

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Факультет математики, механики и компьютерных технологий
Кафедра прикладной математики и программирования
Направление подготовки Прикладная математика и информатика

РАБОТА ПРОВЕРЕНА

Рецензент, _____

« ____ » _____ 20 ____ г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой

_____/А.А.Замышляева

« ____ » _____ 2017 г.

ИССЛЕДОВАНИЕ МОДЕЛИ НЕЙРОННОЙ СЕТИ ДЛЯ
ПРОГНОЗИРОВАНИЯ ЕСТЕСТВЕННОГО ПРИРОСТА НАСЕЛЕНИЯ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУрГУ–01.03.02.2017.23.ПЗ ВКР

Руководитель работы, доцент

_____/ С.У.Турлакова

« ____ » _____ 2017 г.

Автор работы

Студентка группы ЕТ-482

_____/ А.А.Голубева

« ____ » _____ 2017 г.

Нормоконтролер, доцент

_____/Т.Ю. Оленчикова

« ____ » _____ 2017 г.

АННОТАЦИЯ

Голубева А.А. Исследование модели нейронной сети для прогнозирования естественного прироста населения .– Челябинск: ЮУрГУ, ЕТ-482, 41 с., 21 ил., 6 табл., библиогр. список – 16 наим., 1 прил.

В данной работе ставится задача получения прогноза временного ряда, представленного в виде статистических данных о естественном приросте населения, с помощью модели многослойного персептрона прямого распространения (Feed-Forward MLP). Анализ показал, что наилучший прогноз получается при 72 нейронах в скрытом слое в случае обучения алгоритмом Левенберга-Марквардта. Было выявлено, что нейронная сеть, обученная алгоритмом Левенберга-Марквардта, оказывается не трудоемкой и показывает небольшую ошибку прогноза. Таким образом, был получен прогноз естественного прироста населения на 2017 г. Ошибка прогноза составила 2,53%, что свидетельствует об эффективности применения нейронной сети, а именно модели многослойного персептрона прямого распространения для прогнозирования временных рядов.

Также программа была протестирована еще на нескольких примерах и по всем методам показала среднюю ошибку 0,0244.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	6
1 ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ.....	7
1.1 Основные направления в прогнозировании временных рядов с помощью нейронной сети.....	7
1.2 Понятие временного ряда и способы его прогнозирования.....	9
1.3 Понятие нейронной сети и ее особенности.....	10
1.4 Использование нейронной сети для получения прогноза.....	14
1.5 Современные применения искусственной нейронной сети.....	15
1.6 Выводы по разделу.....	16
2 МОДЕЛЬ И ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ.....	18
2.1 Многослойный персептрон прямого распространения.....	18
2.2 Алгоритм Левенберга-Марквардта.....	20
2.3 Выводы по разделу.....	25
3 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ (ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТА).....	26
3.1 Получение прогнозов.....	26
3.2 Полученные результаты.....	28
3.2.1 Пример №1.....	28
3.2.2 Пример №2.....	31
3.2.3 Пример №3.....	35
3.3 Выводы по разделу.....	38
ЗАКЛЮЧЕНИЕ.....	39
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	41
ПРИЛОЖЕНИЕ 1 ТЕКСТ ПРОГРАММЫ.....	42

ВВЕДЕНИЕ

В основе любого социального прогноза и планирования всегда лежат демографическое прогнозирование. Что бы люди ни планировали на будущее: развитие производства товаров, любые социальные процессы — всегда необходимо узнать количество и состав их участников, так как это оказывает сильное влияние на многие социальные процессы.

Поэтому в современном мире возрастает роль моделирования и прогнозирования динамической информации, представленной временными рядами социально-экономических явлений и процессов.

Исследование естественного прироста населения может послужить изменением курса демографической политики в ее субъектах, разработок региональных планов по улучшению демографической ситуации, а также в планировании социально-экономических процессов.

Целью данной работы является построение прогноза и оценка эффективности прогнозирования временного ряда посредством модели нейронной сети многослойного персептрона прямого распространения.

Выделим следующие задачи исследования.

1. Изучить литературу и основные понятия, которые потребуются для прогнозирования с помощью нейронных сетей;
2. Выбрать и исследовать модель нейронной сети;
3. Написать программу на языке Matlab для прогнозирования временного ряда и протестировать ее;
4. Проанализировать полученные результаты, сделать вывод о том, насколько метод нейронных сетей подходит для прогнозирования временных рядов.
5. Оформить необходимую документацию.

В первом разделе работы проводится обзор литературы, описываются понятия временного ряда и нейронной сети; раскрывается содержание метода нейронной сети, подробно описывается алгоритм, выбранный для прогнозирования, а также приводятся примеры современного применения искусственной нейронной сети.

Во втором разделе описывается выбранная модель нейронной сети (Многослойный персептрон прямого распространения) и алгоритм обучения (Левенберга-Марквардта).

В третьем разделе дается описание процедуры построения прогноза для временных рядов на примере статистических данных о естественном приросте населения РФ в программе Matlab с помощью метода нейронных сетей. Также приводятся результаты тестирования программы еще для нескольких временных рядов.

Для нахождения параметров, проведения тестов и прогноза использовались пакет Microsoft Excel и приложение Matlab.

1 ПРОГНОЗИРОВАНИЕ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ НЕЙРОННОЙ СЕТИ

1.1 Основные направления в прогнозировании временных рядов с помощью нейронной сети

Прогнозирование является актуальной задачей в наше время. Это объясняет большое разнообразие исследований на данную тему.

Основные направления в прогнозировании временных рядов, для которых применяют нейронные сети, связаны с предметной областью исследования.

1) Природные или естественные прогнозы. В данной теме существуют такие исследования, как предсказание прогноза погоды, наводнений, землетрясений, осадков, количества радиоактивности и т.д. Например, в статье [2] предсказывают погоду. В качестве нейронной сети был выбран многослойный персептрон с одним скрытым слоем и одним нейроном выходного слоя. Было установлено, что комбинация разных функций активаций в скрытом и выходном слое способствует лучшим результатам обучения. В качестве входных данных был использован архив метеорологических наблюдений национального архива США. Входная выборка представляла собой среднесуточные значения температуры.

В качестве алгоритма корректировки весов был использован метод обратного распространения ошибки, а также его модификация, использующая моменты – инерционную составляющую, позволяющую уменьшить вероятность попадания в локальный минимум.

Был в итоге представлен результат прогнозирования нейронной сетью температуры за период 2002/03 гг. Сеть обучалась на данных о погоде за период 2000/02 гг. Погрешность обучения составила 3 %, погрешность тестирования – 5 %. Разница между итоговыми полученными и расчетными значениями очень мала, что показывает высокую эффективность применения нейронной сети для решения данной задачи.

2) Социальные прогнозы. Чаще всего это может быть оценка группового поведения. Примером может служить исследование о прогнозировании посещаемости футбольного матча. В работе [3] была проанализирована скорость прогнозирования различных моделей нейронных сетей (трехслойный персептрон, трехслойная сеть прямого распространения с задержкой времени, двухслойная нейронная сеть с радиально базисной функцией, рекуррентная нейронная сеть,). Средние ошибки полученных прогнозов варьируются от 4% и до 11% в зависимости от метода. Можно сделать вывод, что данная нейронная сеть может неплохо моделировать динамику посещаемости.

3) Медицинские прогнозы. К этому направлению можно отнести источники, в которых рассматривается прогнозирование количества пациентов, предсказание заболеваемости, смертности, рождаемости. Например, в [4] был произведен прогноз гипополидемического эффекта у больных ишемической болезнью сердца. Для прогнозирования применяли оригинальную разработку – многослойный персептрон с прямыми связями между нейронами и алгоритмом обратного

распространения ошибки с введением в сеть коэффициента крутизны дискриминантной функции модели, позволяющего менять скорость обучения сети.

Разработанные нейросетевые модели прогнозирования гипополидемического эффекта у больных ИБС с ГЛП обладают точностью, при которой чувствительность и специфичность прогноза составляет $\geq 90\%$. Информация о прогнозе гипополидемического эффекта поможет оптимизировать фармакотерапию, исключая неадекватное лечение.

4) Экономические прогнозы. К этому типу относятся прогнозы фондовых рынков, стоимости товаров и услуг и т.д. Одной из главных особенностей таких прогнозов является высокая изменчивость их временных рядов. Примером являются исследования о прогнозировании цен на электроэнергию [5]. Актуальность этой темы заключается в том, что такие прогнозы играют большую роль при ценообразовании на электроэнергию и становятся все более важными для ее производителей, а цены на электроэнергию являются главным показателем состояния рынка электроэнергии. В работе [5] для прогнозирования была выбрана искусственная нейронная сеть (ИНС), которую обучали с помощью алгоритма обратного распространения.

Модели прогнозирования нагрузки рассматривались применительно к Пензенской энергосистеме. Для получения данных о нагрузке использовались телеметрические измерения (ТИ) о перетоках мощности по линиям электропередачи, уровнях напряжения в узлах.

Результаты проведенных исследований показывают, что при хороших входных данных расхождение прогноза по моделям ИНС с реальным потреблением электрической энергии составляет 4-7%. В случае плохих входных данных, когда график нагрузки не характерен для этого дня, расхождение прогноза по моделям ИНС с реальным потреблением составляет 6-9%. В качестве платформы для создания ИНС использовался программный продукт Neural Network Toolbox из пакета MATLAB. Полученная нейронная сеть дает достаточно точный прогноз при условии достоверности входных данных.

5) Физические или механические прогнозы. В данном типе могут быть рассмотрены эффективность использования ветровой фермы, скорости запуска ракеты, предсказание работы газотурбинного двигателя, влияние состава на прочность бетона, расход топлива транспортного средства. Например, в [6] прогнозируется скорость и разлет осколков камней при взрывных работах, так как это необходимо для обеспечения безопасности рабочих и оборудования, т.е для определения пределов безопасности периметра. Начальные данные – это мониторинг взрывных работ в медных рудниках и известняковых карьеров в Сербии за 2011-2013 гг. Для прогноза был выбран двухслойный персептрон с 14 нейронами в скрытых уровнях. Результаты показали, что созданная нейронная сеть дает прогноз скорости с высокой точностью. (Из предсказанных значений выходили за допустимые пределы скорости 6%, то есть не попадали в доверительный интервал). Но к сожалению она не является универсальной: ее

прогноз можно считать верным только для рядов с похожими свойствами. Иначе, для получения точного прогноза сеть необходимо будет дообучать данными.

б) Оптимизационные прогнозы. Термин «оптимизационные» используют, так как по полученным данным в прогнозе можно улучшить результат. Это может быть прогноз транспортных потоков, оценка пропускной способности железных дорог. Например, в [7] находится краткосрочный прогноз транспортного потока с помощью двоичной нейронной сети с помощью метода AURA k-NN, с помощью объединения данных с нескольких датчиков движения и прогнозирования временных рядов. Главная задача – точный и быстрый прогноз транспортного потока. В этой работе немного расширено применение метода AURA k-NN, раньше он использовался только для прогнозирования невременных рядов. Также этот метод применялся только для поиска по шаблону, поэтому авторам пришлось обходить и эту проблему, путем получения многомерных прогнозов. В итоге модифицированный метод AURA k-NN оказался самым эффективным для получения прогноза с точки зрения скорости и точности.

Стоит подчеркнуть, что существует множество работ, цель которых направлена на развитие, модификацию и улучшение самих способов прогнозирования и предсказание конкретных явлений – это не единственное направление развития теории прогнозирования. Перефразируя сказанное, это главным образом изменение уже существующих нейронных сетей, или же создание новых, что реже. Один из примеров модификации существующей сети указан в [7] и уже был описан выше.

1.2 Понятие временного ряда и способы его прогнозирования

Если данные характеризуют какой-то объект за ряд последовательных значений, характеризующих изменение показателя во времени, то эти данные временной ряд.

Временной ряд – это совокупность значений какого-либо показателя за несколько последовательных моментов (периодов) времени. Особенности уровней из которых состоят временные ряды формируются под воздействием большого числа факторов. Эти факторы разбиваются на группы:

- 1) факторы, формирующие тенденцию ряда;
- 2) факторы, формирующие циклические колебания ряда;
- 3) случайные факторы.

Зависимость уровней ряда от времени может быть разная при разных сочетаниях этих факторов:

1) Если некоторые данные рассматривать не в совокупности с другими, они могут показывать другую тенденцию и оказывать разное воздействие на исследуемый показатель.

2) Циклические колебания могут быть характерны для изучаемого показателя. Например, для экономических временных рядов такие колебания носят сезонный характер и их можно корректно распознать только в большом количестве данных за длительное время.

3) У временного ряда может не быть тенденции или циклов, а следующий уровень быть равным сумме среднего уровня ряда и случайной составляющей.

Это теория, а в реальности данные редко можно полностью причислить к одной из вышеперечисленных моделей. Чаще, они содержат в себе сразу все три компонента. Если временной ряд представлен в виде суммы всех перечисленных компонент, то его модель называется аддитивной, а если в виде произведения, то мультипликативной.

Главной задачей исследования временного ряда является придание числового выражения компонентам. Это будет необходимо для прогнозирования будущих значений временного ряда и для других исследований над ним. [8]

Целью прогнозирования временного ряда является по исходным данным предсказать будущие значения характеристик данного объекта. Существуют несколько методов прогнозирования временных рядов. Все их можно поделить на две большие группы: локальные и глобальные по области определения параметров аппроксимирующей функции.

Вот некоторые модели для прогнозирования временного ряда [9]:

- регрессионные модели;
- авторегрессионные модели;
- модели экспоненциального сглаживания;
- нейросетевые модели;
- модели на основе цепей Маркова;
- модели на базе классификационно-регрессионных деревьев (CART);
- метод опорных векторов (SVM);
- генетический алгоритм (GA);
- дискретное преобразование Фурье.

В данной работе используются модели прогнозирования с помощью нейронных сетей.

1.3 Понятие нейронной сети и ее особенности

Причина начала исследования искусственной нейронной сети связаны с тем, что работа человеческого мозга кардинально отличается от работы цифрового компьютера. Мозг – это очень сложный, нелинейный, параллельный компьютер. Он организует свои структурные компоненты, называемые нейронами, так, что они выполняют конкретные задачи (такие как распознавание образов, восприятие, контроль движения) во много раз быстрее, чем самые мощные компьютеры. Мозг имеет структуру, позволяющую строить собственные правила на основе опыта. Опыт накапливается с течением времени.

Пластичность нервной системы позволяет адаптировать ее к окружающей среде. Аналогично в нейронных сетях производится настройка искусственных нейронов и формируется структура нейронной сети. Таким образом, нейронная сеть представляет собой машину, которая моделирует способ обработки мозгом конкретной задачи. Обычно эта сеть реализуется с помощью

электронных компонентов или моделируется программой. В данной работе для создания нейронной сети будет использоваться программа Matlab.

Для достижения высокой производительности, нейронные сети используют массив взаимосвязанных элементов, называемых нейронами. Таким образом, возможно следующее определение нейронной сети, выступающей в роли адаптивной машины: нейронная сеть – это массивный распределенный параллельный процессор, состоящий из элементарных обрабатывающих блоков со способностью хранения данных и обработки их для последующего использования.

У нейронной сети есть общие черты с мозгом:

1) в нее поступают знания из окружающей среды и используются в процессе обучения;

2) для хранения знаний применяется синаптический вес – связь между нейронами.

Процедура, реализующая настройку синаптических весов для получения желаемых результатов, называется алгоритмом обучения.

Опишем некоторые преимущества нейронной сети.

1) Нелинейность. Так как искусственные нейроны могут быть линейными и нелинейными, то нейронные сети могут воспроизводить сложные зависимости, как линейные, так и нелинейные.

2) Параллельная обработка информации. Благодаря этой способности достигается значительное ускорение процесса обработки информации. Часто становится возможной обработка сигналов даже в реальном времени.

3) Обучение с учителем. Этот популярный метод еще называется контролируемым обучением, так как он предполагает изменение синаптических весов за счёт использования набора обучающих примеров. Каждый пример является соответствием входных данных и ожидаемых выходных данных, за счет этого сеть обучается. Начально нейронная сеть инициализируется случайными синаптическими весами, которые потом в процессе обучения модифицируются так, чтобы минимизировать разницу между ожидаемым ответом и реальным выходным сигналом, формируемым нейронной сетью.

4) Адаптивность. Нейронная сеть обладает способностью приспосабливать свои синаптические веса к изменениям в окружающей среде. Также, обученную под определенную среду нейронную сеть можно легко переобучить к немного измененным условиям среды.

5) Корректность. Выход нейронной сети всегда можно проверить на корректность, подав на вход определенный пример.

6) Устойчивость к ошибкам. Реализованная аппаратно нейронная сеть по своей природе устойчива к ошибкам. Функции поврежденных соединений принимают на себя другие элементы, в результате в сети не наблюдаются заметные нарушения, но ее производительность медленно снижается при неблагоприятных условиях.

7) Единообразие анализа и проектирования. Нейронные сети – это универсальный механизм обработки информации. Одно проектное решение нейронной сети может быть использовано в разных предметных областях. Это свойство проявляется так как:

- нейроны являются стандартными составными частями любой нейронной сети;
- можно использовать одни и те же алгоритмы обучения в различных нейросетевых приложениях;
- на основе интеграции целых модулей могут быть построены модульные сети.

Из-за перечисленных свойств за последнее время вырос интерес к нейронным сетям и прогресс в их исследовании. Искусственные нейронные сети используются для аппроксимации функций, классификации и распознавания образов, прогнозирования, идентификации, оценивания и ассоциативного управления. [10]

Основными элементами нейронной сети являются нейроны.

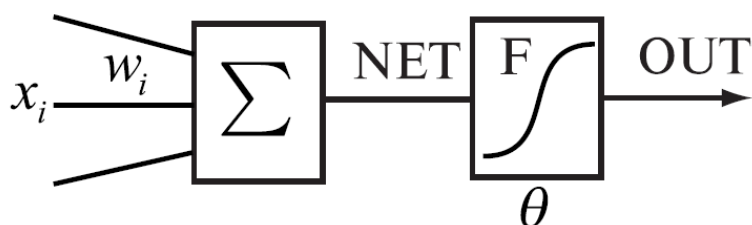


Рисунок 1.1 – Нейрон

Нейрон состоит из взвешенного сумматора и нелинейного элемента. Функционирование нейрона определяется формулами:

$$\text{NET} = \sum_i w_i x_i \quad (1)$$

$$\text{OUT} = F(\text{NET} - \theta) \quad (2)$$

где x_i — входные сигналы, совокупность всех входных сигналов нейрона образует вектор x ;

w_i — весовые коэффициенты, совокупность весовых коэффициентов образует вектор весов w ;

NET — взвешенная сумма входных сигналов, значение NET передается на нелинейный элемент;

θ — пороговый уровень данного нейрона;

F — нелинейная функция, называемая функцией активации.

Нейрон имеет несколько входных сигналов x и один выходной сигнал OUT. Параметрами нейрона, определяющими его работу, являются: вектор весов w , пороговый уровень θ и вид функции активации F.

Выбор функции активации зависит от специфики задачи, удобства реализации на ЭВМ, алгоритма обучения. В данной работе используются линейная и сигмоидальная функции.

1) Линейная функция представлена на рисунке 2 и определяется формулой (3).

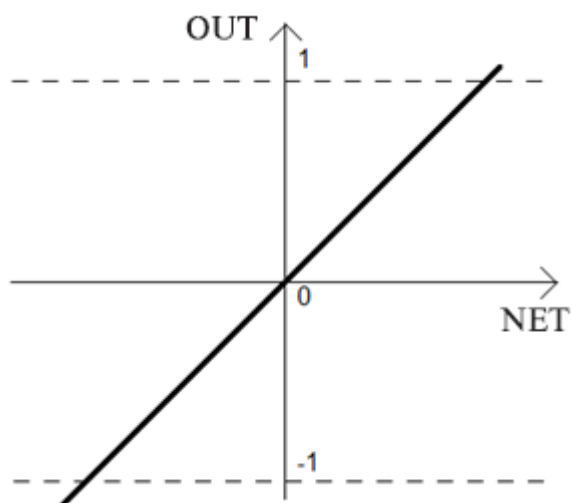


Рисунок 1.2 – Линейная активационная функция

$$OUT = K NET, K = \text{const.}$$

(3)

2) Сигмоидальная функция представлена на рисунке 3 и определяется формулой (4).

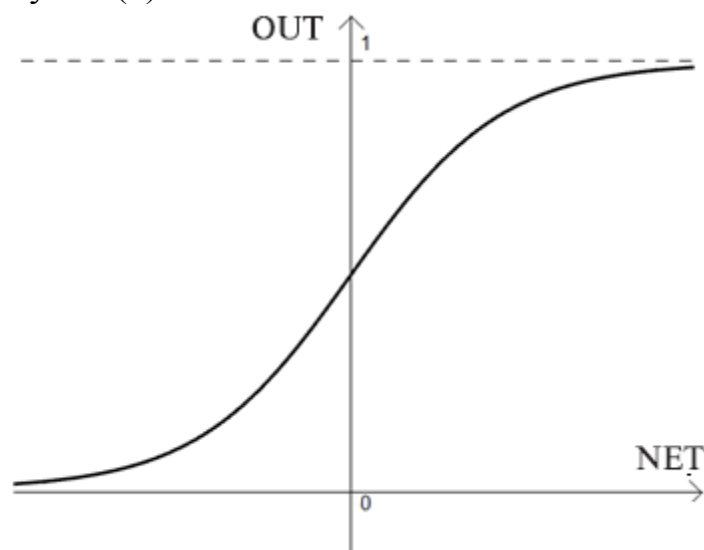


Рисунок 1.3 – Сигмоидальная активационная функция

$$OUT = \frac{1}{1 + e^{-NET}}$$

(4)

Все активационные функции должны обладать двумя важными свойствами.

1) Нелинейность. В случае линейности активационной функции последовательное распространение сигнала по сети означает лишь его масштабирование. Благодаря нелинейным активационным функциям можно получать все более сложную функциональную зависимость по мере прохождения сигнала через каждый нейрон.

2) Ограниченная область допустимых значений при неограниченной области определения. Величина выходного сигнала всегда остается в заданных пределах, не зависимо от того, какое количество пресинаптических нейронов суммируется по

мере прохождения сигнала по сети. В связи с этим нет ограничения на количество связей между нейронами – оно может быть сколь угодно большим. [11]

1.4 Использование нейронной сети для получения прогноза

Пусть на интервале $[0; T_0]$ задана функция $f(x)$, где T_0 – текущее значение времени. Задача состоит в том, чтобы предсказать значение функции при $t > T_0$. Чтобы было возможным применить нейронную сеть, необходимо предварительно обработать входные данные. А именно, необходимо дискретизировать время. Для этого интервал разбивается на $n-1$ временных отрезков. Таким образом, считается известным значение функции в конкретный момент времени t_k :

$$x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix},$$

где $x_k = f(t_k)$, $k = 0, n$;

$$t_0 = 0;$$

$$t_n = T_0;$$

$$t_k = t_{k-1} + 1, k = 1, n.$$

Задача состоит в определении значения функции в момент времени, равный $(T_0 + a_0)$, где $a_0 > 0$. Иными словами, a_0 – интервал прогнозирования.

Таким образом, необходимо решить уравнение:

$$f(T_0 + a_0) = y,$$

где y – решение поставленной задачи.

Пусть вектор X – входной вектор, а Y – выходной. Для того чтобы спрогнозировать некоторую функцию в виде дискретного процесса во времени и предсказать несколько значений функции в некоторых точках, необходимо выбрать все интервалы a_i одинаковыми и постоянными (в нашем случае все $a_i = 1$). Далее необходимо построить и обучить сеть. На вход подается вектор X со значениями функций в известных точках и вычисляется выход сети, т.е. $f(T_0 + a_0)$. Затем «сдвигаются» (т.е. изменяются) компоненты входных и выходных векторов по следующим формулам:

$$\begin{cases} x_0 = x_1 \\ x_1 = x_2 \\ \vdots \\ x_{n-1} = x_n \\ x_n = y \end{cases}$$

Теперь выходное значение стало известным, и можно спрогнозировать следующее значение $f(T_0 + 2a_0)$. Повторяя эту операцию, можно спрогнозировать

функцию в бесконечном количестве точек с дискретным шагом по времени, равным a_0 .

Для многомерных функций задача решается аналогично. [11]

1.5 Современные применения искусственной нейронной сети

За последние годы искусственные нейронные сети проникли во многие сферы жизни общества: от распознавания рукописного текста и голоса до классификации различных видов рака.

Примеры, современного применения искусственных нейронных сетей [12]

1) Для анализа изображений, Google в качестве обучающего набора данных использовал информацию с видеохостинга YouTube. В случае решения задачи распознавания речи, в качестве обучающей выборки использовалась серия аудиоклипов с приложенными к ним описаниями. Первая представленная версия распознавания речи на основе нейронной сети содержала уровень ошибок, достигающий 25 %, через три года результат был улучшен и составлял уже 8 % ошибок.

2) Алгоритм распознавания лиц, предложенный Facebook, работает с точностью в 97 %.

3) Цифровой персональный ассистент Cortana от компании Microsoft достиг такой высокой точности, что может различить две невероятно похожих породы собак.

4) На основе аппарата нейронных сетей проводятся исследования задач прогнозирования динамики цен на фондовом рынке. Точность данных исследований достигает 60–75 % верно угаданных значений цены, а количество верно угаданных тенденций рынка — 90 %.

5) Еще одним применением искусственных нейронных сетей являются задачи в области геологоразведки. К примеру, компанией Amoco используется ИНС для выделения характерных пиков в показаниях сейсмических датчиков. Надежность распознавания пиков — 95 % по каждой сейсмолинии. По сравнению с ручной обработкой скорость анализа данных увеличилась в 8 раз.

6) В некоторых областях искусственный интеллект достиг таких вершин, что стал превосходить профессионалов данной области. Так, например, программа AlphaGo, использующая нейронные сети, обыграла лучшего игрока Го за последнее десятилетие со счётом 4–1. Выигрышную стратегию в го просто невозможно запрограммировать традиционными методами. Количество вероятных комбинаций уже после первого хода здесь равно 129 960 (в шахматах, для сравнения, — четыреста), а за весь матч их набирается больше, чем атомов во Вселенной. Алгоритм просчета ходов не в состоянии продумать все возможные

комбинации и проигрывает профессиональным игрокам, которые полагаются на интуицию, выработанную годами тренировок.

7) Также, нейронные сети используются для решения жизненно важных задач в медицине. Исследователи из университета Дьюка обучили нейронную сеть распознавать маммограммы злокачественной ткани на основе восьми особенностей, с которыми обычно имеют дело радиологи. Данная сеть может решить задачу с почти 100 % чувствительностью, а после лечения рака молочной железы так же помогает эффективно предсказывать возникновение рецидива.

8) Сейчас можно говорить, что нейронные сети уже являются инструментом кардиодиагностики в Англии для предупреждения инфаркта миокарда. Данный пакет был разработан фирмой RESInformatica. Сеть продемонстрировала достаточно высокую точность при обнаружении инфаркта миокарда — 92 % и лишь в 4 % случаев дала ложный сигнал, подтверждающий диагноз. Высокие результаты работы нейронных сетей в области медицины исходят из того, что качество работы экспертной системы всегда стабильно и не зависит от внешних человеческих факторов.

9) В 2012 году произошло событие, которое коренным образом изменило отношение к нейросетям. Сеть SuperVision, разработанная в Торонтском университете, с большим отрывом выиграла конкурс распознавания объектов на изображениях ImageNet LSVRP (Large-Scale Visual Recognition Challenge). Число ее ошибок составило 16,4%, тогда как программа, занявшая второе место, ошибалась в 26% случаев. Для сравнения: человек делает ошибки в 5% случаев. Но удивительнее всего было то, что для предварительного обучения сети использовался не кластер, а обычный компьютер с двумя видеокартами NVIDIA. Тренировка заняла около недели. Это был первый случай, когда нейросеть превзошла классические алгоритмы машинного зрения в очень сложном и специфичном тесте. База изображений включала в себя не только простые объекты вроде автомобилей, автобусов, столов и стульев, но еще и триста пород собак, из которых два десятка — разновидности терьеров.

10) Исследователи из университета Radboud в Голландии, используя сотни тысяч фотографий, [обучили нейронную сеть](#) использовать ручные наброски и превращать их в фотореалистичные изображения. Компьютеры уже давно могут делать из снимка скетчи, но обратный процесс ранее был невозможен. Команда Radboud начала собирать изображения в интернете и накопила 200 тыс. фотографий лиц и преобразовала их в наброски. Использовались эскизы в оттенках серого и цветные эскизы. Затем нашлось обучение того, как превратить эскиз в реалистичный рисунок. Далее прошло тестирование восстановления эскизов, нарисованных от руки. Но опять же, система выполнила свою работу превосходно. Она даже смогла угадать цвет волос и этническую принадлежность на основе черно-белых эскизов. Были созданы точные портреты без какой-либо информации о цвете. Непосредственное применение этой технологии очевидно: превращение эскизов полиции в фотореалистичные изображения подозреваемого. Эта система в будущем может использоваться и в графических редакторах.

Все эти примеры доказывают, что искусственные нейронные сети развиваются с бешеным темпом, а также они стали очень важны в современном мире. Многие стали уже обычным для нас, например, как распознавание речи или рукописного ввода, хотя появились эти примеры работы нейронной сети не так уж и давно.

1.6 Выводы по разделу

Изучение исследований, посвященных прогнозированию рядов с помощью нейронных сетей, выявило, что нейронные сети очень популярны для решения задач прогнозирования в разных сферах, но, к сожалению, пока нет универсальной модели, которая могла бы точно предсказывать любой временной ряд, независимо от его параметров.

Одним из преимуществ нейронной сети является то, что для прогнозирования нет необходимости выделять во временном ряду тренд, случайные и циклические составляющие. Нейронная сеть в процессе обучения сама определяет все вышеуказанное и должна учитывать при построении прогноза. Также, не берется во внимание какие-либо внешние факторы, оказывающие влияние на поведение временного ряда. Задача состоит в том, чтобы получить высококачественный прогноз для нейронной сети, используя минимум входных данных.

Для решения конкретной задачи прогнозирования необходимо выполнить настройку сети для получения наилучших результатов. Под настройкой сети подразумевается определение главным образом количества нейронов в скрытых слоях. Кроме того, сюда же можно добавить выбор алгоритма обучения нейронной сети.

Для прогнозирования временного ряда была выбрана модель многослойного персептрона прямого распространения. Были выбраны линейная и сигмоидальная активационные функции.

Таким образом, для исследуемой задачи необходимо разработать и настроить нейронную сеть, решающую задачу прогнозирования временного ряда.

2 МОДЕЛЬ И ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

2.1 Многослойный персептрон прямого распространения

Для прогнозирования временного ряда была выбрана модель многослойного персептрона прямого распространения (Feed-Forward MLP). Она представляет собой нейронную сеть без обратных связей, в которой входной сигнал преобразуется в выходной, проходя последовательно через несколько слоев.

Первый из таких слоев называют входным, последний - выходным. Эти слои содержат так называемые вырожденные нейроны и иногда в количестве слоев не учитываются. Кроме входного и выходного слоев, в многослойном персептроне есть один или несколько промежуточных слоев, которые называют скрытыми.

В этой модели персептрона должен быть хотя бы один скрытый слой. Присутствие нескольких таких слоев оправдано лишь в случае использования нелинейных функций активации.

Математическая модель используемой нейронной сети представлена ниже (для двухслойного).

Выходные значения сети на объекте x_i вычисляются как суперпозиция:

$$a^m(x_i) = \sigma_m \left(\sum_{h=0}^H \omega_{hm} u^h(x_i) \right) \quad (5)$$

$$u^h(x_i) = \sigma_h \left(\sum_{j=0}^J \omega_{jh} v^j(x_i) \right) \quad (6)$$

Прогнозируемое значение зависит от входных (предыдущих) значений временного ряда. На рисунке 2.1 представлена схема нейронной сети, использованной в данной работе.

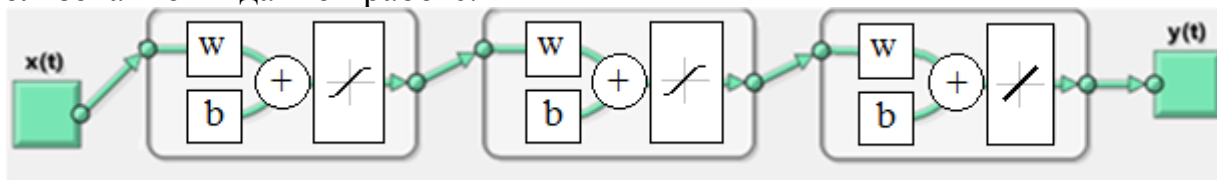


Рисунок 2.1 – Структура используемой нейронной сети

Для построения вышеописанной нейронной сети рассмотрим последовательность действий, которую нужно совершить. Сначала нужно определиться с входными и выходными данными. Так как задача состоит в прогнозировании временного ряда, то в качестве входных данных будут использовано подокно временного ряда размера 48 элементов (данные за 4 года), а в качестве выходных – прогноз на следующие 12 месяцев, следующие за данными 4-х лет, поданных на вход сети.

Использованная в данной работе сеть прямого распространения имеет 3 скрытых слоя. В качестве функций активации в скрытых слоях выступает сигмоидная функция в виде гиперболического тангенса 'tansig', а в выходном слое – линейная 'purelin'. Схема реализации используемой нейронной сети в пакете Matlab представлена на рисунке 2.2.

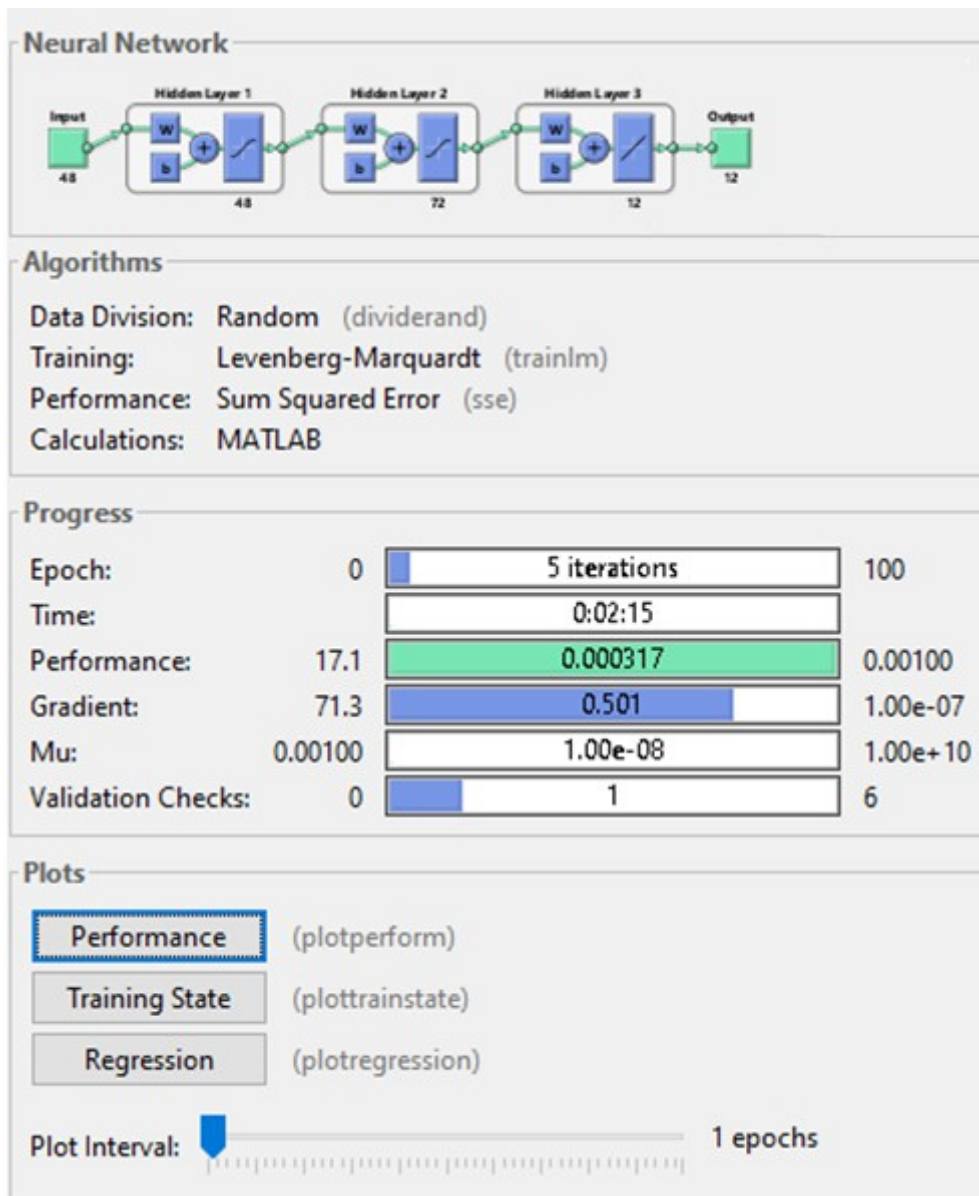


Рисунок 2.2 – Нейронная сеть с тремя скрытыми слоями в Matlab

Количество нейронов на скрытых уровнях будем устанавливать экспериментально.

Теперь осталось выбрать алгоритм обучения нейронной сети. Рассмотрим три алгоритма. Алгоритм Левенберга-Марквардта работает быстро; метод Байесовской регуляризации занимает больше времени, но может показать лучшие результаты для зашумленных данных и небольших задач; методу масштабируемых сопряженных градиентов требуется меньше памяти.

Каждый раз в результате обучения нейронной сети будут получаться другие решения. Это связано с различными начальными весами и различными способами разделения исходных данных на обучающее, проверяющее и тестовое подмножества. В результате для каждой нейронной сети одним и тем же входным данным будут соответствовать разные выходные значения. Чтобы гарантировать хорошую точность обучения сети, необходимо переобучать ее несколько раз. [13]

2.2 Алгоритм Левенберга-Марквардта

Алгоритм Левенберга-Марквардта представляет из себя разновидность метода Ньютона. Он был разработан для минимизации функций, которые представляют собой суммы квадратов других нелинейных функций (то есть решает задачу наименьших квадратов). Этот алгоритм очень хорошо подходит для обучения нейронной сети, при котором в качестве оптимизации используется среднеквадратичная ошибка. [14]

Чтобы описать данный алгоритм, необходимо иметь представление о базовом методе Ньютона, который минимизирует сумму квадратов.

Очередное приближение считается по формуле:

$$x_{k+1} = x_k - A_k^{-1} g_k, \quad (7)$$

где $A_k \equiv \nabla^2 F(x)|_{x=x_k}$,

$$g_k \equiv \nabla F(x)|_{x=x_k}.$$

Полагаем, что $F(x)$ представляет из себя сумму квадратов функций:

$$F(x) = \sum_{i=1}^N v_i^2(x) = v^T(x)v(x)$$

Тогда j -й элемент градиента находится:

$$[\nabla F(x)]_j = \frac{\partial F(x)}{\partial x_j} = 2 \sum_{i=1}^N v_i(x) \frac{\partial v_i(x)}{\partial x_j}$$

или в матричной форме:

$$\nabla F(x) = 2J^T(x)v(x), \quad (8)$$

где

$$J(x) = \begin{pmatrix} \frac{\partial v_1(x)}{\partial x_1} & \frac{\partial v_1(x)}{\partial x_2} & \dots & \frac{\partial v_1(x)}{\partial x_n} \\ \frac{\partial v_2(x)}{\partial x_1} & \frac{\partial v_2(x)}{\partial x_2} & \dots & \frac{\partial v_2(x)}{\partial x_n} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial v_N(x)}{\partial x_1} & \frac{\partial v_N(x)}{\partial x_2} & \dots & \frac{\partial v_N(x)}{\partial x_n} \end{pmatrix} \quad (9)$$

Матрица Гессе (k,j) определяется:

$$[\nabla^2 F(x)]_{k,j} = \frac{\partial^2 F(x)}{\partial x_k \partial x_j} = 2 \sum_{i=1}^N \left[\frac{\partial v_i(x)}{\partial x_k} \frac{\partial v_i(x)}{\partial x_j} + v_i(x) \frac{\partial^2 v_i(x)}{\partial x_k \partial x_j} \right]$$

или в матричной форме:

$$\nabla^2 F(x) = 2J^T(x)J(x) + 2S(x)$$

где

$$S(x) = \sum_{i=1}^N v_i(x) \nabla^2 v_i(x)$$

Если значение $S(x)$ мало, то можно аппроксимировать Гессиан следующим образом:

$$\nabla^2 F(x) \cong 2J^T(x)J(x) \quad (10)$$

При подстановке уравнения (8) и (10) в уравнение (7) и получаем метод Гаусса-Ньютона:

$$x_{k+1} = x_k - [2J^T(x_k)J(x_k)]^{-1} 2J^T(x_k)v(x_k) = \dot{x}_k - [J^T(x_k)J(x_k)]^{-1} J^T(x_k)v(x_k) \quad (11)$$

Преимущество метода Гаусса-Ньютона над классическим методом Ньютона заключается в том, что здесь не нужно вычислять вторые производные.

Единственная проблема данного метода состоит в том, что матрица $H = J^T J$ может быть необратимой. Для того чтобы избежать такой ситуации, модифицируется приближение матрицы Гессе:

$$G = H + \mu I$$

Полученная матрица G всегда будет обратимой. Пусть $\{ \lambda_1, \lambda_2, \dots, \lambda_n \}$ – собственные значения матрицы H , а $\{ z_1, z_2, \dots, z_n \}$ – собственные вектора. Тогда:

$$G z_i = [H + \mu I] z_i = H z_i + \mu z_i = \lambda_i z_i + \mu z_i = (\lambda_i + \mu) z_i$$

Таким образом, собственные вектора матрицы G совпадают с собственными векторами матрицы H , а собственные значения матрицы G равны $(\lambda_i + \mu)$. Матрица G будет положительно определена за счет увеличения μ , пока не будет получено, что $(\lambda_i + \mu) > 0$ для всех i , и, следовательно, полученная матрица G будет обратима.

Вышеописанная модификация приводит к алгоритму Левенберга-Марквардта:

$$x_{k+1} = x_k - [J^T(x_k)J(x_k) + \mu_k I]^{-1} J^T(x_k)v(x_k) \quad (12)$$

$$\Delta x_k = -[J^T(x_k)J(x_k) + \mu_k I]^{-1} J^T(x_k)v(x_k) \quad (13)$$

Полученный алгоритм обладает очень полезным свойством: при увеличении μ_k алгоритм приближается к алгоритму наискорейшего спуска (10), в то время как уменьшение значения μ_k до нуля приводит к алгоритму Гаусса-Ньютона (11).

$$x_{k+1} \cong x_k - \frac{1}{\mu_k} J^T(x_k) J(x_k) = x_k - \frac{1}{2\mu_k} \nabla F(x), \text{ при больших } \mu_k \quad (14)$$

Алгоритм начинается с небольшого значения μ_k (например, $\mu_k = 0.01$). Если на очередном шаге не получается уменьшить значение функции $F(x)$, то шаг повторяется с новым значением μ_k , умноженным на некоторый коэффициент α (например, $\alpha = 10$). В конце концов, значение функции $F(x)$ будет уменьшаться, так как будут совершаться небольшие шаги в направлении наискорейшего спуска. Если на очередной итерации значение функции $F(x)$ действительно уменьшилось, тогда для следующей итерации значение μ_k делится на α , приближая алгоритм к алгоритму Гаусса-Ньютона, который должен обеспечить более быструю сходимость. Таким образом, алгоритм Левенберга-Марквардта обеспечивает компромисс между скоростью сходимости алгоритма Ньютона и гарантией сходимости алгоритма наискорейшего спуска.

Рассмотрим применение алгоритма Левенберга-Марквардта для обучения многослойной нейронной сети. В процессе обучения минимизируется среднеквадратичная ошибка. Если значения выхода равновероятны, то среднеквадратичная ошибка пропорциональна сумме квадратов ошибок всех выходов Q в обучающем множестве:

$$F(x) = \sum_{q=1}^Q \frac{1}{2} (e_q)^T e_q = \sum_{q=1}^Q \sum_{j=1}^{S^M} (e_{j,q})^2 = \sum_{i=1}^N (v_i)^2, \quad (15)$$

где $e_{j,q}$ – j -й элемент ошибки для q -й пары входа/выхода.

Уравнение (15) эквивалентно оценке качества обучения сети, для минимизации которой был разработан алгоритм Левенберга-Марквардта. Её смысл должен быть понятным независимо от способа обучения сети. Оказывается, это верная концепция, но она требует некоторой осторожности в разработке деталей.

Ключевым шагом в алгоритме Левенберга-Марквардта является вычисление якобиана. Для выполнения этого вычисления используется вариация алгоритма обратного распространения, в классическом варианте которого вычисляются производные квадратов ошибок относительно весов и смещений сети. Для вычисления якобиана необходимо вычислить производные от ошибок, а не от их квадратов.

Прежде чем представить саму процедуру вычисления якобиана, необходимо более детально изучить его форму (9). Вектор ошибок представляется в виде:

$$v^T = [v_1 v_2 \dots v_N] = [e_{1,1} e_{2,1} \dots e_{S^M,1} e_{1,2} \dots e_{S^M,Q}], N = Q \times S^M \quad (16)$$

а вектор параметров:

$$x^T = [x_1 x_2 \dots x_n] = [w_{1,1}^1 w_{1,2}^1 \dots w_{S^1,R}^1 b_1^1 \dots b_{S^1}^1 w_{1,1}^2 \dots b_{S^2}^M], \quad (17)$$

где $n = S^1(R+1) + S^2(S^1+1) + \dots + S^M(S^{M-1}+1)$.

Подставлением (16) и (17) в уравнение (9) получается более подробная запись якобиана:

$$\begin{array}{ccccccc}
 & & & & \frac{\partial e_{1,1}}{\partial w_{1,1}^1} & & \\
 & & & & \frac{\partial e_{2,1}}{\partial w_{1,1}^1} & & \\
 & & & & \vdots & & \\
 & & & & \frac{\partial e_{S^M,1}}{\partial w_{1,1}^1} & & \\
 & & & & \frac{\partial e_{1,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,1}}{\partial b_1^1} & \dots \\
 \frac{\partial e_{1,1}}{\partial w_{1,2}^1} & \dots & & & \frac{\partial e_{2,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{2,1}}{\partial b_1^1} & \dots \\
 \frac{\partial e_{2,1}}{\partial w_{1,2}^1} & \dots & & & \vdots & \vdots & \dots \\
 \vdots & & & & \frac{\partial e_{S^M,1}}{\partial w_{S^1,R}^1} & \frac{\partial e_{S^M,1}}{\partial b_1^1} & \dots \\
 \frac{\partial e_{S^M,1}}{\partial w_{1,2}^1} & \dots & & & \frac{\partial e_{1,2}}{\partial w_{1,1}^1} & & \\
 \hline
 & & & & \frac{\partial e_{1,2}}{\partial w_{1,2}^1} & & \\
 & & & & \vdots & & \\
 \dots & & & & \frac{\partial e_{1,2}}{\partial w_{S^1,R}^1} & \frac{\partial e_{1,2}}{\partial b_1^1} & \dots \\
 & & & & \vdots & & \\
 J(x) = & & & & (\vdots \vdots \vdots & \vdots \vdots) &
 \end{array} \quad (18)$$

Элементы якобиана (18) могут быть вычислены с помощью простой модификации алгоритма обратного распространения по формуле:

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial x_l}$$

В алгоритме обратного распространения используется такое понятие, как чувствительность, которая вычисляется через рекуррентное соотношение от последнего слоя к первому. Та же концепция используется для вычисления условий, необходимых для (18), и чувствительность Марквардта:

$$s_{i,h}^{-m} = \frac{\partial v_h}{\partial n_{i,q}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m},$$

где $h = (q-1)S^m + k$

Элемент якобиана вычисляется:

$$[J]_{h,l} = \frac{\partial v_h}{\partial x_l} = \frac{\partial e_{k,q}}{\partial w_{i,j}^m} = \frac{\partial e_{k,q}}{\partial n_{i,q}^m} \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = s_{i,h}^{-m} \times \frac{\partial n_{i,q}^m}{\partial w_{i,j}^m} = s_{i,h}^{-m} \times a_{j,q}^{m-1} \quad (19)$$

Чувствительность Марквардта может быть вычислена через те же рекуррентные соотношения, как и стандартная чувствительность (16) с единственной модификацией на последнем слое. Уравнение для последнего слоя в случае чувствительности Марквардта записано в формуле (17).

$$s^m = F^m(n^m)(W^{m+1})^T s^{m+1} \quad (20)$$

$$\frac{\partial (\dot{i} \dot{i} k, q - a_{k,q}^M)}{\partial n_{i,q}^M} = \frac{\partial}{\partial n_{i,q}^M} \left(-f^M(n_{i,q}^M) \right) = \begin{cases} -f^M(n_{i,q}^M), & \text{для } i=k \\ 0, & \text{для } i \neq k \end{cases} \quad (21)$$

$$s_{i,h}^{-M} = \frac{\partial v_h}{\partial n_{i,q}^M} = \frac{\partial e_{k,q}}{\partial n_{i,q}^M} = \dot{i}$$

Когда на вход сети подается вектор входных значений p_q и на выходе сети получаются соответствующие вычисленные значения a_q^M , то алгоритм обратного распространения с помощью метода Левенберга-Марквардта инициализируется:

$$S_q^M = -F^M(n_q^M), \quad (22)$$

$$\begin{matrix} f^m(n_1^m) & 0 \\ 0 & f^m(n_2^m) \\ \dots & \dot{i} & 0 \\ & \dot{i} \\ & \vdots & \dot{i} & \vdots \\ \dot{i} & \vdots & \dot{i} & \vdots & \dot{i} \\ \dot{i} \dots & \dot{i} & 0 & \dot{i} & 0 \\ \vdots & \vdots & f & \dot{i} & \dot{i} m(n_{S^m}^m) \\ \dots & \vdots & \vdots & \dot{i} & \dot{i} m(n_{S^m}^m) \\ & & & & F^m(n^m) = \dot{i} \end{matrix}$$

Каждый столбец матрицы S_q^M должен пройти по алгоритму обратного распространения, используя формулу (16), чтобы получить одну строку в исходном якобиане. Также столбцы можно распространить обратно с помощью формулы:

$$S_q^m = F^m(n_q^m)(W^{m+1})^T S_q^{m+1} \quad (23)$$

Конечная матрица чувствительностей Марквардта создается путем расширения матриц, вычисленных для каждого входа:

$$S^m = [S_1^m | S_2^m | \dots | S_Q^m] \quad (24)$$

Вектор чувствительности будем вычислять для каждого входа, так как нам нужно вычислить производные непосредственно от каждой ошибки, а не от суммы квадратов ошибок. Каждому входу, подаваемому нейронной сети, будут соответствовать свои ошибки (своя для каждого элемента выходного вектора), а каждой из этих ошибок будет соответствовать одна строка в якобиане. После вычисления чувствительности Марквардта, элементы якобиана вычисляются по формуле (19).

Теперь с учетом вышеописанного, можем сформулировать итерационный алгоритм обратного распространения Левенберга-Марквардта.

1) Для каждого входа нейронной сети вычисляются соответствующие выходы и их ошибки $e_q = t_q - a_q^M$. Вычисляется сумма квадратов ошибок $F(x)$ по всем входам (11).

2) Вычисляется матрица Якоби (18). По рекуррентным формулам (23) вычисляется чувствительность, которая инициализируется значением (22). Полученная матрица дополняется значениями чувствительности Марквардта (24). Вычисляются элементы матрицы Якоби (19).

3) Решается уравнение (13), для получения Δx_k .

4) Пересчитывается сумма квадратов ошибок с учетом $x_k + \Delta x_k$. Если полученная сумма меньше той, что была посчитана на первом шаге, то делится μ на α , изменяется значение $x_{k+1} = x_k + \Delta x_k$ и происходит переход к первому шагу. Если же сумма квадратов ошибок не уменьшилась, то μ умножается на α и происходит переход к третьему шагу.

Алгоритм завершает свою работу, когда градиент (19) станет меньше некоторого наперед заданного значения или сумма квадратов ошибок будет в допустимых пределах.

Ключевым недостатком алгоритма является требование большого объема памяти для хранения всех данных, в частности, необходимо запоминать всю матрицу Гессе, тогда как во многих других методах требуется хранить только лишь градиент. Поэтому когда число параметров нейронной сети очень велико, использование алгоритма Левенберга-Марквардта нецелесообразно.

2.3 Выводы по разделу

Для решения поставленной задачи была выбрана модель нейронной сети многослойный персептрон прямого распространения и определены параметры настройки сети.

Для обучения выбранной модели нейронной сети был выбран алгоритм Левенберга-Марквардта. Главным его преимуществом является его быстрая сходимость. К недостаткам можно отнести необходимость в большом объеме памяти для хранения всех данных. отсутствие параметров, влияющих на успешность работы алгоритма. Был выбран именно этот алгоритм, так как он показывает точный результат за сравнительно небольшое время.

3 РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ (ПРОВЕДЕНИЕ ЭКСПЕРИМЕНТА)

3.1 Получение прогнозов

Целью данной работы является оценка эффективности прогнозирования временного ряда посредством нейронной сети. Для достижения цели необходимо написать программу, моделирующую многослойный персептрон прямого распространения, описанного в п. 2.1 данной работы, который обучается с помощью алгоритма Левенберга-Марквардта, описанный в п. 2.2 данной работы. Результат работы программы должен предоставлять прогнозируемые значения в виде набора значений.

Нейронная сеть была реализована в среде MATLAB с использованием программного пакета Neural Network Toolbox. Код программы представлен в приложении 1. Стоит отметить, что полученная программа строит прогноз для конкретного алгоритма обучения и фиксированного количества нейронов в скрытом слое. Схема алгоритма программы приведена на рисунке 3.1.

В качестве временного ряда взяты статистические данные о естественном приросте населения с 2006 по 2016 гг. с сайта Института демографии Национального исследовательского университета «Высшая школа экономики» [1]. В нашем распоряжении ежемесячные значения естественного прироста населения РФ каждого года из вышеуказанного диапазона. С помощью нейронной сети необходимо получить предполагаемые значения естественного прироста населения на 2013-2016 года и сравнить их с исходными данными, чтобы оценить качество прогноза, а затем получить данные на 2017 год.

В качестве обучающей выборки использовались данные за 2006-2012 года. Так как процесс обучения традиционно можно разделить на три этапа, то исходные данные поделятся следующим образом: 60% отвечают за обучение нейронной сети, 40% – тестируют полученную сеть. Полученный прогноз на 2013-2016 гг. содержит известные данные, они дают возможность оценить качество полученного прогноза. После проверки выводим 2017 год. Но также нужно отметить, что для тестирования можно взять любые 4 года.

Прежде, чем приступить к обучению, необходимо нормировать все данные:

$$\tilde{x} = \frac{x - m}{M - m}, \quad (25)$$

где \tilde{x} – нормированный элемент временного ряда;

x – элемент исходного ряда;

M – значение максимального элемента временного ряда;

m – значение минимального элемента временного ряда.

На вход нейронной сети подается номера по порядку (1, 2, ...), обозначающие порядковый номер момента времени, для которого известны значения естественного прироста. На выходе получают значения естественного прироста населения в следующий момент времени от заданного на входе.

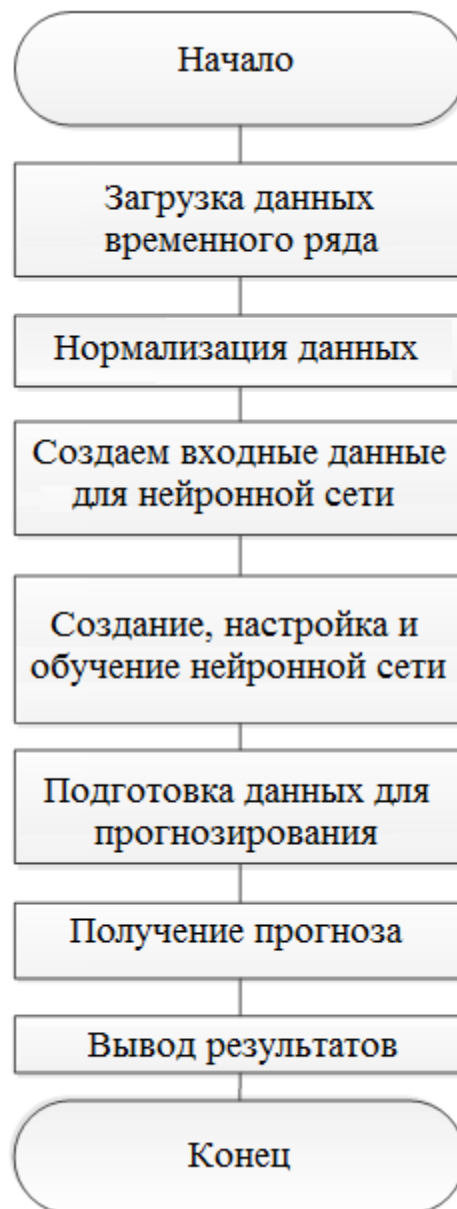


Рисунок 3.1 – Схема алгоритма программы

Чтобы выяснить, насколько точным получился прогноз и можно ли ему доверять, посчитаем среднюю абсолютную ошибку P :

$$\begin{aligned}
 & \hat{x}_i - x_i \vee \frac{\hat{x}_i}{x_i} \cdot 100\% \\
 P &= \frac{1}{n} \sum_{i=1}^n \hat{x}_i
 \end{aligned}
 \tag{26}$$

3.2 Полученные результаты

Нейронная сеть работает с 72 нейронами в скрытом слое с использованием алгоритма обучения – Левенберга-Марквардта.

На выходе из сети получаются нормированные данные, поэтому необходимо с помощью формулы (27) преобразовать выходные значения для дальнейшего сравнения. В таблицах представлены уже преобразованные значения.

$$x = \tilde{x} \cdot (M - m) + m \quad (27)$$

3.2.1 Пример №1

Таблица 1

Данные естественного прироста населения РФ, взятые из [1]

	Год										
	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
январь	-	-	-	-	-	-	-	-	-	-	-
февраль	99636	65331	59214	47976	43945	37554	21840	21804	14727	25453	25719
март	-	-	-	-	-	-	-	-	-	-	-
апрель	67539	45760	35343	24401	25686	21449	-8007	-5608	-1700	12145	-767
май	-	-	-	-	-	-	-	-	-	-	-
июнь	65908	55002	44824	32121	17671	20237	-5103	15157	-6177	14835	20068
июль	-	-	-	-	-	-	-	-	-	-	-
август	59589	51414	38694	27017	20057	21499	13503	-9048	-2526	-7712	-5050
сентябрь	-	-	-	-	-	-	-	-	-	-	-
октябрь	68708	51580	45076	28463	23567	28733	-8493	-8227	-5848	-5673	13848
ноябрь	-	-	-	-	-	-	-	-	-	-	-
декабрь	53946	38076	34464	19821	11226	-9348	19	6992	6309	4497	5358
январь	-	-	-	-	-	-	-	-	-	-	-
февраль	41476	27011	13112	-4237	-3813	-776	13287	23186	23731	26076	18466
март	-	-	-	-	-	-	-	-	-	-	-
апрель	36599	22570	-9811	1050	2341	10742	22761	23066	19854	23541	27879
май	-	-	-	-	-	-	-	-	-	-	-
июнь	40643	22195	18353	-3263	-8550	4154	11479	11709	12060	18543	8557
июль	-	-	-	-	-	-	-	-	-	-	-
август	51480	27747	15841	12170	-9346	-2658	10190	11702	10071	14250	22878
сентябрь	-	-	-	-	-	-	-	-	-	-	-
октябрь	51480	27747	15841	12170	-9346	-2658	10190	11702	10071	14250	22878
ноябрь	-	-	-	-	-	-2302	3810	5839	438	3448	-1328

	51660	36000	22570	25891	10516						
	-	-	-	-	-						
дек	52337	34995	26179	25116	14622	-1548	-7173	263	-2700	8186	-1372

Таблица 2

Полученный прогноз естественного прироста населения РФ на 2013-2017гг.

Месяц	Год				
	2013	2014	2015	2016	2017
январь	-22002	-14578	-25531	-25532	-48217
февраль	-5798	-1844	-12023	-873	-6446
март	-15276	-6082	-14806	-19994	-39115
апрель	-9056	-2656	-7526	-5123	-7456
май	-8154	-5692	-5831	-13982	-3783
июнь	6803	6318	4512	5378	-1892
июль	23301	23744	25903	18513	14626
август	23307	19695	23901	27732	13941
сентябрь	11698	11820	18641	8404	-4638
октябрь	11756	10106	14278	22869	23267
ноябрь	5962	399	3662	-1416	5098
декабрь	254	-2547	8057	-1239	5696

Представим начальные данные и данные прогноза на графиках (рисунки 3.2, 3.3, 3.4, 3.5).

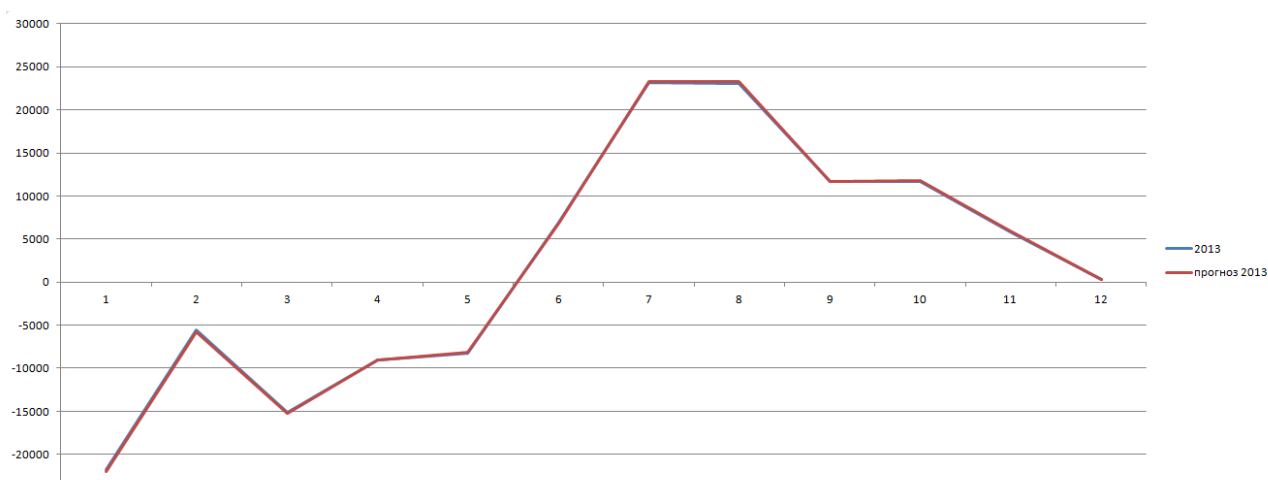


Рисунок 3.2 – Начальные данные и данные прогноза на 2013г

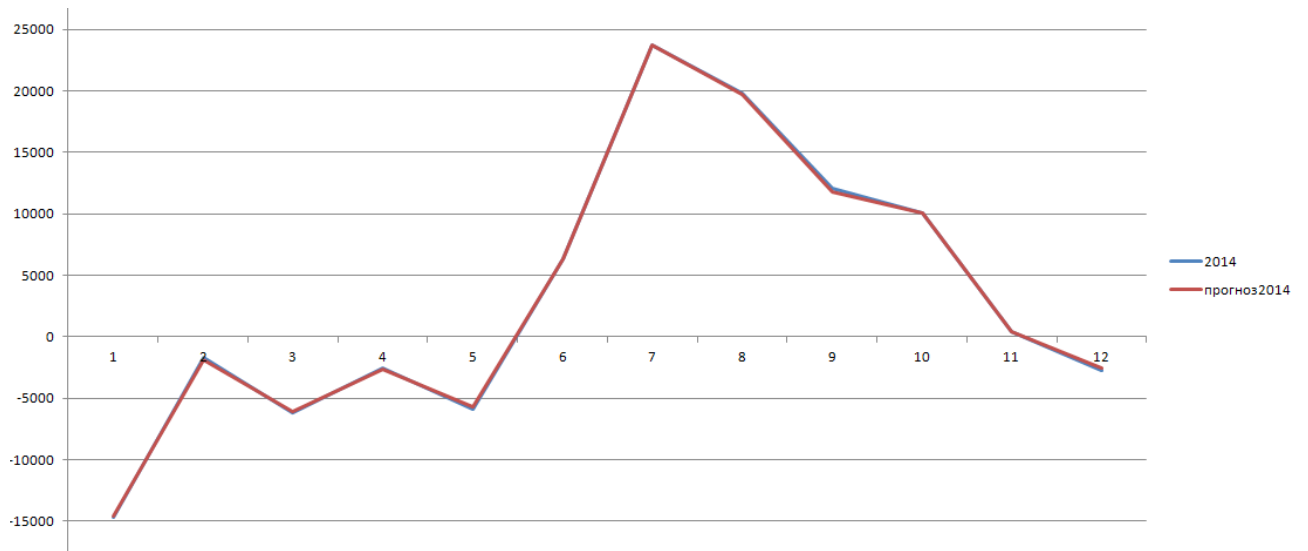


Рисунок 3.3 – Начальные данные и данные прогноза на 2014г

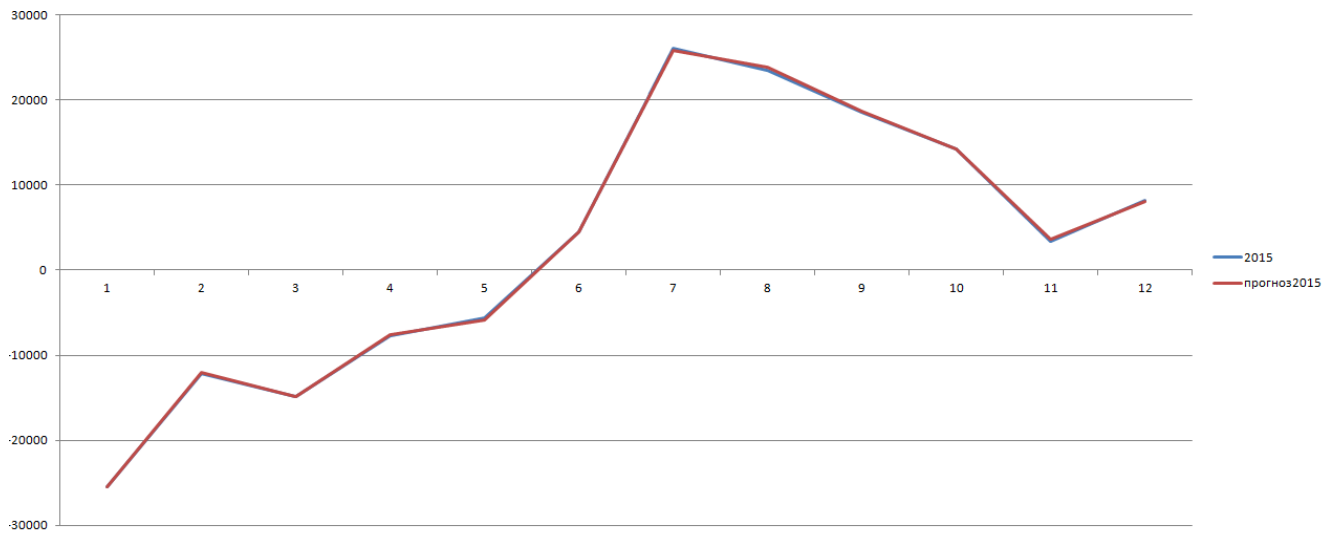


Рисунок 3.4 – Начальные данные и данные прогноза на 2015г

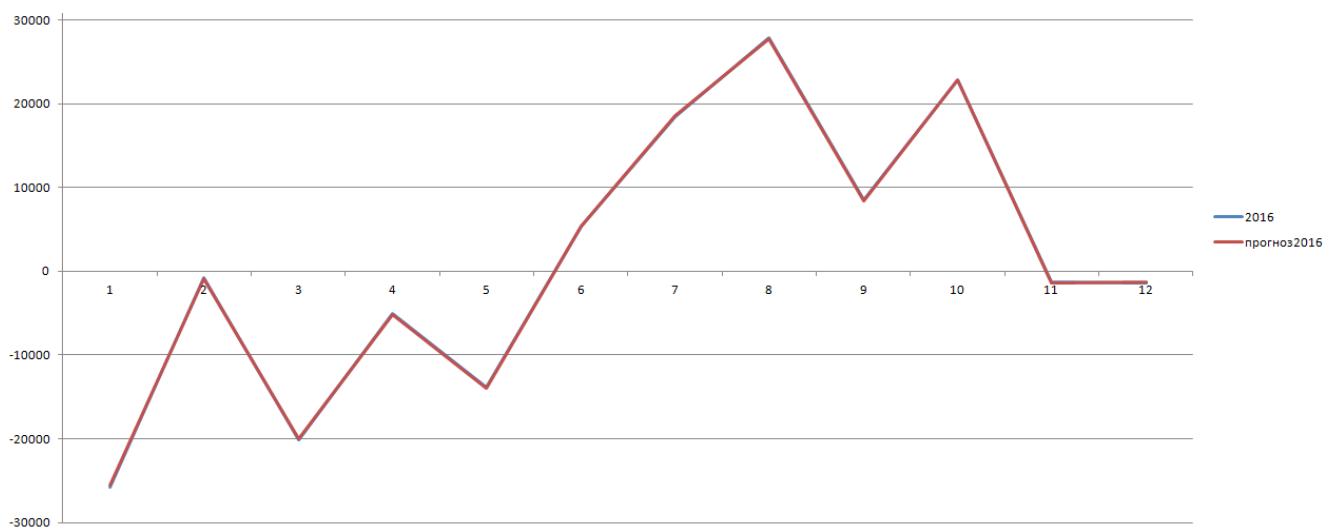


Рисунок 3.5 – Начальные данные и данные прогноза на 2016г

Графики (рисунки 3.2, 3.3, 3.4, 3.5) отличаются на величины незаметные в таком маленьком размере картинок, а это значит, что прогноз удался. Но для того, чтобы признать прогноз нейронной сети корректным необходимо посчитать ошибку полученного прогноза.

Ошибка составила $P=0,0253$. Это достаточно маленькая ошибка, поэтому можно считать, что прогноз оказался корректным.

Для наглядности построим график (рисунок 3.6) прогноза на 2017г.

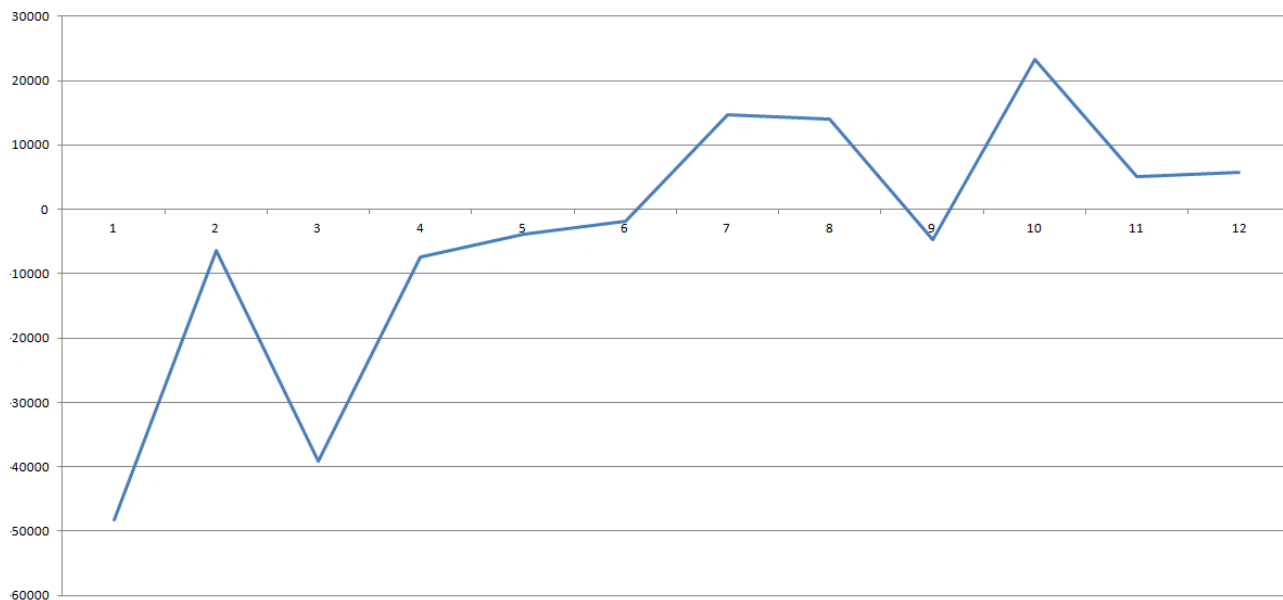


Рисунок 3.6 – Данные прогноза на 2017г

Проверим действие программы на других примерах.

3.2.2 Пример №2

Таблица 3

Данные ежемесячной инфляции РФ, взятые из [15]

	Год											
	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	
янв	2,43	1,68	2,31	2,37	1,64	2,37	0,5	0,97	0,59	2,65	0,96	
февр	1,66	1,11	1,2	1,65	0,86	0,78	0,37	0,56	0,7	2,22	0,63	
март	0,82	0,59	1,2	1,31	0,63	0,62	0,58	0,34	1,02	1,21	0,46	
апр	0,35	0,57	1,42	0,69	0,29	0,43	0,31	0,51	0,9	0,46	0,44	
май	0,48	0,63	1,35	0,57	0,5	0,48	0,52	0,66	0,9	0,35	0,41	
июн	0,28	0,95	0,97	0,6	0,39	0,23	0,89	0,42	0,62	0,19	0,36	
июл	0,67	0,87	0,51	0,63	0,36	-0,01	1,23	0,82	0,49	0,8	0,54	
авг	0,19	0,09	0,36	0	0,55	-0,24	0,1	0,14	0,24	0,35	0,01	
сент	0,09	0,79	0,8	-0,03	0,84	-0,04	0,55	0,21	0,65	0,57	0,17	
окт	0,28	1,64	0,91	0	0,5	0,48	0,46	0,57	0,82	0,74	0,43	

нояб	0,63	1,23	0,83	0,29	0,81	0,42	0,34	0,56	1,28	0,75	0,44
дек	0,79	1,13	0,69	0,41	1,08	0,44	0,54	0,51	2,12	0,77	0,4

Таблица 4

Полученный прогноз месячной инфляции РФ на 2013-2017гг.

Месяц	Год				
	2013	2014	2015	2016	2017
январь	0,93	0,62	2,68	0,92	0,62
февраль	0,51	0,67	2,26	0,59	0,22
март	0,37	0,98	1,17	0,42	0,13
апрель	0,55	0,86	0,49	0,45	0,33
май	0,69	0,87	0,38	0,43	0,43
июнь	0,42	0,62	0,19	0,36	0,25
июль	0,82	0,49	0,8	0,54	0,62
август	0,14	0,24	0,35	0,01	0,19
сентябрь	0,25	0,68	0,59	0,19	0,47
октябрь	0,52	0,8	0,78	0,45	0,66
ноябрь	0,51	1,29	0,71	0,42	0,71
декабрь	0,5	2,14	0,76	0,4	0,79

Представим начальные данные и данные прогноза на графиках (рисунки 3.7, 3.8, 3.9, 3.10).

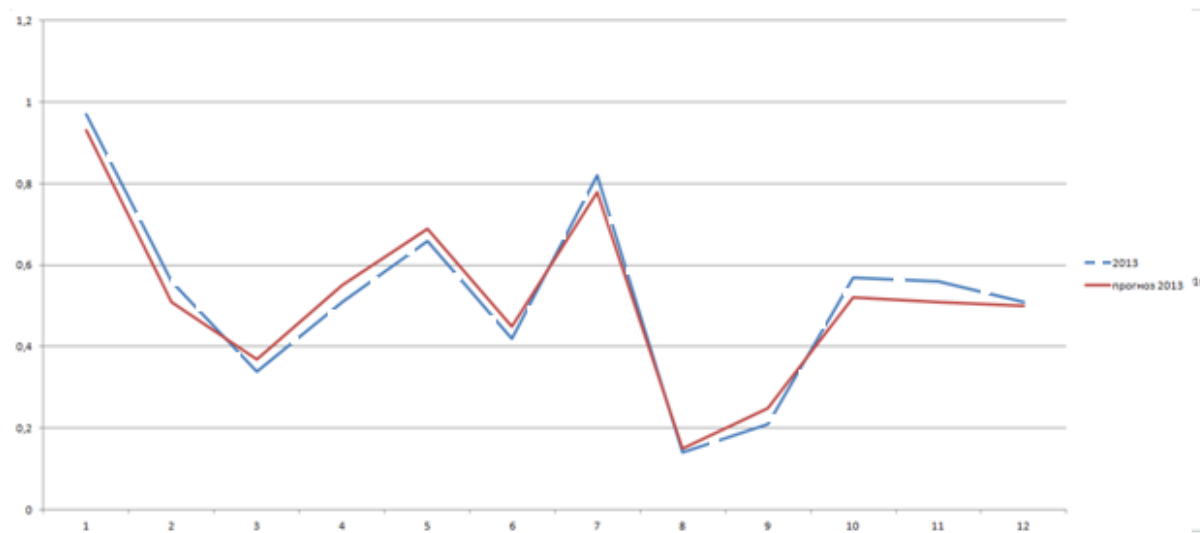


Рисунок 3.7 – Начальные данные и данные прогноза на 2013г

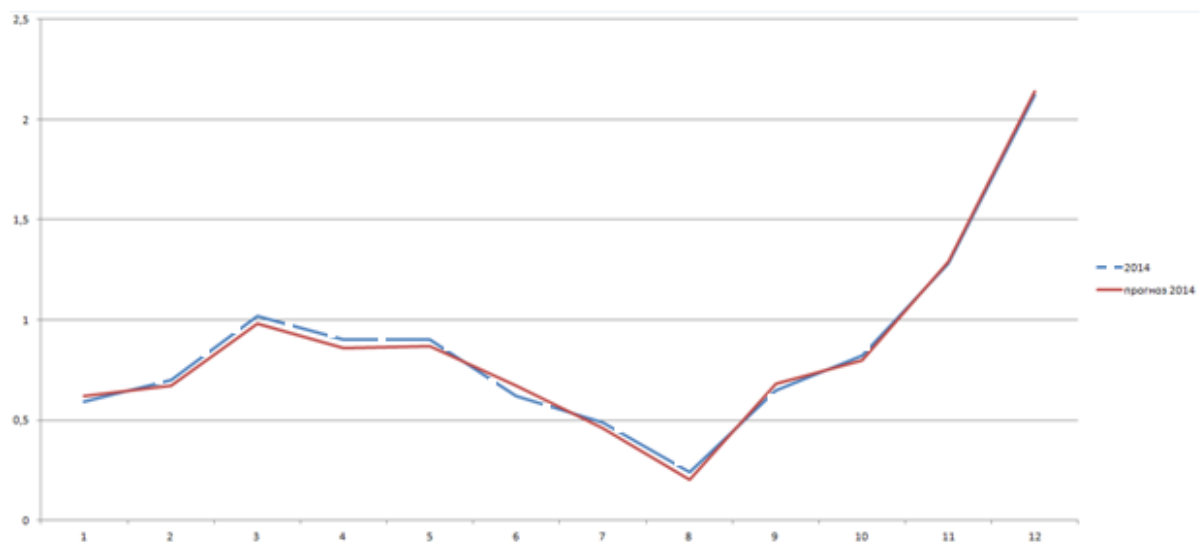


Рисунок 3.8 – Начальные данные и данные прогноза на 2014г

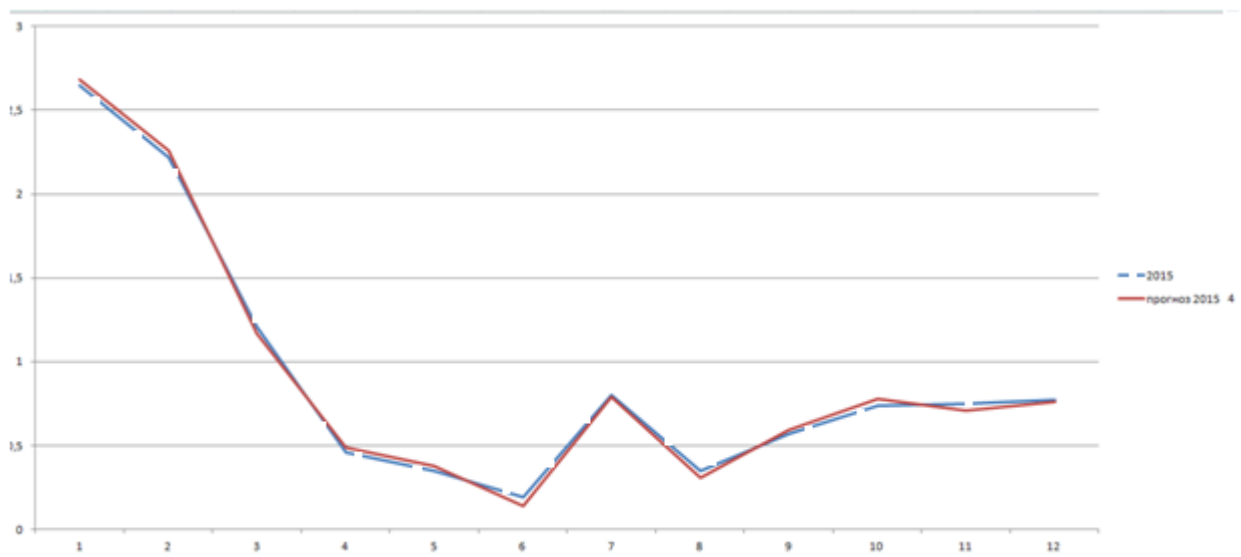


Рисунок 3.9 – Начальные данные и данные прогноза на 2015г

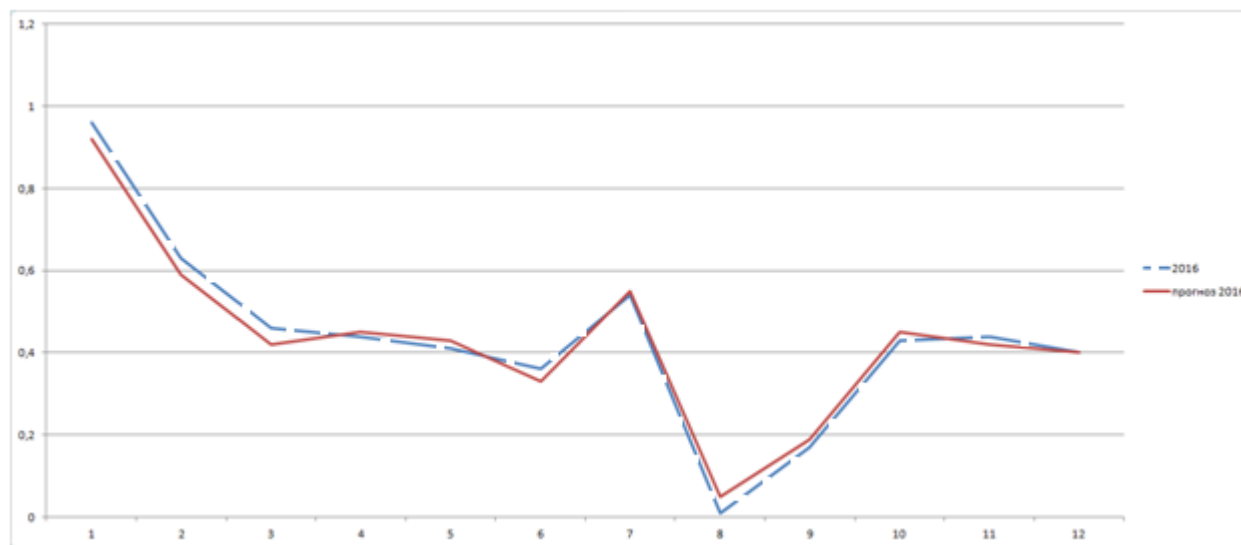


Рисунок 3.10 – Начальные данные и данные прогноза на 2016г

В данных графиках (рисунки 3.7, 3.8, 3.9, 3.10) данные исходные и прогноза отличаются сильнее друг от друга, чем в прошлом примере, но все же, чтобы проверить корректность прогноза нейронной сети необходимо посчитать ошибку полученного прогноза.

Ошибка составила $P = 0,0335$. Это достаточно маленькая ошибка, поэтому можно считать, что прогноз оказался корректным.

Для наглядности построим график (рисунок 3.11) прогноза на 2017г.

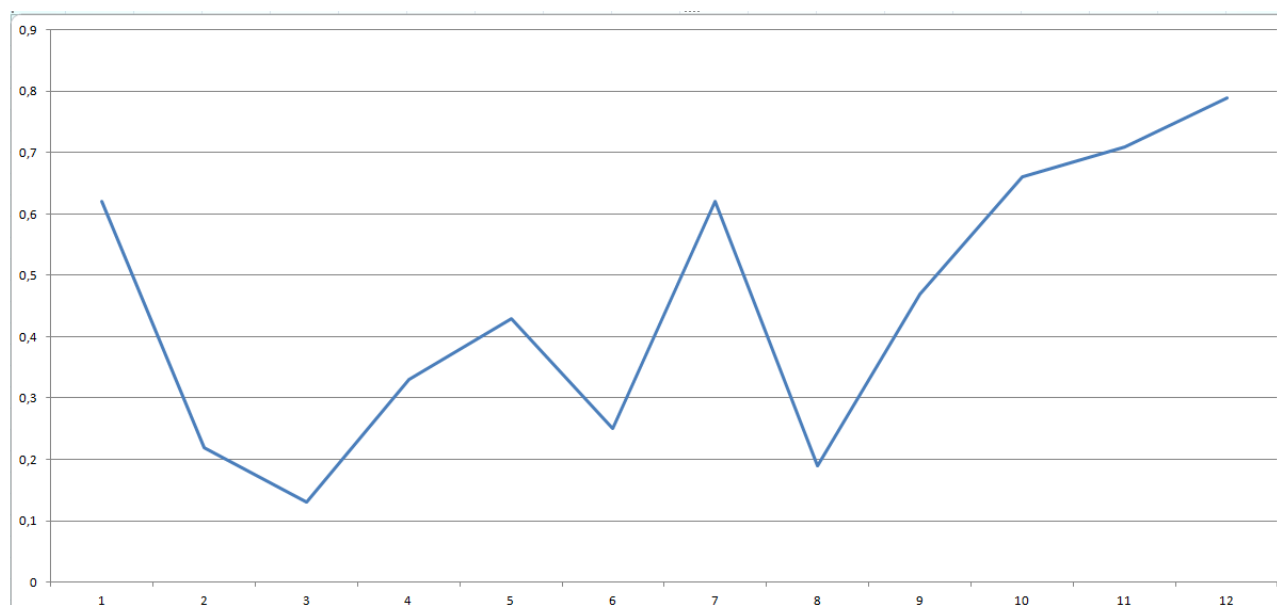


Рисунок 3.11 – Данные прогноза на 2017г

3.2.3 Пример №3

Таблица 5

Данные по распределению населения в Челябинской области по возрастным группам, взятые из [16]

Воз- раст	Год										
	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016
0-9	334011	338329	348304	359465	372289	380684	395543	410373	423461	436593	449096
10-14	195711	181908	169208	165219	159743	158499	159171	163493	166695	172289	175410
15-19	296136	280578	261772	238165	217334	198210	183763	169036	165587	160552	158780
20-24	302218	304257	307907	305990	302190	293751	279879	262834	238641	218857	198964
25-29	273709	273996	277298	284823	290954	294013	301179	307557	307343	304737	298494
30-34	253212	257145	260528	263615	265294	267879	269338	272513	279739	286386	290467

35-44	424476	411644	405641	402716	406219	413243	422176	431305	439554	446165	452734
45-49	287852	286100	280431	271981	259410	242442	228358	216951	208829	206447	209052
50-54	274583	272732	271791	274691	272493	273345	272741	267683	260226	248717	232925
55-59	220956	234418	245046	251537	253703	257212	255917	255369	258474	257068	258210
60-69	304508	298300	291895	285132	289290	296466	309890	334558	361058	391015	415053
>70	299289	307576	319271	331475	342899	349890	352187	343600	330446	318448	311531

Таблица 6

Полученный прогноз распределения населения в Челябинской области по возрастным группам, на 2013-2017гг.

Возраст	Год				
	2013	2014	2015	2016	2017
0-9	411206	424761	437599	449289	458037
10-14	161927	167505	173089	174119	179153
15-19	169912	166327	161502	159088	153029
20-24	265023	239541	220057	197767	180257
25-29	308857	305943	305537	297796	296054
30-34	271213	277739	287086	290535	293156
35-44	431905	440354	445165	454335	454072
45-49	217951	209429	207523	207987	210186
50-54	266083	263226	248517	233919	220032
55-59	256369	256474	255368	256924	257007
60-69	333558	361758	389815	413949	430182
>70	343967	331646	317138	312232	300059

Представим начальные данные и данные прогноза на графиках (рисунки 3.12, 3.13, 3.14, 3.15).

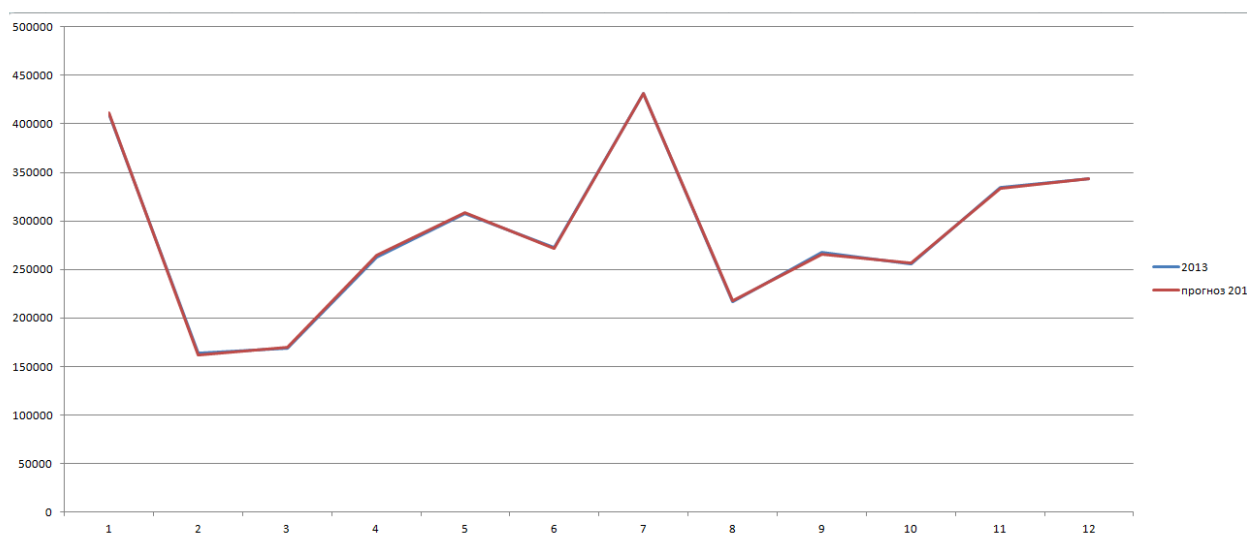


Рисунок 3.12 – Начальные данные и данные прогноза на 2013г

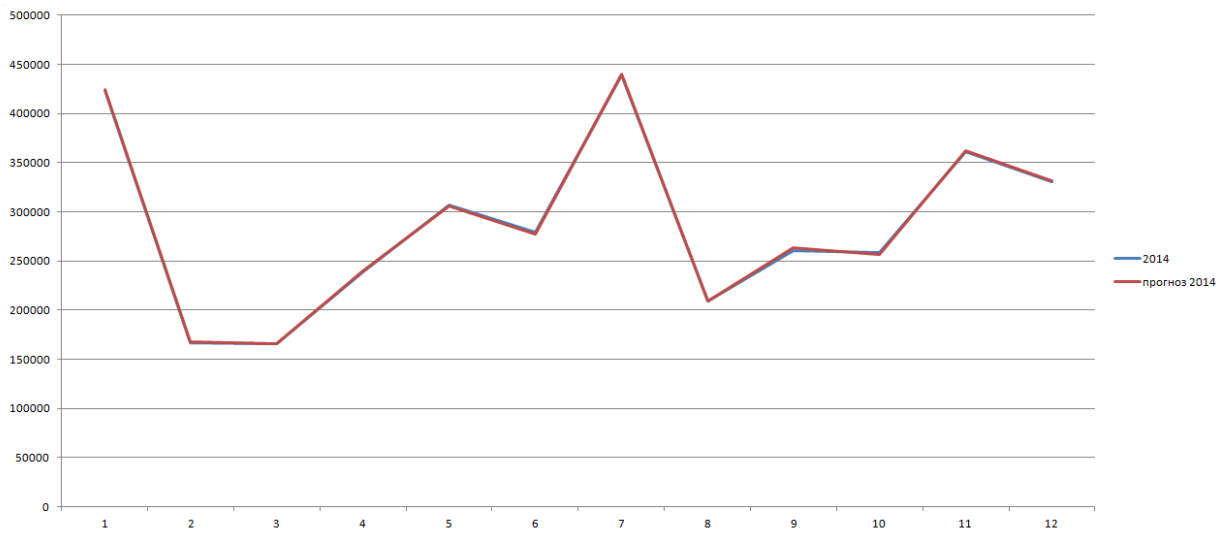


Рисунок 3.13 – Начальные данные и данные прогноза на 2014г

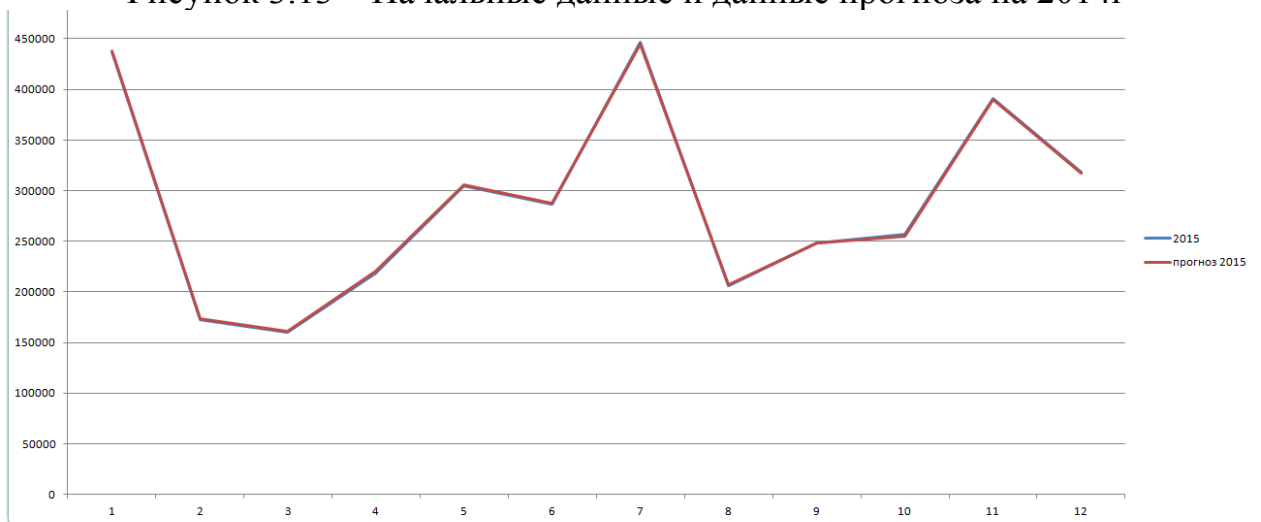


Рисунок 3.14 – Начальные данные и данные прогноза на 2015г



Рисунок 3.15 – Начальные данные и данные прогноза на 2016г

Графики (рисунки 3.12, 3.13, 3.14, 3.15) отличаются на величины незаметные в таком маленьком размере картинок, а это значит, что прогноз удался. Но для того, чтобы признать прогноз нейронной сети корректным необходимо посчитать ошибку полученного прогноза.

Ошибка составила $P = 0,0144$. Это достаточно маленькая ошибка, поэтому можно считать, что прогноз оказался корректным.

Для наглядности построим график (рисунок 3.16) прогноза на 2017г.



Рисунок 3.16 – Данные прогноза на 2017г

Ошибка в первом примере составила 0,0253, во втором – 0,0335, в третьем – 0,0144.

Средняя ошибка $(0,0253 + 0,0335 + 0,0144) / 3 = 0,0244$.

Это показывает эффективность применения модели нейронной сети многослойного персептрона для прогнозирования временного ряда.

Стоит сказать о трудоёмкости выбранного алгоритма обучения. Нейронная сеть с помощью алгоритма Левенберга-Марквардта обучалась на вышеизложенных примерах за 5 итераций.

3.3 Выводы по разделу

При 72 нейронах в скрытом слое в случае обучения алгоритмом Левенберга-Марквардта было выявлено, что нейронная сеть оказывается не трудоемкой и показывает небольшую ошибку прогноза. Таким образом, был получен прогноз естественного прироста населения РФ на 2017 г. Ошибка первого прогноза составила 2,53%, что свидетельствует об эффективности применения многослойного персептрона прямого распространения для прогнозирования временных рядов, так как полученный прогноз получается довольно точным. Также были спрогнозированы данные на 2017 г. еще в нескольких примерах и в них также получился прогноз корректный.

Была разработана программа в среде Matlab с использованием программного пакета Neural Network Toolbox, моделирующая многослойный перцептрон прямого распространения нейронной сети. Результатами работы программы являются данные спрогнозированных временных рядов.

ЗАКЛЮЧЕНИЕ

Для решения конкретной задачи прогнозирования временного ряда с помощью модели многослойного персептрона прямого распространения в ходе работы был выбран алгоритм обучения и значения количества нейронов в скрытом слое. Разработана программа в среде Matlab, моделирующая модель многослойного персептрона прямого распространения. Результатами работы программы являются данные спрогнозированных временных рядов. С помощью нее был получен прогноз естественного прироста населения на 2017 г. с помощью нейронной сети, а также спрогнозированы значения на 2017 г еще в нескольких примерах временных рядов. Ошибка первого прогноза составила 0,0253, а средняя ошибка – 0,0244, что говорит о высокой точности полученного прогноза. Это показывает эффективность применения модели нейронной сети многослойного персептрона для прогнозирования временного ряда.

Все поставленные задачи выполнены и полученные прогнозы удовлетворяют заявленным требованиям.

Самый лучший прогноз получен на выходе нейронной сети с 72 нейронами в скрытом слое и обученной по алгоритму Левенберга-Марквардта. Также, если оценивать вычислительную сложность алгоритмов обучения, то очевидно стоит отдать предпочтение этому же алгоритму. Поэтому и с точки зрения качества прогноза, и с точки зрения эффективности, для рассматриваемой задачи нейронная сеть, обучаемая с помощью вышеуказанного алгоритма, имеет явное преимущество. Полученную ошибку прогноза (2,53%) можно считать приемлемой.

Алгоритм Левенберга-Марквардта является одним из наилучших алгоритмов, который используется при обучении нейронной сети для решения задачи прогнозирования с использованием минимума входной информации.

Еще одним важным заключением данной работы является то, что удалось получить довольно точные прогнозы, используя минимум входной информации. Имеется в виду то, что как описывалось ранее, на вход сети подается лишь время, и все особенности ряда (тренд, циклическая и случайная составляющая), а так же внешние факторы, влияющие на поведение временного ряда, не учитываются. И, тем не менее, полученный прогноз говорит о том, что нейронная сеть способна восстанавливать функцию без дополнительной информации, которая зачастую может быть вовсе неизвестна. В этом заключается несомненное преимущество нейронной сети по сравнению с другими способами прогнозирования. Но в данном случае следует говорить о краткосрочном прогнозировании. Это обусловлено первым делом отсутствием какой-либо вспомогательной информации, помимо времени, которая, безусловно, повысила бы точность полученного прогноза. А во-вторых, сеть получает прогнозируемые значения последовательно, то есть вновь полученный прогноз зависит, в том числе, и от предыдущих полученных значений, которые не могут быть гарантированно точными. Поэтому с каждой итерацией, ошибка получаемого прогноза

увеличивается. А первые прогнозируемые значения получаются с высокой точностью.

В перспективе можно поменять алгоритм для выбранных данных, либо изменить тип данных для выбранного алгоритма и провести анализ полученных данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Институт демографии Национального исследовательского университета «Высшая школа экономики». – URL: <http://www.demoscope.ru/> (дата обращения 03.02.2017)

2 Саиян, С.А. Использование нейронных сетей на основе многослойного персептрона для прогнозирования статистических данных на примере погоды/ С.А Саиян, И.В. Лезина // Актуальные проблемы авиации и космонавтики, т.1, № 9. – 2013. – С. 381-382.

3 Strnad, D. Neural network models for group behavior prediction: a case of soccer match attendance / D. Strnad, A. Nerat, S. Kohek // Neural Comput & Applic. – 2015. – С. 485–492.

4 Маль, Г.С. Прогнозирование гипополидемического эффекта у больных ишемической болезнью сердца с помощью нейронных сетей. / Г.С. Маль// Вестник новых медицинских технологий – 2005 – Т. XII, № 2 – С.96.

5 Чернецов, В. И. Прогнозирование потребления электрической энергии с использованием нейронных сетей/ В. И. Чернецов, Е. Н. Казаковский // Труды Международного симпозиума «Надежность и качество», 2006. – С. 256-271.

6 Stojadinovic, S. Prediction of flyrock launch velocity using artificial neural networks / S. Stojadinovic, N. Lilic, I. Obradovic, R. Pantovic, M. Denic // Neural Comput & Applic. – 2016, №27. – P. 515–524.

7 Hodge, V.J. Short-term prediction of traffic flow using a binary neural network / V.J. Hodge, R. Krishnan, J. Austin, // Neural Comput & Applic. – 2014, №25. – P. 315–322.

8 Эконометрика: учебник / И.И. Елисеева [и др]; под. ред. И.И. Елисеевой. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2005. – 576 с.

9 Чучуева, И.А. Модель прогнозирования временных рядов по выборке максимального правдоподобия: дис. канд. техн. наук: 05.13.18 / И.А. Чучуева– М.: Айрис-пресс. – 2012. – 155 с.

10 Основные свойства нейронных сетей. – Дата обновления 02.02.2014. URL: http://studopedia.ru/2_52381_osnovnie-svoystva-neyronnih-setey.html (дата обращения 17.03.2017)

11 Заенцев, И.В. Нейронные сети. Основные модели: учебное пособие / И.В. Заенцев. – Воронеж: ВГУ, 1999. – 76 с.

12 Журавлева, Л. В. Исследования особенностей развития нейронных сетей в современном мире / Журавлева, Л. В., Стригулин К. А. // Технические науки: проблемы и перспективы: материалы IV Междунар. науч. конф. (г. Санкт-Петербург, июль 2016 г.). — СПб.: Свое издательство, 2016. — С. 9-11.

13 The MathWorks. Documentation : офиц. сайт. – 1994 – 2016. – URL: <http://www.mathworks.com/help/> (дата обращения 01.04.2017)

14 Hagan, M.T. Neural Network Design / M.T. Hagan. – 2nd ed. – MA: PWS, 1996. – 523 p.

15 Уровень инфляции Российской Федерации. – Дата обновления 10.01.2017. URL: <http://уровень-инфляции.рф/> (дата обращения 30.05.2017)

16 Федеральная служба государственной статистики. – Дата обновления 12.01.2017. URL: http://chelstat.gks.ru/wps/wcm/connect/rosstat_ts/chelstat/ru/statistics/employment/ (дата обращения 30.05.2017)

Текст программы

```

Файл train
%% Time Series Forecasting with Neural Network
% Задание: прогнозирование временного ряда естественного прироста
% на 2017 год на основе данных о предшествующих годах.

%% Структура сети
% Многослойный персептрон:
% Входной слой: 12*4=48 нейронов,
% Скрытый слой: 72 нейрона,
% Выходной слой: 12 нейронов.

%% Очистка памяти и командной строки
clear; clc;
close all;

%% Загрузка обучающей выборки
load training_data.mat;

%% Нормализация данных
Min = min(abs(trainingdata(:,1)));
Max = max(abs(trainingdata(:,1)));
B = Min; % Величина смещения
K = 1/(Max - Min); % Коэффициент масштабирования
trainingdata(:,2) = (trainingdata(:,1) - B) .* K; % Масштабирование

%% Создаем входные данные для нейронной сети
NumPats = (length(trainingdata(:,1))-48)-11; % количество элементов в
обучающей выборке
X_Train = zeros(48, NumPats); X_Labels = zeros(12, NumPats);
for i = 1 : NumPats
    % Входные данные
    X_Train(:,i) = trainingdata([i:48 + i - 1],2); % данные
    X_Labels(:,i) = trainingdata([48 + i: 48 + i - 1 + 12],2); % выходные значения
end

%% Создание сети или использование готовой
if ~exist('net.mat','file')
    net=newff(X_Train, X_Labels,[48 72,12],{'tansig' 'tansig' 'purelin'},'trainlm');
    net = init(net);
    save net.mat net;
else

```

```

load net.mat;
end

%% Обучение сети
net.trainFcn = 'trainscg';
net.performFcn='sse';%'msereg';
net.trainParam.lr = 0.1;
net.trainParam.show = 10;

net.trainParam.goal=1e-3;
net.trainParam.epochs=100;
[net,tr]=train(net,X_Train,X_Labels);
save net.mat net;

Файл test
%% Очистка памяти и командной строки
clear; clc;
close all;

%% Загрузка обучающей выборки
load training_data.mat;

%% Загрузка обученной нейронной сети
load net.mat;

%% Нормализация данных
Min = min(abs(trainingdata(:,1)));
Max = max(abs(trainingdata(:,1)));
B = Min; % Величина смещения
K = 1/(Max - Min); % Коэффициент масштабирования
trainingdata(:,2) = (trainingdata(:,1) - B) .* K; % Масштабирование

%% Тестирование работы сети
% 2014 год
X_Test = trainingdata([length(trainingdata(:,1))-48-36+1:length(trainingdata(:,1))-
36],2);
y2014=sim(net,X_Test); % выход сети
y2014 = y2014 ./ K + B; % восстанавливаем результат
y2014=int32(y2014)

```

```
% 2015 год
X_Test = trainingdata([length(trainingdata(:,1))-48-24+1:length(trainingdata(:,1))-
24],2);
y2015=sim(net,X_Test); % выход сети
y2015 = y2015 ./ K + B; % восстанавливаем результат
y2015=int32(y2015)

% 2016 год
X_Test = trainingdata([length(trainingdata(:,1))-48-12+1:length(trainingdata(:,1))-
12],2);
y2016=sim(net,X_Test); % выход сети
y2016 = y2016 ./ K + B; % восстанавливаем результат
y2016=int32(y2016)

% 2017 год
X_Test = trainingdata([length(trainingdata(:,1))-48+1:length(trainingdata(:,1))],2);
y2017=sim(net,X_Test); % выход сети
y2017 = y2017 ./ K + B; % восстанавливаем результат
y2017=int32(y2017)
```