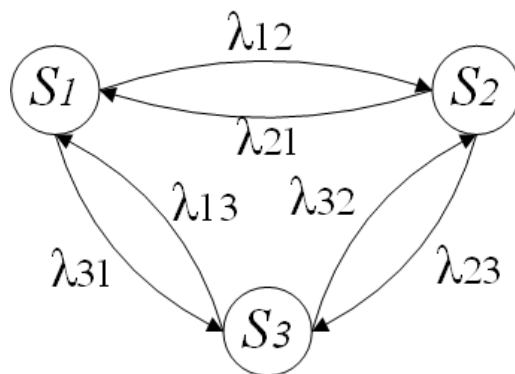


Рис. 1.5. Цепь Маркова с тремя состояниями

Здесь  $S_1, \dots, S_3$  — состояния процесса  $Z(t)$ ;  $\lambda_{12}$  — вероятность перехода из состояния  $S_1$  в состояние  $S_2$ ,  $\lambda_{23}$  — вероятность перехода из состояния  $S_2$  в состояние  $S_3$  и т.д. При построении цепи Маркова определяется множество состояний и вероятности переходов. Если текущее состояние процесса  $S_i$ , то в качестве будущего состояния процесса выбирается такое состояние  $S_j$ , вероятность перехода в которое (значение  $\lambda_{ij}$ ) максимальна [13].



Таким образом, структура цепи Маркова и вероятности перехода состояний определяют зависимость между будущим значением процесса и его текущим значением.

### 1.2.6 Модели на базе классификационно-регрессионных деревьев

Классификационно-регрессионные деревья (classification and regression trees, CART) являются еще одной популярной структурной моделью прогнозирования временных рядов. Структурные модели CART разработаны для моделирования процессов, на которые оказывают влияние как непрерывные внешние факторы, так и категориальные. Если внешние факторы, влияющие на процесс  $Z(t)$ , непрерывны, то используются регрессионные деревья; если факторы категориальные, то — классификационные деревья. В случае, если необходимо учитывать факторы обоих типов, то используются смешанные классификационно-регрессионные деревья.



Рис. 1.6. Бинарное классификационно-регрессионное дерево

Согласно модели CART, прогнозное значение временного ряда зависит от предыдущих значений, а также некоторых независимых переменных. На приведенном на рисунке 6 примере сначала предыдущее значение процесса сравнивается с константой  $Z_0$ . Если значение  $Z(t-1)$  меньше  $Z_0$ , то выполняется следующая проверка:  $X(t) > X_{11}$ . Если неравенство не выполняется, то  $Z(t) = C_3$ , иначе проверки продолжаются до того момента, пока не будет найден лист дерева, в котором происходит определение будущего значения процесса  $Z(t)$ . Важно, что при определении значения в расчет принимаются как непрерывные переменные, например,  $X(t)$ , так и категориальные  $Y$ , для которых выполняется проверка присутствия значения в одном из заранее определенных подмножеств. Значения пороговых констант, например,  $Z_0$ ,  $X_{11}$ , а также подмножеств  $Y_{11}, Y_{12}$  выполняется на этапе обучения дерева [9].

Таким образом, CART моделирует зависимость будущей величины процесса  $Z(t)$  при помощи структуры дерева, а также пороговых констант и подмножеств.

### **1.2.7 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.

Например, 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. В статье указывается, что при

комбинированном подходе удастся повысить точность прогнозирования временных рядов.

В работе предлагается модель для прогнозирования цен на электроэнергию Испании. При помощи вейвлет преобразования (wavelet transform) доступные значения временного ряда разделяются на несколько последовательностей, для каждой из которых строится отдельная модель ARIMA.

В обзоре моделей прогнозирования энергопотребления рассматриваются следующие типы комбинаций:

- 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.

Использование 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 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.

Разработчики подобных моделей утверждают, что предложенные модели просты, прозрачны и эффективны для исследованных временных рядов. При этом очевидно, что главными недостатками данных моделей являются:

- 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 работах.

Развитие искусственного интеллекта – одна из популярнейших сфер исследований в программировании. И хотя сама концепция искусственного интеллекта не является новой, о ней заговорили еще в середине 20 столетия, однако данная задача до сих пор остается актуальной. Это связано с тем, что имеющиеся на сегодняшний день достижения трудно назвать истинным «искусственным интеллектом», к тому же потенциал искусственного интеллекта невероятно огромен.

### **1.3 Обзор искусственных нейронных сетей**

Касательно искусственных нейронных сетях необходимо разобраться поподробнее, так как на сегодняшний день искусственные нейронные сети представляют собой универсальный инструмент.

В 60-80 годах XX века исследователи в сфере искусственного интеллекта возлагали большие надежда на экспертные системы. Которые, в свою очередь, раскрыли свой потенциал в узкоспециализированных областях, но ими и ограничились. Такой результат применения экспертных

систем поднял вопрос о создании более гибкой и универсальной интеллектуальной системы.

Одним из самых результативных направлений в исследовании искусственного интеллекта являются искусственные нейронные сети. Как видно из названия, за основу метода взята нервная система человека, а точнее нейроны и их связи. Нейронная сеть в своем виде появилась как попытка имитации процессов, происходящих в головном мозге человека, на основе математической модели.

Нейрон человека или животного представляет собой клетку, состоящую из ядра, тела клетки и отростков: аксонов и дендритов. Главная задача нейрона заключается в передаче по отросткам электрического сигнала связанным с ним нейронам. В месте контакта нейронов между собой образуется синапс, а связи характеризуются, в свою очередь, силой синаптической связи. Значение связи определяет: вызывает ли синапс деполяризацию (увеличивает сигнал) или гиперполяризацию (затормаживает сигнал).

При создании искусственной нейронной сети в виде математической модели был упрощено представлен описанный выше механизм. Таким образом, сегодня искусственная нейронная сеть представляет собой структуру связанных нейронов (рис. 1.7.).

В простом виде нейронная сеть представляет собой 3 слоя нейронов. Первый слой – слой входных нейронов, представляет собой входные данные для сети. Вторым слоем – скрытый слой нейронов (скрытых слоев может быть больше, их количество зависит от сложности задачи), где каждый нейрон связан с каждым нейроном входного слоя. Третий слой – выходной слой нейронов, связан с нейронами скрытого слоя. Однако, как и в настоящей биологической нейронной сети наиболее важная роль отведена связям нейронов. Так при создании нейронной сети каждой связи задается значение (вес), а в процессе обучения сети веса корректируются.

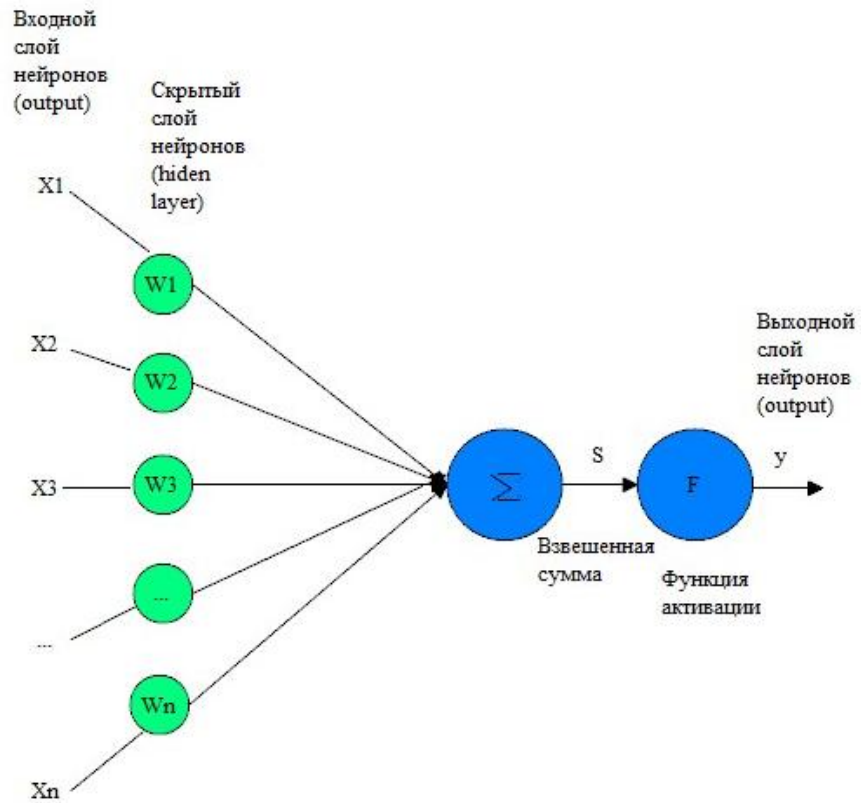


Рис. 1.7. Схема простой искусственной нейронной сети

Нейрон осуществляет нелинейное преобразование  $F$  суммы произведения входных сигналов на весовые коэффициенты (см. рис. 1.7.):

$$y = F(\sum_{i=1}^n w_i x_i) = F(S) = F(WX^r), \quad (1.18)$$

Где  $W = (w_1, w_2, \dots, w_n)$  - вектор весовых коэффициентов,  $X = (x_1, x_2, \dots, x_n)$  - вектор входных сигналов,

$$S = (\sum_{i=1}^n w_i x_i), \quad (1.19)$$

- взвешенная сумма,  $F$  - оператор нелинейного преобразования или *функция активации*. Взвешенную сумму  $S$  иногда представляют в виде

$$S = (\sum_{i=1}^n w_i x_i) - T, \quad (1.20)$$

где  $T$  - порог нейронного элемента, характеризующий сдвиг функции активации по оси абсцисс. Если продолжить биологические аналогии, то входы  $x_i$  являются для искусственного нейрона дендритами, операторы

суммирования и нелинейного преобразования - телом нейрона, а выход  $y$  – аксоном.

В качестве оператора нелинейного преобразования могут выступать различные функции, которые выбираются в соответствии с рассматриваемой задачей и типом НС. Рассмотрим наиболее часто используемые функции активации.

-Линейная функция:  $y = kS$ .

-Пороговая функция  $y = \text{sign}(S) = \begin{cases} 1, S > 0; \\ 0, S \leq 0. \end{cases}$  или  $y = \begin{cases} 1, S > 0; \\ -1, S \leq 0. \end{cases}$

-Сигмоидная функция:  $y = \frac{1}{1+e^{-cy}}$ , где  $c > 0$  - коэффициент, определяющий ширины функции по оси абсцисс[34].

Следует отметить, что в силу монотонности и всюду дифференцируемости сигмоидная функция является наиболее распространенной. Ее график приведен на рис. 1.8.

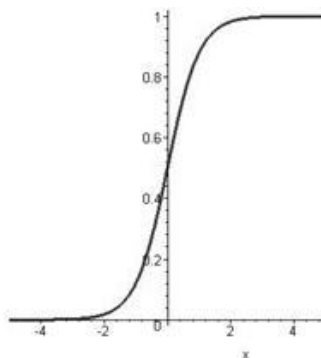


Рис. 1.8. Сигмоидная функция активации ( $c=2$ )

### 1.3.1 Классификация нейронных сетей

Существование множества различных видов искусственных нейронных сетей обусловлено тем, что концепция самой сети заимствована из природного механизма, привести математическую модель которого довольно сложно, и получившаяся модель остается спорной, т.к. сам биологический

механизм не до конца изучен. Такая ситуация породила множество методов в построении искусственных нейронных сетей. Исходя из принципов топологии искусственные нейронные сети можно разделить на:

- слабосвязные (с локальными связями) сети;
- полносвязные сети;
- многослойные или слоистые сети.



Рис. 1.9. – Архитектура слабосвязных сетей

Для архитектуры слабосвязных сетей (рис. 1.9.) характерно расположение нейронов в узлах гексогональной и прямоугольной решетки, в которой каждый нейрон связан несколькими окрестными нейронами. Количество соседних нейронов зависит от окрестности:

- окрестность фон Неймана – 4 нейрона;
- окрестность Голя – 6 нейронов;
- окрестность Мура – 8 нейронов.

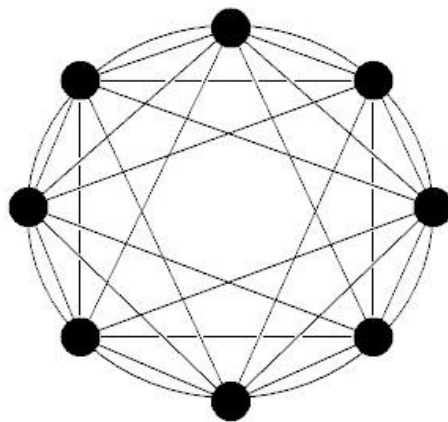


Рисунок 1.10. – Архитектура полносвязных сетей



Архитектура полносвязных искусственных нейронных сетей (рис. 1.10.) подразумевает, что нейроны передают входные сигналы друг другу, а также самому себе. Также все входные сигналы подаются всем нейронам сети. Все или некоторые выходные сигналы нейронов после какого-то количества итераций обучения образуют выходные сигналы нейронной сети.

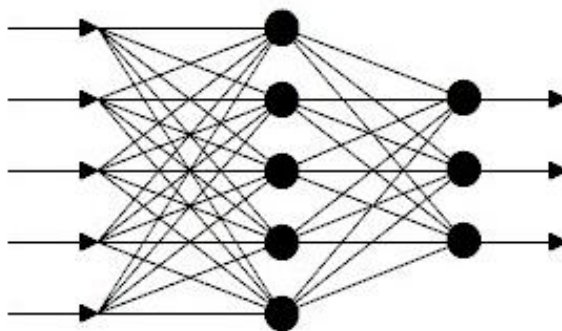


Рис. 1.11. Архитектура многослойных сетей

Для многослойных искусственных нейронных сетей (рис. 1.11.) характерно объединение нейронов в слои. Слой – это набор нейронов, имеющих одинаковые сигналы на входе. Не существует зависимости количества нейронов на конкретном слое от количества нейронов на остальных слоях, это значение может быть абсолютно любым. Также не существует четких правил по расчету количества слоев и количества нейронов на каждом из слоев.

Общий вид сети таков: сеть содержит  $Q$  слоев, каждый из которых пронумерован (нумерация слева направо). На входы нейронов нулевого (входного) слоя подаются внешние (входные) сигналы. Выходные сигналы последнего из слоев представляют собой выходы сети. Помимо перечисленных выше входного и выходного слоев многослойная нейронная сеть содержит хотя бы один скрытый слой, количество скрытых слоев не ограничено. Связи между соседними слоями, имеющие направление слева направо, называются последовательными.

Многослойные искусственные нейронные сети также подразделяются на такие типы как:

1. Монотонные сети – представляют собой особое строение многослойных сетей, которые содержат дополнительные условия для нейронов и связей между ними. Так нейроны на входном и скрытых слоях делятся на возбуждающие и тормозящие. Связи между нейронами также бывают возбуждающими и тормозящими.

В случае, если от нейронов группы  $A$  отходят только возбуждающие сигналы к группе нейронов  $B$  – такое строение эти говорит о том, что каждый выходной сигнал группы образует неубывающую функцию для каждого выходного сигнала группы  $A$ . В случае, если участвуют только тормозящие сигналы, то речь идет о невозрастающей функции между группами нейронов. Таким образом, обеспечивается монотонная зависимость выходного сигнала, от значений входных сигналов, что и характерно для монотонных сетей.

2. Сети без обратных связей – представляют собой такие сети, где на вход нейронов нулевого слоя подаются сигналы, нейроны слоя преобразуют сигналы и передают на вход нейронов первого скрытого слоя и такая эстафета продолжается далее до выходного слоя, задачей которого является передать сигнал для пользователя и интерпретатора. По умолчанию, каждый выходной сигнал слоя переходит на вход всех нейронов соседнего (правого) слоя  $(q+1)$ -го слоя, но помимо такой реализации, возможен вариант, когда слой соединяется с произвольным слоем (справа).

Внутри типа многослойных сетей без обратных связей различают полносвязные сети (выход каждого нейрона слоя связан со входом каждого нейрона соседнего слоя), а также частично полносвязные сети. Полносвязные сети прямого распространения (рис. 1.12.) являются образцовым представителем многослойных сетей.

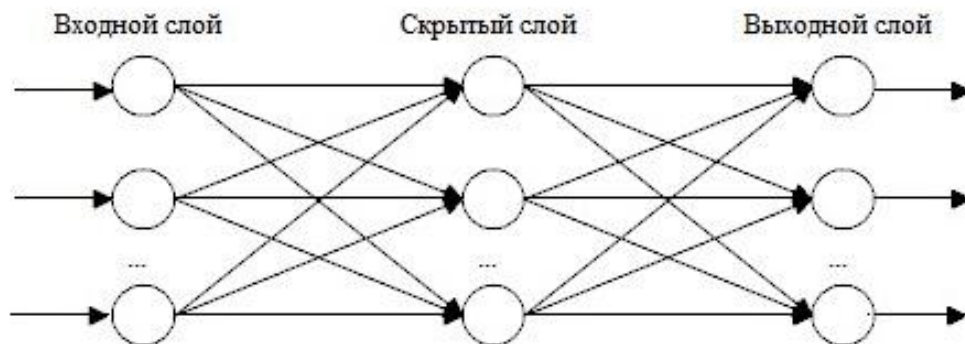


Рис. 1.12. Многослойная (двухслойная) сеть прямого распространения

3. Сети с обратными связями – в данных сетях сигналы последующих слоев поступают на предыдущие слои. Различают несколько видов сетей с обратными связями, а именно:

- слоисто-полносвязные – нейронные сети, состоящие из слоев, каждый из которых – полносвязная сеть, сигналы же передаются между слоями, также сигналы передаются внутри слоев. Каждый цикл работы слоя такой сети содержит три периода: получение сигнала с предыдущего слоя, передача сигналов внутри слоя, формирование и передача сигнала к последующему слою;

- полносвязно-слоистые – во многом схожие по структуре со слоисто-полносвязными, однако, отличие заключается в функционировании сети: циклы работы не разделяются на периоды – в каждом цикле нейроны во всех слоях обмениваются сигналами и внутри своего слоя и обмен сигналами с предыдущим и последующим слоев.

- слоисто-циклические – такие сети отличает то, что слои замкнуты, поэтому последний слой передает выходные сигналы на вход первого слоя. Каждый слой может и получать и отправлять сигналы.

При классификации искусственных нейронных сетей можно поделить сети по типам структур на гомогенные и гетерогенные сети. Состав гомогенных сетей представляет собой нейроны одного типа и единственную

функцию активации. Гетерогенная сеть состоит из нейронов и множества функций активации[39].

По типу входящей информации искусственные нейронные сети можно поделить на бинарные и аналоговые сети. Бинарные сети выполняют операции только с двоичными сигналами, поэтому выход каждого нейрона сети может содержать значение нуля, что обозначает торможение, либо единицы – возбуждение.

Также существует классификация, которая взяла за основу время передачи сигнала, по такому критерию сети делятся на синхронные и асинхронные сети. В синхронных нейронных сетях – состояние меняется у одного нейрона в момент времени, а в асинхронных – изменение состояние происходит одновременно у множества нейронов, зачастую затрагивает весь слой.

### **1.3.2 Алгоритмы обучения нейронных сетей**

Искусственные нейронные сети на данный момент не могут симитировать работу мозга, но в рамках конкретной задачи использование нейронных сетей позволяет достичь положительных результатов. В связи с чем, искусственные нейронные сети получили широкое распространение. Так, например, нейронные сети используются в распознавании образов или интеллектуальном поиске. Основным преимуществом искусственной нейронной сети является её обучаемость и способность решать заранее неизвестные задачи. Однако, у нейронной сети есть и недостаток – возможность переобучения, следствием чего является увеличение погрешности при решении поставленных задач.

Для обучения искусственных нейронных сетей существуют разнообразные методы. Ввиду того, что эти методы созданы на основе общих предпосылок, отличия между ними не так много. В целом все методы можно

разделить на две основные группы: обучение с учителем и обучение без учителя.

При использовании алгоритмов вида «обучение с учителем» существует некий «учитель», который контролирует и корректирует поведение искусственной нейронной сети: задает входные данные, оценивает корректность выходных данных, и в случае. Если результирующие данные далеки от ожидаемых – «учитель» корректирует значения весов сети для уменьшения величины такой погрешности. Однако, такой подход противоречит биологическому строению нейронной сети, а так как искусственная нейронная сеть – это математическая модель настоящей нейронной сети, то противоречие и отражается на искусственной нейронной сети, так по крайней мере, считает ряд исследователей.

Алгоритм обучения «с учителем» нейронной сети наглядно демонстрирует однослойный персептрон (рис. 1.13.)

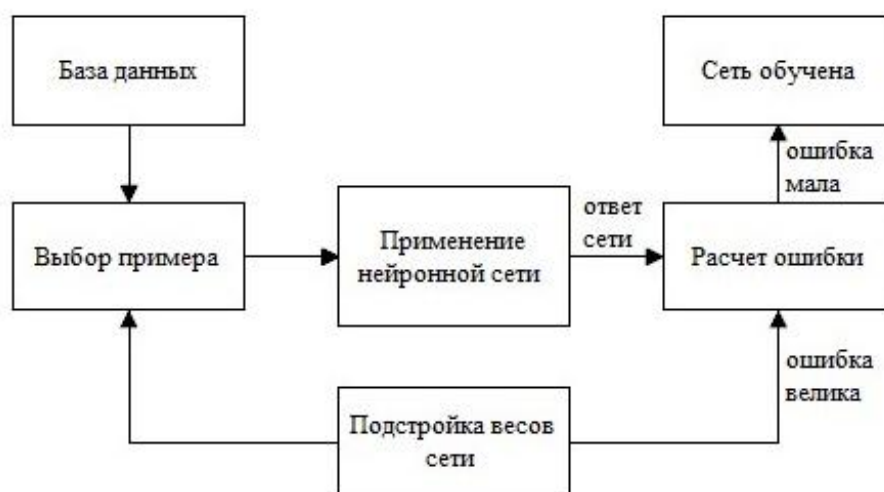


Рис. 1.13. Алгоритм обучения нейронной сети с учителем

Алгоритм обучения искусственной нейронной сети с учителем можно разбить на несколько этапов:

1. Инициализация весов, представленных в виде матрицы (как правило, веса задаются произвольно).

2. На входы нулевого слоя поступает один из наборов значений, которые сеть должна научиться определять, вычисляется выходное значение.

3. В случае, если выходное значение соответствует должному, осуществляется переход к 4 этапу. В противном случае, вычисляется разницу между должным  $y^n$  и полученным  $y$  значениями:

$$\delta = y^n - y ()$$

После чего переопределяются веса по приведенной формуле:

$$w_{ij}(t + 1) = w_{ij}(t) + \eta * \delta * x_i, (1.21)$$

где под  $t$  и  $(t+1)$  понимаются номера итераций (текущей и следующей);  $\eta$  – коэффициент скорости обучения;  $i$  – номер входа;  $j$  – номер нейрона в слое.

Затем значения весовых коэффициентов корректируется в соответствии с формулой.

4. Повторение 2 этапа. Повтор осуществляется до тех пор, пока нейронная сеть не перестанет допускать ошибки.

На втором этапе, в различных итерациях сети задаются все возможные наборы значений в произвольном порядке. Число итераций при обучении заранее неизвестно[34].

Алгоритм «обучение без учителя» не имеет принципиальных отличий от алгоритма «обучение с учителем», однако, является более самостоятельным. Так процедура обучения, также сводится к корректированию весовых коэффициентов синапсов, однако такая коррекция проводится на основании информации о самом нейроне: его состоянии и текущем весовом коэффициенте синапса. Такой принцип наиболее соответствует изначальной идее о подобию биологической нейронной сети. Данные принципы и лежат в основе алгоритма обучения Хебба.

Основой сигнального метода обучения Хебба является коррекция весовых коэффициентов по формуле:

$$w_{ij}(t) = w_{ij}(t - 1) + \alpha * y_i^{(q-1)} * y_i^{(q)}, (1.22)$$

Где  $y_i^{(q-1)}$  – выходное значение для нейрона  $i$ -того слоя ( $q-1$ ),  $y_j^{(q)}$  – выходное значение для нейрона  $j$  слоя  $q$ ; а  $w_{ij}(t)$  и  $w_{ij}(t-1)$  – представляют собой весовые коэффициенты синапса, который соединяет указанные нейроны на итерациях  $t$  и  $t-1$  соответственно;  $\alpha$  – обозначает коэффициент скорости обучения (под  $q$  имеется в виду любой слой сети). Обучение сети по данному алгоритму приводит к усилению связей между нейронами, находящимися в возбужденном состоянии.

Также существует дифференциальный метод обучения Хебба, при использовании которого коррекция весов коэффициентов осуществляется по формуле:

$$w_{ij}(t) = w_{ij}(t-1) + \alpha * [y_i^{(q-1)}(t) - y_i^{(q-1)}(t-1)] * [y_j^{(q)}(t) - y_j^{(q)}(t-1)]$$

, (1.23.)

Где переменные  $y_i^{(q-1)}(t)$  и  $y_i^{(q-1)}(t-1)$  – обозначают выходное значение для  $i$ -того нейрона слоя  $q-1$  на итерациях  $t$  и  $t-1$ ;  $y_j^{(q)}(t)$  и  $y_j^{(q)}(t-1)$  – аналогично по отношению к  $j$ -тому нейрону слоя  $q$ . Из формулы (2) можно сделать вывод, что наиболее подвержены обучению те синапсы, которые соединяют нейроны, значение выходного слоя которых наиболее оперативно увеличилось.

Алгоритм обучения без учителя, с учетом приведенных выше формул в общем виде выглядит так:

1. Инициализация весов, представленных в виде матрицы (как правило, веса задаются произвольно).

2. На входы нулевого слоя поступает входной образ, а сигналы возбуждения передаются по слоям в соответствии с принципом прямого распространения, который заключается в том, что для каждого нейрона производится расчет взвешенной суммы его входов, которая затем пересчитывается с учетом активационной функции, результатом перечисленных манипуляций является выходное значение  $y_i^{(q)}$ , где  $i = 1, \dots, L_q$ , где  $L_q$  – число нейронов в слое  $q$ ; где  $q = 1, \dots, Q$ , а  $Q$  – число слоев в сети.

3. К полученным на предыдущем этапе выходным значениям нейронов применяются формулы (1.22), происходит корректировка весов нейронов.

4. Повторение алгоритма с этапа 2 до тех пор, пока выходные значения нейронов не начнут стабильно выводить результат, соответствующий заданной точности.

Применение данного способа определения момента завершения процесса обучения обеспечивает неограниченность корректировки значений весовых коэффициентов.

В данной главе приведены и описаны некоторые из возможных методов решения поставленной задачи. Приведены классификации моделей прогнозирования и архитектур нейронных сетей. Описаны используемые алгоритмы.



## 2 ОПИСАНИЕ СРЕДСТВ РЕАЛИЗАЦИИ

Инвестиции всегда были самым привлекательным способом получения финансовой выгоды, т.к. вложение инвестиций не требует больших временных и физических усилий. Такой метод получения всегда находит своего пользователя. Но инвестирование сопряжено с большими рисками, т.к. финансовые показатели не проявляют большой стабильности. В попытках уменьшить неопределенность появились прогнозы во всем многообразии своих методов.

Главным инвестиционным объектом на сегодняшний день являются акции, котировки, валюта, драгоценные металлы и прочее. Вопросом прогнозирования данных показателей задаются многие люди. В настоящее время существуют довольно результативные программные решения, позволяющие получить прогноз для выбранного показателя. Однако, данный вопрос не закрыт, т.к. не существует способа, формирующего абсолютно точный прогноз. Поэтому пока погрешность прогнозирования не станет равна нулю, прогнозирование останется актуальной проблемой.

Для решения задачи прогнозирования существуют стандартные и довольно эффективные методы, а именно такие группы методов как фактографические и экспертные методы. Фактографические методы отличаются тем, что данные методы основаны на анализе и использовании объективных данных, в основе которых лежит математический анализ и построение моделей. К фактографическим методам относятся: многомерные регрессионные методы, эконометрические методы и метод компьютерной имитации. Важной характеристикой экспертных методов является наличие человеческого фактора, а именно так называемых «экспертов». Среди экспертных методов можно отметить метод Дельфи (метод экспертных оценок), метод консенсуса и историческая аналогия. Данные методы достаточно эффективны, но имеют значительные недостатки. Например, фактографические методы требуют большой объем исходных данных, а

также требуют громадных затрат времени на проведения анализа, построение моделей и прочее. Экспертные методы не требуют столь значительного объема данных как фактографические, однако использование данных подходов требует наличие некоторого количества «экспертов» и использование данных методов становится невозможным для маленьких компаний (затруднительным для средних), а также для отдельных индивидов.

Существуют смешанные методы прогнозирования, сочетающие экспертное суждение и математические модели. Данные методы были созданы в попытке устранить недостатки обоих подходов и увеличить эффективность прогнозирования.

В данной работе описана методика прогнозирования, основанная в большей части на основании фактографического подхода, так как входные данные подвергаются обработке и структуризации с помощью математических методов, в процессе обработки исходные данные представляются в виде временного ряда для дальнейшего удобства работы с ними. Также методика включает в себя использование нейронной сети, использующая опыт, полученный в результате обучения сети: веса нейронов, входящих в структуру нейронной сети рассчитываются на основании большого потока данных. В каком-то смысле нейронная сеть выступает в роли «эксперта» в рамках данной методики.

Главной причиной представления данных в виде временного ряда в данной методике является тот факт, что курс любой котировки тесно привязан не только к значению, но и ко времени, таким образом, значение курса доллара равное к «28.77» не может быть полезной информацией, так как не обладает второй по важности характеристикой времени. Однако, даже если внести ясность и указать, что данное значение курс доллара имел в период с 12 по 15 февраля 2000 года, останется еще один важный вопрос: по отношению к какой валюте: рублю, гривне, йене? Стоит заметить, что не по отношению к разным валютам определенный показатель ведет себя по-разному. Экономические котировки и показатели чаще всего имеют крайне

сложную зависимость, а иногда не имеют ее вовсе. Изучение такой разности в зависимости между показателями и будет описана в методике ниже.

## **2.1 Описание методики прогнозирования**

Для осуществления краткосрочного прогнозирования в сфере финансов недостаточно лишь выбрать подходящую модель прогнозирования и применить ее на определенном наборе значений. Необходимо разработать структурированный и обоснованный подход к проблеме – методику прогнозирования.

В рамках данной задачи была сформирована методика, приведенная ниже. Однако, стоит отметить, что данная методика имеет примерный характер для остальных задач, так как предназначена указанная методика только для решения конкретной задачи.

Разработанная методика состоит из четырех частей: сбор данных, подготовка данных (формирование временного ряда), формирование структуры нейронной сети и тестирование. Первый и второй этап носят исключительно подготовительный характер, третий и четвертый наиболее подробно описаны в 3 разделе работы. Суть подготовительной части методики состоит в том, что необходимо было подобрать несколько различных экономических показателей и оценить их взаимное влияние с предсказываемым показателем, а также привести подходящие показатели к виду временного ряда.

### **2.1.1 Поиск и обработка показателей**

Первый этап методики – один из самых важных аспектов этапов подготовки. Необходимо было решить несколько проблем на этом этапе: поиск показателей и выбор наиболее подходящих для анализа, анализ показателей для определения самых зависимых показателей.

Для поиска показателей были использованы различные источники, были исследованы различные показатели: валюты и котировки. Однако, после подэтапа поиска показателей оказалось значительно меньше. Самой главной проблемой поиска оказалось то, что значения котировок и валют зависят от работы биржевых рынков, у которых также имеются выходные и праздничные дни. И если выходные дни у них совпадают, то праздничные дни в разных странах отличаются, таким образом, количество значений для каждого показателя за один период времени может сильно варьироваться. Поэтому на данном этапе были отвергнуты многие показатели, у которых отсутствовало больше чем 30% показателей по сравнению с предсказываемым показателем. Таких показателей было достаточно много, так у многих валют отсутствовала значительная часть показателей, возможно, такая проблема обусловлена тем, что в некоторых государствах подобная статистика не ведется в достаточном объеме.

Как было отмечено раньше выходные дни для биржевых рынков разнятся для большинства стран, в связи с чем, не для каждого рабочего дня можно определить значение всех показателей. В таком случае, необходимо устранить несоответствие значений друг другу, у данной проблемы несколько решений: во-первых, можно дополнить ряд значений, продублировав предыдущие значения, во-вторых, можно устранить те данные, которые не имеют соответствий, т.е. если в строчке нет хотя бы одного значения – удалять весь набор данных за этот день. Оба варианта приемлемы, однако, у обоих имеются недостатки и преимущества. Так с помощью первого варианта, возможно, максимально приблизиться к представлению данных в виде временного ряда, так как данные за определенный период времени предоставляются с одинаковой

периодичностью (каждый день). То есть дополнить данными необходимо не только праздничные дни, но и выходные дни. Недостатком такого подхода является то, что при тестировании и использовании в наборе данных для прогноза также необходимо дополнить данные. Преимуществом второго варианта является наиболее высокое значение корреляции между показателями, так как отсутствуют дубли значений, к недостаткам метода можно отнести то, что предоставление данных удалено от концепции временного ряда.

В результате, для снижения недостатков обоих вариантов, было решено использовать их смешение, так ряды были дополнены только на праздничные дни.

12.04.2017	2 299,38	1 446,61	32,85	1916,5	1 728,61	60,3042	56,9552	20592	1078,72	55,62
11.04.2017	2 313,04	1 474,25	33,10	1919	1 752,86	60,7469	57,3896	20651	1091,08	56,29
07.04.2017	2 274,91	1 462,48	33,06	1944	1 738,28	60,0827	56,4369	20656,1	1113,45	55,26
06.04.2017	2 250,78	1 453,80	32,81	1968,5	1 737,73	59,6948	55,894	20663	1148,75	54,9
05.04.2017	2 288,60	1 463,73	33,35	1972	1 738,29	60,2427	56,5553	20648	1152,49	54,01
04.04.2017	2 249,39	1 440,33	32,78	1949	1 720,10	59,8953	56,1396	20689	1135,98	54,26
31.03.2017	2 267,37	1 428,32	32,81	1954,5	1 732,84	60,595	56,3779	20663,22	1113,76	53,62
30.03.2017	2 297,02	1 453,86	33,24	1960	1 750,86	61,5347	57,0241	20728	1137,26	53,07
29.03.2017	2 294,86	1 442,47	32,84	1938,5	1 762,82	61,8102	56,9364	20659	1124,91	52,55
28.03.2017	2 304,33	1 477,67	32,89	1928,5	1 793,01	61,9615	57,0233	20702	1125,58	51,42
24.03.2017	2 307,87	1 466,57	32,51	1930	1 779,12	62,0959	57,5228	20596,72	1124,66	50,97
23.03.2017	2 309,07	1 447,22	32,08	1923	1 791,89	62,2699	57,636	20657	1124,35	50,51
22.03.2017	2 267,05	1 442,61	31,70	1921,5	1 771,98	61,7308	57,2323	20661	1123,17	50,68
21.03.2017	2 270,87	1 425,51	32,05	1930	1 775,44	61,6956	57,2847	20668	1121,63	50,85
17.03.2017	2 295,03	1 430,65	31,67	1911	1 801,42	62,4897	58,2437	20914,62	1111,89	51,67
16.03.2017	2 284,90	1 421,59	32,31	1906	1 786,49	62,8428	59,1128	20935	1096,29	51,74
15.03.2017	2 281,23	1 419,66	32,26	1875,5	1 776,00	62,7447	58,954	20950	1062,61	52
14.03.2017	2 296,22	1 429,67	32,11	1862,5	1 796,60	63,2661	59,1327	20837	1063,69	51,84
11.03.2017	2 278,09	1 420,30	32,63	1897,5	1 789,65	62,7408	59,2174	20902,98	1055,93	51,27
10.03.2017	2 278,48	1 435,64	32,91	1897,5	1 793,13	61,9911	58,8318	20902,98	1055,93	51,27
08.03.2017	2 292,23	1 442,36	33,36	1889	1 817,00	61,7063	58,263	20856	1097,44	53,16
07.03.2017	2 309,12	1 447,95	33,12	1867	1 853,07	61,8606	58,337	20925	1097,44	55,6
03.03.2017	2 334,69	1 457,19	34,42	1913	1 898,47	61,5198	58,4067	21005,71	1108,26	55,71
02.03.2017	2 338,69	1 460,21	34,31	1936	1 927,56	61,5417	58,3776	21003	1099,82	55,03
01.03.2017	2 332,97	1 459,15	34,18	1939	1 911,99	61,3883	57,9627	21116	1109,39	56,28
28.02.2017	2 340,04	1 439,88	34,03	1895,5	1 911,15	61,2569	57,9371	20812	1099,46	56,45
23.02.2017	2 286,78	1 446,91	33,06	1890,5	1 853,45	60,4535	57,4762	20810	1146,01	56,47
22.02.2017	2 285,64	1 419,34	33,45	1877,5	1 847,19	61,209	57,859	20776	1146,01	56,19
21.02.2017	2 307,45	1 447,58	33,63	1890	1 869,72	61,6987	58,0967	20743	1161,65	56,67
17.02.2017	2 272,45	1 451,57	32,85	1879	1 859,49	60,654	57,1507	20624,05	1152,21	55,72
16.02.2017	2 236,21	1 423,70	32,68	1892	1 821,61	60,0249	56,7719	20620	1176,11	55,81
15.02.2017	2 282,65	1 444,24	33,36	1884,5	1 856,34	61,2955	57,7388	20612	1172,59	55,69
14.02.2017	2 294,96	1 448,58	32,89	1901	1 879,80	61,7662	58,0619	20504	1173,97	55,8
10.02.2017	2 356,40	1 466,88	33,66	1858	1 935,60	63,0253	59,0235	20269,37	1164,21	56,61
09.02.2017	2 364,18	1 473,30	33,68	1855,5	1 934,43	63,4467	59,5129	20172	1163,19	55,74
08.02.2017	2 342,73	1 436,85	33,49	1840,5	1 912,62	63,2362	59,1933	20054	1164,65	55,16
07.02.2017	2 306,41	1 442,15	32,62	1824	1 902,74	63,1566	58,7121	20090	1179,21	54,66
03.02.2017	2 360,69	1 463,80	33,94	1815,5	1 934,37	64,7307	59,9858	20071,46	1190,4	56,72
02.02.2017	2 346,20	1 481,40	33,53	1821,5	1 931,25	65,0322	60,3099	19885	1171,42	56,74

Рис.2.1. Дополненные значения показателей (дополненные строки выделены цветом)

## 2.1.2 Анализ взаимного влияния финансовых показателей

После того как были отобраны данные этап сбора данных завершился. Однако, лишь одних данных недостаточно, так как в дальнейших этапах прогнозирования необходимо учитывать только те данные, которые оказывают наибольшее влияние на прогнозируемый показатель. Второй этап методики заключается в преобразовании данных, собранных на первом этапе.

Осуществление анализа взаимного влияния финансовых показателей – важный этап для данной работы, важность заключается в том, что нельзя просто заявить, что показатели влияют друг на друга. Основная задача анализа – определить наиболее влияющие друг на друга показатели. Использование тесной взаимосвязи показателей в прогнозировании может привести к уменьшению ошибки прогноза, что и является главной задачей прогнозирования.

Важным моментом второго этапа является определение взаимосвязи показателей, а именно определении корреляции. Современная статистика предоставляет несколько различных способов для определения корреляции:

- «коэффициент корреляции Пирсона» используется для определения линейной зависимости между двумя показателями;
- «коэффициент корреляции Кендела», «коэффициент корреляции Спирмена», «коэффициент корреляции Фехнера» применяются для определения линейной или нелинейной зависимости между случайными величинами, относятся к коэффициентам корреляции знаков;

Так как показатели в данной работе всегда имеют лишь положительные значения, при определении корреляции (взаимного влияния) показателей будет применяться коэффициент Пирсона.

Для анализа были взяты курсы валют государств, экономическая деятельность оказывает наибольшее воздействие на мировую экономику. Для анализа также использовались цены на драгоценные металлы, нефть (Brent) и

значения индексов ИДД (Индекс Доу-Джонса), РТС (российский фондовый индекс).

Дата	Золото	Палладий	Серебро	Алюмини	Платина	Евро	Доллар	ИДД	РТС	Brent	GBP	CNY	KRW
02.06.2017	2 301,50	1 483,26	31,14	1914	1 714,11	63,531	56,5373	21206,29	1046,02	49,95	72,7974	8,30747	0,050387
01.06.2017	2 303,34	1 467,15	31,55	1930,5	1 715,02	63,4107	56,6876	21144	1045,63	50,4	72,5771	8,31038	0,050659
31.05.2017	2 294,58	1 446,38	31,38	1919	1 718,93	62,9484	56,5168	21009	1053,3	50,98	72,6636	8,24389	0,050211
30.05.2017	2 306,55	1 422,16	31,52	1943,5	1 752,18	63,3684	56,7106	21029	1074,81	52,19	72,7937	8,27216	0,050559
26.05.2017	2 266,17	1 379,06	30,92	1953,5	1 703,55	63,0116	56,0701	21080,28	1073,04	52,58	72,9528	8,16468	0,050134
25.05.2017	2 264,02	1 400,37	30,81	1956	1 704,32	62,9203	56,2743	21083	1083,52	51,64	73,0834	8,1667	0,050076
24.05.2017	2 290,86	1 401,90	31,17	1944	1 720,10	63,6189	56,5552	21012	1087,59	53,86	73,4426	8,20854	0,050363
23.05.2017	2 280,13	1 382,34	30,79	1930	1 703,86	63,1713	56,4988	20938	1096,83	54,16	73,3467	8,19738	0,050528
19.05.2017	2 330,53	1 437,47	31,06	1938	1 729,40	63,9967	57,4683	20804,84	1087,75	53,8	74,6111	8,34059	0,051061
18.05.2017	2 270,37	1 437,45	30,83	1906	1 712,90	62,9568	56,7383	20663	1076,8	52,5	73,391	8,23332	0,050669
17.05.2017	2 232,16	1 454,28	30,24	1926	1 689,43	62,0382	56,2603	20607	1104,3	52,09	72,8233	8,16538	0,050381
16.05.2017	2 238,06	1 468,42	30,15	1912,5	1 682,86	61,8449	56,5258	20980	1112,95	51,23	73,0596	8,19488	0,050359
12.05.2017	2 242,15	1 472,73	30,06	1884	1 683,91	62,1595	57,1161	20896,61	1099,75	50,91	73,8968	8,2729	0,050621
11.05.2017	2 283,73	1 488,31	30,42	1890	1 689,99	63,2634	58,0824	20919	1105,83	50,77	75,4084	8,41347	0,05118
05.05.2017	2 287,51	1 488,17	30,54	1914	1 673,27	62,7298	57,5714	21006,94	1085,68	49,47	74,1692	8,34755	0,050776
04.05.2017	2 301,71	1 479,47	30,93	1916,5	1 677,71	62,3224	57,0927	20951	1081,22	48,35	73,8323	8,28415	0,050453
03.05.2017	2 299,42	1 488,64	31,04	1922,5	1 712,02	62,1515	56,9518	20958	1096,35	50,62	73,3767	8,25759	0,050405
28.04.2017	2 315,75	1 470,81	31,98	1938	1 734,57	62,1664	56,9707	20940,51	1114,43	51,89	73,458	8,26597	0,050259
27.04.2017	2 290,20	1 452,03	31,85	1962	1 725,41	61,5052	56,3131	20981	1106,92	52,07	72,1258	8,17233	0,049867
26.04.2017	2 281,14	1 420,21	32,03	1964	1 718,26	60,7932	55,8453	20975	1119,11	52,07	71,549	8,11009	0,049601
25.04.2017	2 293,05	1 442,39	32,11	1954	1 741,69	60,8461	56,0794	20996	1120,89	52,42	71,8994	8,14386	0,049615
21.04.2017	2 321,52	1 427,49	32,99	1939,5	1 755,79	60,6139	56,4165	20547,76	1084,17	52,05	72,3993	8,19567	0,049549
20.04.2017	2 315,48	1 399,71	32,91	1942,5	1 760,92	60,2649	56,1753	20579	1083,35	52,99	71,9886	8,15648	0,04926
19.04.2017	2 312,71	1 421,82	33,15	1909	1 769,18	59,6124	55,9793	20404	1068,55	53,08	70,4947	8,1305	0,048989
18.04.2017	2 322,38	1 459,45	33,57	1930	1 763,28	59,8393	56,2505	20523	1077,21	54,82	70,4931	8,16704	0,049457
13.04.2017	2 321,61	1 463,44	33,41	1902,5	1 762,69	60,2631	56,7556	20453,25	1089,39	55,64	70,9502	8,23452	0,049813
12.04.2017	2 299,38	1 446,61	32,85	1916,5	1 728,61	60,3042	56,9552	20592	1078,72	55,62	70,7384	8,25091	0,049698
11.04.2017	2 313,04	1 474,25	33,10	1919	1 752,86	60,7469	57,3896	20651	1091,08	56,29	71,1746	8,31071	0,050312
07.04.2017	2 274,91	1 462,48	33,06	1944	1 738,28	60,0827	56,4369	20656,1	1113,45	55,26	70,3655	8,17879	0,049865

Рис.2.2. Рассматриваемые значения показателей

Однако, данные, представленные на рис. 2.2., без дальнейшей обработки могут стать причиной погрешности в результатах, т.к. различные показатели находятся в разных числовых диапазонах. Для того, чтобы избежать подобных ошибок, необходимо прибегнуть к нормализации данных (рис. 2.3.), т.е. привести значение ряда к интервалу [0, 1].

Следующий шаг – рассчитать коэффициент корреляции между значениями.

Корреляция — статистическая взаимосвязь двух или более случайных величин (либо величин, которые можно в некоторой допустимой степени точности считать таковыми).

Золото	Палладий	Серебро	Алюмини	Платина	Евро	Доллар	ИДД	PTC	Brent	Фунт	Юань	Вона
0,1062	1	0,054333	0,854406	0,060458	0,131791	0,038359	1	0,679425	0,654639	0,057478	0,057851	0,070394
0,157515	0,812147	0,155433	0,866858	0,072047	0,143698	0,041101	0,981971	0,728279	0,702749	0,08999	0,061518	0,087919
0,155224	0,833149	0,139615	0,868774	0,08812	0,130048	0,026761	0,966403	0,723008	0,687973	0,070759	0,048705	0,068125
0,149408	0,799512	0,138239	0,869732	0,091342	0,132069	0,029893	0,964828	0,715621	0,699313	0,076178	0,050846	0,082754
0,132203	0,786038	0,130674	0,842912	0,073814	0,132998	0,027748	0,958179	0,727821	0,760137	0,069975	0,047692	0,082814
0,099216	0,688481	0,079092	0,888889	0,049097	0,124128	0,024941	0,970479	0,736266	0,75945	0,06746	0,044007	0,070064
0,101262	0,657779	0,10729	0,920498	0,050191	0,120317	0,030358	0,959579	0,735578	0,774914	0,062955	0,04464	0,083661
0,09152	0,618196	0,095598	0,898467	0,054892	0,105673	0,024202	0,935956	0,749101	0,794845	0,064724	0,030185	0,061221
0,104832	0,572038	0,105227	0,945402	0,094864	0,118977	0,031186	0,939456	0,787024	0,836426	0,067384	0,036331	0,078646
0,059923	0,489899	0,063961	0,964559	0,036402	0,107675	0,008102	0,948429	0,783903	0,849828	0,070638	0,012965	0,057394
0,057532	0,530512	0,056396	0,969349	0,037328	0,104783	0,015462	0,948905	0,80238	0,817526	0,073309	0,013404	0,054488
0,087383	0,533427	0,081155	0,94636	0,056298	0,126912	0,025586	0,936481	0,809556	0,893814	0,080654	0,0225	0,068836
0,075449	0,49615	0,055021	0,91954	0,036775	0,112734	0,023553	0,923532	0,825846	0,904124	0,078693	0,020074	0,077098
0,131502	0,601216	0,07359	0,934866	0,067479	0,13888	0,058495	0,900231	0,809838	0,891753	0,10455	0,051207	0,103801
0,064594	0,601178	0,057772	0,873563	0,047643	0,105939	0,032185	0,875412	0,790532	0,847079	0,079599	0,027887	0,084152
0,022099	0,633252	0,017194	0,911877	0,019427	0,076841	0,014957	0,865613	0,839016	0,83299	0,06799	0,013118	0,069763
0,02866	0,6602	0,011004	0,886015	0,011529	0,070718	0,024526	0,930881	0,854267	0,803436	0,072822	0,019531	0,068661
0,033209	0,668414	0,004814	0,831418	0,012791	0,080684	0,045801	0,91629	0,830994	0,79244	0,089943	0,036492	0,081767
0,079453	0,698106	0,029574	0,842912	0,020101	0,115651	0,080628	0,920207	0,841714	0,787629	0,120855	0,067051	0,109767
0,083657	0,697839	0,037827	0,888889	0	0,098749	0,062211	0,935596	0,806188	0,742955	0,095513	0,05272	0,089527
0,099449	0,681259	0,064649	0,893678	0,005338	0,085844	0,044958	0,925807	0,798325	0,704467	0,088624	0,038937	0,07336
0,096903	0,698735	0,072215	0,905172	0,046585	0,08043	0,03988	0,927032	0,825	0,782474	0,079307	0,033163	0,070935
0,115064	0,664755	0,136864	0,934866	0,073694	0,080902	0,040561	0,923971	0,856876	0,826117	0,080969	0,034985	0,063631
0,086649	0,628964	0,127923	0,980843	0,062682	0,059958	0,01686	0,931056	0,843635	0,832302	0,053726	0,014628	0,044012
0,076572	0,568322	0,140303	0,984674	0,054086	0,037404	0	0,930007	0,865127	0,832302	0,04193	0,001098	0,030651
0,089818	0,610592	0,145805	0,965517	0,082253	0,039079	0,008437	0,933681	0,868265	0,84433	0,049096	0,008439	0,031352
0,121481	0,582196	0,206327	0,937739	0,099204	0,031724	0,020587	0,855247	0,803526	0,831615	0,059319	0,019702	0,028076
0,114764	0,529254	0,200825	0,943487	0,105371	0,020669	0,011894	0,860713	0,80208	0,863918	0,05092	0,011183	0,013562
0,111683	0,57139	0,217331	0,87931	0,115301	0	0,00483	0,830091	0,775987	0,86701	0,02037	0,005535	0

Рис. 2.3. Нормализованные значения

Коэффициент корреляции Пирсона рассчитывается по формуле:

$$r_{xy} = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (y_i - \bar{y})^2}} \quad (2.1)$$

где  $x^m = (x_1, \dots, x_m)$ ,  $y^m = (y_1, \dots, y_m)$  – две выборки,  $\bar{x}, \bar{y}$  – выборочные средние,  $r_{xy} \in [-1, 1]$ .

Таблица 2.1

Корреляция показателей по отношению к курсу доллара

Показатель	Коэффициент корреляции
Золото	0,75443129
Палладий	-0,567541
Серебро	0,33509102
Алюминий	-0,8606588
Платина	0,67092738
Евро	0,97462993
ИДД	-0,9233111
PTC	-0,9237439
Brent	-0,9015729
Фунт	0,94775356
Юань	0,98978624
Вона	0,95513638



Как видно из таблицы 2.1, значения курса доллара в наибольшей степени связаны с курсом китайского юаня, евро и корейской вонны.

Итоги, полученные в результате данного анализа, имеют не игнорируемое значение, т.к. такая тесная взаимосвязь между величинами должна быть учтена в краткосрочном прогнозировании.

### 2.1.3 Формирование структуры нейронной сети

Как уже отмечалось ранее, нейронная сеть имеет определенное строение, однако, наиболее важным структурным элементом нейронной сети является скрытый слой или слои (в зависимости от сложности задачи). Скрытый слой также состоит из нейронов, как и входной и выходной слои, однако, если количество нейронов во внешних слоях довольно легко определить исходя из постановки задачи, то определение количества нейронов на скрытых слоях представляет собой наиболее неоднозначный и творческий процесс. Самой большой проблемой является тот факт, что на настоящее время не существует строго определенных правил для определения количества нейронов на скрытых слоях, а имеющиеся правила носят скорее рекомендательный характер. Самым распространенным правилом для определения количества нейронов в скрытых слоях является правило «геометрической пирамиды», при использовании данного правила количество нейронов для сети с одним скрытым слоем рассчитывается как:

$k = \sqrt{n \cdot m}$ , где  $k$  – количество нейронов скрытого слоя, а  $n$  и  $m$  – количество нейронов для входного и выходного слоев соответственно.

При построении нейронной сети с двумя скрытыми слоями, при определении количества нейронов скрытых слоев данное правило используется следующим образом:

$$r = \sqrt[3]{\frac{n}{m}}$$

$$k_1 = m \cdot r^2, k_2 = m \cdot r,$$

где  $k_1$  и  $k_2$  – это число нейронов первого и второго скрытого слоя соответственно.

Впрочем, задачи, решаемые с помощью нейронных сетей, невероятно разнятся, поэтому приведенные выше формулы скорее являются рекомендацией для определения числа нейронов, так как большинство выбирает наиболее трудоемкий процесс при определении числа нейронов: начиная с одного нейрона, добавлять по одному нейрону на скрытый слой или слои до тех пор, пока полученный результат не будет достаточно точным.

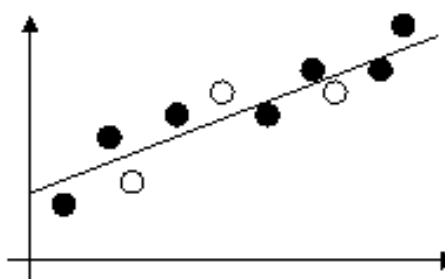


Рис. 2.4. Пример правильно обученной сети

Важно точно определить количество нейронов, так как их большее количество может привести к переобучению сети. Переобучение сети (рис.2.5.) столь же вредит результату, как и недообучение, которые и являются главными проблемами всех нейронных сетей.

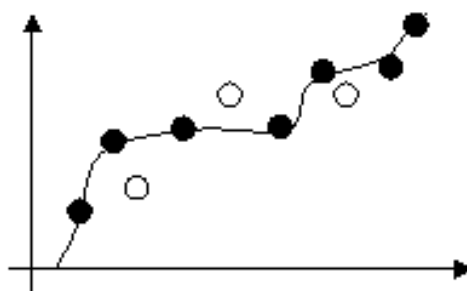


Рис. 2.5. Пример переобученной (недообученной) сети

Проблема переобучения решается все так же методом подбора количества нейронов на скрытых слоях.

Для определения количества нейронов скрытого слоя в данной работе было решено использовать метод перебора значений количества нейронов.

Так как основной минус данного подхода – это его трудоемкость, то решение данной проблемы необходимо реализовать на программном уровне, на стадии обучения нейронной сети.

#### **2.1.4 Апробация полученного инструментального средства**

Тестирование или апробация полученной модели – самый простой на первый взгляд этап представленной методики, однако, это не так. Тестирование программного продукта со стороны визуальной выглядит довольно простым, но тестирование программного продукта лишь итог самого этапа апробации. Главнейшей задачей является выбор наиболее подходящих наборов данных для тестирования. Соответственно, тестирование программного средства, представленное в разделе 3, не является первым тестированием программного продукта, оно итоговое – показательное. Для тестирования же программного продукта необходимо проведение множества тестов. В самом первом варианте было необходимо оценить степень обученности нейронной сети, таким образом, на данном этапе был использованы данные обучающей выборки, таких тестов было очень много, т.к. необходимо было проверить весь массив обучающей выборки, то есть для тестирования используется каждый набор значений. Однако, несмотря на полноту такого подхода, он требует значительных временных затрат, поэтому для тестирования использовались выборки с самыми наименьшими значениями (в выборке эти значения характеризуются нулем) и выборки с наибольшими (в выборке эти значения характеризуются единицей). Данные значения первостепенны, т.к. являются наиболее редкими наборами значений для всей выборки. Помимо приведенных выборок для тестирования использовались произвольные наборы значений (равномерно по всей обучающей выборке).

Затем необходимо тестирование на данных, не включенных в выборку, однако, до того как провести тестирование – необходимо убедиться, что данные не содержатся в выборке, это обусловлено спецификой задачи, так как, текущие данные могут совпадать с данными другого года. Данное совпадение довольно редко, однако, для всех наборов тестирования данная проверка была необходима для получения достоверных данных о работе полученного приложения.

## **2.2 Описание используемых программных продуктов**

На начальном этапе в работе был использован программный продукт Microsoft Excel, являющийся стандартным программным продуктом. Excel – это программное средство для работы с электронными таблицами, оно также имеет набор стандартных функций для работы с различными данными, а также некоторые аналитические инструменты. Потребность в данном продукте возникла в связи с необходимостью использования статистических и экономических функций. В качестве альтернативного варианта рассматривался программный продукт Statistica, имеющий схожий функционал, но обладающий большим потенциалом при работе со статистическими данными. Но конечный выбор пал на Microsoft Excel, такой выбор обусловлен тем, что интерфейс данного программного обеспечения крайне прост, возможна работа с большими массивами однотипных данных, к тому же мелкие стандартные функции позволяют использовать программный продукт за пределами его основного назначения. Недостатком же продукта можно считать недостаточную точность расчетов, особенно в отношении встроенных функций, результат вычислений, полученный с использованием которых разнится с расчетами по формулам вручную.

Для реализации самой нейронной сети была выбрана среда разработки Microsoft Visual Studio. Выбор на данную среду разработки пал в основном благодаря тому, что для реализации данной работы был выбран язык программирования C++. Однако, приведенная причина не была единственной – на сегодняшний день среда Microsoft Visual Studio представляет наиболее широкие возможности для разработки, так, например, при использовании среды возможно найти необходимую библиотеку в сети Интернет не выходя из приложения. Основным достоинством среды является возможность работы с различными библиотеками (в частности fann-библиотекой). К тому же данную среду поддерживают ее разработчики (актуальные версии), что препятствует возникновению ошибок среды при отладке кода. Очевидным преимуществом также является удобный интерфейс. К недостаткам же можно отнести сложность интерфейса, так поиск необходимого пункта меню отнимает много времени.

В данной главе рассмотрены программные продукты и средства, с помощью которых было реализовано программное средство. Приведены преимущества и недостатки выбранных программных средств.

Приведено подробное описание разработанной методики прогнозирования, а именно приведен анализ взаимного влияния показателей (курсов и котировок), описаны этапы формирования структуры нейронной сети и апробации полученной модели прогнозирования.

## 3 РАЗРАБОТКА И ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА ПРОГНОЗИРОВАНИЯ ИНВЕСТИЦИЙ

### 3.1 Описание разработанного приложения

Исходя из различных целей, методы прогнозирования делятся по периоду применения на: краткосрочные, среднесрочные и долгосрочные прогнозы. Однако, использование нейронной сети позволяет осуществлять любые виды прогнозирования, но, важно отметить, что выбранная специфика накладывает ограничение на использование нейронной сети, т.к. долгосрочный и среднесрочный прогноз финансовых показателей носят более статистический характер, нежели точное предсказание. Наиболее эффективным является краткосрочный прогноз.

Разработанный программный продукт предоставляет пользователю возможность получить краткосрочный прогноз (зависит от обучающей выборки) будущих значений курса доллара. Описание интерфейса приведено на рис.3.1.:



Рис. 3.1. Структура приложения

Как можно увидеть из рис. 3.1. – приложение имеет простую структуру. Однако, такое строение обусловлено задачей программного продукта – получение прогноза. Прогноз представляется в трех вариантах: предсказание на основе значений курса доллара, прогноз на основе значений курса доллара, коррелированного по отношению к курсу юаня/ евро.

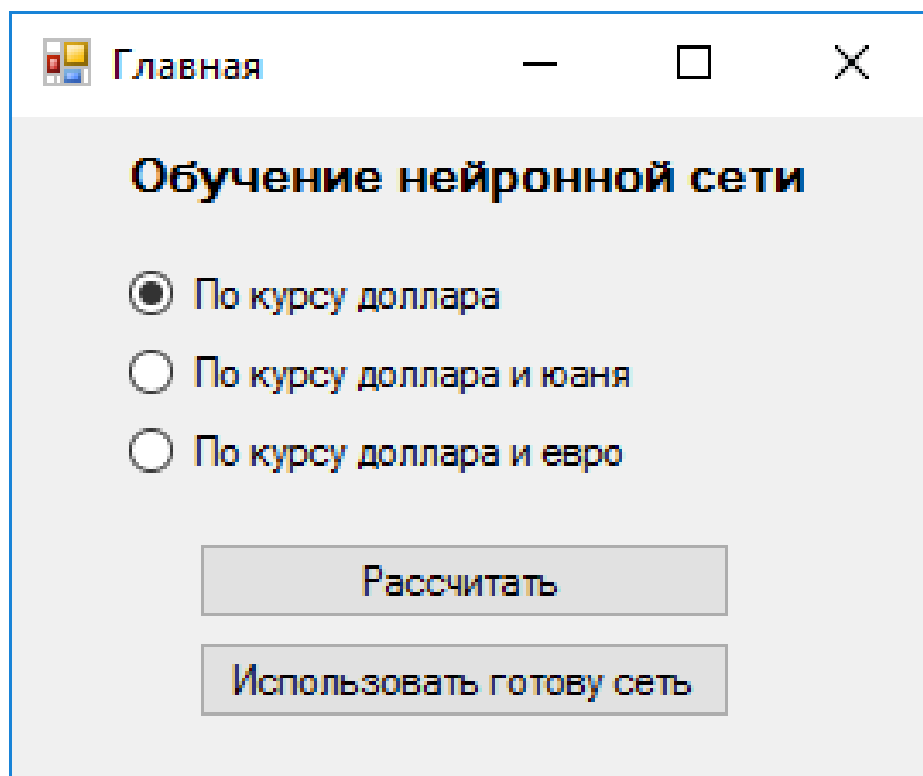


Рис. 3.2. Интерфейс программного продукта

На рис. 3.2. приведен интерфейс главной формы приложения. Так пользователю необходимо выбрать одно из значений:

- по курсу доллара;
- по курсу доллара и юаня;
- по курсу доллара и евро.

Затем нажать кнопку «Рассчитать». Затем откроется соответствующее выбору окно (рис. 3.3).

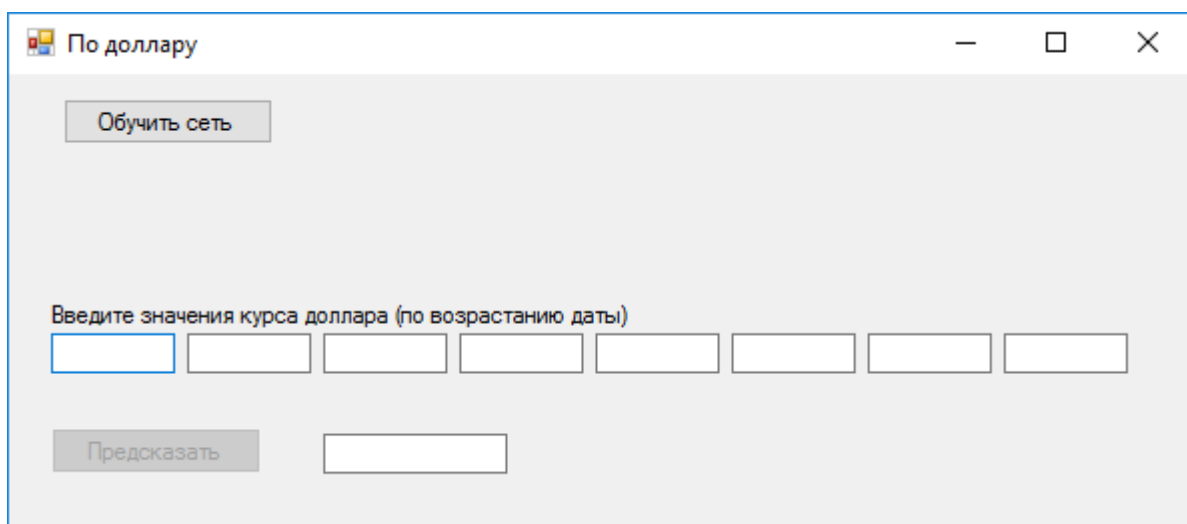


Рис. 3.3. Окно приложения «По доллару»

При переходе в данную форму (рис. 3.3.) необходимо последовательно выполнить следующие действия:

- нажать кнопку «Обучить сеть»;
- подождать пока появится надпись «Обучение завершено» (обучение может занять длительное время);
- ввести значения курса доллара;
- нажать кнопку «Предсказать».

В поле должно отобразиться полученное значение прогноза.

Описанные выше действия совершает пользователь, однако для приложения функционирование отличается. Работа приложения при выполнении описанных выше действий отражена на рисунке 3.4.

Аналогичная работа, как со стороны приложения, так и со стороны пользователя, осуществляется на остальных формах: «по доллару и юаню», «по доллару и евро»





Рис. 3.4. Работа с приложением в окне «По доллару»

Самая важная часть работы программного продукта – обучение искусственной нейронной сети. Обучение искусственной нейронной сети производится по алгоритму «без учителя» на основе выборки данных. Схема процесса обучения представлена на рис. 3.5.:

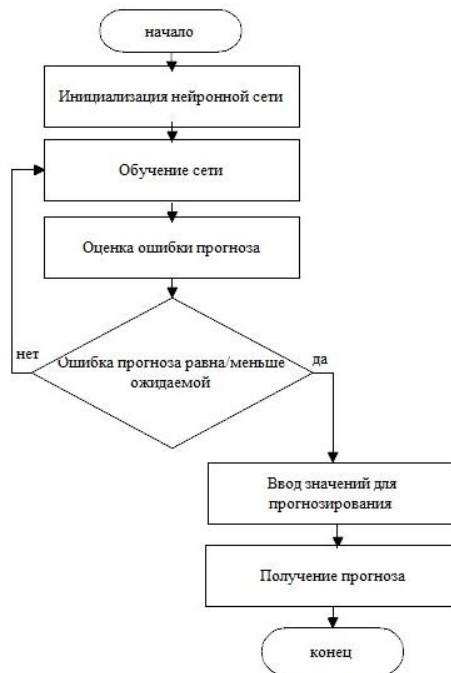


Рис. 3.5. Схема обучения нейронной сети

Процесс обучения нейронной сети во многом зависит от обучающей выборки, а именно количества входных нейронов и количества наборов входных данных. Обучающая выборка для нейронной сети окна «По доллару» представлена на рис. 3.6.

Представленные в выборке данные – нормализованные значения показателя доллара. Первая строка содержит три числа, описывающих выборку: первое число – количество наборов входных данных, второе число – количество входов сети, третье число – количество выходов сети. В случае, если необходимо обучить сеть на меньшем объеме данных не обязательно менять файл выборки – достаточно уменьшить первое число. В случае, если нейронная сеть успешно обучается – созданная сеть сохраняется в файл.

Возвращаясь к главной форме, на ней к тому же размещена кнопка «Использовать сеть», ее функционал аналогичен нажатию кнопки «Рассчитать».

Открывшаяся форма также похожа на формы обучения (рис. 3.7.). Однако есть существенные различия.

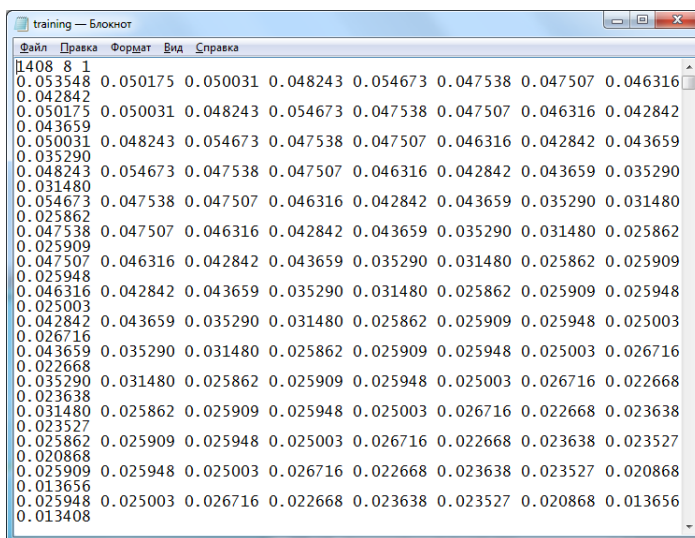


Рис. 3.6. Обучающая выборка

Прогноз, осуществляемый на форме, не намного отличается от описанных выше форм, однако, занимает меньше времени, т.к. использует уже готовую (обученную) нейронную сеть.

Для работы с окном необходимо нажать кнопку «Использовать сеть», ввести данные и нажать кнопку «Предсказать», которая станет доступной после нажатия кнопки «Использовать сеть».

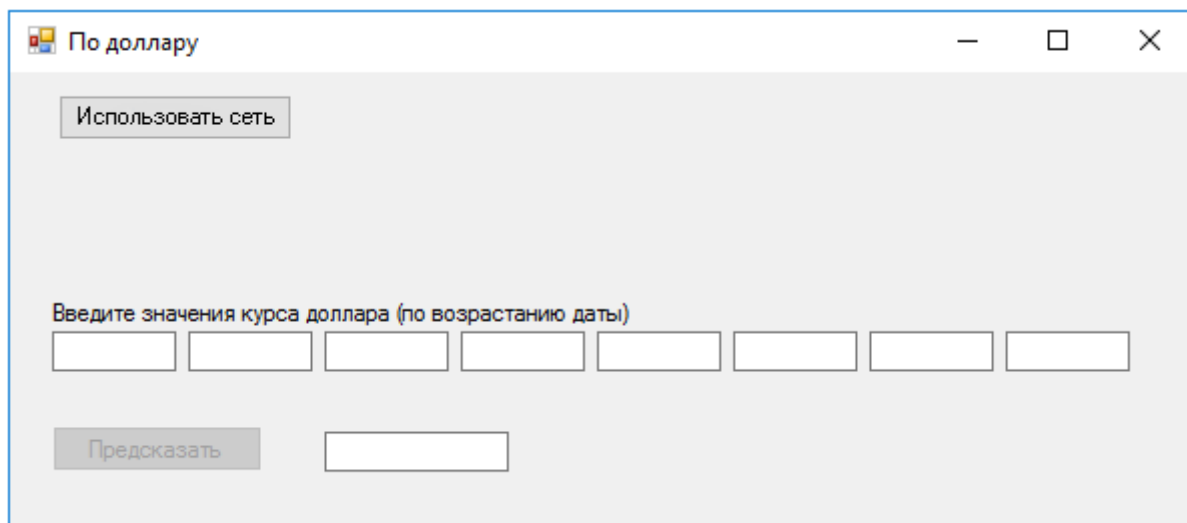


Рис. 3.7. Окно использования готовой нейронной сети

Итак, можно заключить, что использование разработанного программного приложения не составляет труда, т.к. на формах расположен минимум объектов, не позволяющих пользователю теряться при работе. Использование неактивных кнопок (которые неактивны до тех пор, пока в них не возникнет необходимость) также облегчает ориентацию в формах.

### 3.2 Тестирование программного средства

Тестирование готового приложения – один из важнейших этапов при разработке программного обеспечения, т.к. тестирование позволяет определить – соответствует ли программа заданным к ней требованиям. Предпочтительным является системный подход к тестированию, т.к. результат зависит не от особенностей самого тестирования, а от работы системы.

Для начала необходимо определить наборы входных данных для тестирования системы. К ним предъявляются следующие требования:

- наборы данных не должны входить в обучающую выборку;
- наборы данных должны содержать значения за соответствующий период.

Таблица 3.1

Тестовые выборки данных

№	доллар	юань	евро	результат
1 выборка	70,8295	10,9609	78,2312	67,7691
	70,2244	10,8717	76,9659	
	69,1755	10,7161	75,7472	
	69,2151	10,7504	76,0674	
	69,2	10,7652	75,5595	
	69,3026	10,7983	75,3111	
	68,5156	10,6915	74,4079	
	67,6698	10,5693	73,6721	
2 выборка	69,1755	10,7161	75,7472	66,2584
	69,2151	10,7504	76,0674	
	69,2	10,7652	75,5595	
	69,3026	10,7983	75,3111	
	68,5156	10,6915	74,4079	
	67,6698	10,5693	73,6721	
	67,7691	10,5922	71,7404	
	66,7402	10,4301	70,8647	
3 выборка	70,2244	10,8717	76,9659	66,7402
	69,1755	10,7161	75,7472	
	69,2151	10,7504	76,0674	
	69,2	10,7652	75,5595	
	69,3026	10,7983	75,3111	
	68,5156	10,6915	74,4079	
	67,6698	10,5693	73,6721	
	67,7691	10,5922	71,7404	
4 выборка	69,2151	10,7504	76,0674	66,737
	69,2	10,7652	75,5595	
	69,3026	10,7983	75,3111	
	68,5156	10,6915	74,4079	
	67,6698	10,5693	73,6721	
	67,7691	10,5922	71,7404	
	66,7402	10,4301	70,8647	
	66,2584	10,355	70,1345	
5 выборка	69,2	10,7652	75,5595	66,588
	69,3026	10,7983	75,3111	

	68,5156	10,6915	74,4079
	67,6698	10,5693	73,6721
	67,7691	10,5922	71,7404
	66,7402	10,4301	70,8647
	66,2584	10,355	70,1345
	66,737	10,4306	70,6478

Для достижения наиболее достоверного результата необходимо тестировать каждый набор данных на каждой нейронной сети.

Результат тестирования на наборе данных 1 для предсказания по значению показателя доллара представлен на рис. 3.8.

Рис. 3.8. Тестирование на выборке 1, окно «по доллару»

Результат тестирования на наборе данных 1 для предсказания по значению показателей доллара и юаня представлен на рис. 3.9.

Рис. 3.9. Тестирование на выборке 1, окно «по доллару и юаню»

Результат тестирования на наборе данных 1 для предсказания по значению показателей доллара и евро представлен на рис. 3.10.

Рис. 3.10. Тестирование на выборке 1, окно «по доллару и евро»

Результаты тестирования приведены в таблице 3.2 и приложении 1.

Таблица 3.2

### Результаты тестирования

	выборка 1	выборка 2	выборка 3	выборка 4	выборка 5	средняя погрешность
эталон	67,7691	66,2584	66,7402	66,737	66,588	
результат 1	66,6938	66,2491	66,816	66,8137	66,7122	
результат 2	66,6975	66,2613	66,6349	66,8	66,5208	
результат 3	66,79	66,2874	66,67	66,6808	66,4592	
погрешность 1	1,586711348	0,01418688	0,113574727	0,11492875	0,186520094	0,40318436
погрешность 2	1,581251632	0,004376804	0,157775973	0,094400408	0,100919085	0,38774478
погрешность 3	1,444758747	0,043768035	0,105183982	0,084211157	0,193428245	0,374270033

Из значений таблицы можно заключить, что результаты, полученные при тестировании нейронных сетей, использующих корреляцию при обучении, ненамного превзошли результаты тестирования нейронной сети № 1. Из чего можно заключить, что, несмотря на скромные результаты, дальнейшее исследование в данной сфере может привести к лучшим результатам.

Таким образом, третья глава посвящена разработке структуры нейронной сети и тестированию полученного программного продукта, оценке погрешностей, возникших при тестировании и расчет средней погрешности. Также приведено обоснование по поводу потенциала данного метода

## ЗАКЛЮЧЕНИЕ

Главным инвестиционным объектом на сегодняшний день являются акции, котировки, валюта, драгоценные металлы и прочее. Вопросом прогнозирования данных показателей задаются многие люди. В настоящее время существуют довольно результативные программные решения, позволяющие получить прогноз для выбранного показателя. Однако, данный вопрос далек от закрытия закрыт, так как способа, способного дать абсолютно точный прогноз не существует, поэтому задача прогнозирования инвестиций заключается в следующем – уменьшение погрешности до нулевой. Поэтому пока погрешность прогнозирования не станет равна нулю, финансовое прогнозирование останется актуальной проблемой.

В процессе осуществления первоначального обзора были рассмотрены методы, позволяющие решить задачу прогнозирования финансовых показателей. Также был осуществлен анализ взаимного влияния финансовых показателей. Так для курса доллара, который был взят за основу, самыми связанными показателями оказались китайский юань, европейский евро и корейская вона. Такой результат не был неожиданным, однако, целью анализа были выяснить тесноту связи, что и было достигнуто.

На основании проведенного анализа, были выбраны показатели евро и юаня для создания обучающей выборки для искусственной нейронной сети. Как показали результаты тестирования, такой подход имеет потенциал. Так как результаты, полученные по выборкам, скорректированным по курсу второй валюты, продемонстрировали более точный результат. Из чего можно заключить, что задачи работы были выполнены.

Из результатов можно предположить, что дальнейшее исследование взаимного влияния финансовых показателей имеет место быть, однако, не каждое влияние, оказываемое на прогнозируемый показатель можно



обличить в количественный показатель, что и служит камнем преткновения в области прогнозирования финансовых показателей.

Подводя итог, можно с уверенностью сказать, что разработанный программный продукт имеет потенциал к развитию, так как пути для модернизации этого приложения существуют, и такая модернизация не затронет функциональной ценности средства.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Аверченков В. И. Информационные системы в производстве и экономике: учебное пособие / В. И. Аверченков, Ф. Ю. Лозбинев, А. А. Тищенко – М.: Флинта, 2011. – 273 с.
2. Ажеронок В. Разработка управляемого интерфейса/ В. Ажеронок, А. Островерхов, М.Г. Радченко и др.– СПб.: 1С-Паблишинг, 2010 – 156 с.
3. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - Москва: Высшая школа, 2015. - 882 с.
4. Афанасьев, В.Н. Анализ временных рядов и прогнозирование: учебник / В.Н. Афанасьев, М.М. Юзбашев. - М.: ФиС, ИНФРА-М, 2012. - 320 с.
5. Ахременко, А.С. Политический анализ и прогнозирование в 2 ч. часть 1 2-е изд., испр. и доп. учебник и практикум для бакалавриата и магистратуры / А.С. Ахременко. - Люберцы: Юрайт, 2016. - 224 с.
6. Бабаев А.А. Информационные технологии и методы принятия решений: Учебный курс / А.А. Бабаев СПб.: СПбГУ, 2007 – 156 с.
7. Бабич, Т.Н. Прогнозирование и планирование в условиях рынка: Учебное пособие / Т.Н. Бабич, И.А. Козьева, Ю.В. Вертакова, Э.Н. Кузьбожев. - М.: НИЦ ИНФРА-М, 2013. - 336 с.
8. Балена, Франческо Современная практика программирования на Microsoft Visual Basic и Visual C# / Франческо Балена , Джузеппе Димауро. - М.: Русская Редакция, 2015. - 640 с.
9. Барский, А.Б. Логические нейронные сети / А.Б. Барский. - М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2013. - 0 с.
10. Боровский, А. С++ и Pascal в Kylix 3. Разработка интернет-приложений и СУБД / А. Боровский. - М.: БХВ-Петербург, 2015. - 544 с.
11. Бубнова Н.Г. Информатика в экономике: учебное пособие: / Н. Г. Бубнова и др. – М: Вузовский учебник, 2011. – 476 с.

12. Бунаков, В. Е. Нейронная физика. Учебное пособие: моногр. / В.Е. Бунаков, Л.В. Краснов. - М.: Издательство Санкт-Петербургского университета, 2015. - 200 с.
13. Головинский, П. А. Математические модели. Теоретическая физика и анализ сложных систем. Книга 2. От нелинейных колебаний до искусственных нейронов и сложных систем / П.А. Головинский. - М.: Либроком, 2012. - 234 с.
14. Давыдов, В. Visual C++. Разработка Windows-приложений с помощью MFC и API-функций / В. Давыдов. - М.: БХВ-Петербург, 2014. - 576 с.
15. Девятков В.В. Системы искусственного интеллекта: Учеб. пособие/ В.В. Девятков; М-во образования РФ. - М.: МГТУ им. Н.Э. Баумана, 2011. - 352 с.
16. Довбуш, Галина Visual C++ на примерах / Галина Довбуш , Анатолий Хомоненко. - М.: БХВ-Петербург, 2012. - 528 с.
17. Егорова, Н.Е. Прогнозирование фондовых рынков с использованием экономико-математических моделей / Н.Е. Егорова, А.Р. Бахтизин, К.А. Торжевский. - М.: Красанд, 2013. - 216 с.
18. Зиборов, В. MS Visual C++ 2010 в среде .NET / В. Зиборов. - М.: Питер, 2012. - 320 с.
19. Иопа, Н.И. Информатика: (для технических специальностей): учебное пособие / Н. И. Иопа. – М.: КноРус, 2011. – 469 с.
20. Казначеева А.О. Основы информационных технологий: Учебное пособие / А.О. Казначеева – СПб.: СПбГУ ИТМО, 2009. - 44 с.
21. Кетков, Юлий Практика программирования: Visual Basic, C++ Builder, Delphi. Самоучитель (+ дискета) / Юлий Кетков , Александр Кетков. - М.: БХВ-Петербург, 2012. - 464 с.
22. Клещина, М.Г. Экономическое прогнозирование: Учебное пособие / М.Г. Клещина. - М.: ИД МИСиС, 2012. - 88 с.

23. Круглов, В.В. Искусственные нейронные сети. Теория и практика: моногр. / В.В. Круглов, В.В. Борисов. - М.: Горячая линия - Телеком; Издание 2-е, стер., 2002. - 382 с.
24. Мадера, А.Г. Риски и шансы: Неопределенность, прогнозирование и оценка / А.Г. Мадера. - М.: Красанд, 2014. - 448 с.
25. Мешков, А. Visual C++ и MFC / А. Мешков, Ю. Тихомиров. - М.: БХВ-Петербург, 2013. - 546 с.
26. Нейронные сети. Statistica Neural Networks. Методология и технологии современного анализа данных. - М.: Горячая линия - Телеком, 2008. - 392 с.
27. Неформальное введение в C++ и Turbo Vision. - Москва: ИЛ, 2010. - 384 с.
28. Павлов С.И. Системы искусственного интеллекта : Учебное пособие / С.И. Павлов – Томск : ТУСУР, 2011. – 415 с.
29. Павлова Е.А. Современные информационные технологии: Учебный курс / Е.А. Павлова – М.: МИФИ, 2007 – 116 с.
30. Панюкова, Т. А. Языки и методы программирования. Создание простых GUI-приложений с помощью Visual C++. Учебное пособие / Т.А. Панюкова, А.В. Панюков. - Москва: Мир, 2015. - 144 с.
31. Пахомов, Б. C/C++ и MS Visual C++ 2010 для начинающих / Б. Пахомов. - М.: БХВ-Петербург, 2011. - 736 с.
32. Пахомов, Борис C/C++ и MS Visual C++ 2012 для начинающих / Борис Пахомов. - Москва: СИНТЕГ, 2015. - 518 с.
33. Пахомов, Борис C/C++ и MS Visual C++ 2012 для начинающих / Борис Пахомов. - М.: "БХВ-Петербург", 2013. - 502 с.
34. Позин, Н. В. Моделирование нейронных структур / Н.В. Позин. - М.: Наука, 1970. - 264 с.
35. Полубенцева, М. C/C++. Процедурное программирование / М. Полубенцева. - М.: БХВ-Петербург, 2014. - 448 с.

36. Поляков, А. Методы и алгоритмы компьютерной графики в примерах на Visual C++ / А. Поляков, В. Брусенцев. - М.: БХВ-Петербург, 2011. - 560 с.
37. Понамарев, В. Программирование на C++/C# в Visual Studio .NET 2003 / В. Понамарев. - М.: БХВ-Петербург, 2015. - 917с.
38. Рассел, Джесси Вербализация нейронных сетей / Джесси Рассел. - М.: VSD, 2013. - 0 с.
39. Рассел, Джесси Искусственный нейрон / Джесси Рассел. - М.: VSD, 2013. - 0 с.
40. Роберт, С. Сикорд Безопасное программирование на С и С++ / Роберт С. Сикорд. - Москва: РГГУ, 2014. - 496 с.
41. Рыбина Г.В. Технология построения динамических интеллектуальных систем: Учебное пособие / Г.В. Рыбина, С.С. Паронджанов – М.: МИФИ, 2011 – 280 с.
42. Садовникова, Н.А. Анализ временных рядов и прогнозирование / Н.А. Садовникова, Р.А. Шмойлова. - М.: МФПУ Синергия, 2016. - 152 с.
43. Секунов, Н. Программирование на C++ в Linux / Н. Секунов. - М.: БХВ-Петербург, 2016. - 425 с.
44. Сидорина, Татьяна Самоучитель Microsoft Visual Studio C++ и MFC / Татьяна Сидорина. - М.: "БХВ-Петербург", 2014. - 848 с.
45. Толкачев, С. Нейронное программирование диалоговых систем / С. Толкачев. - Москва: РГГУ, 2016. - 192 с.
46. Трофимова В. В. Информационные системы и технологии в экономике и управлении / В. В. Трофимова – М.: Изд-во: Юрайт-Издат, 2009.
47. Чураков, Е.П. Прогнозирование экономических временных рядов / Е.П. Чураков. - М.: Финансы и статистика, 2008. - 208 с.
48. Юревич, Артур Нейронные сети в экономике / Артур Юревич. - М.: LAP Lambert Academic Publishing, 2014. - 80 с.

# ПРИЛОЖЕНИЕ 1

## Результаты тестирования

По доллару

Обучить сеть **Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

69.1755	69.2151	69.2	69.3026	6805156	67.6698	67.7691	66.7402
---------	---------	------	---------	---------	---------	---------	---------

Предсказать 66.2491 Очистить

Рис.1. Тестирование на выборке 2, окно «по доллару»

По доллару

Обучить сеть **Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

70.2244	69.1755	69.2151	69.2	69.3026	68.5156	67.6698	67.7691
---------	---------	---------	------	---------	---------	---------	---------

Предсказать 66.816 Очистить

Рис.2. Тестирование на выборке 3, окно «по доллару»

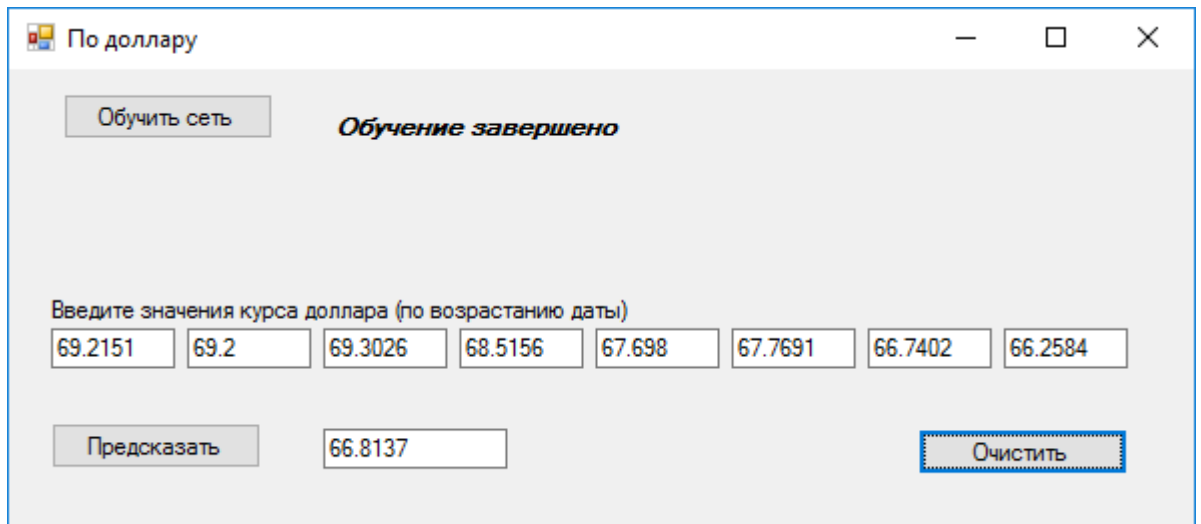


Рис.3. Тестирование на выборке 4, окно «по доллару»

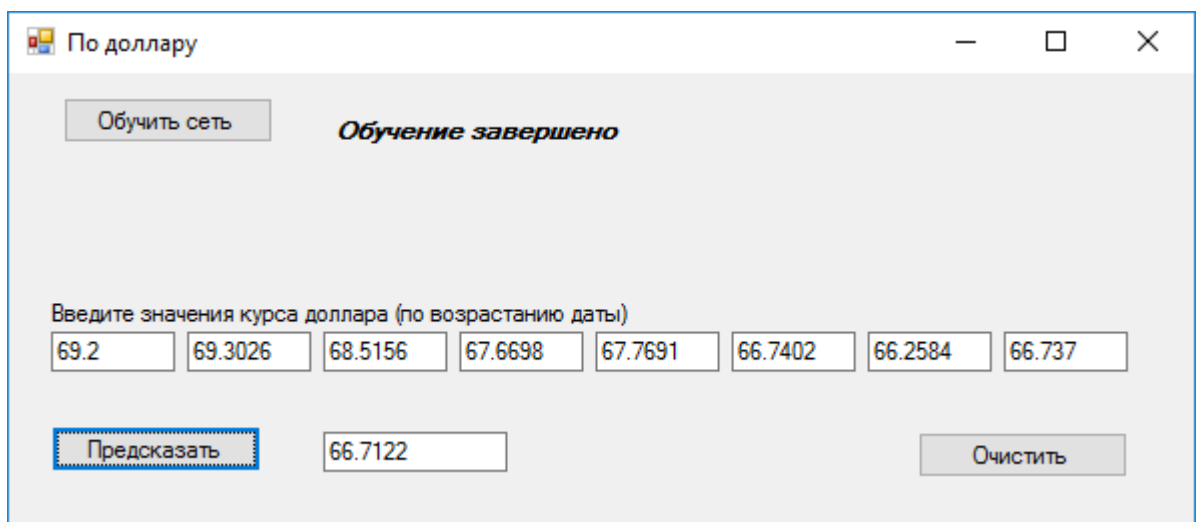


Рис.4. Тестирование на выборке 5, окно «по доллару»

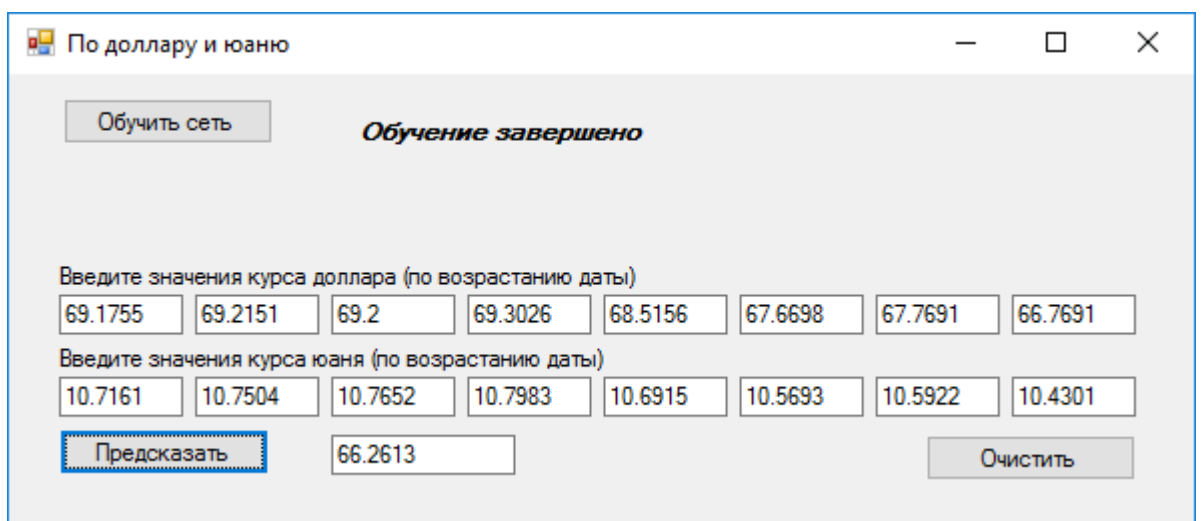


Рис.5. Тестирование на выборке 2, окно «по доллару и юаню»

По доллару и юаню

Обучить сеть **Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

70.2244	69.1755	69.2151	69.2	69.3026	68.5156	67.6698	67.7691
---------	---------	---------	------	---------	---------	---------	---------

Введите значения курса юаня (по возрастанию даты)

10.8717	10.7161	10.7504	10.7652	10.7983	10.6915	10.5693	10.5922
---------	---------	---------	---------	---------	---------	---------	---------

Предсказать 66.6349 Очистить

Рис.6. Тестирование на выборке 3, окно «по доллару и юаню»

По доллару и юаню

Обучить сеть **Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

69.2151	69.2	69.3026	68.5156	67.6698	67.7691	66.7402	66.2584
---------	------	---------	---------	---------	---------	---------	---------

Введите значения курса юаня (по возрастанию даты)

10.7504	70.7652	10.7983	10.6915	10.5693	10.5922	10.4301	10.355
---------	---------	---------	---------	---------	---------	---------	--------

Предсказать 66.8 Очистить

Рис.7. Тестирование на выборке 4, окно «по доллару и юаню»

По доллару и юаню

Обучить сеть **Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

69.2	69.3026	68.5156	67.6698	67.7691	66.7402	66.2584	66.737
------	---------	---------	---------	---------	---------	---------	--------

Введите значения курса юаня (по возрастанию даты)

10.7652	10.7983	10.6915	10.5693	10.5922	10.4301	10.355	10.4306
---------	---------	---------	---------	---------	---------	--------	---------

Предсказать 66.5208 Очистить

Рис.8. Тестирование на выборке 5, окно «по доллару и юаню».



По доллару и евро

Обучить сеть **Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

69.1755	69.2151	69.2	69.3026	68.5156	67.6698	67.7691	66.7402
---------	---------	------	---------	---------	---------	---------	---------

Введите значения курса евро (по возрастанию даты)

75.7472	76.0674	75.5595	75.3111	74.4079	73.6721	71.7404	70.8647
---------	---------	---------	---------	---------	---------	---------	---------

Предсказать 66.2874 Очистить

Рис.9. Тестирование на выборке 2, окно «по доллару и евро»

По доллару и евро

Обучить сеть **Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

70.2244	69.1755	69.2151	69.2	69.3026	68.5156	67.6698	67.7691
---------	---------	---------	------	---------	---------	---------	---------

Введите значения курса евро (по возрастанию даты)

76.9659	75.7472	76.0674	75.5595	75.3111	74.4079	73.6721	71.7404
---------	---------	---------	---------	---------	---------	---------	---------

Предсказать 66.67 Очистить

Рис.10. Тестирование на выборке 3, окно «по доллару и евро»

По доллару и евро

Обучить сеть **Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

69.2151	69.2	69.3026	68.5156	67.6698	67.7691	66.7402	66.2584
---------	------	---------	---------	---------	---------	---------	---------

Введите значения курса евро (по возрастанию даты)

76.0674	75.5595	75.3111	74.4079	73.6721	71.7404	70.8647	70.1345
---------	---------	---------	---------	---------	---------	---------	---------

Предсказать 66.6808 Очистить

Рис.11. Тестирование на выборке 4, окно «по доллару и евро»

По доллару и евро

Обучить сеть

**Обучение завершено**

Введите значения курса доллара (по возрастанию даты)

69.2	69.3026	68.5156	67.6698	67.7691	66.7402	66.2584	66.737
------	---------	---------	---------	---------	---------	---------	--------

Введите значения курса евро (по возрастанию даты)

75.5595	75.3111	74.4079	73.6721	71.7404	70.8647	70.1345	70.6478
---------	---------	---------	---------	---------	---------	---------	---------

Предсказать

66.4592

Очистить

Рис.12. Тестирование на выборке 5, окно «по доллару и евро»

## ПРИЛОЖЕНИЕ 2

### Листинг программы

```
Код файла MyForm1.cpp
#include "StdAfx.h"
#include "MyForm1.h"
#include "iostream"
#include "stdio.h"
#include "fann.h"
#include "floatfann.h"
#include "math.h"
using std::cout;
using std::endl;
namespace ChildToParent {
    System::Void Form1::button1_Click(System::Object^ sender,
System::EventArgs^ e) {
        int main()
        {
            int k, p;
            float a=0;
            const unsigned int num_input = 8;
            const unsigned int num_output = 1;
            const unsigned int num_layers = 4;
            const unsigned int num_neurons_hidden = k;
            const float desired_error = (const float) 0.0001;
            const unsigned int max_epochs = 2000000;
            const unsigned int epochs_between_reports = 1000;
            fann_type *calc_out;
            fann_type input[8];
            k = 5;
            p = 0;
            struct fann *ann = fann_create_standard(num_layers, num_input,
                num_neurons_hidden, num_output);
            fann_set_activation_function_hidden(ann,
FANN_SIGMOID_SYMMETRIC);
            fann_set_activation_function_output(ann,
FANN_SIGMOID_SYMMETRIC);
            while (p>1) {
                fann_train_on_file(ann, "training.txt", max_epochs,
                    epochs_between_reports, desired_error);
                input[0] = 0.492245;
                input[1] = 0.500806;
                input[2] = 0.512840;
```

```

        input[3] = 0.513080;
        input[4] = 0.512494;
        input[5] = 0.515073;
        input[6] = 0.523833;
        input[7] = 0.541526;
        calc_out = fann_run(ann, input);
        a = a+abs(0.533184 - calc_out[0]);
        input[0] = 0.098396;
        input[1] = 0.102889;
        input[2] = 0.100266;
        input[3] = 0.093861;
        input[4] = 0.089460;
        input[5] = 0.093539;
        input[6] = 0.098211;
        input[7] = 0.099287;
        calc_out = fann_run(ann, input);
        a = a+abs(0.099525 - calc_out[0]);
        input[0] = 0.906576;
        input[1] = 0.924422;
        input[2] = 1.000000;
        input[3] = 0.944736;
        input[4] = 0.893955;
        input[5] = 0.967940;
        input[6] = 0.915922;
        input[7] = 0.886102;
        calc_out = fann_run(ann, input);
        a = a+abs(0.845931 - calc_out[0]);
        a = a / 3.000000;
        if (a > 0.001) {
            fann_destroy(ann);
            k++;
            p++;
            if (p == 4) {
                l++;
                p = 1;
            }
        }
        else {
            p = 0;
        }
    }

    fann_save(ann, "set.net");
    fann_destroy(ann);
}
this->label1->Visible = true;
this->button2->Enabled = true;

```

```

    }
}
System::Void Form1::button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    input[0] = System::Convert::ToFloat(textBox1->Text);
    input[1] = System::Convert::ToFloat(textBox2->Text);
    input[2] = System::Convert::ToFloat(textBox3->Text);
    input[3] = System::Convert::ToFloat(textBox4->Text);
    input[4] = System::Convert::ToFloat(textBox5->Text);
    input[5] = System::Convert::ToFloat(textBox6->Text);
    input[6] = System::Convert::ToFloat(textBox7->Text);
    input[7] = System::Convert::ToFloat(textBox8->Text);
    calc_out = fann_run(ann, input);
    textBox8->Text = System::Convert::ToString(calc_out);
}
Код файла MyForm2.cpp
#include "StdAfx.h"
#include "MyForm2.h"
#include "iostream"
#include "stdio.h"
#include "fann.h"
#include "floatfann.h"
#include "math.h"
using std::cout;
using std::endl;
namespace ChildToParent {
    System::Void Form1::button1_Click(System::Object^ sender,
System::EventArgs^ e) {
        int main()
        {
            int k, p;
            float a=0;
            const unsigned int num_input = 8;
            const unsigned int num_output = 1;
            const unsigned int num_layers = 4;
            const unsigned int num_neurons_hidden = k;
            const float desired_error = (const float) 0.0001;
            const unsigned int max_epochs = 2000000;
            const unsigned int epochs_between_reports = 1000;
            fann_type *calc_out;
            fann_type input[8];
            k = 5;
            p = 0;
            struct fann *ann = fann_create_standard(num_layers, num_input,
                num_neurons_hidden, num_neurons_hidden1, num_output);
            fann_set_activation_function_hidden(ann,
FANN_SIGMOID_SYMMETRIC);

```

```

    fann_set_activation_function_output(ann,
FANN_SIGMOID_SYMMETRIC);
    while (p>1) {
        fann_train_on_file(ann, "training3.txt", max_epochs,
epochs_between_reports, desired_error);
        input[0] = 0.479756;
        input[1] = 0.488100;
        input[2] = 0.499829;
        input[3] = 0.500063;
        input[4] = 0.499491;
        input[5] = 0.502005;
        input[6] = 0.510543;
        input[7] = 0.527787;
        calc_out = fann_run(ann, input);
        a = a+abs(0.519657 - calc_out[0]);
        input[0] = 0.095899;
        input[1] = 0.100278;
        input[2] = 0.097722;
        input[3] = 0.091479;
        input[4] = 0.087190;
        input[5] = 0.091165;
        input[6] = 0.095719;
        input[7] = 0.096768;
        calc_out = fann_run(ann, input);
        a = a+abs(0,097- calc_out[0]);
        input[0] = 0.883576;
        input[1] = 0.900969;
        input[2] = 0.974629;
        input[3] = 0.920767;
        input[4] = 0.920767;
        input[5] = 0.943383;
        input[6] = 0.892684;
        input[7] = 0.863621;
        calc_out = fann_run(ann, input);
        a = a+abs(0.824469- calc_out[0]);
        a = a / 3.000000;
        if (a > 0.001) {
            fann_destroy(ann);
            k++;
            p++;
            if (p == 4) {
                l++;
                p = 1;
            }
        }
        else {
            p = 0;
        }
    }

```

```

    }
    fann_save(ann, "set2.net");
    fann_destroy(ann);
    }
    this->label1->Visible = true;
    this->button2->Enabled = true;
} }
System::Void Form1::button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    System::Void Form1::button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    float mas[8];
    float min1=35.4465, min2=6.2569, max1=93.2569, max2 = 15.3645
    int s=0, k=0,m=0,n=0, l=0;
    input[0] = System::Convert::ToFloat(textBox1->Text);
    input[1] = System::Convert::ToFloat(textBox2->Text);
    input[2] = System::Convert::ToFloat(textBox3->Text);
    input[3] = System::Convert::ToFloat(textBox4->Text);
    input[4] = System::Convert::ToFloat(textBox5->Text);
    input[5] = System::Convert::ToFloat(textBox6->Text);
    input[6] = System::Convert::ToFloat(textBox7->Text);
    input[7] = System::Convert::ToFloat(textBox8->Text);
    mas[0] = System::Convert::ToFloat(textBox9->Text);
    mas[1] = System::Convert::ToFloat(textBox10->Text);
    mas[2] = System::Convert::ToFloat(textBox11->Text);
    mas[3] = System::Convert::ToFloat(textBox12->Text);
    mas[4] = System::Convert::ToFloat(textBox13->Text);
    mas[5] = System::Convert::ToFloat(textBox14->Text);
    mas[6] = System::Convert::ToFloat(textBox15->Text);
    mas[7] = System::Convert::ToFloat(textBox16->Text);
    input[0]=(max1- input[0])/(max1/min1);
    input[1]=(max1- input[1])/(max1/min1);
    input[3]=(max1- input[2])/(max1/min1);
    input[3]=(max1- input[3])/(max1/min1);
    input[4]=(max1- input[4])/(max1/min1);
    input[5]=(max1- input[5])/(max1/min1);
    input[6]=(max1- input[6])/(max1/min1);
    input[7]=(max1- input[7])/(max1/min2);
    mas[0]=(max2- mas[0])/(max2/min2);
    mas[1]=(max2- mas[1])/(max2/min2);
    mas[3]=(max2- mas[2])/(max2/min2);
    mas[3]=(max2- mas[3])/(max2/min2);
    mas[4]=(max2- mas[4])/(max2/min2);
    mas[5]=(max2- mas[5])/(max2/min2);
    mas[6]=(max2- mas[6])/(max2/min2);
    mas[7]=(max2- mas[7])/(max2/min2);

    for (a=0; a<8; a++){

```

```

s=s+ input[a];
k=k+ mas[a];
}
s=s/8;
k=k/8;
for (a=0; a<8; a++){
m=m+ (input[a]-s)*(mas[a]-k);
n=n+ (input[a]-s)^2;
l=l+ (mas[a]-s)^2;
}
m=m/(n*1)^0.5;
input[0] = input[0]*m;
input[1] = input[1]*m;
input[2] = input[2]*m;
input[3] = input[3]*m;
input[4] = input[4]*m;
input[5] = input[5]*m;
input[6] = input[6]*m;
input[7] = input[7]*m;
calc_out = fann_run(ann, input);
textBox17->Text = System::Convert::ToString(calc_out);
}
}
}
}
}

```

Код файла MyForm3.cpp

```

#include "StdAfx.h"
#include "MyForm1.h"
#include "iostream"
#include "stdio.h"
#include "fann.h"
#include "floatfann.h"
#include "math.h"
using std::cout;
using std::endl;
namespace ChildToParent {
    System::Void Form1::button1_Click(System::Object^ sender,
System::EventArgs^ e) {
        int main()
        {
            int k, l, p;
            float a=0;
            const unsigned int num_input = 8;
            const unsigned int num_output = 1;
            const unsigned int num_layers = 4;
            const unsigned int num_neurons_hidden = k;
            const unsigned int num_neurons_hidden1 = l;
            const float desired_error = (const float) 0.0001;

```



```

const unsigned int max_epochs = 2000000;
const unsigned int epochs_between_reports = 1000;
fann_type *calc_out;
fann_type input[8];
k = 5;
l = 1;
p = 0;
struct fann *ann = fann_create_standard(num_layers, num_input,
    num_neurons_hidden, num_neurons_hidden1, num_output);
fann_set_activation_function_hidden(ann,
FANN_SIGMOID_SYMMETRIC);
fann_set_activation_function_output(ann,
FANN_SIGMOID_SYMMETRIC);
while (p>1) {
    fann_train_on_file(ann, "training3.txt", max_epochs,
    epochs_between_reports, desired_error);
    input[0] = 0.492245;
    input[1] = 0.500806;
    input[2] = 0.512840;
    input[3] = 0.513080;
    input[4] = 0.512494;
    input[5] = 0.515073;
    input[6] = 0.523833;
    input[7] = 0.541526;
    calc_out = fann_run(ann, input);
    a = a+abs(0.533184 - calc_out[0]);
    input[0] = 0.098396;
    input[1] = 0.102889;
    input[2] = 0.100266;
    input[3] = 0.093861;
    input[4] = 0.089460;
    input[5] = 0.093539;
    input[6] = 0.098211;
    input[7] = 0.099287;
    calc_out = fann_run(ann, input);
    a = a+abs(0.099525 - calc_out[0]);
    input[0] = 0.906576;
    input[1] = 0.924422;
    input[2] = 1.000000;
    input[3] = 0.944736;
    input[4] = 0.893955;
    input[5] = 0.967940;
    input[6] = 0.915922;
    input[7] = 0.886102;
    calc_out = fann_run(ann, input);
    a = a+abs(0.845931 - calc_out[0]);
    a = a / 3.000000;
    if (a > 0.001) {

```

```

        fann_destroy(ann);
        k++;
        p++;
        if (p == 4) {
            l++;
            p = 1;
        }
    }
    else {
        p = 0;
    }
}

fann_save(ann, "set3.net");
fann_destroy(ann);
}
this->label1->Visible = true;
this->button2->Enabled = true;
} }
System::Void Form1::button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    float mas[8];
    float min1=35.4465, min2=6.2569, max1=93.2569, max2 = 15.3645
    int s=0, k=0,m=0,n=0, l=0;
    input[0] = System::Convert::ToFloat(textBox1->Text);
    input[1] = System::Convert::ToFloat(textBox2->Text);
    input[2] = System::Convert::ToFloat(textBox3->Text);
    input[3] = System::Convert::ToFloat(textBox4->Text);
    input[4] = System::Convert::ToFloat(textBox5->Text);
    input[5] = System::Convert::ToFloat(textBox6->Text);
    input[6] = System::Convert::ToFloat(textBox7->Text);
    input[7] = System::Convert::ToFloat(textBox8->Text);
    mas[0] = System::Convert::ToFloat(textBox9->Text);
    mas[1] = System::Convert::ToFloat(textBox10->Text);
    mas[2] = System::Convert::ToFloat(textBox11->Text);
    mas[3] = System::Convert::ToFloat(textBox12->Text);
    mas[4] = System::Convert::ToFloat(textBox13->Text);
    mas[5] = System::Convert::ToFloat(textBox14->Text);
    mas[6] = System::Convert::ToFloat(textBox15->Text);
    mas[7] = System::Convert::ToFloat(textBox16->Text);
    input[0]=(max1- input[0])/(max1/min1);
    input[1]=(max1- input[1])/(max1/min1);
    input[3]=(max1- input[2])/(max1/min1);
    input[3]=(max1- input[3])/(max1/min1);
    input[4]=(max1- input[4])/(max1/min1);
    input[5]=(max1- input[5])/(max1/min1);
    input[6]=(max1- input[6])/(max1/min1);
    input[7]=(max1- input[7])/(max1/min2);

```

```

mas[0]=(max2- mas[0])/(max2/min2);
mas[1]=(max2- mas[1])/(max2/min2);
mas[2]=(max2- mas[2])/(max2/min2);
mas[3]=(max2- mas[3])/(max2/min2);
mas[4]=(max2- mas[4])/(max2/min2);
mas[5]=(max2- mas[5])/(max2/min2);
mas[6]=(max2- mas[6])/(max2/min2);
mas[7]=(max2- mas[7])/(max2/min2);

for (a=0; a<8; a++){
s=s+ input[a];
k=k+ mas[a];
}
s=s/8;
k=k/8;
for (a=0; a<8; a++){
m=m+ (input[a]-s)*(mas[a]-k);
n=n+ (input[a]-s)^2;
l=l+ (mas[a]-s)^2;
}
m=m/(n*l)^0.5;
input[0] = input[0]*m;
input[1] = input[1]*m;
input[2] = input[2]*m;
input[3] = input[3]*m;
input[4] = input[4]*m;
input[5] = input[5]*m;
input[6] = input[6]*m;
input[7] = input[7]*m;

calc_out = fann_run(ann, input);
textBox17->Text = System::Convert::ToString(calc_out);
}
}

```