

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт энергетики и электротехники

(институт)

Кафедра Промышленная электроника

11.04.04 – Электроника и наноэлектроника

(код и наименование направления подготовки, специальности)

Электронные приборы и устройства

(направленность (профиль))

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

на тему «Исследование влияния структуры искусственной нейронной сети и параметров ее обучения на точность моделирования источника питания для контактной сварки»

Студент

Карчевский А.П.

(И.О. Фамилия)

(личная подпись)

Руководитель

Глибин Е.С.

(И.О. Фамилия)

(личная подпись)

Консультанты

(И.О. Фамилия)

(личная подпись)

Руководитель программы \_\_\_\_\_

(ученая степень, звание, И.О.Ф.)

(личная подпись)

**Допустить к защите**

Заведующий кафедрой

«Промышленная электроника» к.т.н, доцент А.А. Шевцов \_\_\_\_\_

(ученая степень, звание, И.О.Ф.)

(личная подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 2016 г.

Тольятти 2016

## **Аннотация**

Исследование влияния структуры искусственной нейронной сети и параметров ее обучения на точность моделирования источника питания для контактной сварки: магистерская диссертация по направлению «Электронные приборы и устройства». — Тольятти, ТГУ, Кафедра «Промышленная электроника», 2015. Так как имеется сложность традиционного математического моделирования в проекте была создана и рассмотрена искусственная нейронная сеть, которая предназначена для имитации высокой электрической нагрузки. Для осуществления выполнения данной работы был использован программный комплекс MATLAB 2014a вкпе с пакетом Simulink и компонентом Neural Network Toolbox. В результате проделанной работы была получена искусственная нейрона сеть, которая способна адекватно реагировать на изменения питающего напряжения и параметров источника питания для контактной сварки.

## Содержание

Введение.....	5
1 Знакомство с объектом исследования.....	9
1.1 Трансформаторы для дуговой и контактной сварки с электронным регулированием .....	13
1.2 Временные диаграммы работа сварочного источника питания	15
1.3 Схемы источников питания для сварки с прерывистым током .	18
2 Обзор способов моделирования электрических нагрузок с помощью искусственных нейронных сетей.....	21
2.1 Структура искусственной нейронной сети .....	21
2.2 Моделирование с помощью алгоритма обратного распространения ошибки .....	29
2.3 Метод Левенберга-Маркара.....	33
2.4 Алгоритм байесовской регуляризации .....	37
2.5 Масштабируемый алгоритм сопряженных градиентов. ....	42
3 Модель RL-цепи в MATLAB Simulink .....	47
3.1 Анализ влияния структуры нейронной сети на точность при моделировании RL-ветви .....	53
3.2 Синтаксис кода нейронной сети, обученной по методу Левенберга-Маркара .....	71
4. Моделирование тиристорного трансформатора контактной машины на основе нейронной сети.....	72
4.1 Обучение искусственной нейронной сети методом Левенберга-Маркара.....	75

4.2 Обучения искусственной нейронной сети методом байесовской регуляризации.....	83
4.3 Обучение искусственной нейронной сети методом масштабируемого алгоритма сопряженных градиентов .....	90
Заключение .....	96
Список литературы .....	98

## Введение

**Актуальность работы.** Одним из основополагающих этапов проектов электроснабжения на предприятиях является вычисление ожидаемых электрических нагрузок. Они определяют необходимые характеристики элементов электрических сетей — это типы проводников и проводов, мощности трансформаторов и т.д. Превышение таких нагрузок может привести к увеличению расходов, вложенных в стоимость работ, а также к неоправданным потерям в сетях.

В этих случаях вложения, которые являются критерием экономности в проектах, а также себестоимости электроэнергии повышаются. Точное вычисление электрических нагрузок обеспечивает правильное решение в выборе и экономную работу устройств регулировки напряжения, защиты и автоматики, а также устройств регулировки напряжения.

Проблемами изучения закономерностей работы электрических нагрузок, прогнозированию режимов их работы посвящено большое количество работ. Значительный вклад в изучение вопроса внесли Ю.М. Коган, Д.А. Арзамасцев, А.А. Бесчинский, А.З. Гамм, Б.И. Кудрин, А.М. Меламед, А.С. Некрасов, В.Ф. Тимченко, С.А. Совалов, Л.А. Мелентьев, Т.А. Филлипова, Г.П. Шумилова, П.И. Головкин и др.

Вычисление прогноза нагрузки относится к тем задачам, в которых задача поиска зависимости между входными и выходными данными нетривиальна, а нахождение зависимостей и закономерностей в больших объёмах данных требует трудоёмких и неочевидных алгоритмов и занимает достаточно большое количество времени. В настоящий момент имеется множество методик

прогнозирования. Однако сегодня наибольшее распространение получили информационные технологии (ИТ), которые в свою очередь связаны с использованием искусственных нейронных сетей (ИНС).

В наши дни нейронные сети считаются перспективной разработкой и являются альтернативой традиционным методом решения нелинейных задач вычисления прогноза временных рядов. Преимущество использования нейронных сетевых алгоритмов обусловлено простой и эффективной реализацией вычислений и возможности применения как при непрерывном, так и при дискретном характере переменных.

Использование ИНС вызвано тем, что нейронные сети позволяют воспроизводить крайне сложные зависимости, которые сопровождают плохо формализованным задачам. Также выбор в пользу ИНС вместо традиционных моделей обусловлено тем, что не нужно использовать построение модели объекта, и не происходит потери работоспособности при недостаточной полноте входной информации.

В сравнении с линейными методами статистики нейронные сети осуществляют эффективное построение нелинейных зависимостей, которые более точно описывают наборы данных.

Осуществление прогноза режимных параметров и экономических показателей является важной задачей при планировании и ведении режимов электроэнергетических систем. Большой эффект приносит уточнение краткосрочного прогнозирования графиков потребления мощности.

Таким образом, актуальным на сегодня является совершенствование анализа и прогнозирования электрической мощности на основе искусственных нейронных сетей, а также

уменьшение ошибки прогнозирования при использовании небольшого объема информации.

**Цели и задачи работы.** Целью диссертационной работы является моделирование источника питания контактной сварки и последующего исследования влияния структуры искусственной нейронной сети и параметров её обучения на точность моделирования, а также оценка возможности применения методов ИНС для снижения технико-экономических затрат в условиях современного энергетического рынка. На основе использования искусственного интеллекта возможно прогнозирования более точного значения переменных.

Для достижения цели решены следующие этапы исследования:

- 1) Было проведено исследование структуры источника питания однофазной контактной машины переменного тока.
- 2) Произведен обзор способов моделирования электрических нагрузок с помощью искусственных нейронных сетей.
- 3) Составлена модель резистора и RL-цепи, работающая на постоянном и переменном напряжении.
- 4) Выполнен анализ влияния структуры нейронной сети на точность при моделировании RL-ветви.
- 5) Получена модель тиристорного трансформатора контактной машины на основе нейронной сети.
- 6) Проанализировано влияние параметров обучения и структуры нейронной сети на точность при моделировании источника питания контактной машины.

**Методы исследований.** При решении поставленных задач были использованы методы математического моделирования и

прогнозирования, теория искусственных нейронных сетей, а также математические пакеты MATLAB и Neural Network Toolbox.

**Научная новизна.** Научная новизна выпускной квалификационной работы заключается в том, что предложены к внедрению технология моделирования и прогнозирования аварийных электрических нагрузок, позволяющая экономически и технологически выгодно произвести планирование и разработки проектов, где требуется максимальная точность и экономичность в разработке проектов.

**Практическая значимость.** Выполненный анализ, а также разработанные модели исследования электрической нагрузки с использованием нейронных сетевых моделей для цели краткосрочного прогнозирования нагрузки электрических аппаратов могут использоваться в проектных и исследовательских организациях.

Предложенные модели нейронного сетевого прогноза являются более эффективными при прогнозе параметров электропотребления по сравнению с методом использования математического аппарата дифференциального исчисления.

## 1 Знакомство с объектом исследования

Чтобы получить достаточно качественное сварное соединение, требуется то, чтобы источники питания дуги должны обладать свойствами, которые требуются для процесса сварки, которые, в свою очередь, являются проявляющимися при высоких технико-экономических показателях. С другой же стороны, как всякое электротехническое устройство, источник питания изначально должен быть рассчитан на конкретные режимы работы, то есть, иными словами, на определенную нагрузку и определенные условия эксплуатации, при которых все его свойства будут проявляться оптимально.

Технологические требования задаются техническими потенциалами достижения технологических свойств, которые обуславливаются, в свою очередь, статическими и динамическими свойствами источника питания.

Статические свойства источника воссозданы в его внешней статической характеристике и в ее соответствии вольтамперной характеристике сварочной дуги, так как способность энергетической системы «источник — дуга — ванна» поддерживать устойчивое горение дуги и заданный режим зависит от видов и взаимного расположения этих характеристик.

О динамических свойствах источника следует судить по характеру и скорости протекания переходных процессов в системе «источник — дуга — ванна». Они сопровождаются резкими изменениями сварочного тока при ступенчатых изменениях проводимости разрядного промежутка, а также вызываются резкими переходами из одного установившегося режима в другой. Более того, при сварке

могут замечаться сравнительно небольшие колебания напряжения на дуге и тока дуги, которые вызываются процессами в разрядном промежутке.

Электротехническая установка, в том числе и источник питания для сварки, рассчитывается на установленную нагрузку, при которой она работает. При этом предполагается, что она работает без перегрева выше установленных норм, то есть рассчитывается по нагреву на определенный режим работы. Это определяется характером изменения нагрузки по времени  $P = f(t)$ .

Для ручной сварки ответственных соединений применяют электроды. Данные электроды обеспечивают необходимое качество и достаточно высокую устойчивость процесса сварки только на постоянном токе. Здесь надлежит отметить, что устойчивость процесса характеризуется таким отрицательным явлением, как так называемым магнитным дутьем. Оно является следствием взаимодействия собственного магнитного поля дуги и поля сварочного контура. Магнитное дутье ограничивает применение высокопроизводительных режимов при сварке на постоянном токе.

Самое простое и надежное средство борьбы с данным явлением при процессе сварке — это применение переменного тока. В данном случае электромагнитная сила, которая действует на сварочную дугу, определяется результирующим магнитным потоком, равным геометрической сумме магнитных потоков сварочного и вихревых токов, индуцируемых в массе основного металла. Так как составляющие этих потоков находятся почти в противофазе, то результирующее значение их невелико, поэтому появление магнитного дутья значительно ослабляется по сравнению с постоянным током. Таким образом, в случае использования переменного тока можно

применить более производительные режимы сварки. Известны и другие преимущества переменного тока. Например, сварка в этом случае значительно экономичнее с точки зрения расхода электроэнергии: КПД сварочных трансформаторов, как правило, выше, чем у источников питания постоянного тока. Сварочные трансформаторы имеют простое устройство, требуют минимального ухода и текущего ремонта, поэтому расходы на их эксплуатацию и обслуживания сравнительно невелики.

Существует два принципиально отличных пути создания источников переменного тока для ручной дуговой сварки плавящимся электродом. Первый — на основе трансформатора с падающей внешней характеристикой, которая обеспечивается созданием повышенных магнитных полей рассеяния. Такие трансформаторы имеют только механическое регулирование.

Второй — это на основе трансформатора с жесткой внешней характеристикой. Падающая характеристика источника обеспечивается в этом случае дополнительным включением в цепь питания дуги катушки с ферромагнитным сердечником — дросселем (т.е. большого индуктивного сопротивления) или изменением проводимости магнитопровода. Эти трансформаторы могут иметь как механическое, так и электронное регулирование сварочного тока.

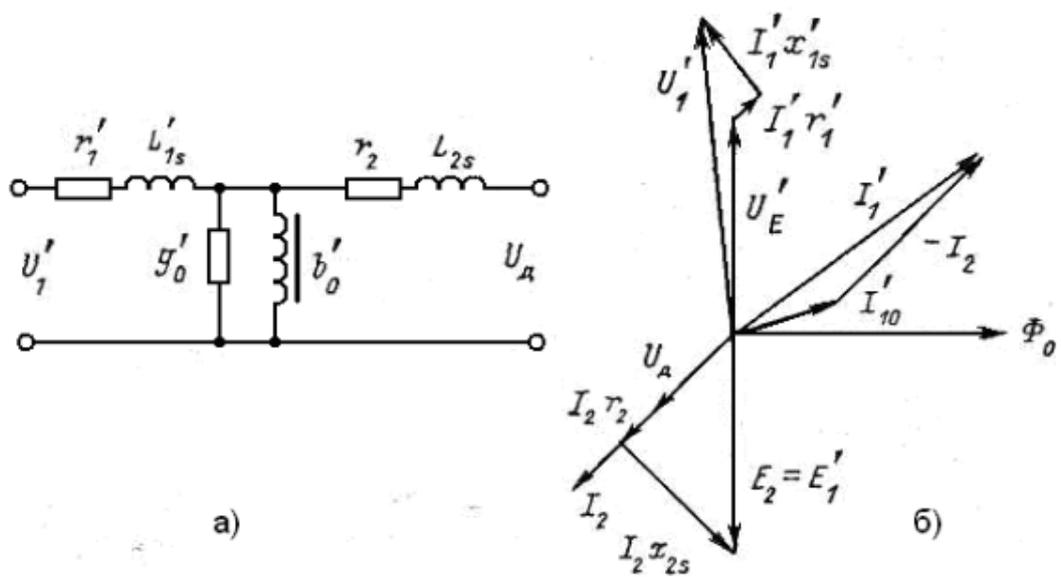


Рисунок 1 — Схема замещения (а) и векторная диаграмма (б) трансформатора с повышенным рассеянием

## 1.1 Трансформаторы для дуговой и контактной сварки с электронным регулированием

Схема простейшего тиристорного трансформатора прерывистого тока представлена на рисунке 2.

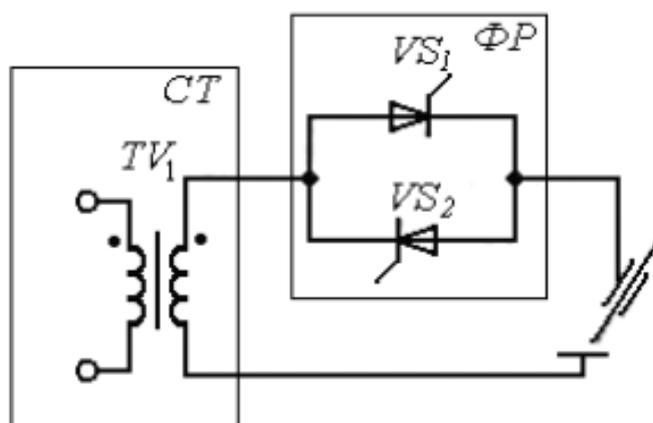


Рисунок 2 — Схема тиристорного трансформатора: СТ — силовой трансформатор; ФР — фазовый регулятор

Основными элементами этой схемы являются фазовый регулятор на тиристорах и силовой трансформатор. Способ фазового регулирования основан на преобразовании синусоидального тока в знакопеременные импульсы. Длительность и амплитуда импульсов определяются углом включения тиристоров. Способ управления при дуговой сварке считался неприемлемым, поскольку в паузах между импульсами тока происходит быстрая деионизация дугового промежутка, затрудняющая повторные возбуждения дуги.

Особенно большие паузы имеют место в устройствах, близких к идеальным, где используется силовой трансформатор с нормальным магнитным рассеянием.

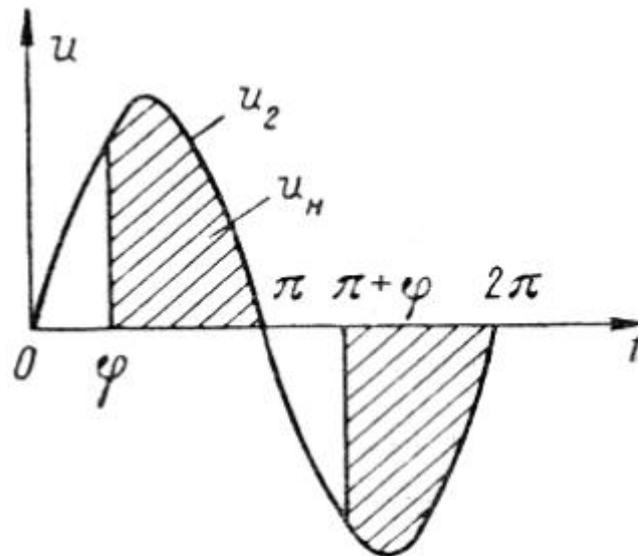


Рисунок 3 — Диаграмма напряжений

Заштрихованная часть - напряжение осциллограмм приложено к нагрузке, незаштрихованная — к запертым тиристорам. Среднее (за полупериод) действующее значение напряжения на нагрузке определяются следующим соотношением:

$$U_{н.ср.} = \frac{\sqrt{2} \times U_2}{\pi} \times (1 + \cos \varphi), \text{ В,}$$

Среднеквадратичное напряжение при нагрузке

$$U_{н.д.} = U_2 \times \sqrt{\frac{\pi - \varphi + \sin 2\varphi / 2}{\pi}}, \text{ В,}$$

коэффициент формы

$$k_\varphi = \frac{U_{н.д.}}{U_{н.ср.}} = \frac{1}{1 + \cos \varphi} \sqrt{\frac{\pi}{2} \times \left( \pi - \varphi - \frac{\sin 2\varphi}{2} \right)},$$

активная мощность:

$$P = I_{2ср} \times U_{н.ср.}, \text{ Вт.}$$

## 1.2 Временные диаграммы работа сварочного источника питания

Для сварки применяются специальные схемы и конструкции с использованием трансформаторов: тиристорные трансформаторы. Отличительной особенностью этих устройств является наличие силового трансформатора с развитым электромагнитным рассеянием или дополнительного дросселя в сварочной цепи.

Силовые схемы тиристорного трансформатора делятся: способу обеспечения непрерывности процесса сварки и месту установки тиристорного фазорегулятора — в первичной или вторичной цепи.

По первому признаку тиристорные трансформаторы делятся на две группы. В трансформаторах одной группы стабильность процесса сварки обеспечена непрерывностью сварочного тока (рис. 4, а). Для заполнения пауз с отсутствием токов в интервалы непроводимости тиристоров использована специальная цепь, называемая цепью подпитки дуги, по которой проходит минимально необходимый для устойчивого процесса сварки ток (ток подпитки  $i_{20}$ ). Сдвиг между током  $i_{20}$  и напряжением  $u_{20}$  здесь определяется углом  $\psi$ , зависящим от соотношения напряжений дуги и цепи подпитки:

$$\cos\psi = \frac{\pi \times U_d}{2 \times U'_{20m}}$$

где  $U'_{20m}$  — амплитудное значение напряжения цепи подпитки.

В трансформаторах другой группы сварочный ток  $i_2$  прерывистый (рис. 4, б).

Длительность протекания тока определяется углом включения силовых тиристоров, который здесь является также углом сдвига между током и напряжением холостого хода трансформатора. Для

надежного повторного возбуждения применена импульсная стабилизация горения дуги. В момент окончания паузы с отсутствием тока на дугу накладывается стабилизирующий импульс тока.

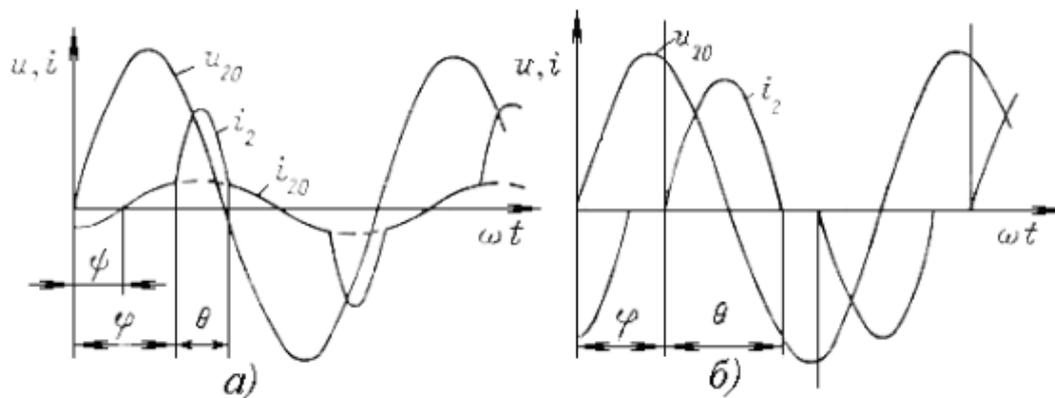


Рисунок 4 — Диаграммы напряжений и токов трансформаторов: а) с цепью подпитки; б) с импульсной стабилизацией

Фазовый регулятор совмещает функции управления значением тока и коммутирующего элемента импульсного стабилизатора горения дуги.

В промышленных сварочных установках наибольшее распространение получили тиристорные трансформаторы с прерывистым регулированием тока и импульсной стабилизацией, так как применение цепи подпитки связано с дополнительным расходом активных материалов, с применением дополнительного коммутатора цепи подпитки для снятия напряжения холостого хода, а относительно низкая скорость нарастания тока подпитки при смене полярности обуславливает невысокие сварочные качества источников.

Размещение тиристорного фазового регулятора в первичной или вторичной (сварочной) цепи трансформатора во многом обуславливает конструкцию тиристорного трансформатора. Во всех случаях, когда не требуется управлять значением постоянной составляющей сварочного тока, предпочтительнее включение тиристора в сравнительно

слаботочную цепь первичной обмотки трансформатора. Кроме снижения габаритов регулятора и потерь в тиристорах, такое решение позволяет уменьшить или исключить потери холостого хода трансформатора. Также это обеспечивает оперативное отключение его от питающей сети. Тем не менее при включении тиристорov в первичную цепь трансформатора становится невозможным регулирование (компенсирование) постоянной составляющей сварочного тока. Для этого в трансформаторах с цепью подпитки уменьшается диапазон регулирования сварочного тока, повышаются требования к симметрии импульсов управления тиристорами, а также к надежности и электрической прочности управления.

### 1.3 Схемы источников питания для сварки с прерывистым током

На рисунке 5 приведены схемы сварочных трансформаторов.

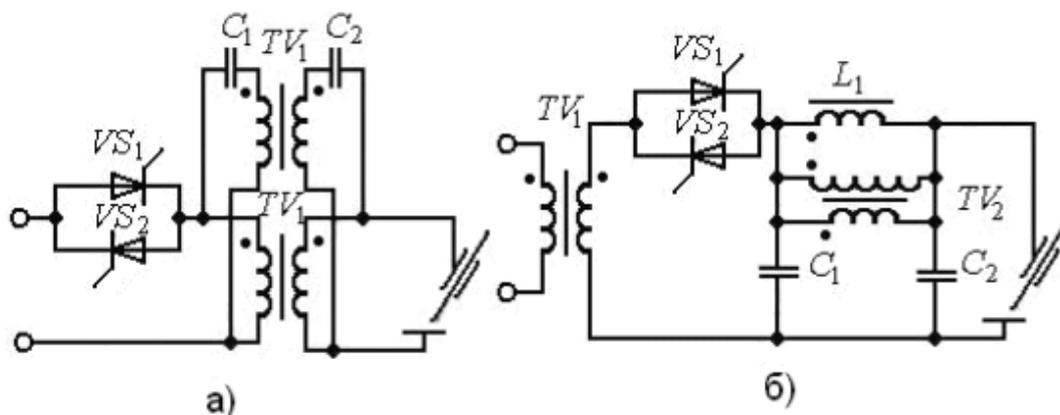


Рисунок 5 — Трансформаторы с прерывистым питанием дуги

Параметры импульса определяются соотношением витков импульсной и вторичной обмоток и емкостью конденсатора. Амплитуда импульса зависит от угла включения тиристоров, однако незначительно, поскольку в трансформаторе с кратностью регулирования сварочного тока около двух, угол включения тиристоров изменяется в пределах 20-60 градусов. Это соответствует колебаниям амплитуды импульса приблизительно на 14% от максимального значения.

В тех случаях, когда выполнение дополнительной импульсной обмотки почему-либо нежелательно, может быть использован отдельный импульсный трансформатор, вторичная обмотка которого включена через разделительный конденсатор параллельно вторичной обмотке сварочного трансформатора.

Принцип импульсной стабилизации дуги прерывистого переменного тока может быть реализован и при установке фазового регулятора во вторичной цепи сварочного трансформатора. На рисунке 5, б дана схема тиристорного трансформатора с устройством генерации

высоковольтных импульсов, обеспечивающих первоначальное зажигание и повторное возбуждение дуги. В момент включения тиристора зарядный ток конденсатора  $C_2$  наводит во вторичной обмотке высоковольтного трансформатора импульс высокого напряжения, достаточный для пробоя межэлектродного промежутка. Конденсатор фильтра  $C_1$  защищает источник питания от перенапряжений.

Принцип импульсной стабилизации дуги прерывистого переменного тока может быть реализован и при установке тиристорного фазорегулятора во вторичной цепи сварочного трансформатора. Данная схема (рисунок 5, б) тиристорного трансформатора имеет устройство генерации высоковольтных импульсов, обеспечивающих первоначальное зажигание дуги. В момент включения тиристора зарядный ток конденсатора  $C$  наводит во вторичной обмотке высоковольтного трансформатора импульс высокого напряжения, достаточный для пробоя межэлектродного промежутка. Конденсатор фильтра  $C_\phi$  защищает источник питания от перенапряжений.

На рисунке 6 приведены внешние характеристики тиристорных трансформаторов с цепью подпитки и без нее для углов включения от 0 до 120 градусов.

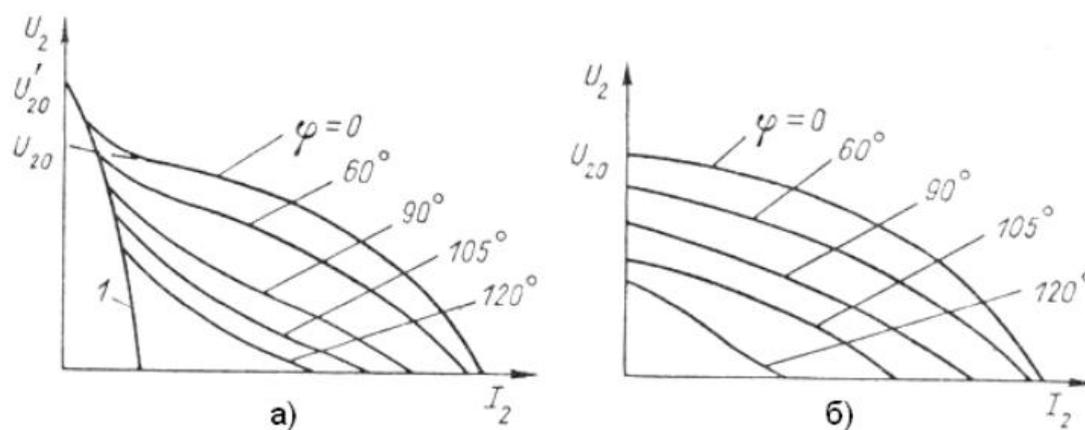


Рисунок 6 — Внешние характеристики трансформаторов: а) с цепью подпитки; б) с импульсной стабилизацией дуги.

Как видно на рисунке 6, а напряжение холостого хода трансформатора с цепью подпитки зависит от напряжения подпитки и не зависит от угла отпирания тиристоров, а при импульсной стабилизации горения дуги (рисунок 6, б) изменение угла отпирания влияет на напряжение холостого хода источника питания.

На это закончим рассмотрение устройства тиристорного трансформатора и перейдем к обзору способов моделирования электрических нагрузок с помощью искусственных нейронных сетей.

## 2 Обзор способов моделирования электрических нагрузок с помощью искусственных нейронных сетей

### 2.1 Структура искусственной нейронной сети

Искусственная нейронная сеть — это математическая модель, в том числе её программная или аппаратная реализация, сформированная по принципу функционирования биологических нейронных сетей, то есть сетей нервных клеток живых организмов. Данное понятие появилось при изучении процессов, которые протекают в мозге, и последующих попытка промоделировать данные процессы. После разработки алгоритмов данные модели стало возможно использовать в множестве практических целях, например, в задачах прогнозирования, управления, распознавания образов и т.п.

Нейронные сети активно применяются в сегодняшнем мире, например, в сферах техники, бизнеса, медицине, геологии, физике. Это связано с тем, что нейронные сети имеют богатые возможности, т.к. являются очень мощным инструментом моделирования, который позволяет воссоздавать чрезвычайно сложные зависимости. Нейронные сети по своей сути нелинейны. На протяжении много времени линейное моделирование было основным методом во множестве областей, т.к. для него были хорошо разработаны различные процедуры оптимизации. Но есть задачи, в которых линейная аппроксимация совершенно неудовлетворительна, и из-за этого линейные модели работают плохо. Кроме того, нейронные сети справляются с задачами при большом числе переменных, при которых линейные зависимости не позволяют моделировать.

Также преимущество нейронных сетей состоит в том, что они учатся на примерах. Пользователь может подбирать данные, а затем

запускает алгоритм обучения, на которых нейронная сеть автоматически воспринимает структуру данных. От пользователя требуются знания о том, как нужно выбрать и подготавливать данные, а также уметь выбирать нужную архитектуру нейронной сети и интерпретировать полученные результаты. Однако уровень знаний, которые необходимы для применения нейронных сетей, гораздо проще, чем при использовании традиционных методов статистики.

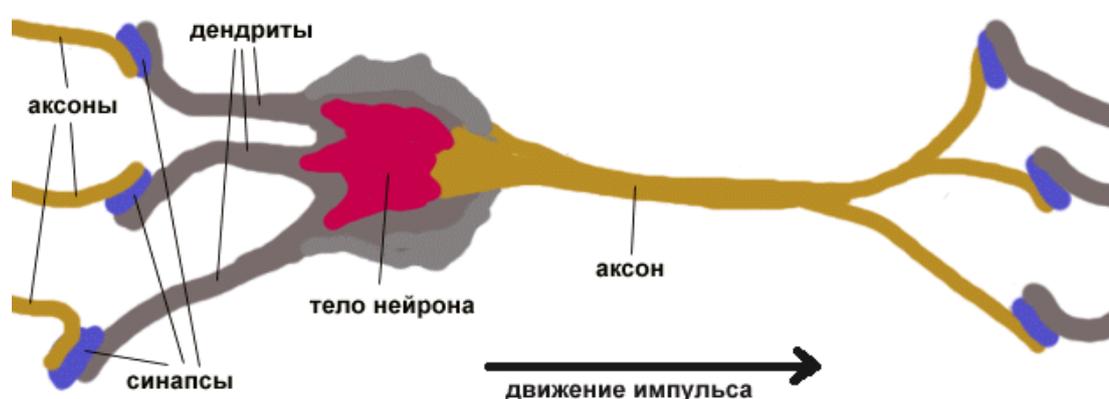


Рисунок 7 — Структура биологической единицы (нейрона).

Мозг живого организма состоит из огромного количества нейронов, которые в свою очередь соединены множеством связей. Нейрон — это клетка, которая имеет способность распространять электрохимические сигналы. Он имеет разветвленную структуру, которая состоит из ввода информации (дендриты), ядро и разветвляющийся выход (аксон). В свою очередь аксоны клетки соединяются с дендритами других клеток при помощи т.н. синапсов. Каждый входной сигнал проходит через соединение, которое имеет определенную интенсивность, т.е. вес. Вес соответствует синаптической активности биологического нейрона. При активации нейрон отправляет сигнал по своему аксону. Через синапсы он

достигает другие нейроны, которые также активируются. Сам нейрон активируется тогда, когда суммарный уровень сигналов, которые пришли в его ядро из дендритов, превысит определенный уровень, т.е. порог активации. Интенсивность сигнала сильно зависит от активности синапсов. Если использовать ступенчатую функцию активации, иными словами выход нейрона равняется нулю, если вход имеет отрицательное значение, и единице, если нейрон имеет нулевое или положительное значение, то такой нейрон будет работать в точности, как описанный выше естественный нейрон. Стоит учесть, что веса могут быть также отрицательными, то есть синапс будет оказывать на нейрон не возбуждающее, а тормозящее действие.

Таким образом, мозг, будучи построенным из очень большого количества простых элементов, где каждый из которых берет сумму входных сигналов, и, если суммарный вход превышает порог активации, передает далее двоичный сигнал, способен решать сложные задачи.

Известным подходом является представление электрической нагрузки как черный ящик, внутренняя структура которого полностью неизвестна. В этом случае математически описывается зависимость тока нагрузки от приложенного к ней напряжения, времени и других параметров. В программах имитационного моделирования нагрузка может быть представлена идеальным управляемым источником тока, мгновенная величина тока которого математически вычисляется на каждом шаге работы модели (рисунок 8).

Структурная схема моделирования нагрузки идеальным источником тока

Основной задачей такого подхода является нахождение выражения для описания мгновенного значения тока. Выражение для

тока конкретной нагрузки может быть получено с помощью расчёта эквивалентной схемы замещения с учетом допущений. Другим способом, описанном в, является расчет ряда коэффициентов (сопротивлений) для нагрузки по известным временным кривым напряжений и токов (например, осциллограммам работы устройства) с последующим их использованием для расчёта мгновенного тока для произвольного питающего напряжения. Такой способ обладает высокой способностью к автоматизации с помощью ЭВМ: компьютерная программа может автоматически по осциллограммам с помощью известных выражений рассчитать коэффициенты, а затем использовать их в известной формуле расчета мгновенного тока имитационной модели. Далее в программе формируется блок нагрузки, который разработчик подключает к источнику питающего напряжения, другим нагрузкам и выполняется моделирование и анализ системы в целом. Однако способ применим только к стационарным (значения тока при одинаковом напряжении в разные моменты времени должны быть также одинаковыми) периодическим пассивным нагрузкам, неприменим при расчете переходных процессов включения и выключения устройств. Большинство устройств, применяемых сегодня в автомобилестроении, авиастроении являются нестационарными и непериодическими электрическими нагрузками.

Задача автоматизированного создания компьютерных имитаций электрических нагрузок путем нахождения выражений для мгновенных значений тока с учетом разнообразных параметров, в том числе и неэлектрических, является актуальной и важной. А учитывая возрастающую степень внедрения сложного электрооборудования, она станет еще актуальней в будущем.

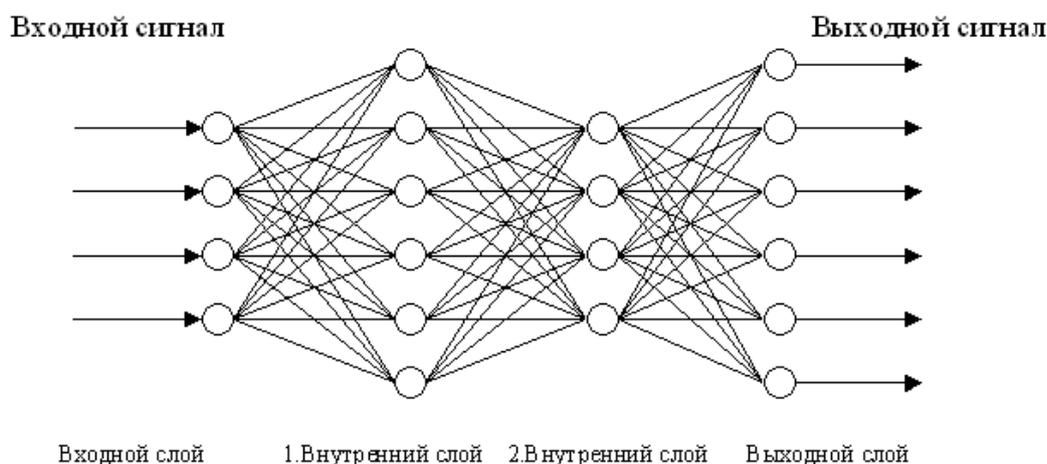
Математическим аппаратом, позволяющим автоматизировано найти функциональную зависимость любой сложности между большим количеством параметров, используя известные экспериментальные данные, является аппарат искусственных нейронных сетей (ИНС). Сегодня нейронные сети используются для распознавания речи и визуальных образов, в экономике и медицине для обнаружения скрытых зависимостей и прогнозирования временных рядов. Представляется возможным использовать нейронную сеть, обученную по экспериментальным данным (например, осциллограммы работы в разных режимах), для расчета мгновенных значений тока при воздействии произвольного напряжения в установившихся и переходных режимах с учетом неэлектрических параметров (например, скорости вращения руля и обороты двигателя при анализе работы электрического усилителя руля). Правильно обученная сеть может вычислять ток произвольной электрической нагрузки в любом режиме. Очевидными недостатками такого способа по сравнению с аналитическими способами являются огромные требования к вычислительным ресурсам даже по современным меркам, необходимым при обучении сети, и сложность анализа адекватности результатов моделирования.

Рассмотрим создание модели электрической нагрузки в программе MATLAB Simulink, расчет тока которой осуществляется с помощью искусственной нейронной сети.

В данном случае было описано действие отдельного нейрона. Самая простая сеть имеет структуру прямой передачи сигнала, т.е. сигналы проходят от входов через скрытые элементы и проходят на выходные элементы. Данная структура имеет устойчивое поведение. Если сеть рекуррентная, то есть содержит связи, которые ведут назад

от более дальним к более ближним нейронам, то она будет неустойчивой и будет иметь сложную динамику поведения. Данный тип сегодня представляет большой интерес в области исследования искусственных нейронных сетей, но на сегодняшний день оказываются полезными структуры прямой передачи. Такой тип нейронных сетей моделируется в пакете MATLAB Neural Network Toolbox.

А):



Б):

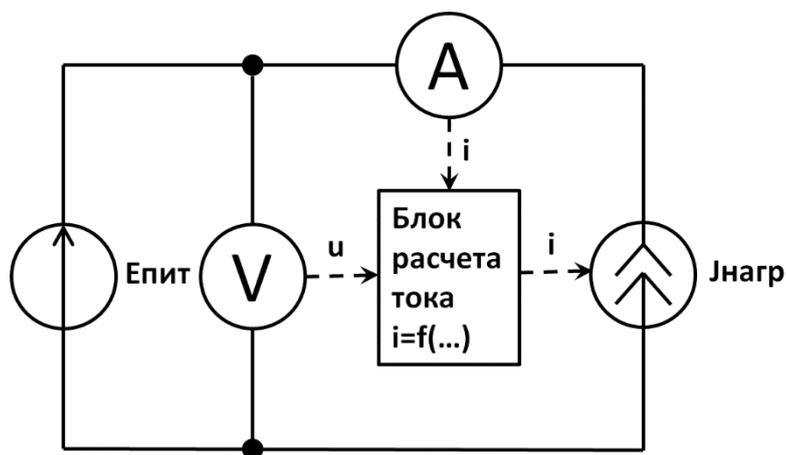


Рисунок 8 — Сеть с прямой передачей данных сигнала

Пример сети с прямой передачей сигнала изображен на рисунке 8. Нейроны организованы в слой. Входной слой служит для ввода

значений входных переменных. Каждый скрытый и выходной нейрон связан со всеми элементами предыдущего слоя. При использовании сети во входные элементы вводятся значения входных переменных, после этого последовательно обрабатываются нейроны промежуточных и выходных слоев. Каждый нейрон вычисляет значение своей активации, беря взвешенную сумму выходов элементов предыдущего слоя, при этом вычитая из нее пороговое значение. После этого значение активации преобразуется с помощью функции активации, в результате чего получается выход нейрона. После отработки всей сети, выходные значения элементов выходного слоя принимаются за выход всей сети в целом.

Для обучения искусственных нейронных сетей используются алгоритмы двух типов — это управляемое обучение, т.н. «обучение с учителем» и неуправляемое обучение, т.е. «обучение без учителя». Чаще всего на практике применяется обучение с учителем.

Для управляемого обучения пользователю требуется самостоятельно подготовить набор требуемых обучающих данных. Они являют собой входные данные и соответствующие им выходы. Сама сеть устанавливается связь между первыми и вторыми. Затем искусственная нейронная сеть обучается с помощью алгоритма управляемого обучения, при котором используются данные для корректировки весов сети и ее пороговые значения для того, чтобы максимально уменьшить ошибку прогноза на обучающем множестве. Если же нейронная сеть успешно обучается, то следом она приобретает возможность моделировать функцию, которая связывает значения входных и выходных переменных, а впоследствии эту сеть можно использовать для прогнозирования ситуации, когда выходные значения неизвестны.

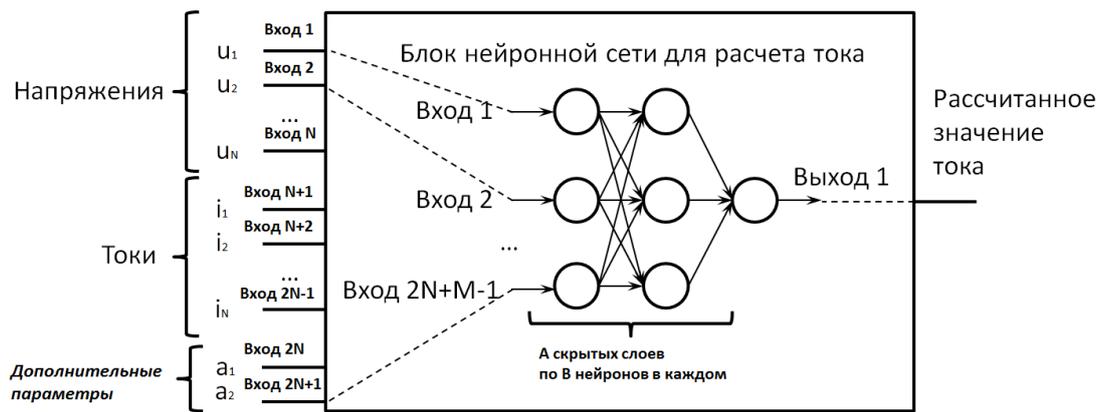


Рисунок 9 — Структура применяемой нейронной сети

## 2.2 Моделирование с помощью алгоритма обратного распространения ошибки

Алгоритм обратного распространения ошибки — это метод обучения многослойного персептрона. Впервые метод был описан в 1974 г. А.И. Галушкиным, а также Полом Дж. Вербосом. Далее был существенно развит в 1986 г. Дэвидом И. Румельхартом, Дж. Е. Хинтоном и Рональдом Дж. Вильямсом, С.И. Барцевым и В.А. Охониным.

Особенности построения искусственных нейронных сетей наводят исследователей на идею использования все более сложных иерархических структур. Сама идея состоит в том, что на низших уровнях иерархии классы преобразуются таким образом, что формируются линейно разделимые множества, которые в свою очередь будут успешно распознаваться нейронами на следующих, то есть высших уровнях иерархии сети.

Однако же основной проблемой, которая традиционно ограничивает возможности сетевой топологии простейшими структурами, является проблема обучения. На этапе обучения сети предъявляются входные образы, которые называются обучающей выборкой, и далее исследуются получающиеся выходные реакции. Сама же цель обучения состоит в том, чтобы привести наблюдаемые реакции на заданной выборке к требуемым адекватным реакциям с помощью изменения состояний синаптических связей. Сама же сеть является обученной, если в ней все реакции на заданном наборе стимулов являются адекватными. В данном случае классическая схема обучения с учителем требует конкретного знания ошибок при функционировании каждого нейрона, что, естественно, является

затруднительным для иерархических систем, в которых непосредственно контролируются только входы и выходы сети. Кроме этого, необходимая избыточность в иерархических сетях может привести к тому, что состояние обучения сети может быть реализована многими способами, что само по себе делает понятие «ошибка, делаемая данным нейроном» весьма и весьма неопределенным.

Для обучения многослойной сети в 1986 году Руммельхартом, Макклеландом, Вильямсом и Хинтоном был предложен алгоритм обратного распространения ошибок. Многочисленные публикации о применениях многослойных нейронных сетей с этим алгоритмом обучения подтвердили его принципиальную работоспособность на практике. Это ознаменовало собой повторное возрождение интереса к искусственным нейронным сетям, который стал угасать в 70-х годах. Позже было обнаружено то, что Паркер опубликовал похожие результаты в 1982 году, а Вербос выполнил подобную работу в 1984 году. Однако же так сложилось в науке, что ученые, которые работали независимо друг от друга, не смогли использовать все то прогрессивное, что есть в других областях, поэтому во многих случаях получается повторение уже достигнутого. Однако статья Руммельхарта и др. до сих пор является наиболее цитируемой в данной области.

Изначально встает вопрос, почему же для обучения многослойного персептрона невозможно применить d-правило Розенблатта, которое состоит в методе коррекции с обратной передачей сигнала ошибки? Ответ заключается в том, что для корректного применения данного метода необходимо знать не только лишь текущие выходы нейронов  $y$ , но и требуемые правильные значения  $Y$ . В случае же многослойной нейронной сети данные правильные значения имеются только лишь для нейронов выходного

слоя. Требуемые значения выходов для нейронов скрытых слоев неизвестны, что и ограничивает применение d-правила.

Основной смысл метода обратного распространения состоит как раз в том, чтобы получить оценку ошибки для нейронов скрытых слоев. Стоит отметить, что известные ошибки, которые выполняют нейроны выходного слоя, происходят вследствие же неизвестных пока ещё ошибок нейронов скрытых слоев. Чем больше само значение синаптической связи между нейроном скрытого слоя и выходным нейроном, тем сильнее ошибка первого влияет на ошибку второго. Поэтому оценку ошибки элементов скрытых слоев можно получить в качестве взвешенной суммы ошибок следующих слоев. При обучении же информация распространяется от низших слоев данной иерархии к высшим слоям, а оценки ошибок, которая делает сеть, в обратном направлении, отсюда и происходит название данного метода.

В данной работе был использован данный алгоритм. В нем вычисляется вектор градиента поверхности ошибок. Он указывает направление самого короткого спуска по поверхности из данной точки.

Если имеется большая длина шага, то сходимость будет более быстрой. Однако есть опасность того, что можно перепрыгнуть через решение при условии, что поверхности ошибок имеет особо сложную форму или уйти в неправильном направлении. Но при маленьком шаге, возможно, будет найдено верно направление, но при этом потребуется большое количество итераций. На практике величина шага берется пропорциональной крутизне склона с некоторой постоянной, которая называется скоростью обучения. Ее правильный выбор зависит от конкретной задачи и определяется опытным путем, т.к. она зависит от времени, уменьшая по мере продвижения алгоритма.

Данный алгоритм обычно изменяется так, чтобы включать слагаемое импульса. Данный импульс заставляет двигаться в фиксированном направлении, поэтому алгоритм ускоряется, при условии того, что было сделано несколько шагов в одном и том же направлении, что позволяет избежать локального минимума и быстрее проходить плоские участки.

Данный алгоритм действует итеративно, поэтому его шаги называются эпохами. На каждой такой эпохе на вход сети поочередно подаются все обучающие наблюдения. При этом выходные значения сети сравниваются с целевыми, а после этого вычисляется ошибка. Значение этой ошибки, а также градиента поверхности ошибки используется для корректировки весов сети, после этого все действия повторяются. Начальная конфигурация нейронной сети выбирается произвольным путем в том случае, когда ошибка достигнет некоторого определенного минимального уровня, либо же когда она перестанет уменьшаться.

На основе алгоритма обратного распространения ошибки были основаны три метода, которые могут быть использованы для обучения искусственной нейронной сети, которые, в свою очередь, были включены в программный пакет MATLAB. Это метод Левенберга-Маркара, алгоритм Байесовской регуляризации и масштабируемый алгоритм сопряженных градиентов. Рассмотрим каждый метод и алгоритм по отдельности.

## 2.3 Метод Левенберга-Маркара

Данный метод предназначен для оптимизации параметров нелинейных регрессионных моделей. В данном случае предполагается, что в качестве критерия оптимизации используется среднеквадратичная ошибка модели на обучающей выборке. Метод заключается в том, что имеется последовательное приближение заданных начальных значений параметров к исходному локальному оптимуму. Метод считается наиболее быстрым и надежным алгоритмом обучения. Стоит отметить, что его применение связано с определенными ограничениями:

- 1) Метод Левенберга-Маркара можно применять только для сетей с одним выходным элементом.
- 2) Небольшие сети. Метод Левенберга-Маркара требует памяти, которая является пропорциональной квадрату числа весов в сети.
- 3) Среднеквадратичная функция ошибок. Метод Левенберга-Маркара применим только для среднеквадратичной функции ошибок. Поэтому данный метод обычно подходит только для регрессионных сетей.

Этот метод отличается от метода сопряженных градиентов тем, что в нем для модели используется матрица Якоби, а не градиент вектора параметров. В отличие от алгоритма Гаусса-Ньютона, этот метод использует параметр регуляризации. Рассмотрим данный метод.

Для постановки задачи задается регрессионная выборка, которая состоит из множества пар  $D = \{(x_n, y_n)\}_{n=1}^N$ , свободной переменной  $x \in R^M$  и зависимой переменной  $y \in R$ . Также задана регрессионная

модель, т.е. функция  $f(w, x_n)$ , которая является непрерывно дифференцируемой в области  $W \times X$ .

В данном случае требуется найти такое значение вектора параметров  $w$ , которое бы доставляло локальный минимум функции ошибки

$$E_D = \sum_{n=1}^N (y_n - f(w, x_n))^2$$

Перейдем к описанию алгоритма. Перед началом его работы задается начальный вектор параметров  $w$ . На каждом шаге итерации этот вектор заменяется на  $w + \Delta w$ . Для оценки приращения  $\Delta w$  используется линейное приближение функции

$$f(w + \Delta w, x) \approx f(w, x) + J\Delta w,$$

в которой  $J$  — это якобиан функции  $f(w, x_n)$  в точке  $w$ .  $(N \times R)$ -матрицу  $J$  отлично можно представить в виде

$$J = \begin{bmatrix} \frac{\partial f(w, x_1)}{\partial \omega_1} & \dots & \frac{\partial f(w, x_1)}{\partial \omega_R} \\ \dots & \dots & \dots \\ \frac{\partial f(w, x_N)}{\partial \omega_1} & \dots & \frac{\partial f(w, x_N)}{\partial \omega_R} \end{bmatrix}$$

В данном случае вектор параметров  $w = [w_1, \dots, w_R]^T$ .

Приращение  $\Delta w$  в точке  $w$ , доставляющий минимум  $E_D$  равно нулю. Поэтому для нахождения последующего значения  $\Delta w$  приравнивается нулю вектор частных производных  $E_D$  по  $w$ . Для этого  $E_D = \sum_{n=1}^N (y_n - f(w, x_n))^2$  представляется в виде

$$E_D = |y - f(w + \Delta w)|^2,$$

где  $y = [y_1, \dots, y_n]^T$  и

$$f(w + \Delta w) = [f(w + \Delta w, x_1), \dots, f(w + \Delta w, x_N)]^T.$$

Преобразовывая это выражение

$$\begin{aligned} |y - f(w + \Delta w)|^2 &= (y - f(w + \Delta w))^T (y - f(w + \Delta w)) \\ &= f^T(w + \Delta w)f(w + \Delta w) - 2y^T f(w + \Delta w) + y^T y \end{aligned}$$

и дифференцируя, то получим

$$\frac{\partial E_D}{\partial w} = (J^T J)\Delta w - J^T(y - f(w)) = 0.$$

Таким образом, для того, чтобы найти значение  $\Delta w$  потребуется решить систему линейных уравнений

$$\Delta w = (J^T J)^{-1} J^T (y - f(w)).$$

Так как число обусловленности матрицы  $J^T J$  есть квадрат числа обусловленности матрицы  $J$ , то матрица  $J^T J$  может оказаться существенно вырожденной. Поэтому Маркар предложил ввести параметр регуляризации  $\lambda \geq 0$ .

$$\Delta w = (J^T J + \lambda I)^{-1} J^T (y - f(w)),$$

где  $I$  — это единичная матрица. Данный параметр назначается на каждой итерации этого метода. Если значение ошибки  $E_D$  убывает быстро, то малое значение  $\lambda$  сводит данный метод к алгоритму метода Гаусса-Ньютона. Алгоритм метода останавливается в том случае, если же приращение  $\Delta w$  в последующей итерации меньше заданного значения, либо если параметры  $w$  доставляют ошибку  $E_D$ , которая оказывается меньше заданной величины. Значение вектора  $w$  на последней итерации считается искомым.

Недостатком алгоритма данного метода является то, что имеется значительное увеличение параметра  $\lambda$  при плохой скорости аппроксимации. При этом обращение матрицы  $J^T J + \lambda I$  становится бессмысленным. Этот недостаток можно устранить, при этом используя диагональ матрицы  $J^T J$  в качестве регулязирующего слагаемого:

$$\Delta w = (J^T J + \lambda \text{diag}(J^T J))^{-1} J^T (y - f(w))$$

## 2.4 Алгоритм байесовской регуляризации

Задача аппроксимации скалярной функции многих переменных состоит в отыскании аналитической зависимости  $y = h(x)$ , генерирующей данные, представленные конечным числом точек  $D = \{y_i, x_i\}_{i=1}^L$  в  $(m + 1)$ -мерном пространстве, состоящем из подпространства аргумента  $X$  размерности  $m$  и одномерного подпространства значений функции  $Y$ . Предполагается, что в данных присутствует шум, то есть искомая функция необязательно точно проходит через все заданные точки  $D$ .

Задача аппроксимации функции по сгенерированным ею данным, то есть восстановление причины по следствия, как и большинство обратных задач, является некорректной, то есть имеющей множество решений. Так, если  $h(x)$  — это решение, которое точно проходит через все точки, то для любой функции  $f(x)$  комбинация  $h(x) + f(x) \prod_{i=1}^L (x - x_i)^2$  также будет решением, которое точно проходит через все точки.

Математически задача чаще всего формулируется как минимизация среднеквадратичной ошибки аппроксимации данных по адаптивным параметрам модели, а для обеспечения единственности решения на множестве функций вводится стабилизирующий функционал  $\Omega(x)$ , который отражает степень предпочтительности выбора той или иной функции  $h(x)$ . В результате задача сводится к минимизации ошибки аппроксимации при дополнительных условиях:

$$F = \sum_{i=1}^L (y_i - h(x_i))^2 + \lambda \Omega(h)$$

Данное выражение по сути формулирует задачу поиска компромисса между точностью описания данных (минимизацией первого слагаемого) и априорной предпочтительностью того или иного решения (второго слагаемого). Параметр регуляризации  $\lambda$  в данной формуле определяет, какому из двух факторов отдать большее предпочтение. Однако оптимальное значение этого параметра можно оценить, исходя из данных  $D$ . Для этого можно воспользоваться методом Байеса.

В байесовском подходе выбор того или иного решения  $h$  ведется в рамках некоторой модели  $H$ , задающей ограничения на его вид. Решение выбирается как наиболее вероятное по формуле Байеса:

$$P(h|D, H) = \frac{P(D|h, H)P(h|H)}{P(D|H)}$$

Здесь  $P(h|D, H)$  — это вероятность выбора решения  $h$  для описания данных  $D$  в рамках модели (набора ограничивающих условий)  $H$ ,  $P(D|h, H)$  — это вероятность генерации данных  $D$  решением  $h$ ;  $P(h|H)$  — априорная вероятность выбора  $h$  в рамках модели  $H$ , коэффициент  $P(D|H)$  необходим для того, чтобы нормировать множество значений произведения  $P(D|h, H)P(h|H)$  на единицу, придав ему таким образом смысл вероятности.

Насколько подходит та или иная модель для описания данных определяет знаменатель формулы Байеса:

$$P(D|H) = \sum_h P(D|h, H)P(h|H) = \sum_h P(D, h|H)$$

В литературе он получил название Evidence. Чем он больше, тем соответствующая модель является более предпочтительнее. Действительно, формула Байеса, но уже на уровне моделей

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)}$$

дает возможность сравнивать между собой различные заранее известные ограничения  $H$ . Решение обратной задачи для модели, естественно, требует введения приоритетов уже на их множестве. В отсутствие каких-то предпочтений между несколькими различными наборами, ограничивающих условий  $P(H) = \text{const}$  имеется возможность обоснованно выбрать тот из них, в пользу которого свидетельствуют эмпирические данные — с максимальной Evidence. Выбрав таким образом модель, нужно перейти к поиску решения  $h$  путем максимизации выражения

$$P(h|D, H) = \frac{P(D|h, H)P(h|H)}{P(D|H)}$$

Начнем поиск решения  $h(x)$  в виде ряда по набору функций

$$h(x) = \sum_{n=1}^N a_n \psi_n(x)$$

Предположим, что данные зашумлены гауссовым шумом, тогда вероятность генерации решением  $h(x)$  данных  $D = \{y_i, x_i\}_{i=1}^L$  можно оценить следующим выражением

$$P(D|h) = \frac{1}{Z_x} \exp\left(-\beta \sum_{i=1}^L (y_i - h(x_i))^2\right)$$

Возьмем стабилизирующий функционал в гауссовой форме

$$\Omega(h) = \sum_{n=1}^N a_n^2$$

Ему будет соответствовать априорная вероятность решения

$$P(h|H) = \frac{1}{Z_A} \exp\left(-\alpha \sum_{n=1}^N a_n^2\right)$$

По формуле Байеса вероятность решения  $h(x)$  будет равна

$$P(h|D, H) = \frac{1}{Z_M} e^{-M}$$

$$\text{где } M = \beta \sum_{i=1}^L (y_i - h_i)^2 + \alpha \sum_{n=1}^N a_n^2.$$

Здесь  $Z_X$ ,  $Z_A$ ,  $Z_M$  — это нормировочные коэффициенты, которые получаются из условий нормировки соответствующих вероятностей на единицу. Максимизация вероятности  $P(h|D, H)$  сводится к решению системы линейных алгебраических уравнений (СЛАУ)

$$\sum_m a_m A_{mn} = B_n$$

$$\text{где } A_{mn} = \beta \sum_{i=1}^L \psi_{mi} \psi_{ni}, n \neq m, A_{nn} = \beta \sum_{i=1}^L \psi_{ni}^2 + \alpha,$$

$$B_n = \beta \sum_{i=1}^L y_i \psi_{ni}.$$

Параметры  $\alpha$  и  $\beta$  находятся путем максимизации Evidence или, что тоже самое, его логарифма. Легко доказать, что

$$\ln P(D|H) = \ln Z_M - \ln Z_A - \ln Z_X$$

$$\text{где } Z_A = \int_{-\infty}^{+\infty} \exp(-\alpha \sum_{n=1}^N a_n^2) d^N a_n = \left(\frac{2\pi}{\alpha}\right)^{\frac{N}{2}}, Z_X = \left(\frac{\pi}{\beta}\right)^{\frac{L}{2}},$$

$$Z_M = \int d^N a_n \times \exp(-\beta \sum_{n=1}^N (y - \sum_{n=1}^N a_n \psi_n)^2 - \alpha \sum_{n=1}^N a_n^2).$$

Здесь через  $\psi_n$  обозначены векторы значений  $n$ -й базисной функции в точках  $\{x_i\}_{i=1}^L$ , а через  $y$  — вектор значений искомой функции в этих точках.

Последнее выражение считается приближенно по методу перевала. В результате приходим к следующему приближенному выражению для логарифма Evidence:

$$\ln P(D|H) = \sum_{n=1}^N \left( \frac{(\beta y \psi_n)^2}{A_{nn}} - \frac{1}{2} \ln A_{nn} - \beta \sum_{m \neq n}^N \psi_n \psi_m a_n a_m \right) + \frac{N}{2} \ln \alpha - \beta y^2 + \frac{L}{2} \ln \beta - \frac{L}{2} \ln \Pi$$

Данный функционал содержит коэффициенты разложения  $\{a_n\}_{n=1}^N$ , для нахождения которых, в свою очередь, необходимо знать значения параметров  $\alpha$  и  $\beta$ . В таких случаях обычно пользуются так называемым EM-алгоритмом:

- 1) На первом шаге некоторым образом, возможно даже случайным, фиксируют параметры модели  $\alpha$  и  $\beta$  и вычисляют значения  $\{a_n\}_{n=1}^N$ .
- 2) На втором шаге находят новые значения для  $\alpha$  и  $\beta$  путем максимизации Evidence при текущих значениях коэффициентов разложения.

При многократном повторении этих шагов алгоритм сходится к некоторому решению  $h(x)$ .

## 2.5 Масштабируемый алгоритм сопряженных градиентов.

Как было сказано ранее, алгоритм обратного распространения ошибки является реализацией классического метода наискорейшего спуска. Данный алгоритм обучения относительно прост в реализации и применении, что и объясняет его широкое использование в области искусственных нейронных сетей.

Спустя некоторое время был разработан другой более эффективный метод обучения, который является альтернативой методу градиента — это метод сопряженных градиентов.

Метод сопряженных градиентов дает улучшения скорости сходимости по сравнению с методом наискорейшего спуска. Однако же, как и метод наискорейшего спуска, он является методом локальной оптимизации.

В искусственных нейронных сетях имеется целевая функция, которую необходимо минимизировать, так как она является средней ошибкой на всем множестве обучающих образцов. Она равна

$$E_{\Sigma}(W) = \sum_{t=1}^T \sum_{k=1}^M (d_{t_k} - y_{t_k}(W))^2$$

Где  $t = 1, 2, \dots, T$  — это множество обучающих образцов.

Алгоритм масштабируемых сопряженных градиентов, как и более общий алгоритм сопряженных направлений, получил применение в области оптимизации благодаря широкому классу проблем, для которых он способен обеспечивать сходимость к оптимальному решению за конечное число шагов. Это серьезное улучшение по сравнению с методом наискорейшего спуска, которое требует бесконечного числа итераций для поиска минимума функции  $f$ .

В векторном пространстве размерности  $D$  множество векторов  $\{P_1, P_2, \dots, P_D\}$  образует множество сопряженных направлений относительно матрицы  $A$ , если

$$P_i A P_j = 0 \text{ для } i = j$$

где  $A$  — это положительно определенная матрица размером  $D \times D$ . Вектора, удовлетворяющие данному условию, называют  $A$ -сопряженными.

Тут может возникнуть вопрос, каким же образом алгоритм масштабируемых сопряженных градиентов достигает сходимости за конечное число шагов и на каких задачах? Допустим, что понадобилось минимизировать функцию

$$F(W) = (b - AW)^T (b - AW)$$

где  $b$  и  $(W-D)$ -мерные векторы, а матрица  $A$  размерностью  $D \times D$  определена выше. Итак, имеется квадратичная функция. Предположим, что нужно найти итерационно оптимальный вектор  $W$ , который минимизирует  $E(W)$ , и нужно начать поиск с начальной точки  $W_0$ . Тогда нужно выбрать ненулевой вектор  $p_1$ , который служит направлением поиска на следующей итерации, при этом совсем неважно, каким образом были выбраны  $W_0$  и  $p_1$ . Следует задать  $W_1$  как следующий вектор

$$W_1 = W_0 + \alpha p_1$$

где скаляр  $\alpha$  выбирается так, чтобы минимизировать  $E(W_0 + \alpha p_1)$ . Сейчас произойдет подход к главному пункту. Оптимальное направление, в котором необходимо двигаться на следующей итерации, является направлением, в котором требуется только один шаг прямо в точку оптимального решения  $W^*$ , и оно должно образовывать  $A$ -сопряженную пару с вектором  $p_1$ .

Оптимальное направление — это  $W^* - W_1$ , поэтому условие, что  $W^* - W_1$  есть  $A$ -сопряженное направление, эквивалентно утверждению, что должно выполняться условие

$$(W^* - W_1)Ap_1 = 0$$

Конечно, в этой точки неизвестно оптимальное решение  $W^*$ , в противном случае не потребовалось никакого алгоритма. Данное условие считается важным по причине того, что в  $D$ -мерном пространстве имеется ровно  $D - 1$ -независимых векторов. Они образуют  $A$ -сопряженную пару с вектором  $p_1$ .

Подытоживая это, потребуется только конечное число направлений для нахождения оптимального решения.

Масштабируемый алгоритм сопряженных направлений создает множество  $A$ -сопряженных векторов. Спустя достижения максимума  $D$ -шагов, алгоритм найдет оптимальное направление, что приведет к его сходимости.

Для задач в форме  $F(W) = (b - AW)^T(b - AW)$  минимизация выполняется непосредственно и образует часть классического алгоритма сопряженных градиентов.

В разбираемой задаче обучения искусственной нейронной сети не существует в явной форме уравнение  $F(W) = (b - AW)^T(b - AW)$ , в частности матрица  $A$  не имеет явного выражения. Несмотря на это, градиент ошибки  $\nabla E$  может выполнять данную роль.

Стоит заметить, что в уравнении  $F(W) = (b - AW)^T(b - AW)$   $AW$  есть множитель, который является пропорциональным градиенту функции  $E(W)$ .

Для подобных задач общего вида конечная сходимость не дает гарантии.

Масштабируемый алгоритм сопряженных градиентов, подобно методу градиентного спуска, обеспечивает нахождение лишь локально оптимальных решений. Тем не менее, метод дает значительное ускорение сходимости по сравнению с методом наискорейшего спуска.

Подойдем к краткому описанию алгоритма.

1. Предположим, что  $K = 0$ . Следует инициализировать весовой вектор  $W$  и вычислить градиент  $G = \text{grad } E(W)$ . Положить вектор начального направления

$$p_H = -\frac{G}{\|G\|}$$

2. Найти скаляр  $\alpha$ , который минимизирует  $E(W_0 + \alpha p)$ , для чего можно использовать метод Фибоначчи или золотого сечения
$$W(K + 1) = W(K) + \alpha p(K)$$
3. Если  $E(W(K + 1)) < \varepsilon_{\text{доп}}$ , где  $\varepsilon_{\text{доп}}$  — это допустимая точность нахождения минимума, то STOP. Иначе же следует вычислить новое направление:

$$G(k + 1) = \text{grad } E(W(k + 1))$$

4. Если  $\text{mod}_D(K + 1)$ , где  $D$  — это размерность пространства весов  $W$ , то новый вектор направления

$$P(k + 1) = -\frac{G(k + 1)}{\|KG(k + 1)\|}$$

иначе же положить

$$\beta = \frac{G(K + 1)^T G(K + 1)}{G(K)^T G(K)}$$

и вычислить новый вектор направления

$$P_{k+1} = - \frac{G(k+1) + \beta p(k)}{\| -G(k+1) + \beta p(k) \|}$$

5. Заменить  $p(k)$  на  $p(k+1)$  и  $G(k)$  на  $G(k+1)$ . Далее осуществить переход к шагу 1 следующей итерации.

### 3 Модель RL-цепи в MATLAB Simulink

Исходными данными при создании данной модели и обучения сети являются осциллограммы ее напряжений и токов, которые сохранены в текстовый файл с помощью цифрового осциллографа в виде множества напряжений  $u_1, u_2, \dots, u_n$  и множества токов  $i_1, i_2, \dots, i_n$  вместе с временными отсчетами. Значением единственного необходимого выхода является мгновенный ток нагрузки. Входным же значением являются аргументы искомой функции, от которых зависит ток.

Данная сеть, как уже было сказано выше, обучается методом обратного распространения ошибки.

В данной работе для моделирования и прогнозирования был использован программный пакет MATLAB, который позволяет смоделировать и обучить искусственную нейронную сеть с заданными параметрами.

Тестирование, анализ и обучение нейронных сетей, а также составление электрических схем производилась на ПЭВМ, которая имеет следующую конфигурацию:

- процессор: 4-ядерный производства AMD A6-6310 с тактовой частотой 1,8 ГГц
- оперативная память: 4 ГБ типа DDR3L с частотой 1600 МГц
- ПЗУ: твердотельный накопитель Kingston SSDNow V300 объемом 120 ГБ
- операционная система: Microsoft Windows 10 Enterprise 64-разрядная

Программа Simulink является встроенным приложением к пакету MATLAB. Simulink реализует принцип визуального программирования

при создании моделей. Пользователь из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от классических способов моделирования, не обязательно изучать язык программирования и численные методы математики. Достаточно общих знаний, которые требуется при работе на компьютере, а также знаний той области, в которой он работает.

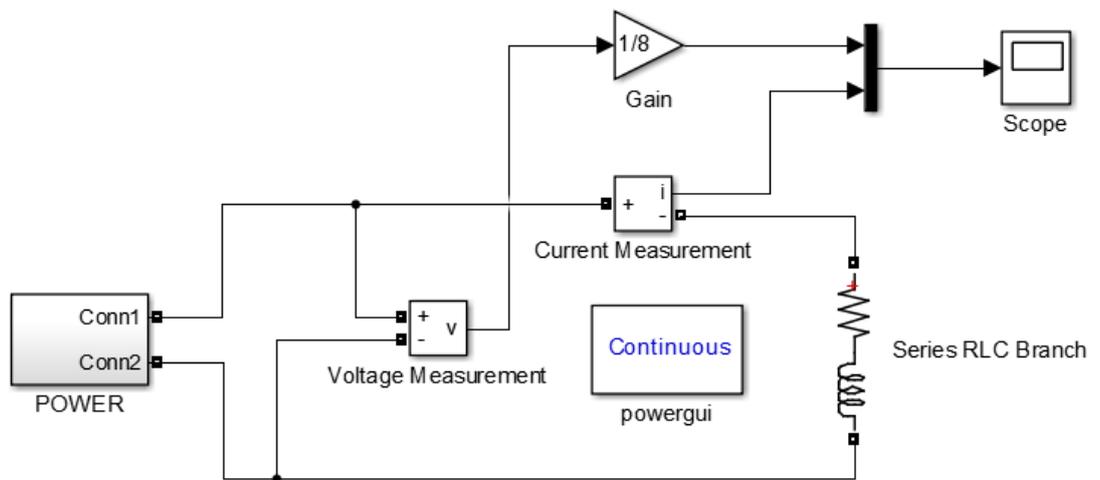


Рисунок 10 — Исходная схема электрической цепи, построенная в Simulink

На рисунке 10 изображена исходная схема электрической цепи для последующего моделирования. Для моделирования использовался инструмент Neural Net Fitting.

Перед моделированием искусственной нейронной сети, требуется снять входные значения напряжений и токов с их осциллограмм. Они изображены на рисунке 11.

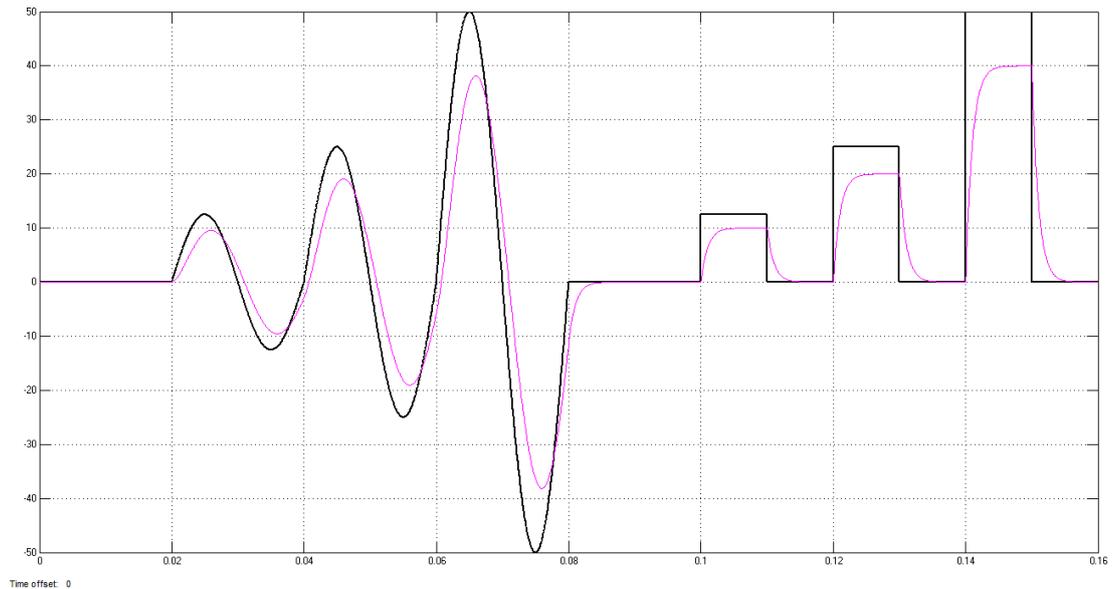


Рисунок 11 — Осциллограмма напряжений и токов

Полученные данные с осциллограммы представляют собой многомерный массив размерностью 351\*99. Извлекается он при помощи блока Save to workspace. Для моделирования с помощью нейронных сетей такой вид массива неудобен, поэтому с помощью языка программирования С (Си) была составлена программа, которая выполняет операцию транспонирования:

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

float* voltage;
float* current;
int numOfPoints;
int N;

float* OpenCSV(const char* fileName)
{
    FILE* file = fopen(fileName, "r");
    if (!file)
        return NULL;

    float dummy = 0.0f;
    numOfPoints = 0;
    while (fscanf(file, "%f;", &dummy) == 1)
```

```

        ++numOfPoints;

    if (numOfPoints == 0)
        return NULL;

    float* points = new float[numOfPoints];
    rewind(file);
    for (int i = 0; i < numOfPoints; ++i)
        fscanf(file, "%f;", &points[i]);
    fclose(file);
    return points;
}

void SaveInputCSV()
{
    FILE* file = fopen("input.csv", "w");
    if (!file)
        return;

    int numSamples = numOfPoints - N;
    int j;

    for (int i = 0; i < numSamples; ++i)
    {
        for (j = 0; j < N; ++j)
            fprintf(file, "%f;", voltage[i + j]);
        for (j = 0; j < N - 2; ++j)
            fprintf(file, "%f;", current[i + j]);

        fprintf(file, "%f", current[i + (N - 2)]);

        if (i < numSamples - 1)
            fprintf(file, "\n");
    }
    fclose(file);
}

void SaveOutputCSV()
{
    FILE* file = fopen("output.csv", "w");
    if (!file)
        return;

    int numSamples = numOfPoints - N;
    int numInputs = 2 * N - 1;
    int j;

    for (int i = 0; i < numSamples; ++i)
    {
        for (j = 0; j < numInputs - 1; ++j)
            fprintf(file, "%f;", current[i + (N - 1)]);
        fprintf(file, "%f", current[i + (N - 1)]);
    }
}

```

```

        if (i < numOfSamples - 1)
            fprintf(file, "\n");
    }
    fclose(file);
}

void main()
{
    while (true)
    {
        voltage = OpenCSV("voltage.csv");
        if (!voltage) break;

        current = OpenCSV("current.csv");
        if (!current) break;

        printf("Enter N: ");
        scanf("%d", &N);
        if (N < 1) break;

        SaveInputCSV();
        SaveOutputCSV();
        break;
    }

    delete voltage;
    delete current;
    _getch();
}

```

### 3.1 Анализ влияния структуры нейронной сети на точность при моделировании RL-ветви

Для обучения искусственной нейронной сети методом обратного распространения ошибки была составлена одномерная матрица выходных данных размерностью  $351 \times 1$ .

Далее запускаем приложение Neural Fitting. Сделать это можно как с помощью GUI (graphical user interface, пер. «графический интерфейс пользователя») через вкладку Apps с последующим выбором приложения, так и введя в командную строку рабочей области MATLAB команду “nftool”.

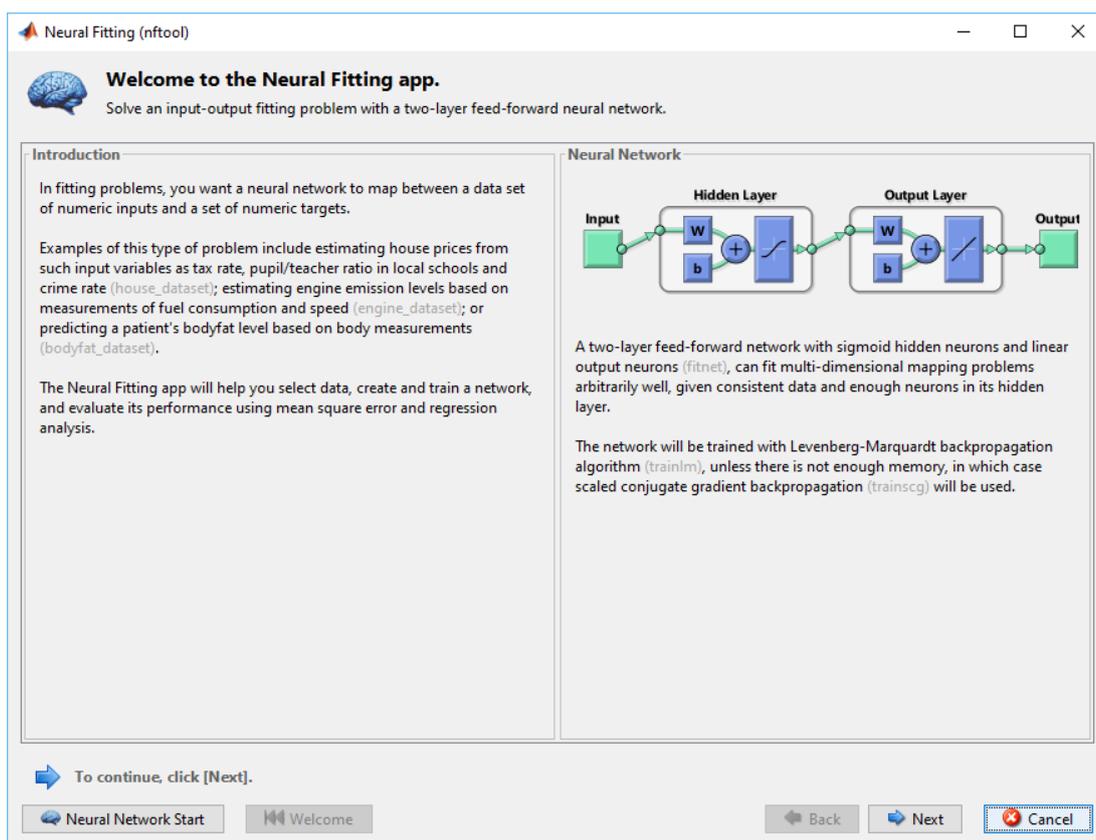


Рисунок 12 — Окно приглашения

На рисунке 12 представлено окно приглашения к использованию инструмента. В нем указывается, что будет применена двухслойная

однонаправленная сеть с сигмоидальной функцией активации нейронов скрытого слоя и линейной функцией активации нейронов выходного слоя (для этого используется функция fitnet). Также здесь представлена визуализация самой нейронной сети. Данная сеть позволяет множество раз решать задачи многомерной аппроксимации, если соблюдено условие согласованности данных и имеется достаточное количество нейронов в скрытом слое. Сеть обучается с использованием алгоритма Левенберга-Маркара (при этом используется функция trainlm).

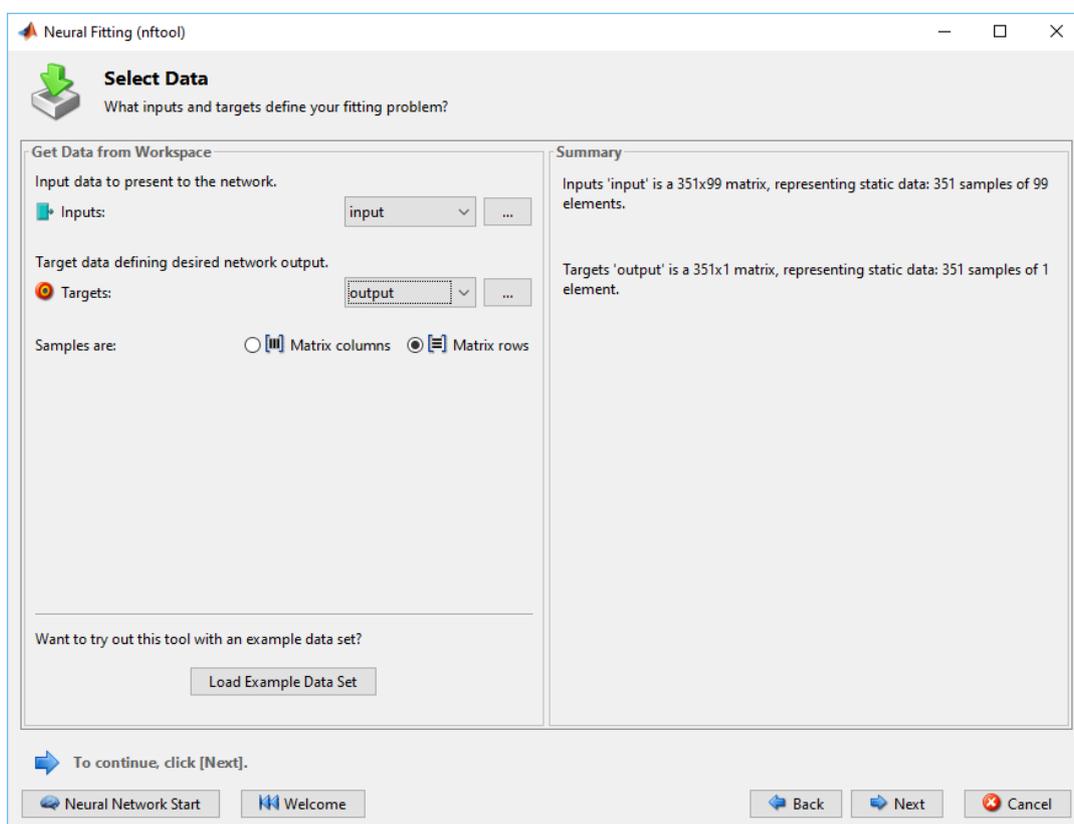


Рисунок 13 — Окно выбора данных

Перейдем в окно выбора данных Select Data, представленному на рисунке 13, в котором надлежит осуществить выбор массивов входных и выходных данных. После нажатия на кнопку Next, происходит переход в

окно Validation and Test Data Window, которое представлено на рисунке 14. В нем отображено разделение набора данных между обучающим набором (Training) и проверочным набором (Validation). Данное разделение применяется для того, чтобы оценить обобщающие свойства сети, а также остановить обучение, когда обобщение прекращает улучшаться. Также имеется тестовое множество (Testing), которое не оказывает влияния на обучение, но служит для проверки качества обучения на данных, которые не использовались в обучении сети.

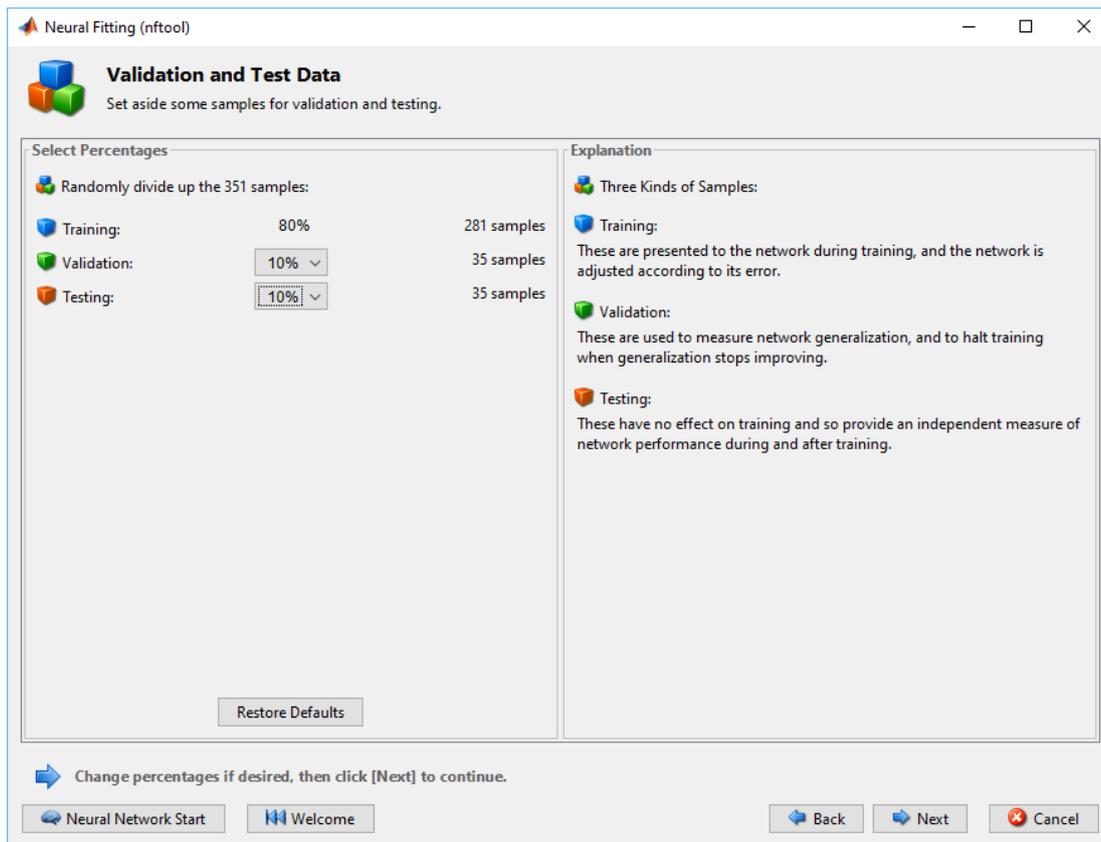


Рисунок 14 — Окно распределения набора данных

Далее настройка искусственной нейронной сети переходит в окно выбора количества нейронов. По умолчанию число нейронов в скрытом слое равно 10. Данное число возможно изменить, изменяя опцию Number Hidden Neurons. Число нейронов выходного слоя равно 1.

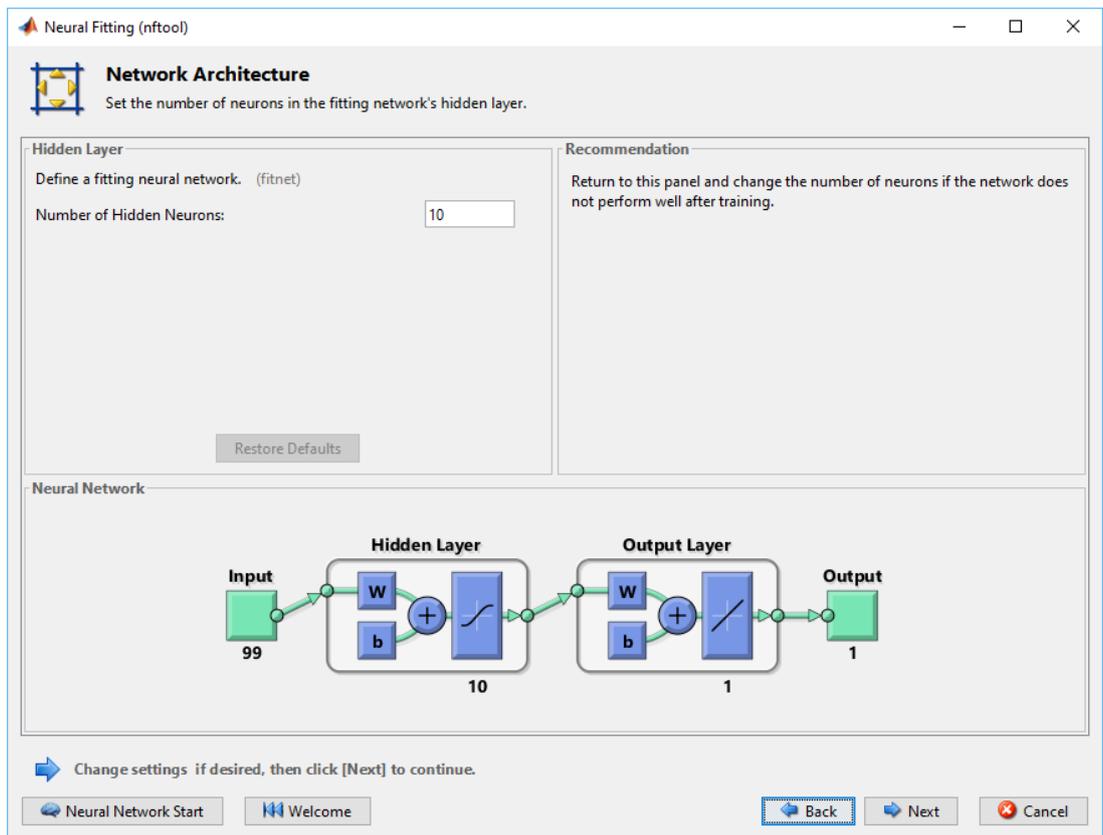


Рисунок 15 — Выбор количества нейронов в скрытом слое и структура нейронной сети

В следующем шаге осуществляется переход к окну Train Network, то есть к окну обучения сети (рисунок 17).

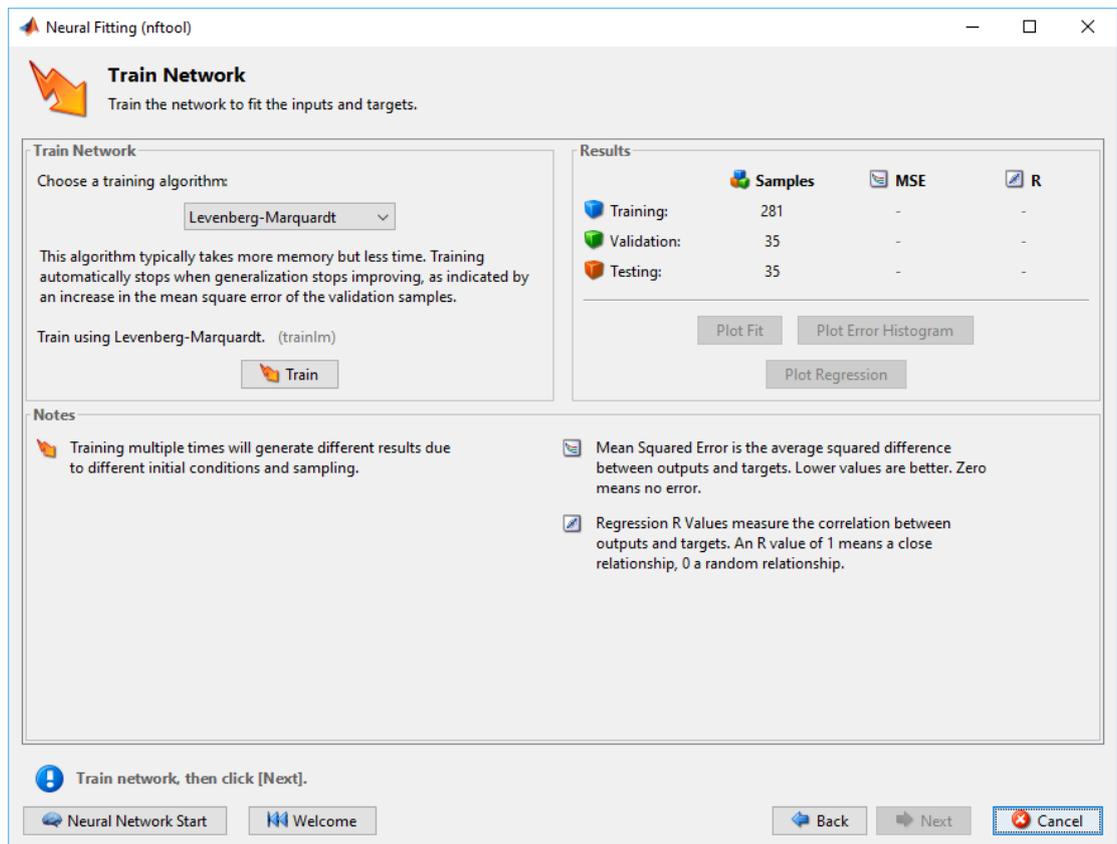


Рисунок 16 — Окно обучения сети

В данном окне из выпадающего списка осуществляется выбор алгоритма обучения. По умолчанию предлагается выбран алгоритм Левенберга-Маркара (Levenberg-Marquardt). Здесь можно выбрать алгоритм Байесовской регуляризации (Bayesian Regularization) и масштабируемый алгоритм сопряженных градиентов (Scaled Conjugate Gradient). Ниже окна выбора алгоритма при этом выводится краткая характеристика алгоритма.

При нажатии на кнопку Train открывается окно процесса обучения сети Neural Network Training. После обучения в окне выводятся результаты обучения: для обучающего, проверочного и тестового множеств выводится среднеквадратическая ошибка (MSE) и значение коэффициента корреляции (R) между выходами сети и целевыми значениями. Данное диалоговое окно позволяет составить графики,

которые характеризуют качество обучения. Если результат не устраивает пользователя, то для можно повторить обучение (кнопка Retrain), используя другой алгоритм.

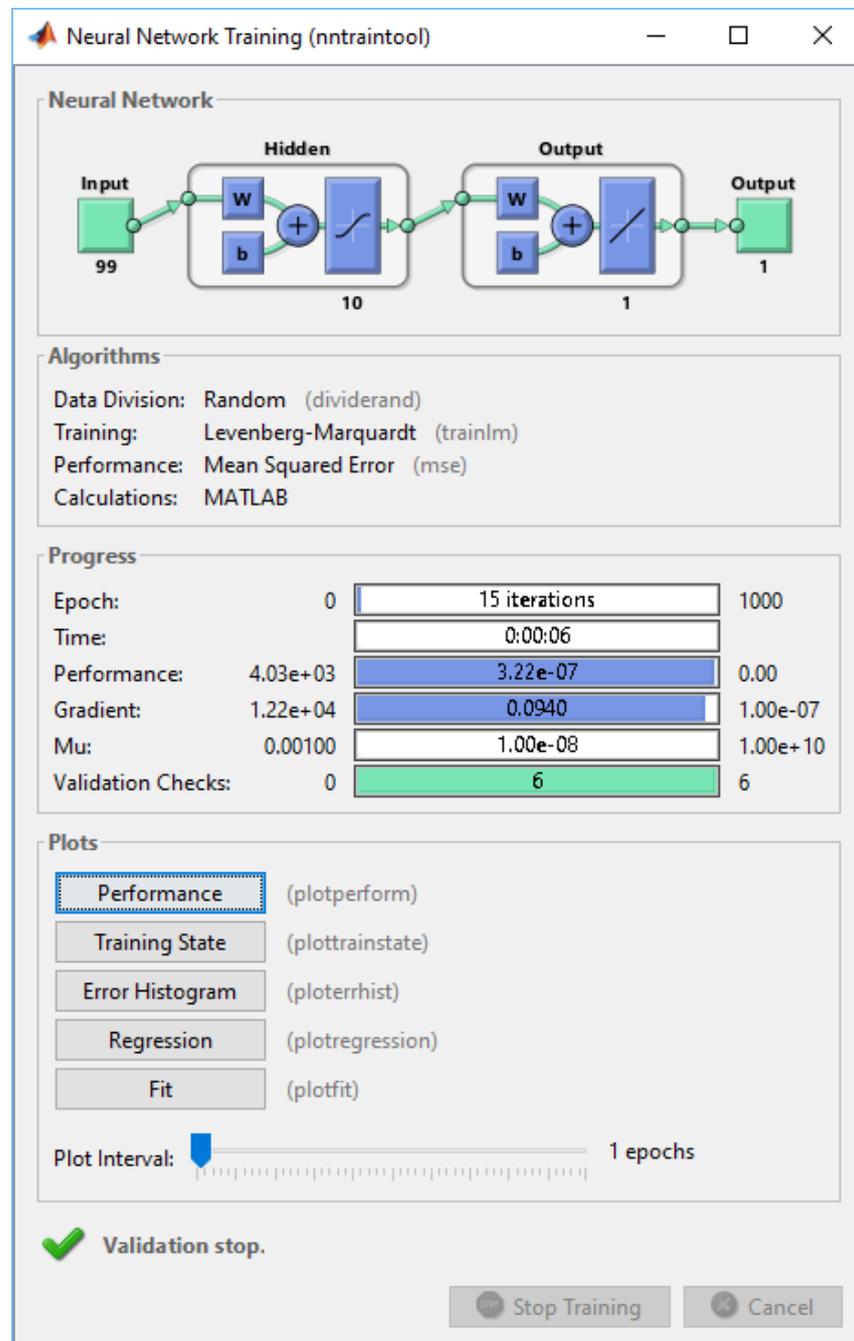


Рисунок 17 — Окно процесса обучения сети

Здесь же по нажатию на кнопку Performance предлагается возможность увидеть график обучения сети, который показывает как повела себя ошибка обучения сети. Из данного графика, который представлен на рисунке 15, видно, что за 15 эпох достигнуто среднеквадратическое значение ошибки равной  $8,3404 * 10^{-2}$ . Функция обучения предлагает в качестве средства борьбы с переобучением сети обучение с ранней остановкой (early stopping). Ошибка на проверочном множестве перестала уменьшаться, вместе с этим обучение сети было остановлено.

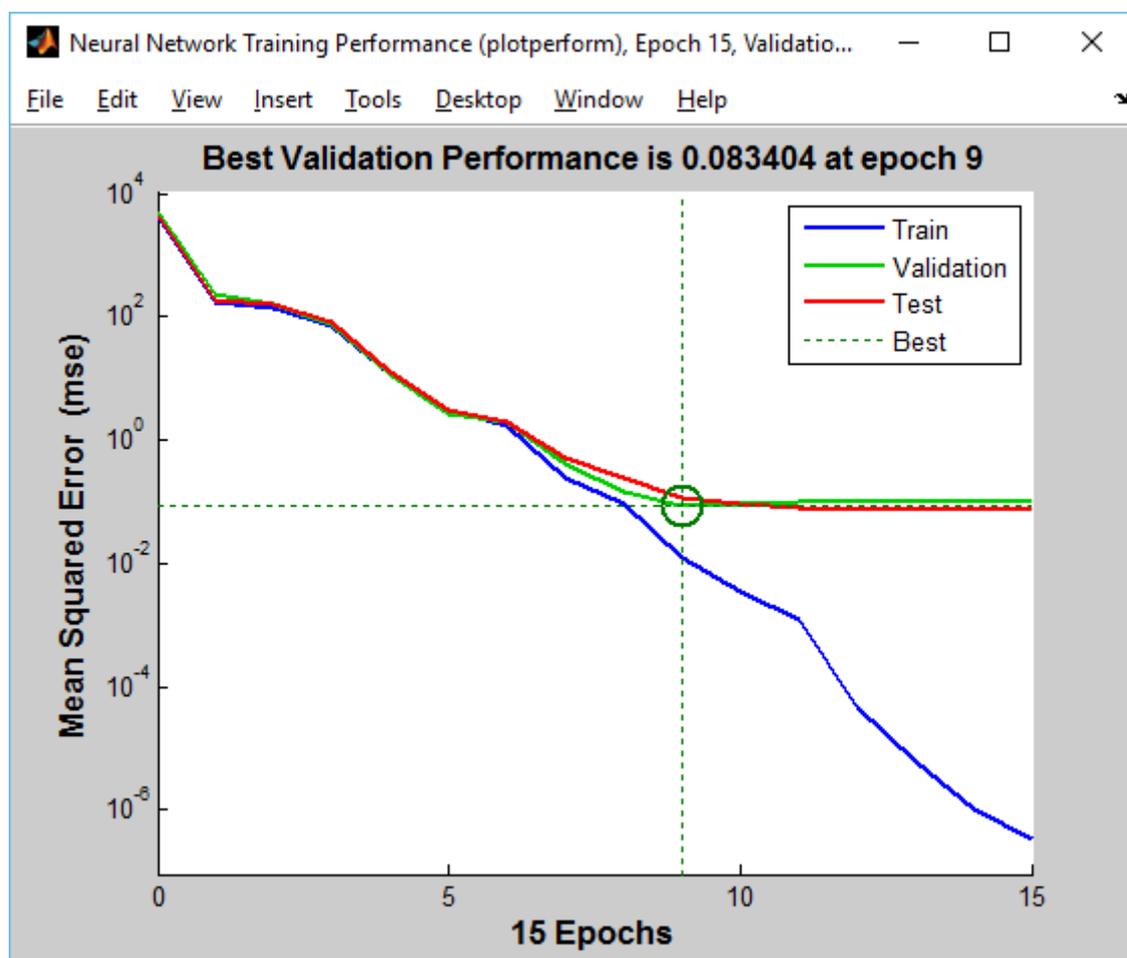


Рисунок 18 — График обучения сети

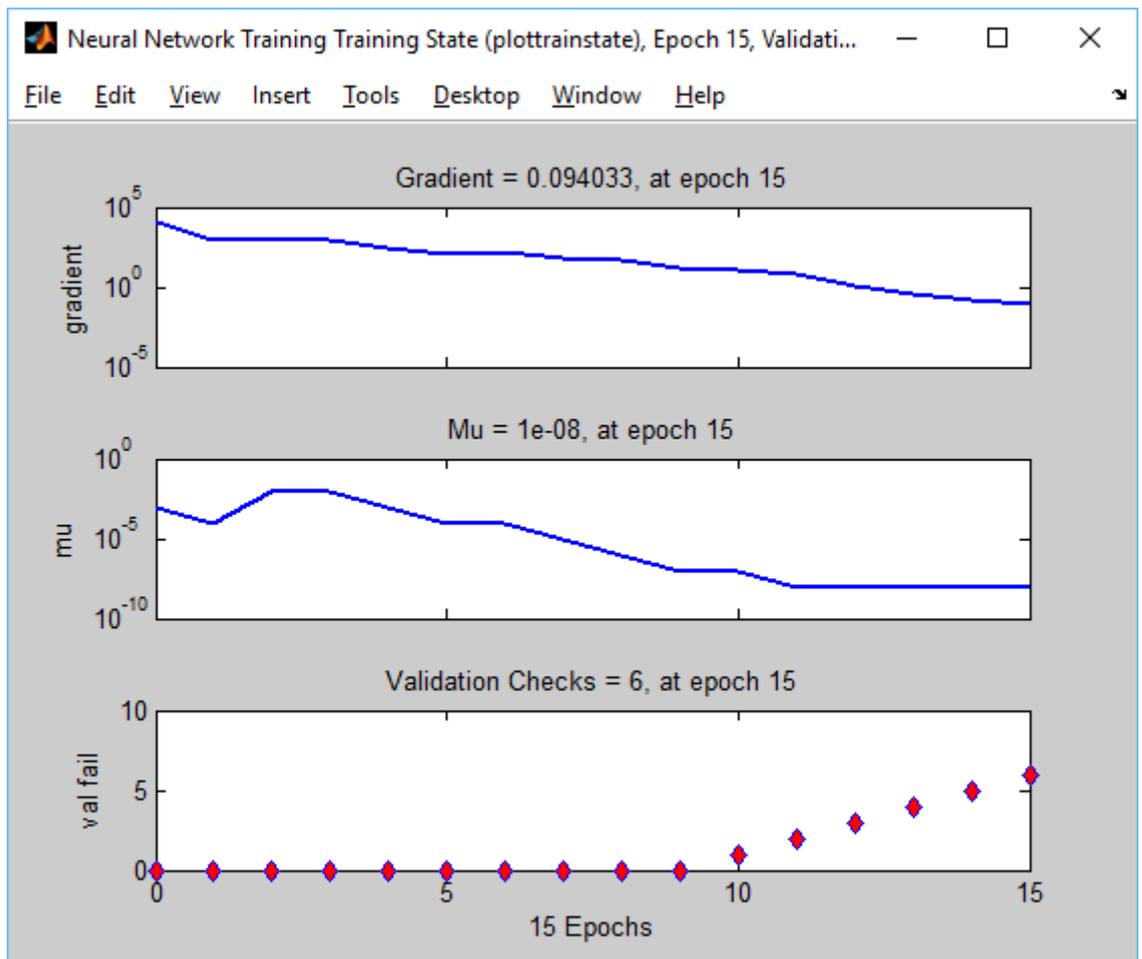


Рисунок 19 — Графики состояния обучения Training State

Перейдем к графикам состояния обучения, изображенному на рисунке 19. На третьем графике val fail показано изменение ошибки на контрольном множестве. Видно, что после 9 точки ошибка возрастает. График gradient отражает изменение градиента функционала ошибки обучения по весам сети. График mu показывает изменение параметра обучения  $\mu$  методом Левенберга-Маркара.

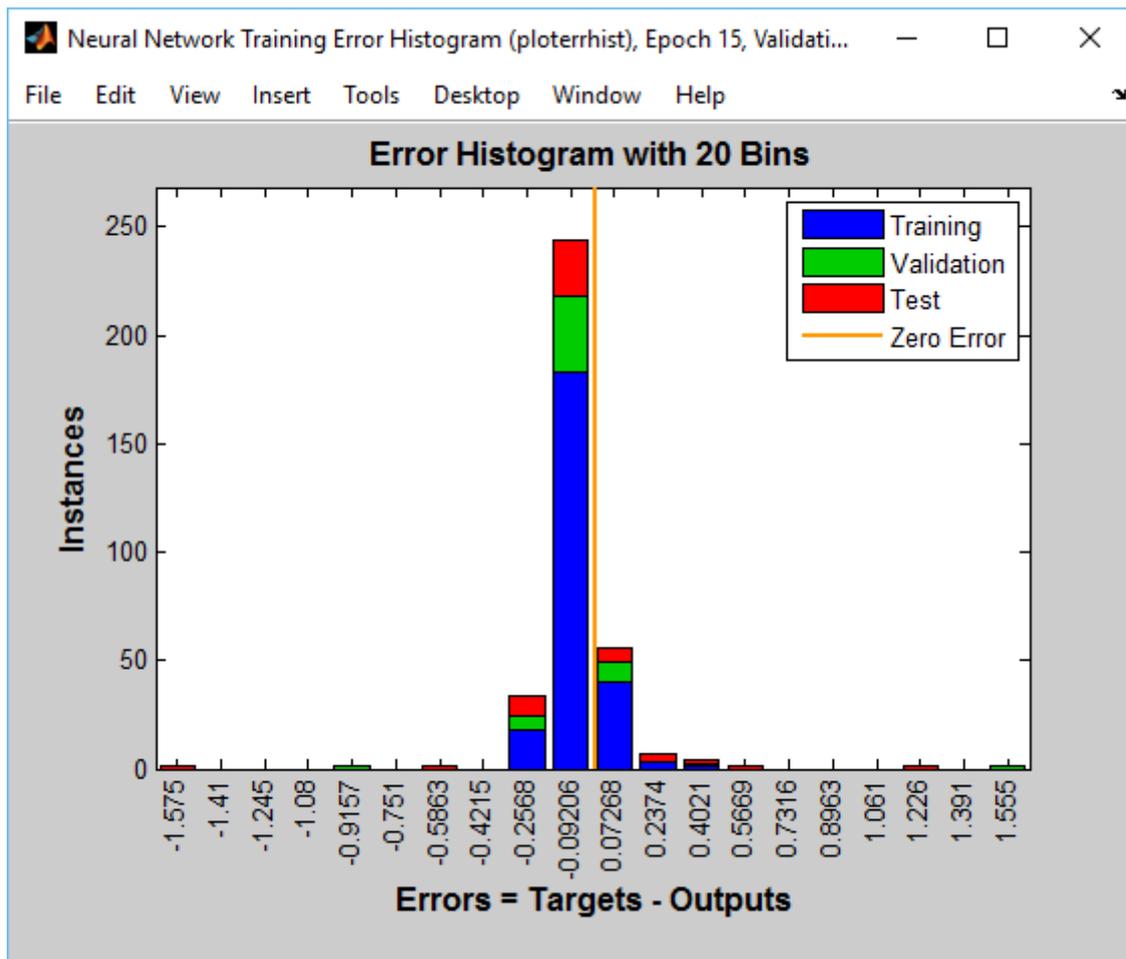


Рисунок 20 — Гистограмма ошибок

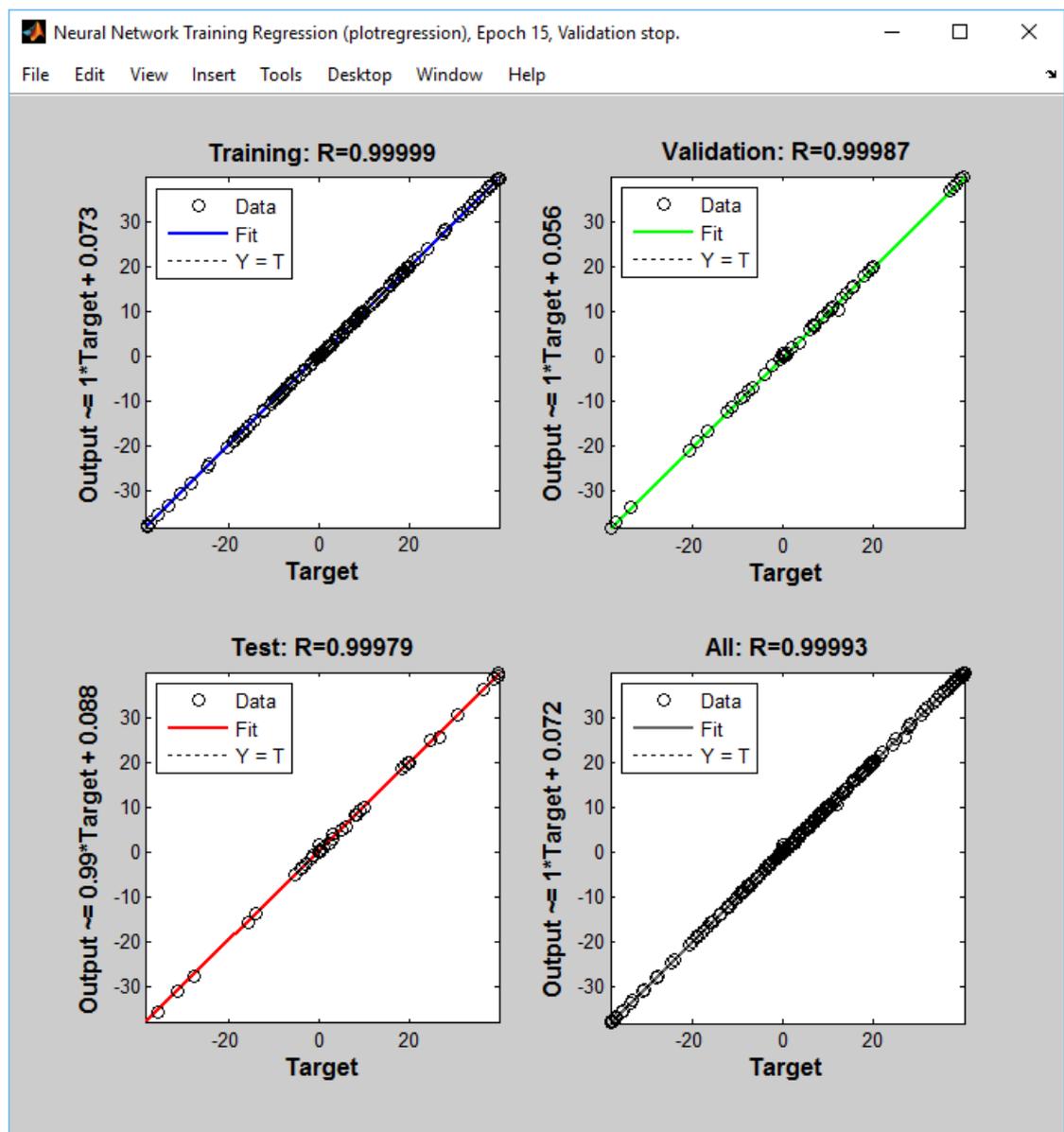
Гистограмма ошибок (рисунок 20) Error Histogram показывает, на каком числе примеров (Instances) сеть дает ту или иную погрешность. Данная погрешность высчитывается как разность между целевым значением и выходом сети. На гистограмме также отображаются погрешности для обучающего, проверочного и тестового множеств.

Из нее видно, что большинство ошибок лежит в диапазоне от -0,2568 до 0,07268. Имеется небольшое количество ошибок, превышающих данные значения.

График Regression показывает линейную регрессию результатов обучения сети на трех анализированных подмножествах и на всех

множествах. Для каждого результата рассчитывается коэффициент корреляции  $R$ , создается график и выводится уравнение регрессии вида  $Output = a \times Target + b$ , если происходит полное совпадение выходов сети с целевыми значениями  $R = 1, a = 1, b = 0$ . Исходя из изображенного на рисунке 21, можно сделать вывод, что сеть практически идеально аппроксимирует функцию.

Функция аппроксимации Fit представляет значения ошибок сети при различных входных значениях. На графике видно, при каких входных значениях получаются большие ошибки.



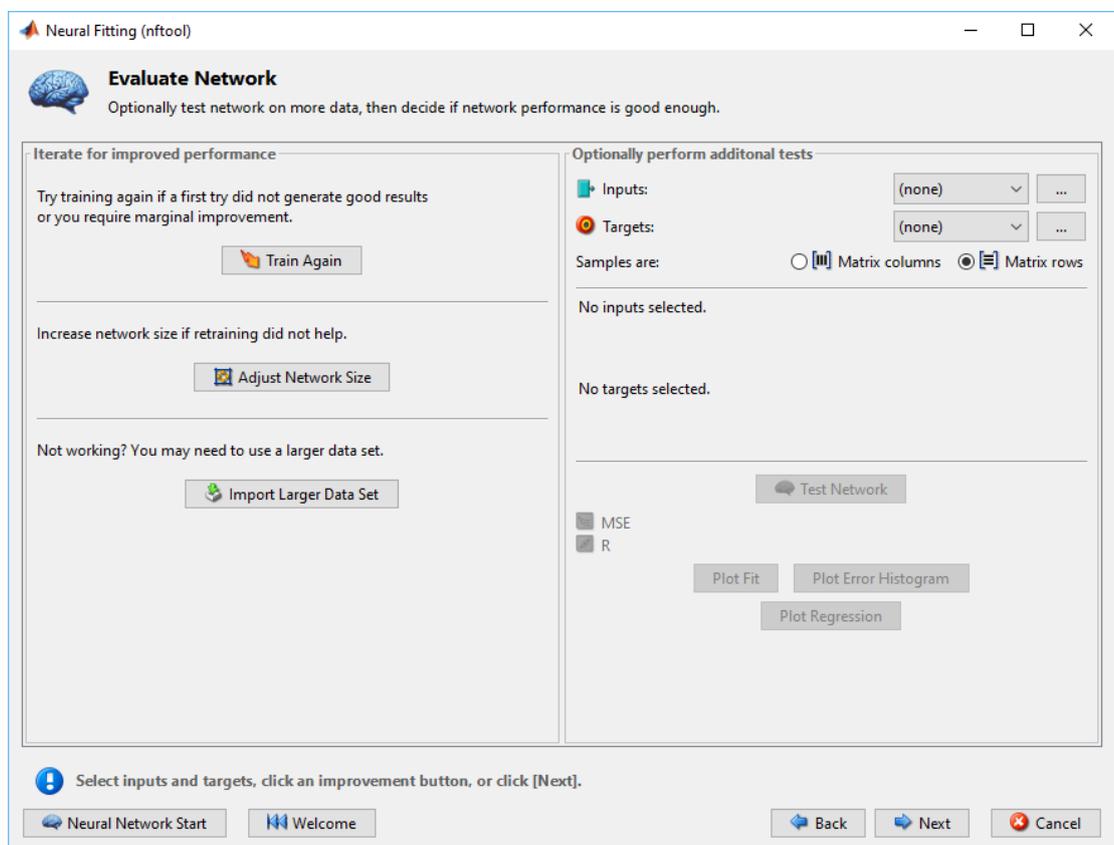
## Рисунок 21 — Графики линейной регрессии результатов обучения

Результаты обучения оказались подходящими для дальнейшей работы. Переходим в окно обучения сети после завершения обучения. Появляется окно оценки (Evaluate Network).

Левая часть окна предлагает повторить для улучшения производительности (Iterate for improved performance):

В правой части окна предлагается провести дополнительные испытания (Optionally perform additional tests), выбрав входные и целевые значения.

Для окончания работы с сетью следует нажать на кнопку Next. После этого будет произведен переход в окно Deploy Solution развертывания (размещения) решения.



## Рисунок 22 — Окно оценки сети после завершения обучения

Здесь программа позволяет сформировать функции для MATLAB Compiler, который в свою очередь дает создавать независимые приложения в формате исполняемого файла Windows (.exe), компоненты и Windows-библиотеки (.dll), а для MATLAB Builder — для создания пакетов, которые расширяют возможности MATLAB Compiler средствами создания компонентов различных фреймворков, например, Microsoft .NET Framework. Возможно создание функции для MATLAB Coder, создающий автономный C и C++ код из кода MATLAB.

В последнем окне, который является окном сохранения результатов можно сохранить результаты, а также создать программу, воспроизводящую создание сети.

Возможно создать простой или усовершенствованный код MATLAB, повторяющий действия Neural Network Pattern Recognition Tool. Код отличается более детальным заданием параметров функций. В простом коде многие параметры устанавливаются по умолчанию.

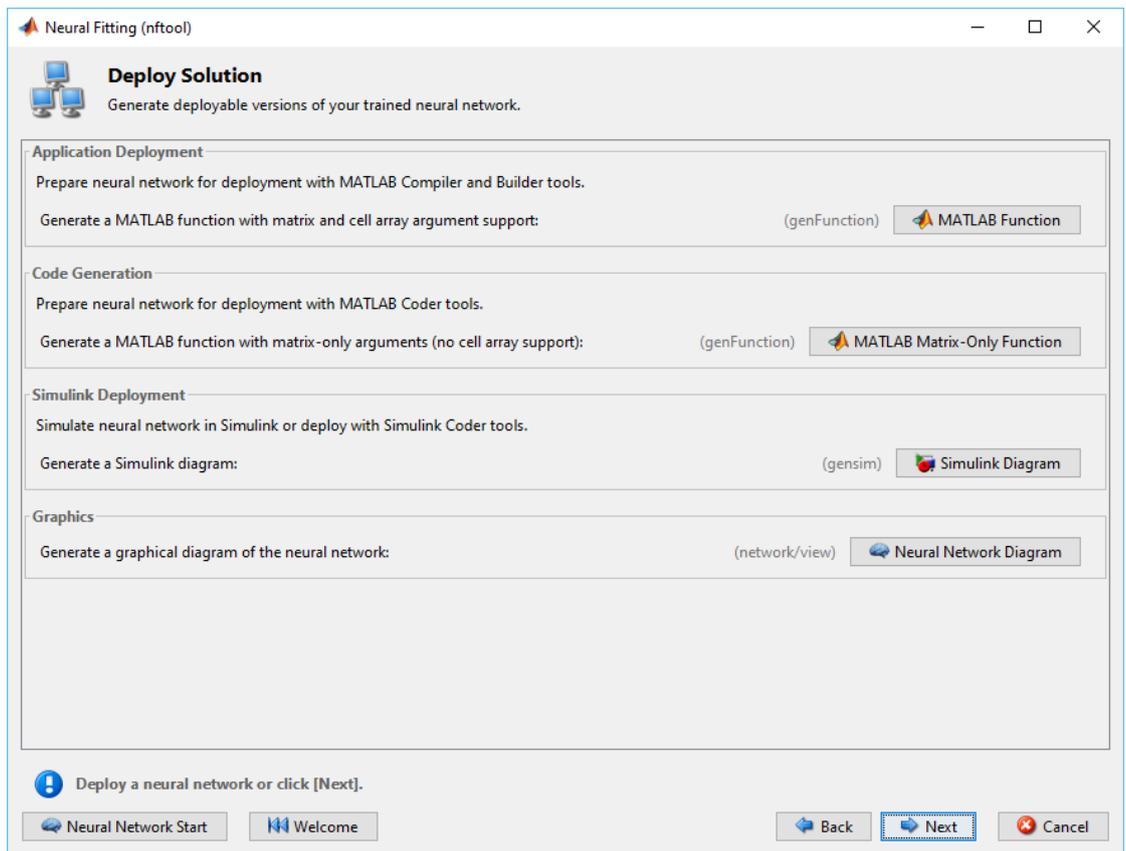


Рисунок 23 — Окно развертывания решения

Также можно сохранить отмеченные результаты в рабочем пространстве MATLAB. При этом рекомендуется заменить имена, которые присваиваются по умолчанию, на свои имена.

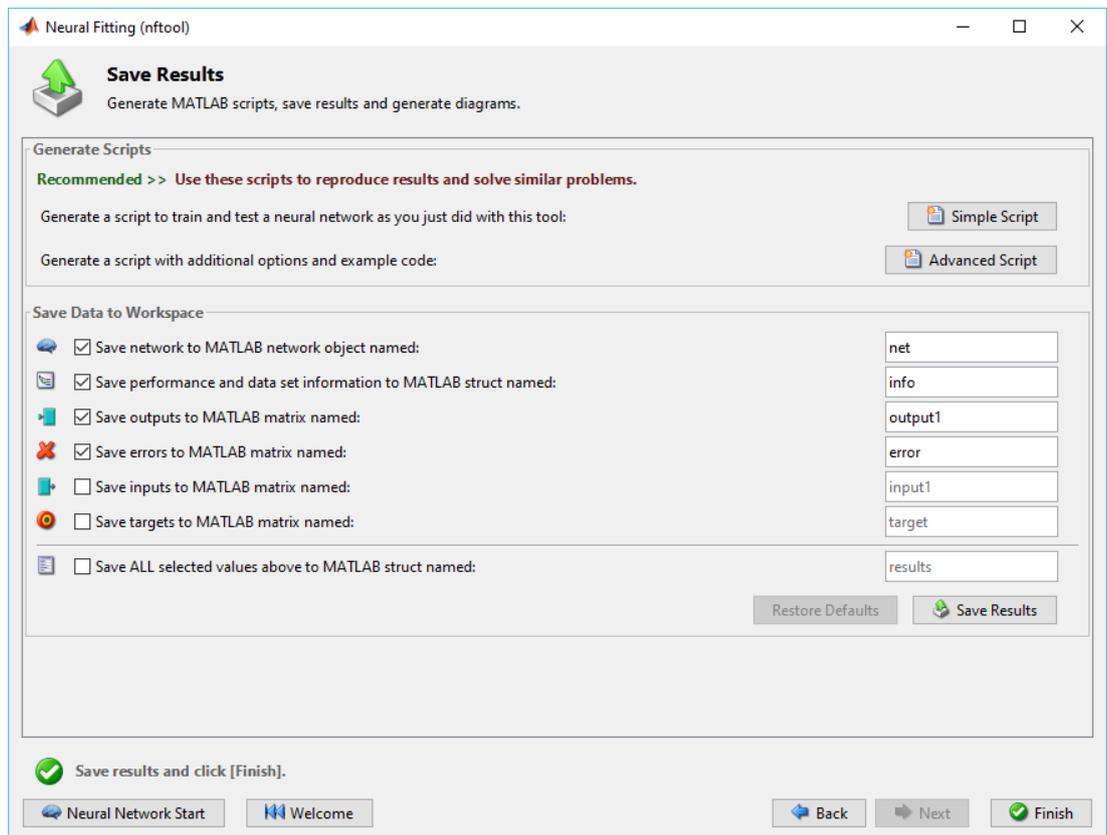


Рисунок 24 — Окно сохранения результатов

Для последующего использования нейронной сети необходимо экспортировать сведения о ней, которые находят в менеджере нейронных сетей в саму рабочую область MATLAB. Для этого нужно в окне Deploy Solution нажать на кнопку Simulink Diagram. На рисунке 25 изображена получившаяся схема нейронной сети, которая пригодна для дальнейшего применения в электрической модели.

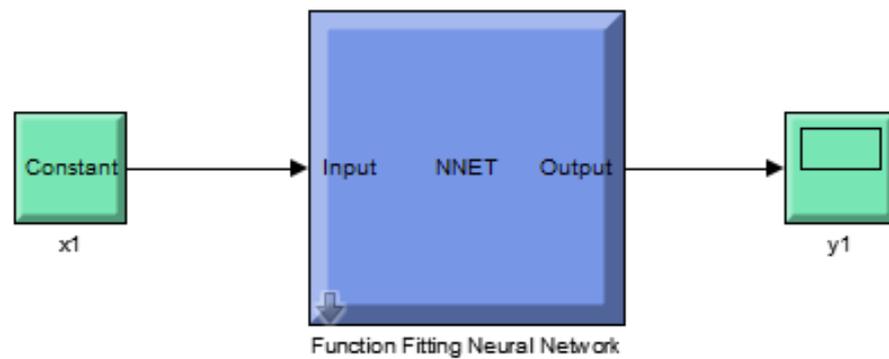


Рисунок 25 — Схема искусственной нейронной сети

Если осуществить двойной щелчок левой кнопкой мыши на блоке Function Fitting Neural Network, то откроется его структура, которая представлена на рисунке 26. В ней содержатся входные и выходные данные, а также два слоя и веса нейронной сети.

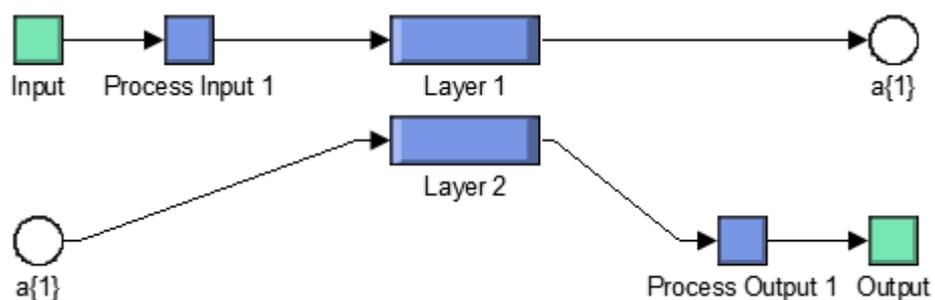


Рисунок 26 — Структура блока Function Fitting Neural Network

После этого этапа данный блок был помещен в изначальную схему RL-цепи с внесением некоторых изменений. Так как требовалось получить множества предыдущих значений напряжений и токов были включены в схему блоки Tapped Delay, который позволяет формировать вектор предыдущих значений с заданным в нем интервалом времени. Блок искусственной нейронной сети реализуется

с помощью Neural Network Fitting, который был получен на этапе создания нейронной сети. На вход поступает вектор напряжений и токов, а также дополнительные параметры. Так как сеть управляет источником тока, то значение ее выхода через блок задержки передается на вход сети. Отсутствует возникновение алгебраической петли вследствие отсутствия передачи текущего значения тока, потому что передается только предыдущее, что и указывается в настройках блоков Tapped Delay. В данном случае происходит осуществляется механика памяти модели для мгновенных значений, поскольку сама сеть не обеспечивает этого. Также имеются дополнительный блок масштабирования, которые нужен на выходе сети, т.к. ее входные и выходные значения ограничены интервалом значений, которыми пользуется сигмоидальная функция активации нейронов. Получившаяся структурная схема изображена на рисунке 26.

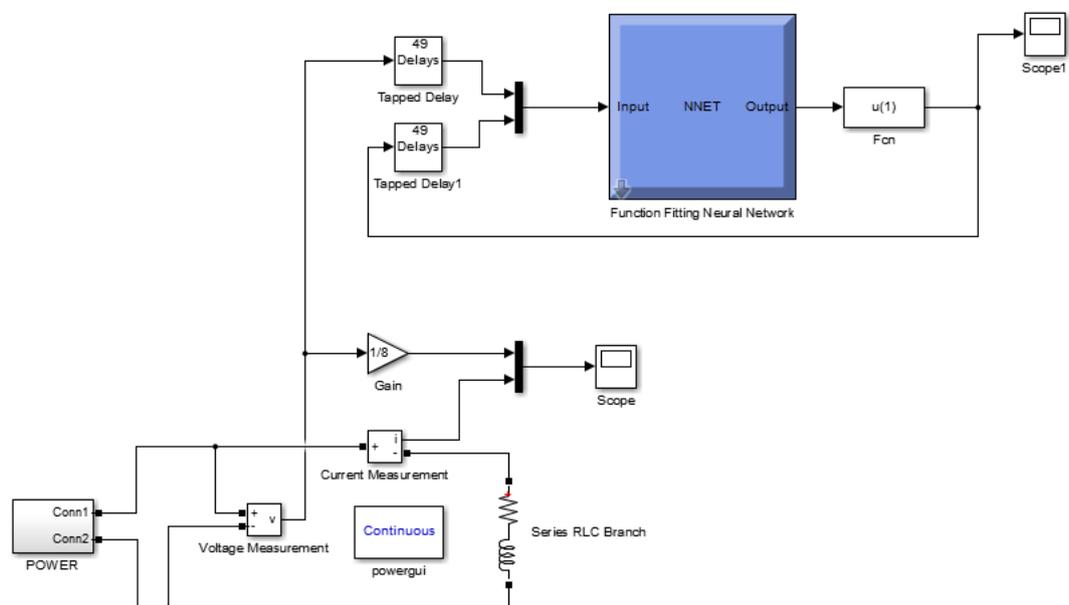


Рисунок 26 — Схема с включенным в нее блоком искусственной нейронной сети и блоками временной задержки

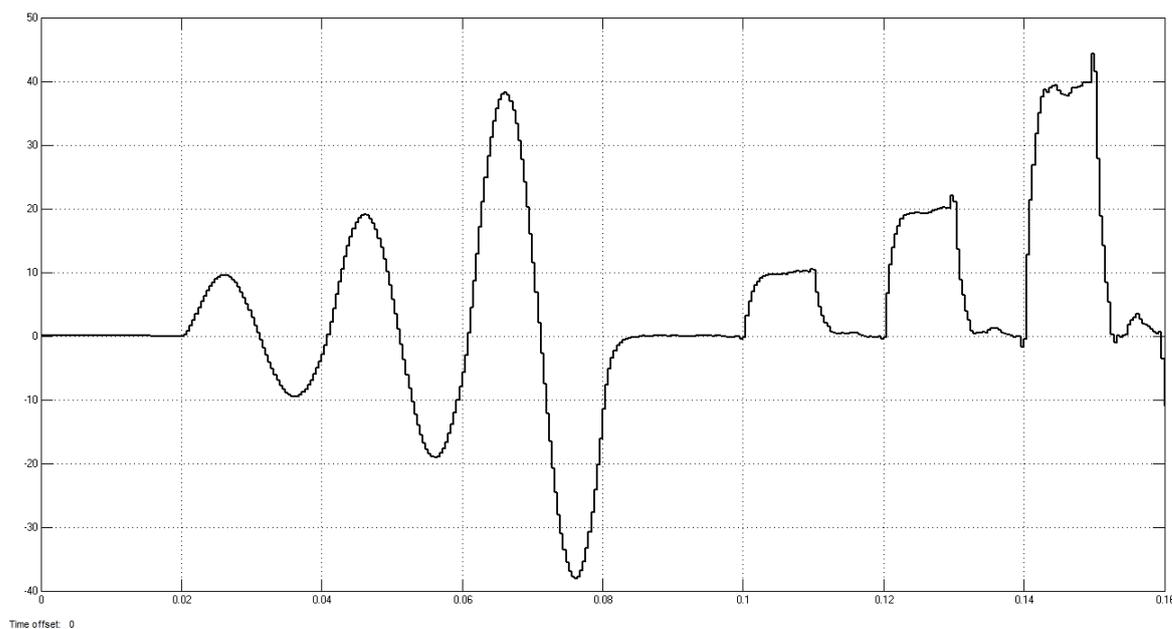


Рисунок 27 — Рассчитанный ток искусственной нейронной сетью

На основе диаграммы, которая представлена на рисунке 27, можно сделать вывод, что нейронная сеть, которая была обучена методом Левенберга-Маркара, дает высокий результат. Из этого следует, что искусственные нейронные сети пригодны для использования для моделирования электрических нагрузок. В следующей главе конкретно рассматривается использование искусственных нейронных сетей для моделирования электрических нагрузок тиристорного трансформатора контактной машины, из которого выясним, насколько будет результативным использование того или иного метода обучения нейронных сетей.

### 3.2 Синтаксис кода нейронной сети, обученной по методу Левенберга-Маркара

Также представлен исходный код блока получившейся обученной нейронной сети. Данный код позволяет более тонко варьировать параметры нейронной сети.

```
x = input';
t = output';

trainFcn = 'trainlm'; % Levenberg-Marquardt

hiddenLayerSize = 30;
net = fitnet(hiddenLayerSize,trainFcn);

net.divideParam.trainRatio = 90/100;
net.divideParam.testRatio = 5/100;
net.divideParam.valRatio = 5/100;

[net,tr] = train(net,x,t);

y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y);

view(net)
```

Изначально сам синтаксис, который отличается от графической настройки нейронной сети, и имеет вид:

```
[net,tr,Ac,EL] = trainlm(net,Pd,Tl,Ai,Q,TS,VV,TV)

info = trainlm(code)
```

TRAINLM — это функция обучения сети. Справочные сведения по ней приведены в документации Matlab [4].

#### 4. Моделирование тиристорного трансформатора контактной машины на основе нейронной сети

Для начала необходимо составить схему тиристорного трансформатора контактной машины. Она была представлена на рисунке 28. Составлена она также в программном пакете MATLAB Simulink.

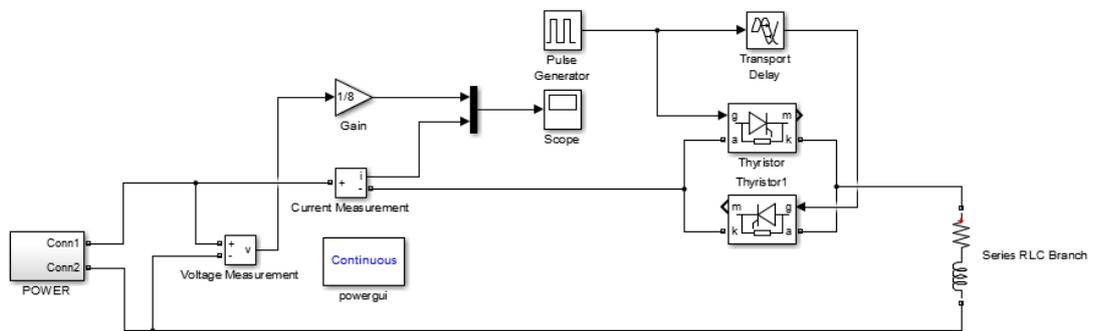


Рисунок 28 — Схема тиристорного трансформатора контактной машины

Далее потребуется снять данные с осциллограммы токов и напряжений, которая была продемонстрирована на рисунке 29.

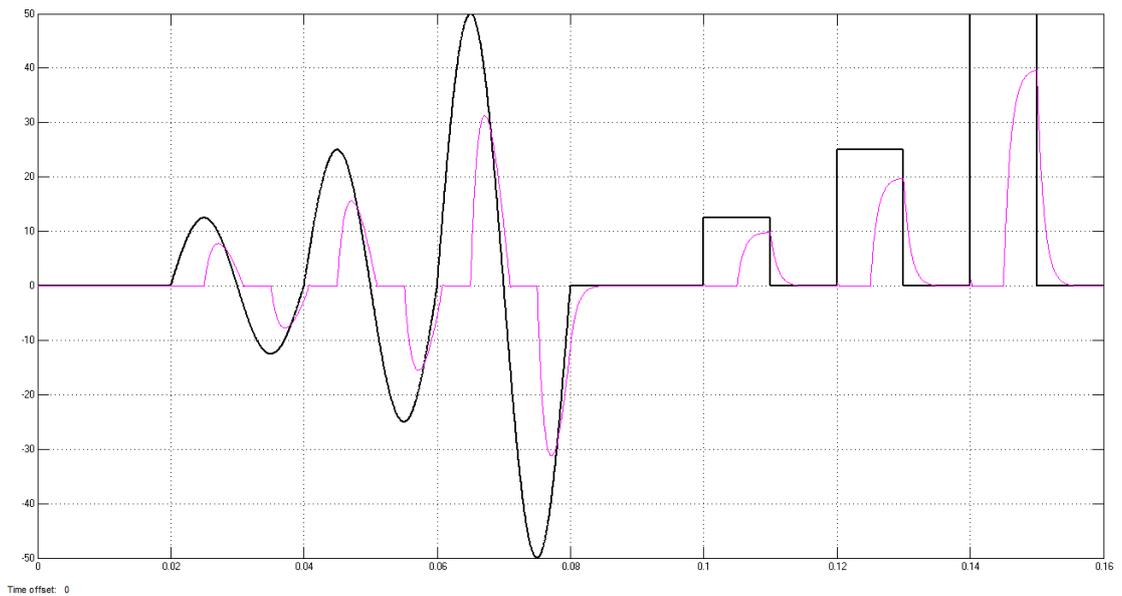
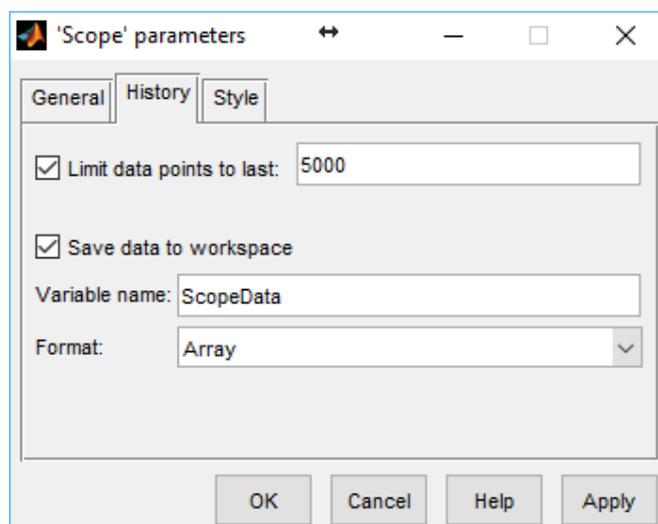


Рисунок 29 — Осциллограмма токов и напряжений тиристорного трансформатора контактной машины

После этого необходимо произвести экспорт полученных данных с осциллограммы. Для этого в свойствах осциллографа на вкладке History следует поставить галочку на параметре Save data to workspace и выбрать формат данных Array. Данное свойство показано на рисунке 30. После этого массив появляется в рабочей области MATLAB.



### Рисунок 30 — Параметр осциллографа

Следующим шагом следует произвести операцию транспонирования получившихся временных матриц токов и напряжений с помощью программы, код которой был изложен в главе 3.

После получения данных в нужном виде произведем экспорт в рабочую область MATLAB с целью обучения нейронной сети на основе полученных входных и выходных данных. Саму искусственную нейронную сеть обучим тремя методами с целью выяснения, какой способ покажется наиболее эффективным. Для наглядности результатов и сравнения между собой каждого метода в работу включены: график обучения сети, график состояния обучения Training State, гистограмма ошибок, а также графики линейной регрессии результатов обучения.

Как и в предыдущей главе, для исследования влияния нейронной сети на точность моделирования тиристорного источника питания контактной сварки будет использоваться сеть прямого распространения с двумя слоями и с сигмоидальной функцией активации на первом слое и линейной функции активации в выходном слое.

В окне распределения набора данных оставим значения по умолчанию, так как они дают наиболее высокую эффективность обучения. Далее в окне выбора количества нейронов для наиболее точных результатов обучения выберем число нейронов 20 штук. В следующих главах рассмотрим эффективности данных методов.

#### 4.1 Обучение искусственной нейронной сети методом Левенберга-Маркара

Первым методом для обучения искусственной нейронной сети будет методом Левенберга-Маркара (функция `trainlm`). На рисунке 31 изображено окно обучения сети. Из информации в нем можно увидеть, что при увеличении количества скрытых нейронов прямо пропорционально увеличивается количество итераций, при этом существенно увеличивается нагрузка на процессор. Для полного обучения нейронной сети хватило всего 138 итераций.



Рисунок 31 — Окно процесса обучения методом Левенберга-Маркара

По итогам обучения проанализируем полученные результаты с помощью предоставляемых модулем графиков. На рисунке 32 изображен график состояния обучения. В данном случае среднеквадратичное значение ошибки достигло значения  $2,178 \times 10^{-2}$  на 132 эпохе, при этом сеть обучилась всего за 138 эпох. Далее ошибка перестала уменьшаться на проверочном множестве.

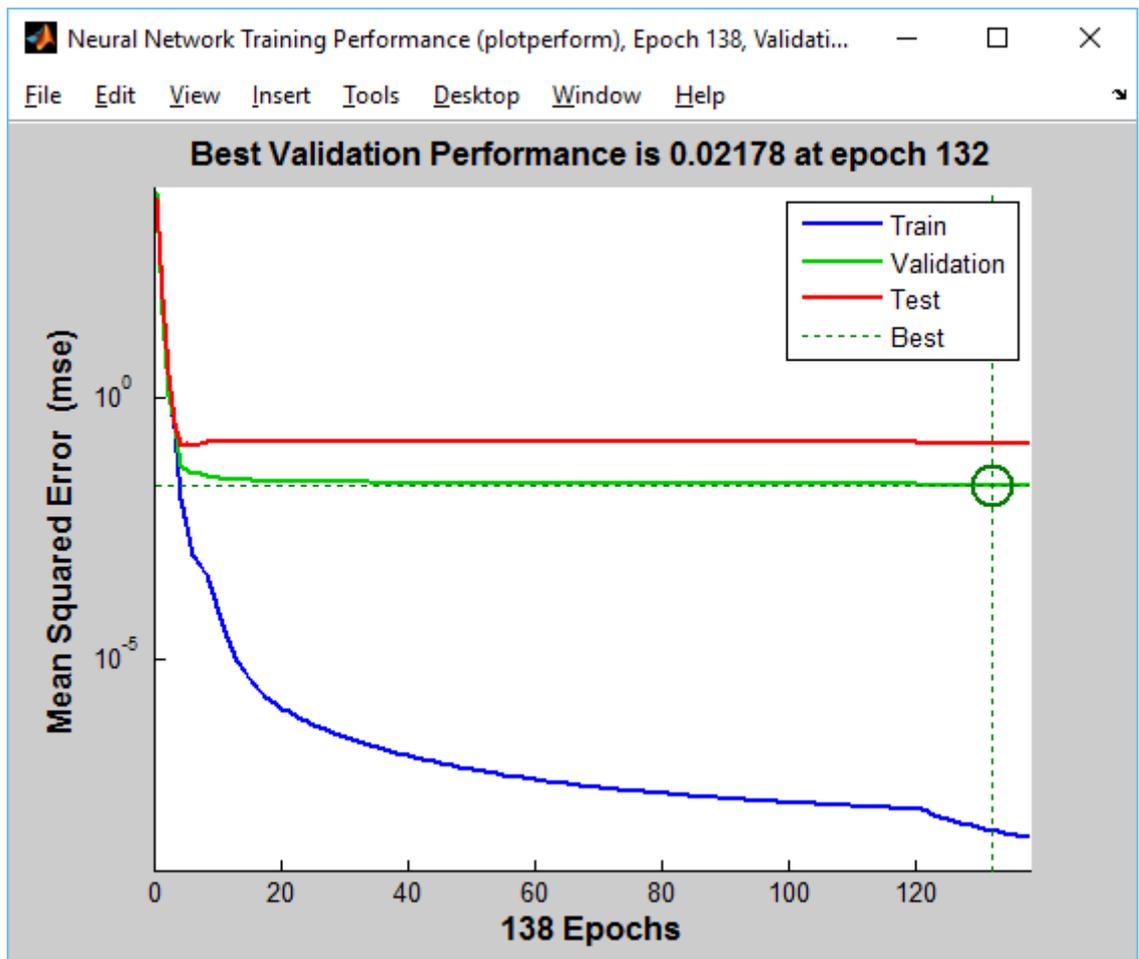


Рисунок 32 — График обучения сети

На рисунке 34 изображен график линейной регрессии состояния обучения. Из нижнего графика val fail находим подтверждение того, что после 132 эпохи ошибка резко возрастает. При этом градиент достиг максимального значения  $8,3104 \times 10^{-4}$ .

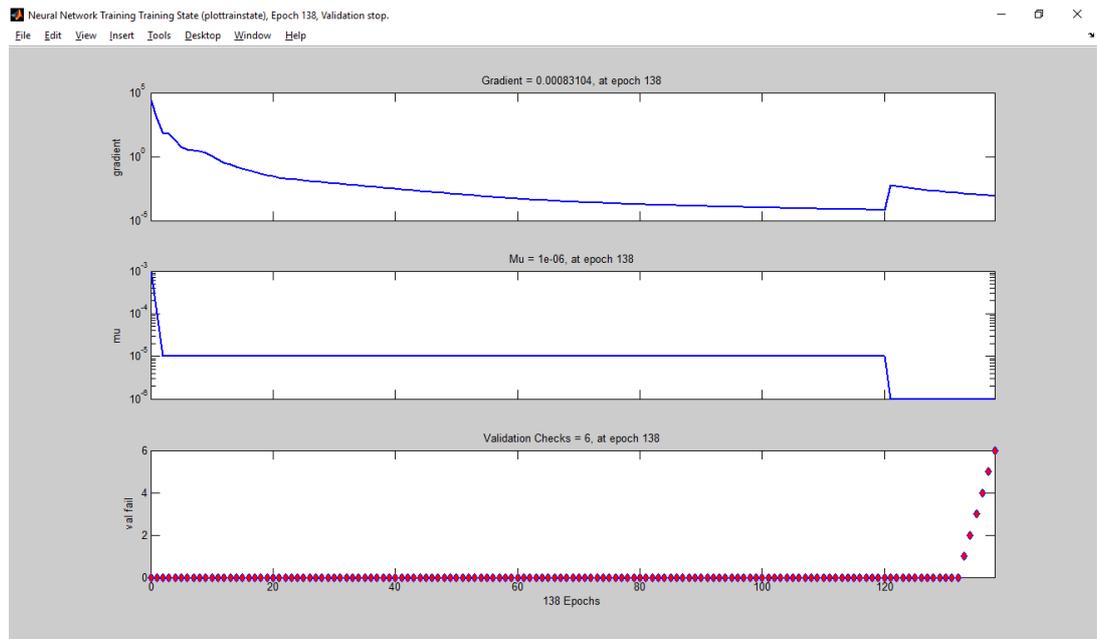


Рисунок 33 — График состояния обучения сети

Перейдем к рисунку 35, на котором изображена гистограмма ошибок обученной нейронной сети. В данном случае присутствует совсем небольшой диапазон ошибок в районе значения 0,6295, который лежит в пределах погрешности обучения.

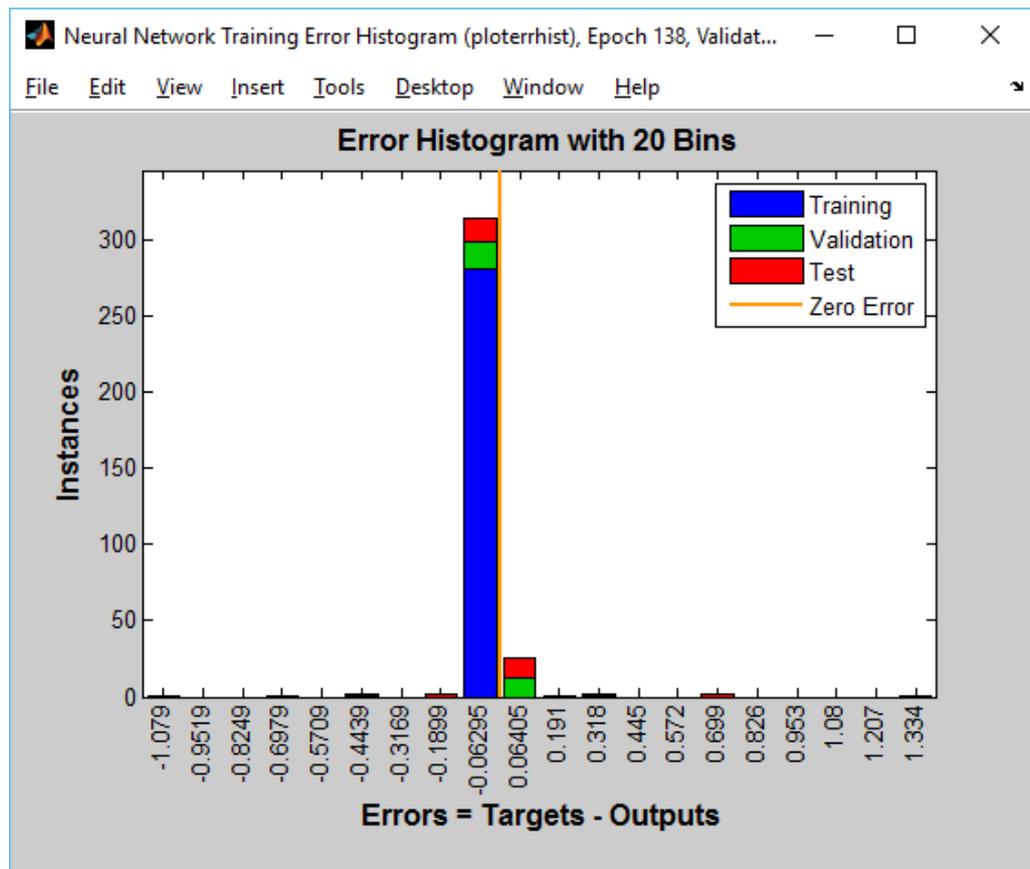


Рисунок 34 — Гистограмма ошибок

На следующем графике, который называется графиком линейной регрессии, сеть на трех рассмотренных подмножествах идеально аппроксимирует функцию, т.к. произошло практически полное совпадение выходов сети с целевыми значениями  $R = 1, a = 1, b = 0$ .

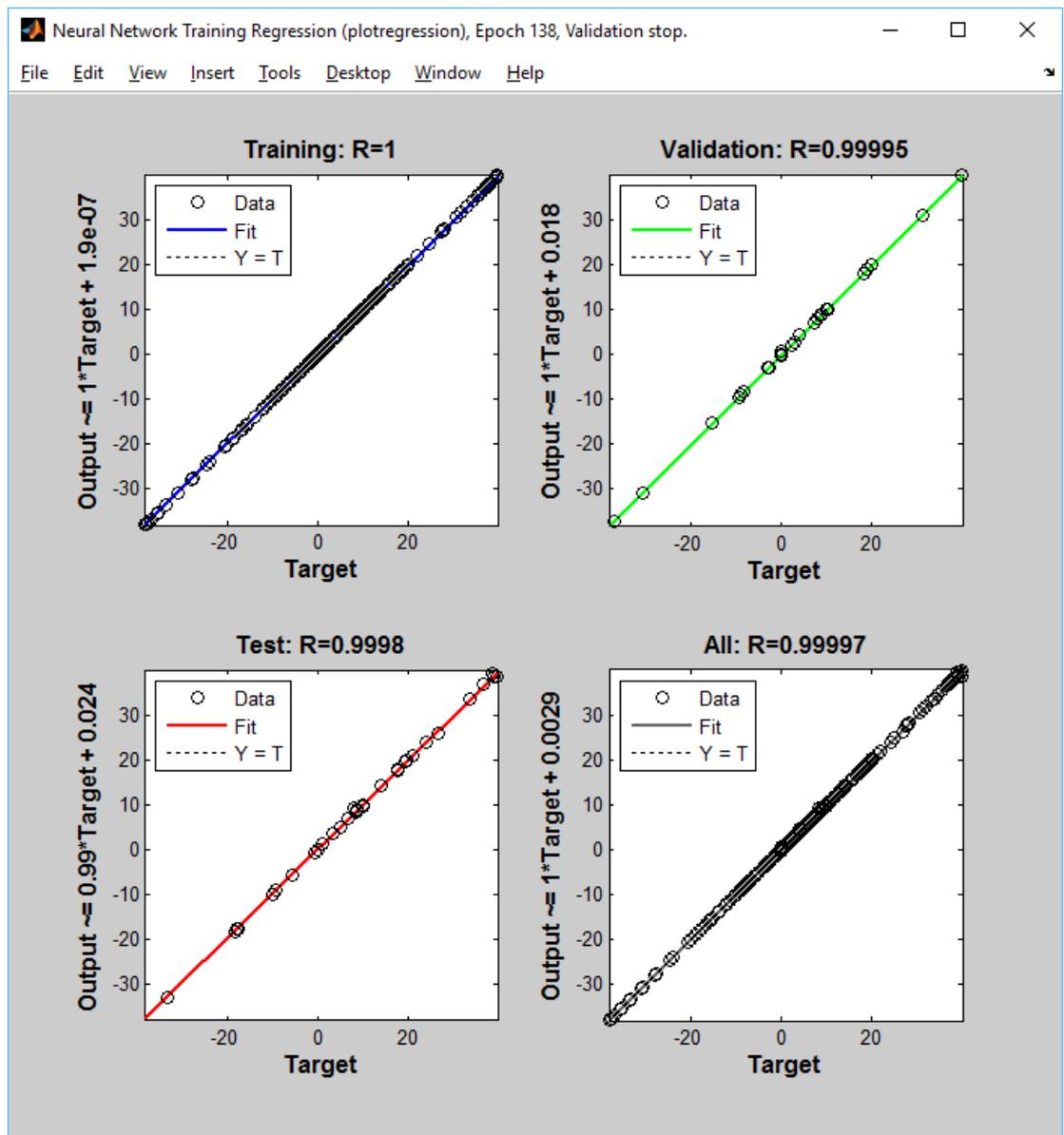


Рисунок 35 — Графики линейной регрессии результатов обучения

Сформируем блок обученной искусственной нейронной сети и перенесем его в схему RL-цепи, которая изображена на рисунке 35, так как она является основной для проверки результатов обучения.

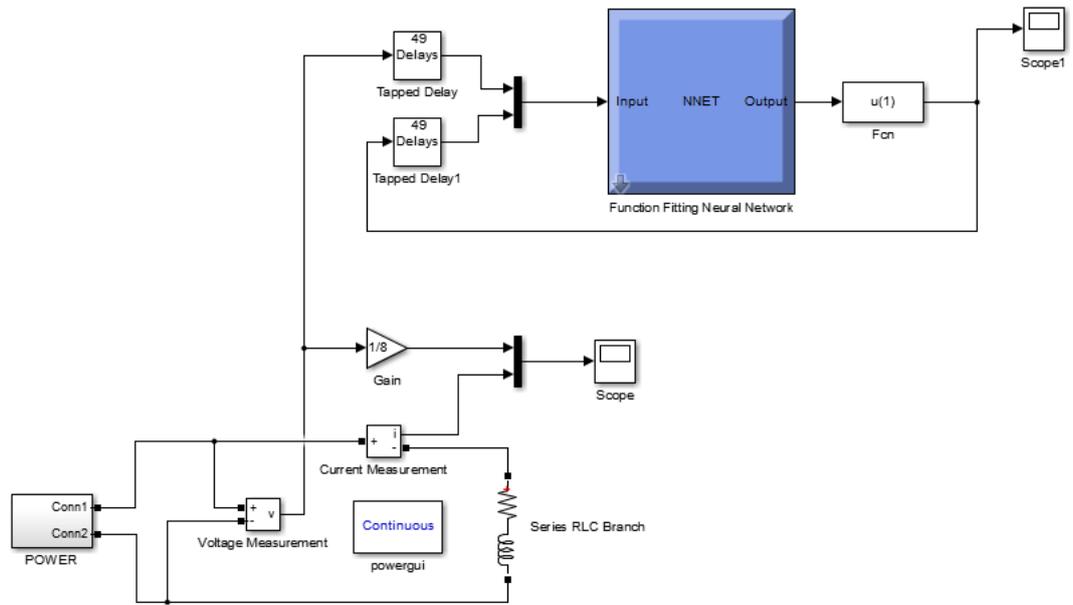


Рисунок 36 — Схема RL-цепи для проверки результатов правильности обучения нейронной сети

После запуска данной схемы следует рассмотреть получившуюся осциллограмму рассчитанного тока с помощью искусственной нейронной сети по методу Левенберга-Маркара. Она изображена на рисунке 36.

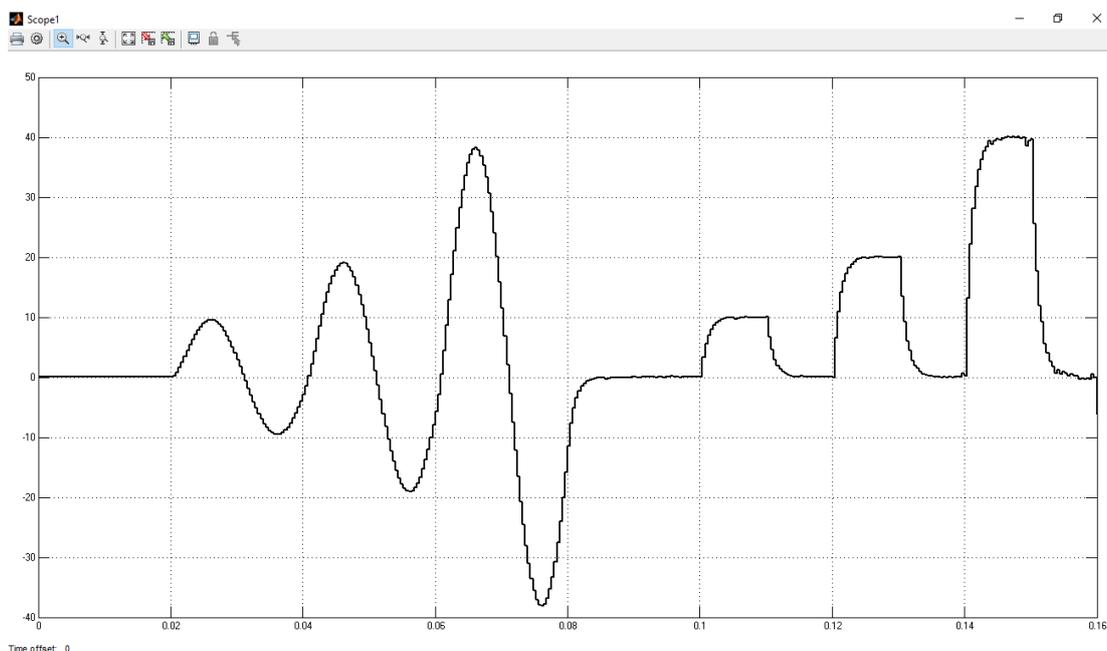


Рисунок 36 — Рассчитанный ток искусственной нейронной сетью методом Левенберга-Маркара

Сравнивая исходные данные токов тиристорного источника питания (рисунок 29) с полученными данными от обучения нейронной сети, следует отметить, что данный метод дает преимущественно отличный результат, который можно сравнить с методом традиционного математического моделирования для прогнозирования электрических нагрузок при помощи дифференциальных уравнений при небольших ресурсных и временных затратах.

## 4.2 Обучения искусственной нейронной сети методом байесовской регуляризации

На тех же данных обучим сеть с помощью метода байесовской регуляризации (функция `trainbr`). Сразу же следует отметить, что данный способ является наиболее «тяжелым» в плане нагрузки на оборудование. Отсюда происходит достаточно длительное количество времени на обучение. В данном случае было затрачено полное количество эпох, имеющее значение 1000.

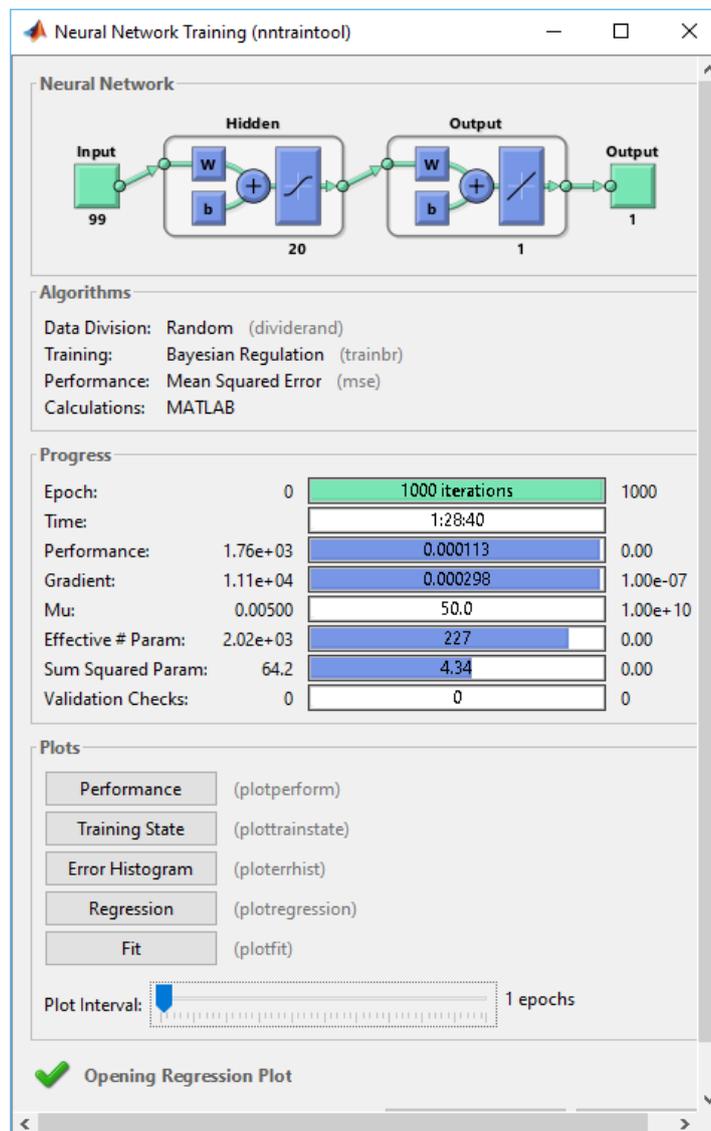


Рисунок 37 — Окно процесса обучения методом байесовской регуляризации

На графике обучения сети, изображенному на рисунке 38, можно увидеть, что среднеквадратичная ошибка достигла своего значения  $11,254 \times 10^{-4}$  на 1000-й эпохе. Также, как и в предыдущем случае функция обучения здесь используется функция ранней остановки в целях борьбы с переобучением сети.

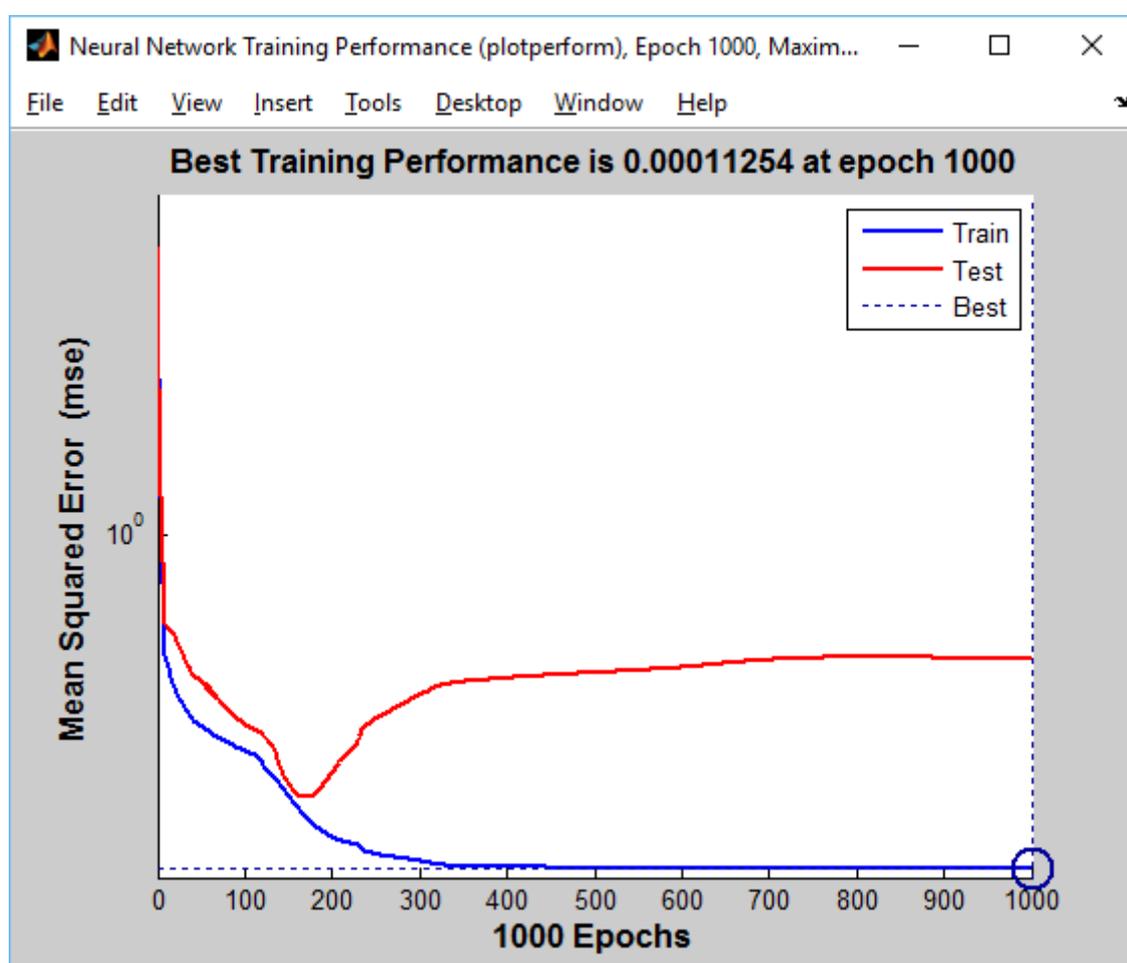


Рисунок 38 — График обучения сети с помощью метода байесовской регуляризации

На графике состояние обучения сети (рисунок 39) градиент достиг значения  $29,806 \times 10^{-5}$ , при этом на графике val fail можно увидеть ошибка себя никак не проявляет. На графике gamk, который обозначает эффективность обучения сети, достигает значения 226,8649.

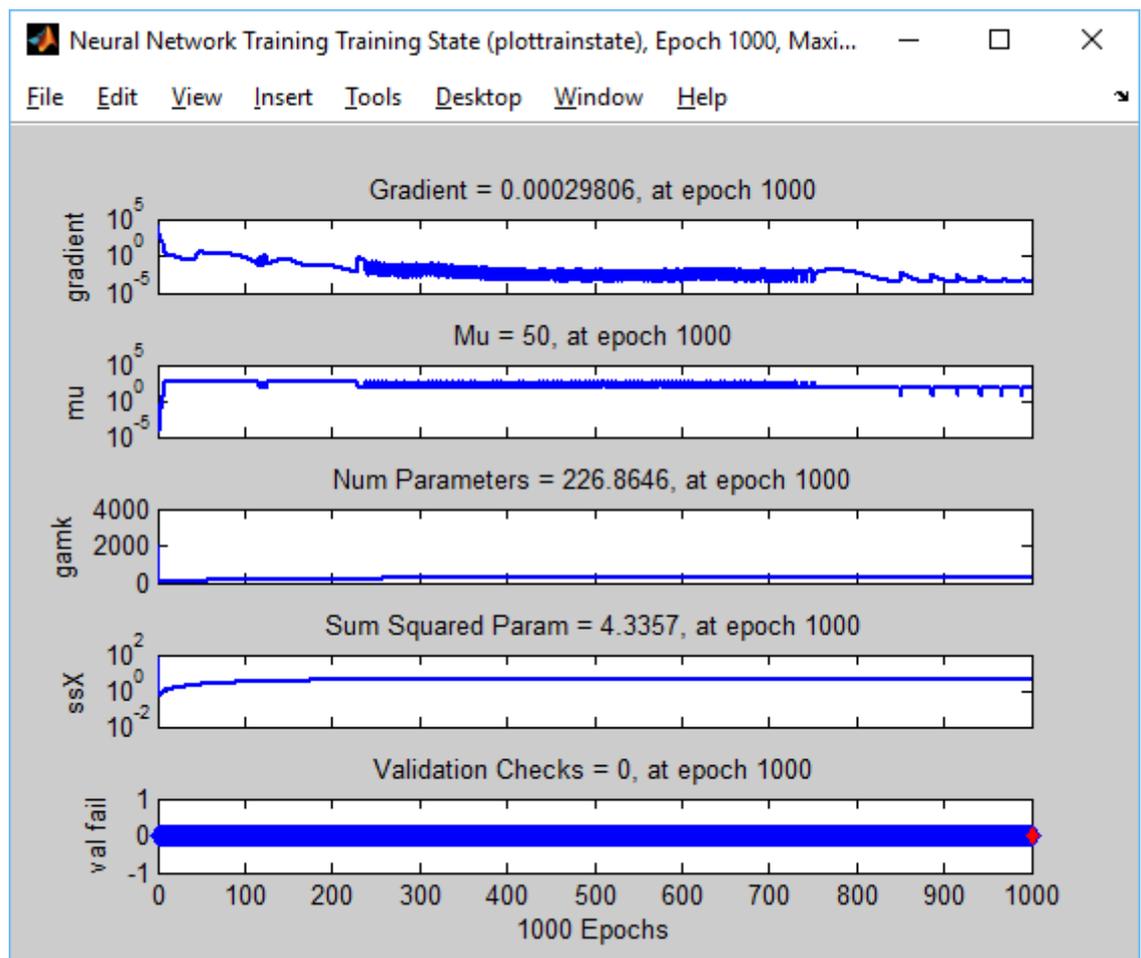


Рисунок 39 — График состояния обучения сети

Далее на гистограмме ошибок (рисунок 40) также, как и с использованием метода Левенберга-Маркара, ошибка не выходит за пределы своего диапазона погрешности, который составляет  $-0.00509$ .

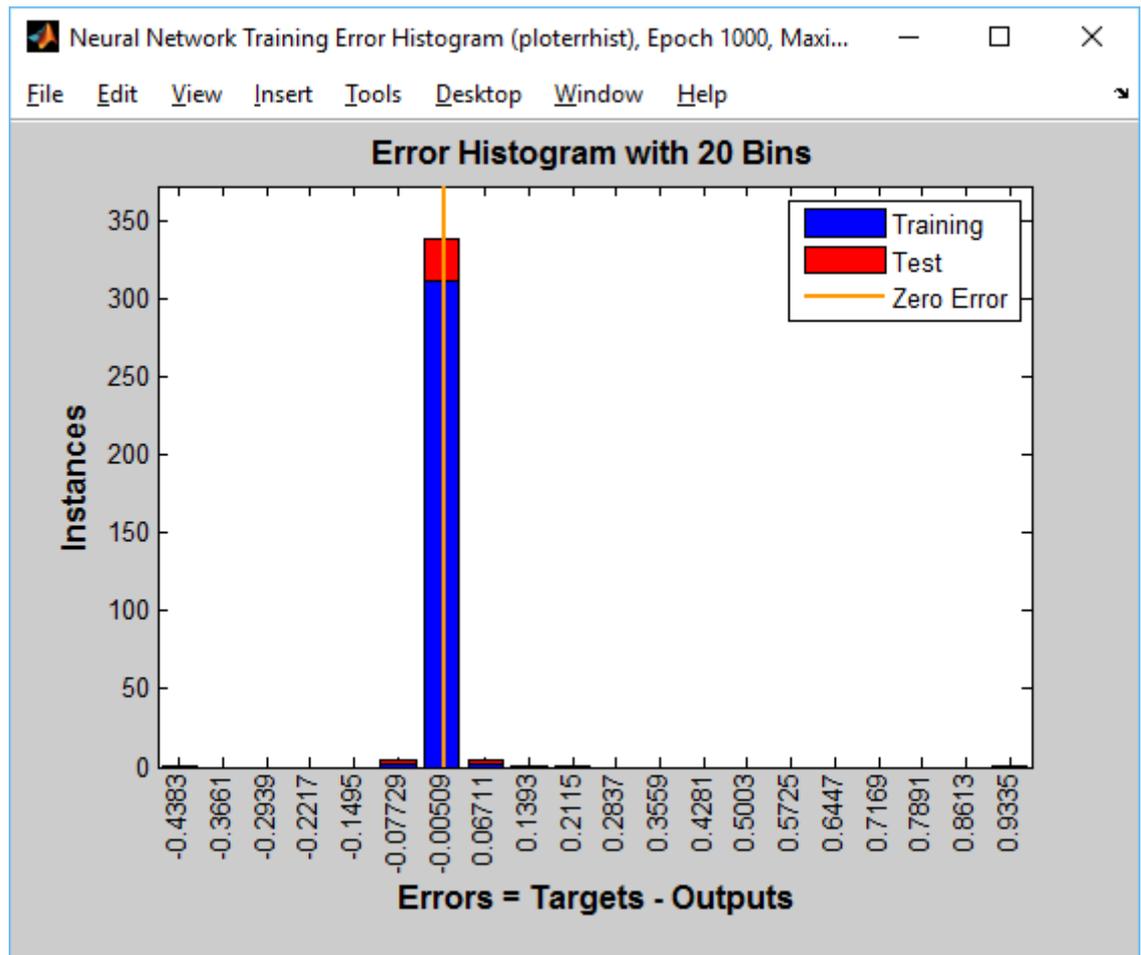


Рисунок 40 — Гистограмма ошибок сети, обученной по методу байесовской регуляризации

На получившемся графике линейной регрессии обученной нейронной сети, который изображен на рисунке 41, результаты обучения оказались практически идеальными, так как целевое значение  $R = 1$ .

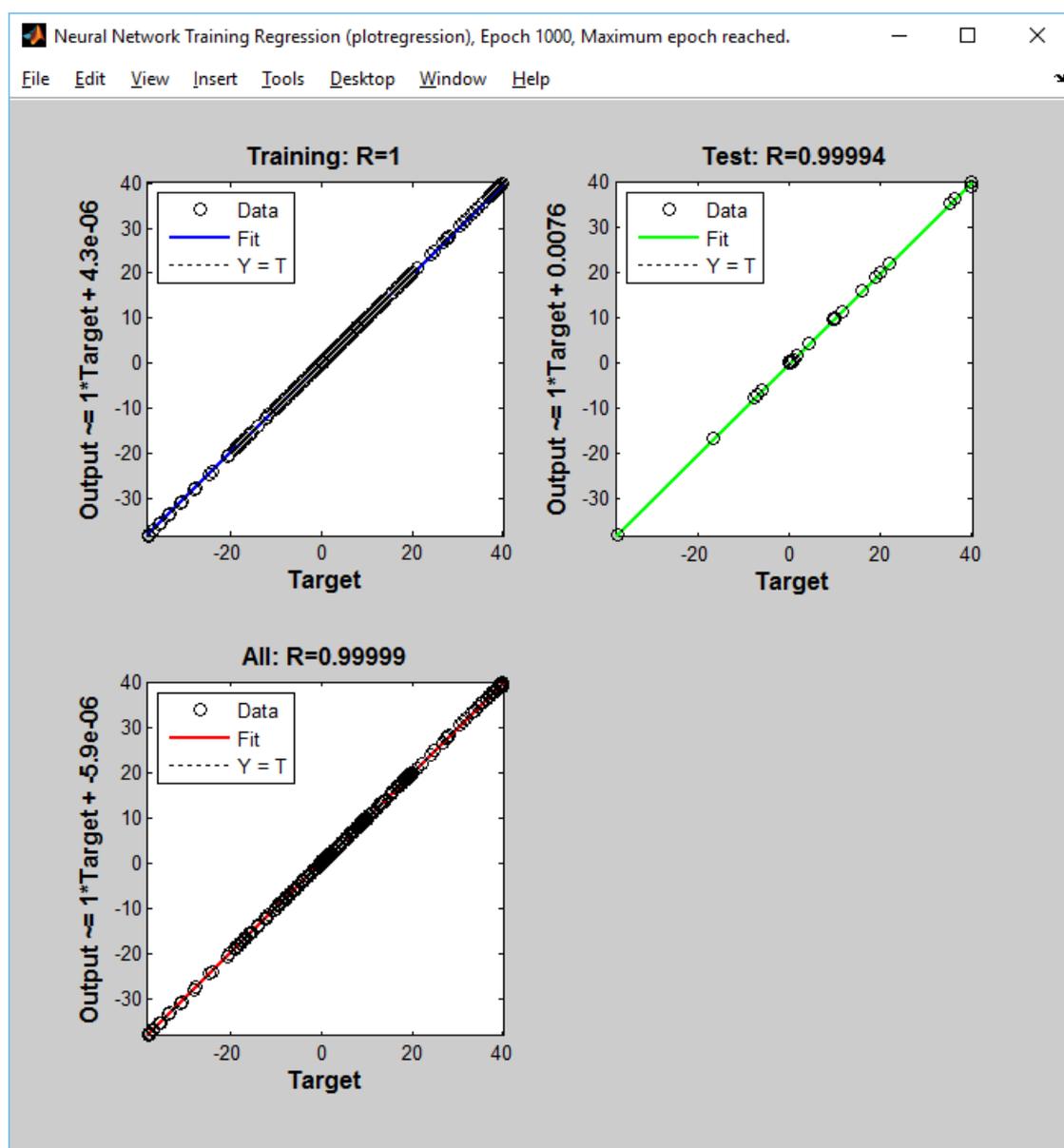


Рисунок 41 — График линейной регрессии обученной сети

Установим полученный блок обученной нейронной сети в схему RL-цепи для выяснения, насколько данный метод подходит для обучения. Полученная осциллограмма токов изображена на рисунке 42.

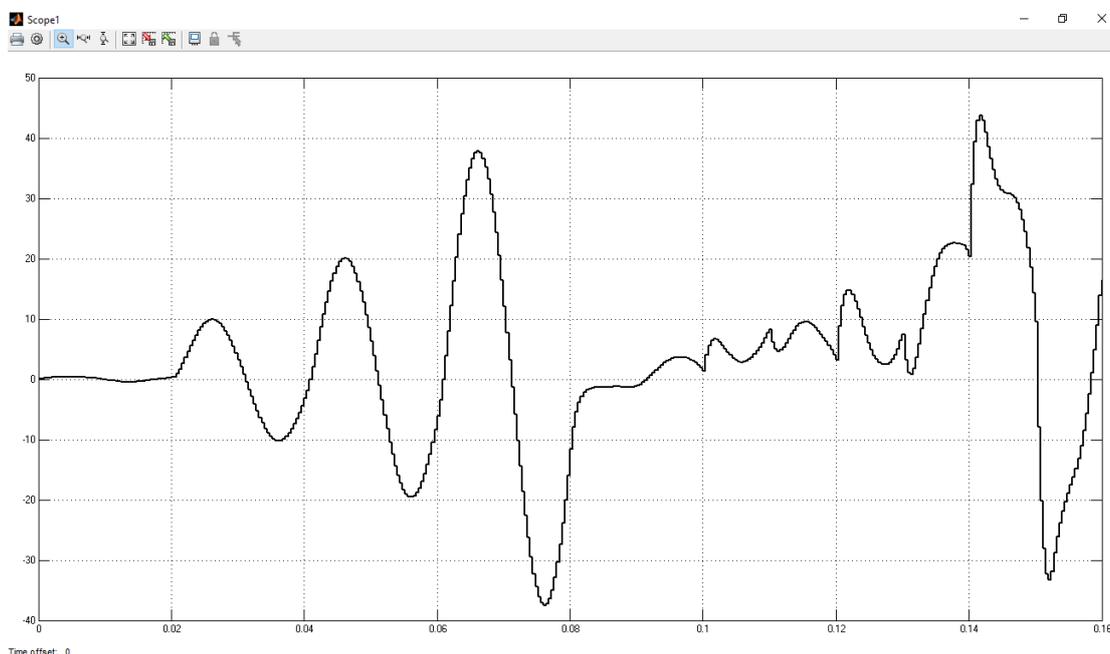


Рисунок 42 — Осциллограмма токов после использования нейронной сети, обученной методом байесовской регуляризации

Несмотря на положительный прогноз данного метода в обучении сети и последующего его применения на практике, из получившейся осциллограммы видно, что данный метод недостаточен для правильного последующего прогнозирования электрической нагрузки, несмотря на то что данный метод минимизирует комбинацию квадратов ошибок и весов. К сожалению, несколько попыток переобучения не дали положительного результата.



### 4.3 Обучение искусственной нейронной сети методом масштабируемого алгоритма сопряженных градиентов

Рассмотрим третий и последний метод, который носит названия масштабируемого алгоритма сопряженных градиентов (функция `trainscg`). В процессе изучения данный алгоритм показал себя наиболее быстрым в работе и наименее требовательным к оборудованию. Было использовано всего 72 эпохи для обучения.

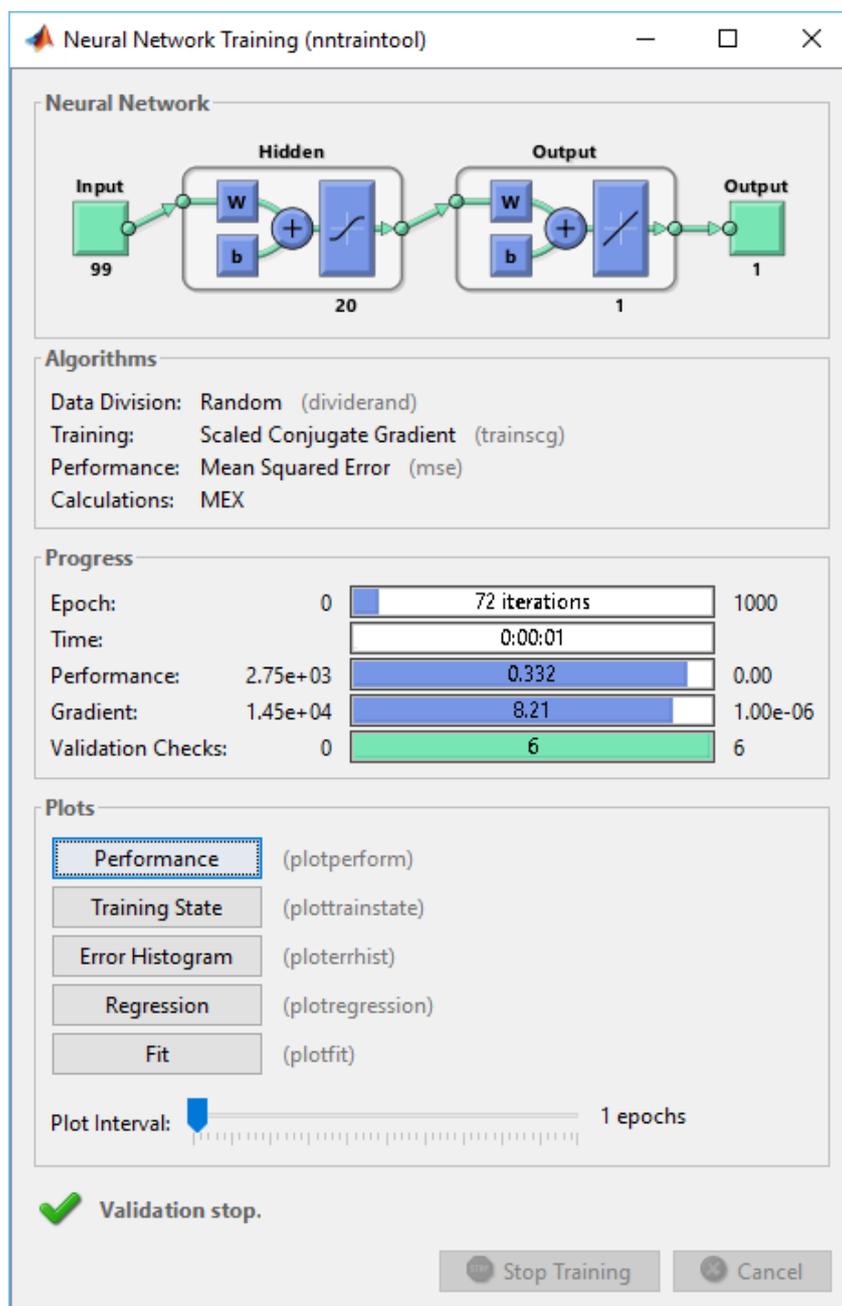


Рисунок 43 — Окно обучения нейронной сети с помощью метода алгоритма сопряженных градиентов

Перейдем непосредственно к графику обучения сети, который изображен на рисунке 44. В данном случае среднеквадратичная ошибка достигла значения  $5.3515 \times 10^{-2}$  на 66-й эпохе, что является наибольшим результатом среди других способов обучения.

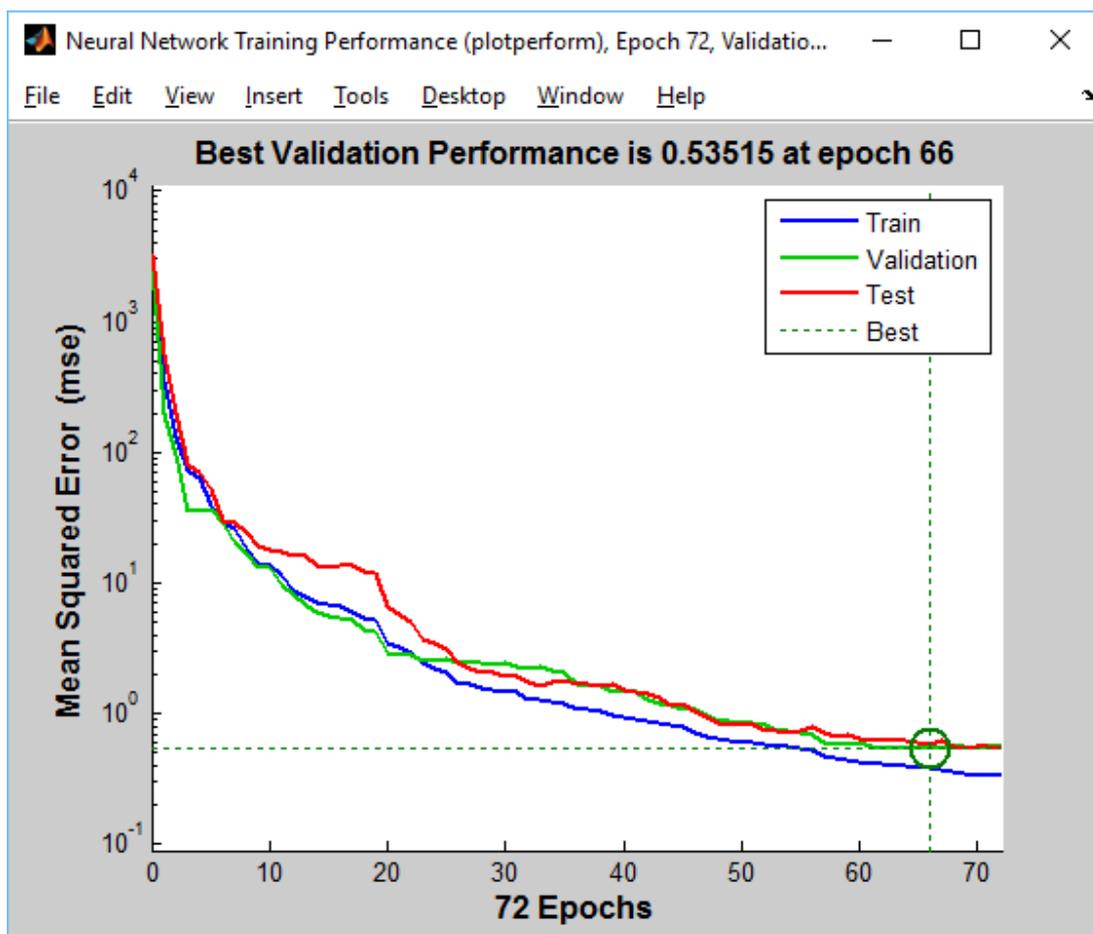


Рисунок 44 — График обучения сети с помощью метода алгоритма сопряженных градиентов

На рисунке 45 изображен график состояния обучения сети, который в свою очередь состоит из графика градиента и графика изменения ошибки на проверочном множестве. В данном случае градиент был равен 8,2147, а график val fail показал, что ошибка начала значительно расти только после 68 эпохи.

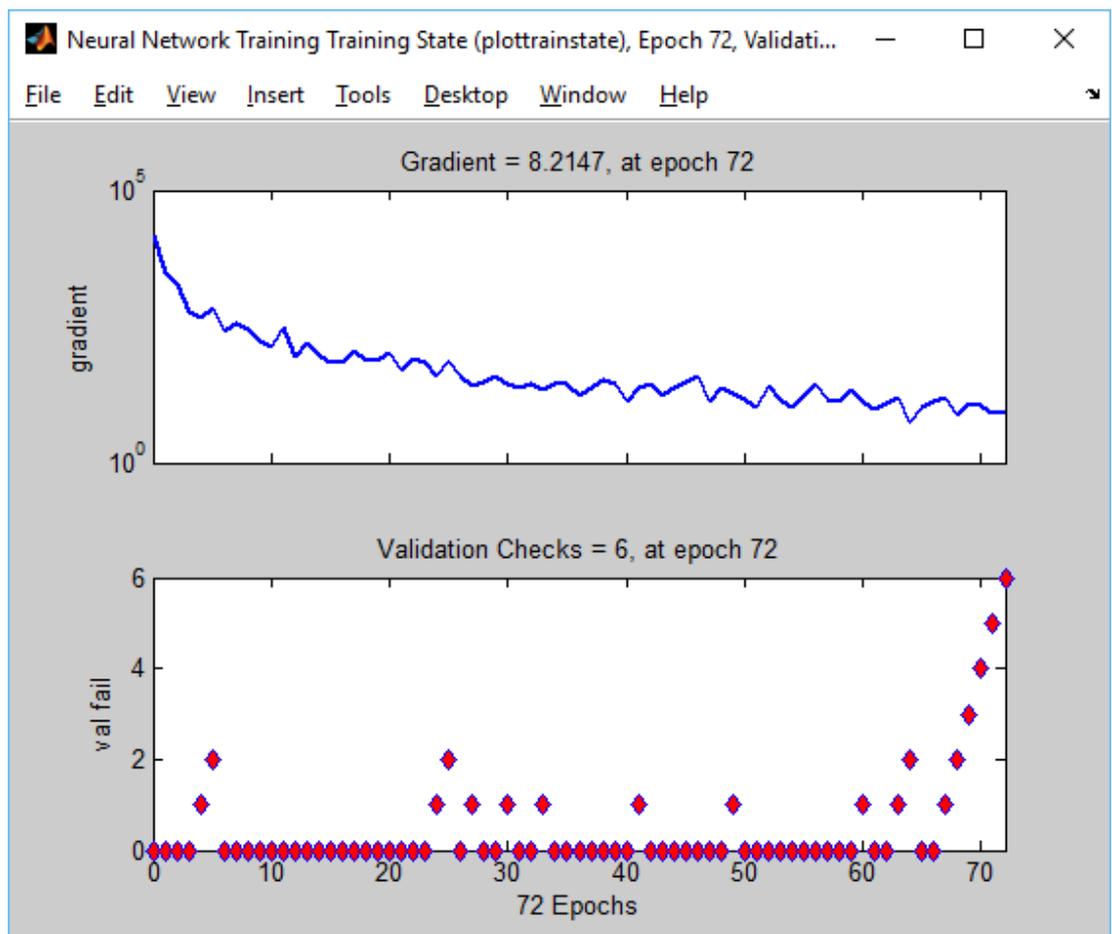


Рисунок 45 — График состояния обучения сети

Перейдем к гистограмме ошибки обученной сети (рисунок 46). В отличие от предыдущих методов, здесь можно увидеть большой диапазон погрешностей сети, который лежит в диапазоне от -0,6628 до 0,8176.

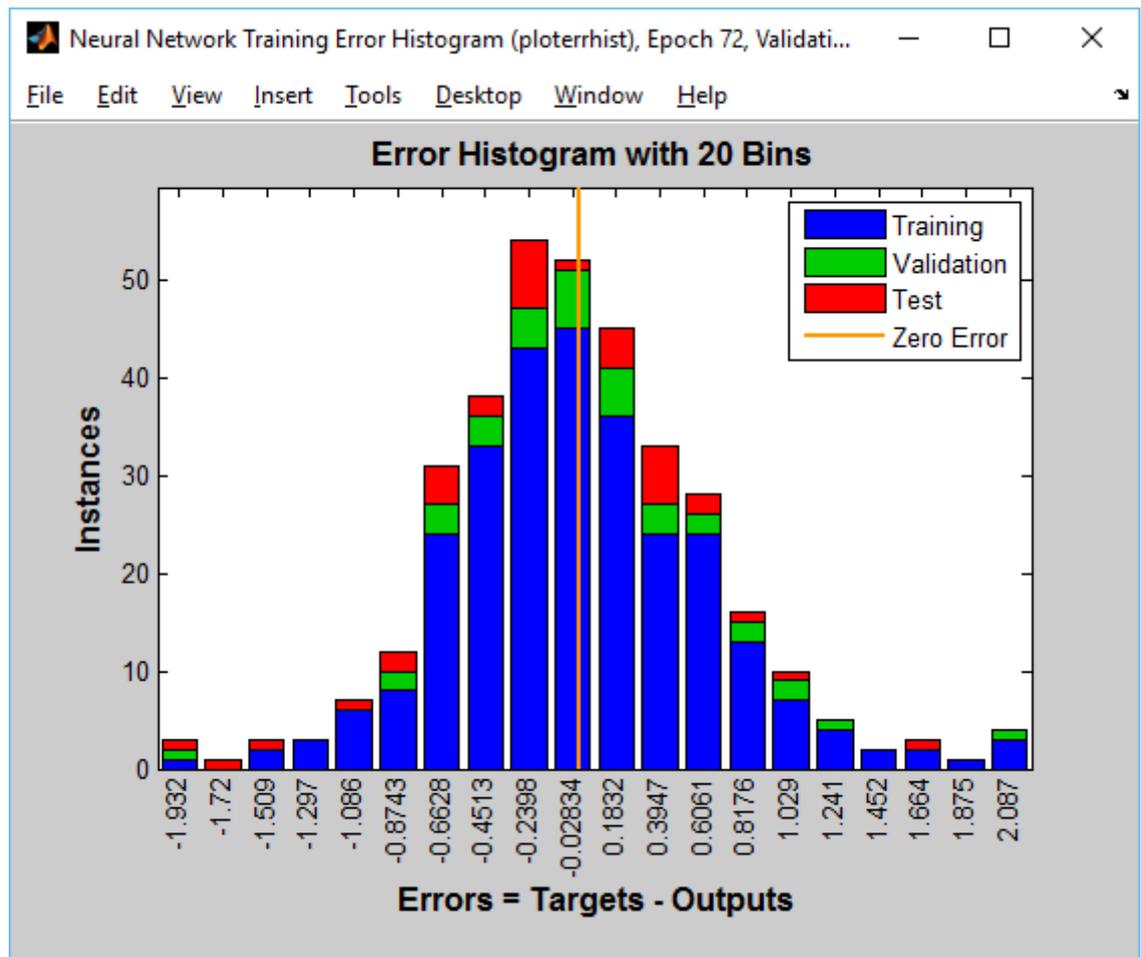


Рисунок 46 — Гистограмма ошибок обученной сети

Однако на графике линейной регрессии, изображенному на рисунке 47, отчетливо видно, что сеть, как и в предыдущих случаях, имеет практически идеальное значение регрессии для данного случая.

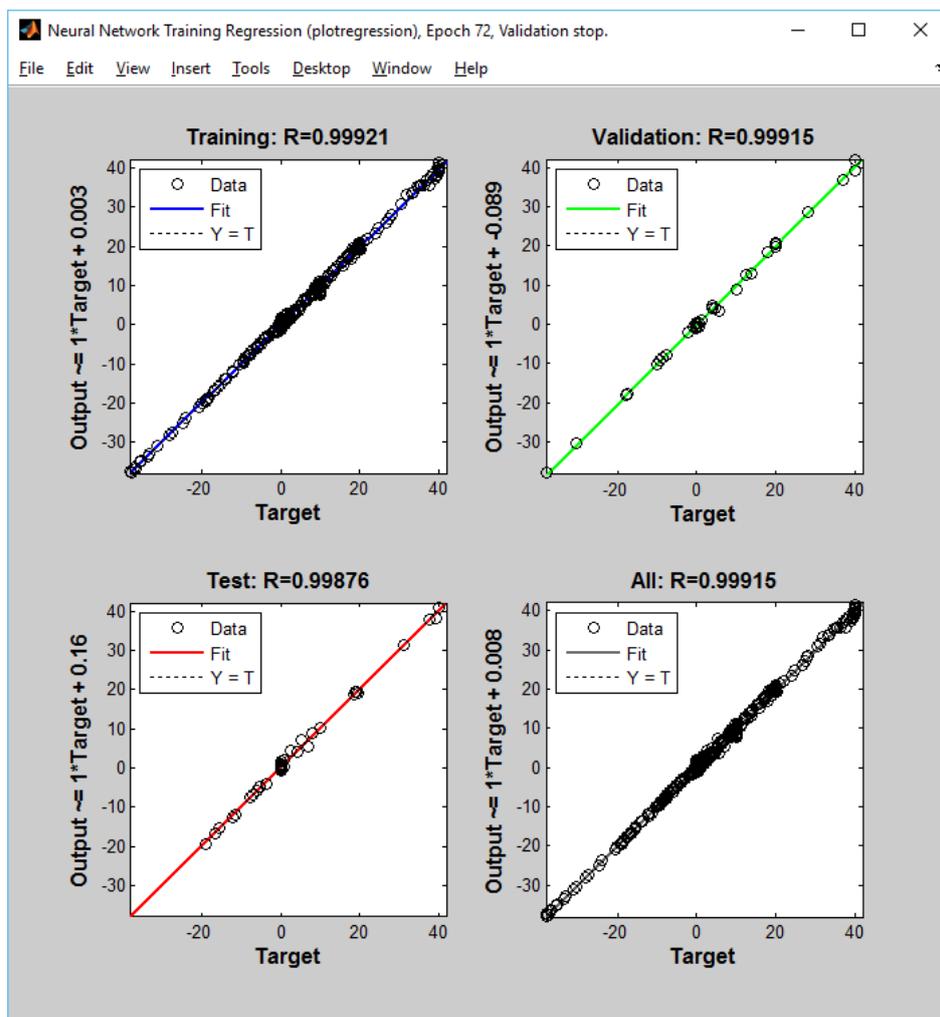


Рисунок 47 — Окно графика линейной регрессии

Применим полученный блок обученной нейронной сети и посмотрим на получившуюся осциллограмму токов после обучения нейронной сети, которая изображена на рисунке 48. Результат оказался еще более негативным, чем в случае применения метода байесовской регуляризации. Данная ситуация не исправилась и после нескольких

попыток переобучения с различными параметрами распределения данных.

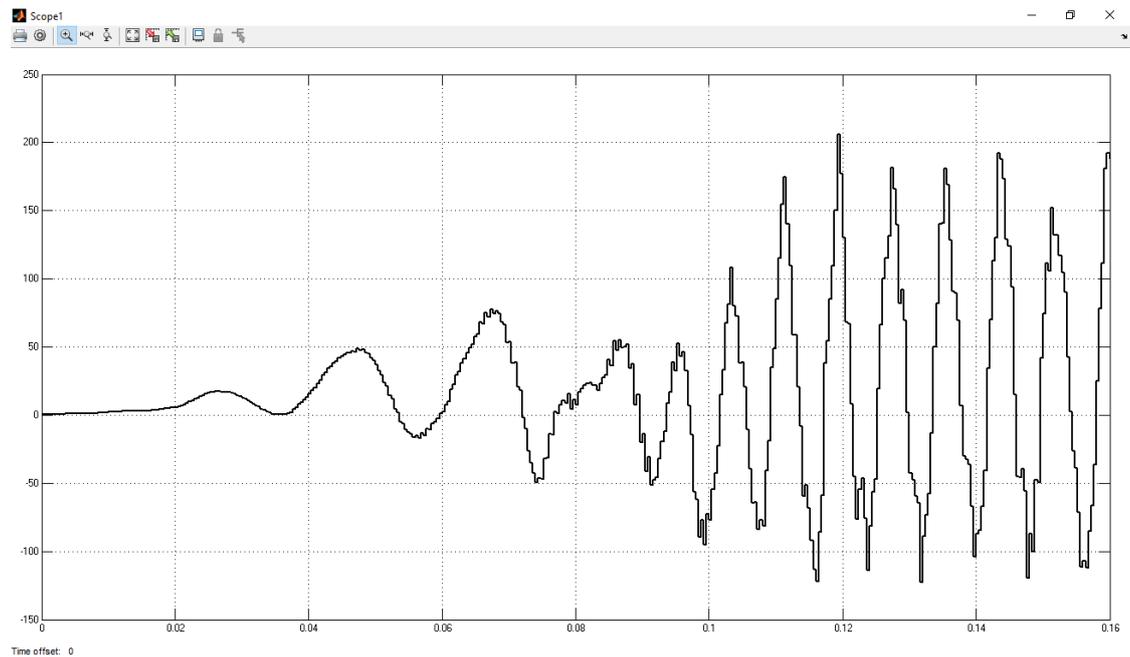


Рисунок 48 — Осциллограмма токов после использования обученной нейронной сетью с помощью метода алгоритма сопряженных градиентов

## Заключение

В данной диссертации была освещена проблема моделирования электрических нагрузок с помощью искусственных нейронных сетей, в частности моделирование источника питания контактного сварочного аппарата. Работа была проведена посредством применения программного обеспечения MATLAB версии 2014a с помощью включенного в него пакета Simulink и модуля для моделирования нейронных сетей Neural Network Toolbox.

Как видно из полученных осциллограмм токов, которые были обучены тремя предоставляемыми MATLAB методами, наибольшую эффективность в прогнозировании электрических нагрузок дает метод Левенберга-Маркара как с точки зрения точности данного моделирования, так и с точки зрения ресурсоемкости.

Это позволяет сделать вывод о том, что применение нейронных сетей для подобного рода моделирования является оправданным.

Ошибка при использовании данного метода является минимальной и ей можно пренебречь.

Нейронная сеть получилась универсальной, поэтому может работать как с различными значениями действующего напряжения, так и с различными по номиналу электрическими нагрузками.

Для достижения поставленной цели, а именно снижения трудовых, экономических и временных затрат на производстве, был выполнен ряд задач:

1. Была произведена разработка и моделирование структуры активно-индуктивной нагрузки в программном пакете MATLAB Simulink, которая в свою очередь была

использована как база для изучения эффективности использования искусственных нейронных сетей.

2. В процессе обучения нейронной сети различными способами был произведен анализ структуры и алгоритмов обучения на результат работы нейронной сети. В результате проведенного анализа выбрана топология в виде двухслойной сети с обратным распространением ошибки, которая была обучена по алгоритму Левенберга-Марквара.
3. В итоге была получена искусственная нейронная сеть, которая дала адекватные результаты работы при различных входных параметрах. Целесообразность применения ИНС была подтверждена посредством опытов.

Все выше перечисленные пункты проделанной работы позволяют убедиться в том, что искусственные нейронные сети имеют ценность в моделировании электрических нагрузок. Данное применение в целом снижает затраты на разработку и внедрение новых электротехнических установок и устройств в связи с ускорением их разработки, а также позволяет рассчитывать исходные и оптимальные режимы их работы.

## Список литературы

1. Шевцов А.А., Глибин Е.С. Имитационное моделирование совместной работы статических компенсаторов и контактной сварочной машины // Электротехника № 4, 2010 г. – с. 34 – 38.
2. Шевцов А.А., Глибин Е.С. Моделирование работы компенсационных устройств совместно с контактными сварочными установками // Сварочное производство №5, 2009 г. – с. 17 – 21
3. Моделирование нейронных сетей в MATLAB [Электронный ресурс]. URL: <http://robocraft.ru/blog/algorithm/560.html>
4. Neural Network Toolbox — Математика [Электронный ресурс]. URL: <http://matlab.exponenta.ru/neuralnetwork/index.php>
5. Тиристорные сварочные трансформаторы [Электронный ресурс]. URL: [http://www.welding.su/additional/additional\\_222.html](http://www.welding.su/additional/additional_222.html)
6. Использование Neural Network Fitting Tool [Электронный ресурс]. URL: <http://megaobuchalka.ru/5/14821.html>
7. Солодский С.А., Брунов О.Г., Ильященко Д.П. Источники питания для дуговой сварки // Издательство Томского политехнического университета, 2012 г.
8. Нейронные сети [Электронный ресурс]. URL: <http://www.statsoft.ru/home/textbook/modules/stneunet.html>
9. Агунов М.В. Представление составляющих электрического сопротивления для моделей нелинейных нагрузок. - Электричество №4/2004, с. 48-50.
10. Rosenblatt R. 1959. Principles of neurodynamics. New York: Spartan Books.

11. Медведев В. С. Нейронные сети / В. С. Медведев, В. Г. Потемкин. – М.: Диалог МИФИ, 2002.
12. Сапожников А.М., Лукашенко А.В. Использование нейронных сетей для краткосрочного прогнозирования нагрузки в энергетике. XXI Международная научная конференция «Математические методы в технике и технологиях». — Тамбов: Издательство ТГТУ, 2008.
13. Боровиков В.П. Нейронные сети. STATISTICA Neural Networks: Методология и технологии современного анализа данных – М.: Горячая линия —Телеком, 2008.
14. Мальцев В.В. Повышение эффективности энергопотребления в промышленности // Доклад на Международной конференции по энергоэффективности и возобновляемой энергетике ENES и REnergy 2011. – М., 2011. – С. 2.
15. Жданов А.А. Автономный искусственный интеллект. — М.: БИНОМ. Лаборатория знаний, 2008. – 359 с.
16. Пономарев В.А., Суворов И.Ф. Комплексный метод диагностики асинхронных электродвигателей на основе использования искусственных нейронных сетей//Новости электротехники.– № 5 (71). – 2011.
17. Данилюк А.В. Опыт применения искусственных нейронных сетей в электроэнергетике // Вестник гос. Ун-та "Львовская политехника": компьютерная инженерия и информационные технологии, 1999. – №380. – С. 116-120.
18. Гордиенко Е.К., Куссуль М.Э., Иваницкая Т.В., Сычев А.С. Моделирование нейронного классификатора для решения задач прогноза в электроэнергетике // Техническая электродинамика. – Киев: Институт электродинамики НАНУ, 1995. – №4, сс.61-65.

19. Осовский, С. Нейронные сети для обработки информации [Текст] / С. Осовский; пер. с польского И. Д. Рудинского. – М.: Фи-нансы и статистика, 2002. – 344 с.
20. Медведев, В. Г. Нейронные Сети Matlab 6 / В. Г. Медведев; под общ. ред. к.т.н. В. Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002. – 496 с. – (Пакеты прикладных программ; Кн.4).
21. Introduction to Neural Networks Using Matlab 6.0 [Электронный ресурс]. URL: <https://books.google.com/books?id=jJTN8RPgyXgC>
22. Галушкин А.И. Нейронные сети: основы теории. – М.: Горячая линия – Телеком, 2010. – 496 с.: ил.
23. Круглов В.В. Искусственные нейронные сети. Теория и практика. – 2-е изд., стереотип. – М.:Горячая линия – Телеком, 2002. – 382с.:ил.
24. Хайкин С. Нейронные сети: полный курс, 2-е изд., испр.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2006. – 1104 с.:ил.
25. Hebb D. O. 1949. Organization of behavior. New York: Science Editions.
26. Kohonen T. 1984. Self-organization and associative memory. Series in Information Sciences, vol. 8. Berlin: Springer verlag.
27. Rosenblatt R. 1959. Principles of neurodynamics. New York: Spartan Books.
28. Дьяконов В. П. MATLAB 5.3.1 с пакетами расширений /В. П. Дьяконов, И. В. Абраменкова, В. В. Круглов. – М: Нолидж, 2001.
29. Комашинский В. И. Нейронные сети и их применение в системах управления и связи / В. И. Комашинский, Д. А. Смирнов. – М.:Горячая линия – Телеком, 2002.
30. Области применения нейронных сетей [Электронный ресурс]. URL: [http://otherreferats.allbest.ru/mathematics/00074207\\_0.html](http://otherreferats.allbest.ru/mathematics/00074207_0.html)

