**«Санкт-Петербургский государственный электротехнический университет**

**«ЛЭТИ» им. В.И.Ульянова (Ленина)»**

**(СПбГЭТУ «ЛЭТИ»)**

|  |  |  |
| --- | --- | --- |
| **Направление** | 13.03.02–Электроэнергетика и электротехника | |
| **Профиль** | 13.03.02-10- Электропривод и автоматика | |
| **Факультет** | ЭА | |
| **Кафедра** | РАПС | |
| *К защите допустить* |  | |
| Зав. кафедрой |  | Белов М.П. |

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

БАКАЛАВРА

Тема: Разработка микропроцессорной системы управления вентиляцией для самосвала

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент(ка) |  |  |  | Дубовцев Н.К. |
|  |  | *подпись* |  |  |
| Руководитель | д.т.н., доцент |  |  | Белов М.П. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
| Консультанты | к.т.н., доцент |  |  | Козлова Л.П. |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |
|  |  |  |  |  |
|  | *(Уч. степень, уч. звание)* | *подпись* |  |  |

Санкт-Петербург

2021

**ЗАДАНИЕ**

**на выпускную квалификационную работу**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой аббревиатура названия кафедры |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Иванов И.И. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Студент(ка) | Иванов И.И. | | | |  | Группа | 0000 |
| Тема работы: Наименование темы | | | | | | | |
| Место выполнения ВКР: место выполнения ВКР | | | | | | | |
| Исходные данные (технические требования):  кратко указываются основные требования к ВКР | | | | | | | |
| Содержание ВКР:  Кратко указывается основное содержание ВКР, перечисляются ее основные разделы | | | | | | | |
| Перечень отчетных материалов: текст ВКР, иллюстративный материал, иные отчетные материалы | | | | | | | |
| Дополнительные разделы: указывается наименование дополнительного раздела | | | | | | | |
|  | | | | | | | |
| Дата выдачи задания | | | Дата представления ВКР к защите | | | | |
| «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. | | | | |
|  | | |  | | | | |
| Студент(ка) | |  | | Иванов И.И. | | | |
| Руководитель | |  | | Иванов И.И. | | | |
| *(Уч. степень, уч. звание)* | |  | |  | | | |
| Консультант | |  | | Иванов И.И. | | | |

**календарный план выполнения**

**выпускной квалификационной работы**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой аббревиатура названия кафедры |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Иванов И.И. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент(ка) | Иванов И.И. |  | Группа | 0000 |
| Тема работы: Наименование темы | | | | |

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование работ | Срок выполнения |
| 1 | Обзор литературы по теме работы | 00.00 – 00.00 |
| 2 | Наименование раздела | 00.00 – 00.00 |
| 3 | Наименование раздела | 00.00 – 00.00 |
| 4 | Наименование раздела | 00.00 – 00.00 |
| 5 | Оформление пояснительной записки | 00.00 – 00.00 |
| 6 | Оформление иллюстративного материала |  |

|  |  |  |
| --- | --- | --- |
| Студент(ка) |  | Иванов И.И. |
| Руководитель |  | Иванов И.И. |
| Консультант |  | Петров П.П |

*(Уч. степень, уч. звание)*

**РЕФЕРАТ**

Пояснительная записка 55 стр., 38 рис., 8 табл., 17 ист., 1прил.

СИСТЕМА УПРАВЛЕНИЯ, СИЛОВОЙ ШКАФ,ВЕНТИЛЯЦИЯ   
САМОСВАЛА, ОХЛАЖДЕНИЕ КЛЮЧЕЙ, МОДЕЛИРОВАНИЕ   
ЭЛЕКТРОПРИВОДА, АСИНХРОННЫЙ ЭЛЕКТРОВДИГАТЕЛЬ

Объектом исследования является СУ вентиляции силовых шкафов карьерных самосвалов.

Целью работы является применение на практическом задании, знаний полученных в процессе обучения, разработка СУ вентиляции силового шкафа карьерного самосвала, выбор основных средств автоматизации, а так же модуляция разработанной системы управления.

В процессе выполнения ВКР был произведен анализ существующих решений для вентиляции силовых шкафов самосвалов. Основываясь на проведенном анализе, был выбран способ охлаждения силовых ключей и разработана система управления, отвечающая за охлаждение силового шкафа грузового самосвала. Была разработана математическая модель данной СУ в математическом пакете MatLabSimulink и получены графики переходных процессовв различных режимах работы электропривода.

**ABSTRACT**

Explanatory note 55 pages,38 figures, 8 tables, 17sources, 1appendix.

CONTROL SYSTEM, POWER CABINET, VENTILATION DUMPING,  
 COOLING KEYS, MODELING ELECTRIC DRIVE, ASYNCHRONOUS ELECTRIC DRIVE

The object of research is the SU of ventilation of power cabinets of mining dump trucks.

The purpose of the work to consolidate and apply on practical tasks, the knowledge obtained in the learning process, the development of control systems for ventilation of the power truck of a mining truck.

In the course of the implementation of the WRC, an analysis was made of the existing solutions for ventilation of the power cabinets of dump trucks. Based on the analysis performed, the method of cooling the power switches was selected and a control system was developed that is responsible for cooling the power cabinet of the cargo dump truck. A mathematical model of this control system was developed in the mathematical package MatLab Simulink and graphs of transients were obtained in various operating modes of the electric drive.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 8](#_Toc67419750)

[1.Теория о нейронных сетях 9](#_Toc67419751)

[1.1 Математическое представление нейронных сетей 9](#_Toc67419752)

[1.2 Состав нейронной сети 11](#_Toc67419753)

[1.3 Методы обучения нейронных сетей 13](#_Toc67419754)

[2. Специализированные программы для работы с нейросетями 19](#_Toc67419755)

[2.1 Нейронные сети в Matlab 19](#_Toc67419756)

[2.2 Python для работы с нейронными сетями 20](#_Toc67419757)

[3 Создание и обучение нейронной сети 22](#_Toc67419758)

[3.1 Работа с нейронной сетью в Keras 22](#_Toc67419759)

[3.2 Создание модели нейросети в Matlab Simulink 23](#_Toc67419760)

[Обучение нейронной сети на однофазной модели 27](#_Toc67419761)

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

НС – Нейронная сеть

МОР – Метод обратного распространения ошибки

ПИД -

ВВЕДЕНИЕ

1.Теория о нейронных сетях

## Математическое представление нейронных сетей

. Нейронные сети – это попытка математиков различными законами и формулами описать работу биологической нейронной сети. Нейронная сеть представляет из себя последовательно соединенные между собой нейроны. Основной задаче нейронных сетей является получение определенных результатов, по которым будет выполняться определенное предположение, по набору значений подающихся на вход нейронной сети. Предположения могут быть различными от того какой цвет показывают НС, до того как правильно ответить на то или иное предложение.

Базовые компоненты, из которых строятся все нейронные сети это синопсис, нейрон и функции активации. Все эти компоненты отвечают за важные функции без которых нейронные сети не смогут корректно функционировать.

Синопсис это связи между нейронами. Они обладают всего одним параметром, который принято называть вес синопсиса или просто вес. Вес необходим для изменения входной информации передающейся от нейрона к нейрону. Этот параметр показывает насколько входная информация будет влиять на выход нейрона. Ниже представлена формула по которой происходит расчет параметров входящих в нейрон. 



где - вход нейрона;

 - вес синопсиса.

Нейрон это определенный математический оператор, который складывает значения входов. После к полученному значению прибавляется значение “смещения, если и формируется значения выхода. Рисунок нейрона представлен ниже.

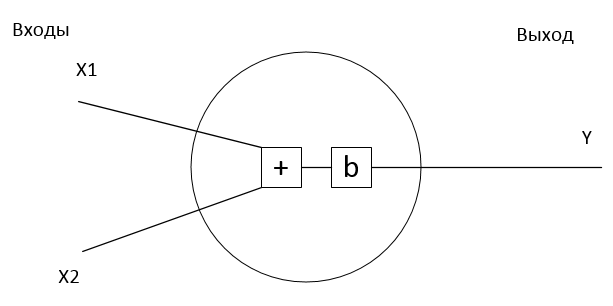


Рисунок 1 – Визуализация нейрона

Далее значение выхода подается на функцию активации для того, что бы упорядочить результат работы нейрона к определенному диапазону значений, что бы в дальнейшем передавать упорядоченный ответ следующим нейронам, для их более правильной работы и обучения. В таблице 1 представлены наиболее часто используемые функции активации.

Таблица 1 Функци Активации

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Функция | Диапазон значений | График функции |
| Линейная | F(x)=x | (;) |  |
| Сигмоид |  | [0;1] | Sigmoid function - Wikipedia |
| Название | Функция | Диапазон значений | График функции |
| Гиперболический тангенс |  | [-1;1] |  |
| Линейный выпрямитель |  | [0;) | How ReLU works in convolutional neural network - knowledge Transfer |

Функции активации являются важным инструментом в нейронных сетях на ровне с нейроном и синопсисам, влияющими на нейронную сеть, на ее работу и на ее обучение. Эти функции от части определяют какие нейроны будут активированы и какая информация будет передаваться дальше последующим нейронам.

Линейную функцию активации использую редко. В основном ее используют если необходимо передать данные с выходов нейронов дальше без изменений. Чаще всего в качестве функции активации применяют сигмоиду, её также называют логической функцией. Так как регуляторы могут выдавать и отрицательные и положительные значения поэтому функции Сигмоид в работе не подойдет для работы с числами в данном диапазоне лучше всего использовать гиперболический тангенс.

## 1.2 Состав нейронной сети

Для того чтобы правильно описать состав нейронной сети необходимо ввести понятие слоя. Слой – это набор, не связанных между собой нейронов, на который подаются значения либо от предыдущих нейронов, либо из в не сети. Ниже на рисунке представлены, какие слои входят в состав нейронной сети.

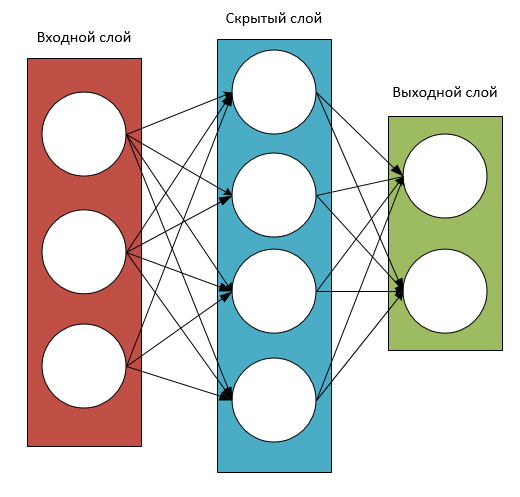


Рисунок 2 – Слои в нейронной сети

Нейронные сети состоят из трех основных слоев:

Входной слой – служит для подачи значений в нейронную сеть;

Скрытый слой – основной вычислительный слой, в котором происходит основная работа нейронной сети. Скрытый слой может состоять из одного или нескольких слоев;

Выходной слой – предназначен для вывода результатов работы нейронной сети.

Выходы нейронов каждого слоя связаны синопсисами с входом каждого нейрона следующего слоя.

Нейросеть может иметь любое количество слоев, и в них может быть любое количество нейронов. Эти параметры влияют на работу нейронной сети и на ее обучения. Не существует формул по которым от количества входных параметров или от типа передаваемых данных или от их диапазона можно было бы рассчитать количество слоев и количество нейронов в каждом слое для корректной работы сети. Именно поэтому значения этих параметров подбираются эмпирическим путем.

## 1.3 Методы обучения нейронных сетей

Выделяют четыре основных метода обучения нейронных сетей, каждый из них имеет свои плюсы и свои минусы.

**Машинное обучение с учителем.**

Этот тип обучения подразумевает, что во время обучения вместе с входными параметрами нейронной сети в данных для обучения, как их еще называют дата сеты, присутствуют значения выходных параметров сети. Во время обучения нейроная сеть сравнивает полученные данные на своем выходном слое с данными, которые у нее должны были получиться при данных значениях входного слоя. И по методу обратного распространения ошибки меняет весовые коэффициенты своих синопсисов. После этого данные ещё раз подаются, и происходит сравнение выходов и так до тех пор, пока значение разницы не достигнет нуля или не будет близкой к нулю. Тогда можно считать, что сеть обучилась.

Так же для дальнейшего понимания методов обучения необходимо рассказать про метод обратного распространения ошибки. Для подсчёта разницы между выходом и значением, к которому она стремится, используют следующую формулу.

1)

2),

где  - это разница для выходного слоя;

 – разница для скрытых слоев;

 - это идеальный выход;

- это то значение которое рассчитала НС;

 – это производная функции активации слоя;

 – вес синапса;

 – дельта ошибки предыдущего слоя.

Так как нейроны, содержащиеся на выходном слое, не имеют исходящих синапсов, то для подсчета дельты необходимо пользоваться первой формулой. Для нейронов, содержащихся во входном и скрытом слое, второй.

Так как при расчете МОР необходимо использовать производную функции активации, то функций активации необходимо выбирать такие, что бы их можно было продифференцировать. Так же для упрощения расчетов для некоторых функций активации формулу производной можно заменить на упрощенную:





Далее необходимо найти градиент для каждого входящего в нейрон синапса, что бы в дальнейшем правильно изменить его вес. Значение градиента будем искать по следующей формуле.



,где А - начало синапса;

B - конец синапса.

Теперь по формуле представленной ниже мы можем рассчитать вес, на который нам необходимо изменить вес синапса.



где  – скорость обучения;

 – момент обучения;

 – это вес синапса на предыдущем шаге итерации.

Далее необходимо полученное значение сложить с весом синопсиса.



Чаще всего обучение с учителем используется для решения двух типовых задач это задачи классификации и задачи регрессии.

Задачи классификации чаше всего применяются, когда нужно определить тот или иной предмет на фото. Фотография разбивается на пикселе и каждый пиксель по отдельности подается на вход нейронной сети, таким подходом можно добиться от нейронной сети четкого определения цвета пиксели, и по расположению пикселей обучить на распознавание предметов. Каждое изображение связанно с определённым наборов нейронов в выходном слое. Например, необходимо различить автотранспорт, изображенный на фото. На вход нейронной сети будут подаваться фотографии с изображением поезда, автомобиля или мотоцикла. Каждому изображению будет дана метка соответствующему определенным значениям выходного слоя. Например, когда на вход подаётся изображения поезда, то на выходном слое нейроны должны выдать результат [1;0;0]. При подаче на вход изображения автомобиля на выходе должны быть значения [0;1;0], а при загрузке изображения мотоцикла [0;0;1] соответственно.

После обучения нейронной сети и при загрузке на ее входной слой изображения, не участвовавшего в обучения, автомобиля нейросеть должна выдать результат похожим на следующий:[0.05;0.99;0.05], что говорит о том, что нейросеть на 99% уверена, что на изображении автомобиль.

Задача регрессии связана с непрерывной подачей данных на вход нейросети. Одним из примеров регрессии является линейная регрессия. Нейросеть при получение конкретного значения x должна вычислить ожидаемое значение.

Обучение без учителя.

Иногда собрать обучающую выборку для нейросети, где каждому значению входа соответствует определенное значение выходов достаточно сложно, потому что это может занять много времени или потратить огромное количество ресурсов. Именно поэтому используются метод обучения без учителя. Перед нейросетью ставится задача самостоятельно найти корреляции в данных, извлекая полезные признаки и анализируя их.

Одним из примеров такого типа обучения является кластеризация. Нейросеть находит общие признаки у данных и группирует их вместе. На рисунке ниже приведен пример кластеризации.



Рисунок 3 – Пример кластеризации данных

По большому количеству данных нейронная сеть разделила в разные группы обувь и подарки.

Обучение с частичным привлечением учителя.

Этот метод признано считать самым лучшим, когда обучение происходит на большой выборке данных, потому что он включает в себя и обучение с учителем, и обучение без учителя. Обучающая выборка при таком обучение имеет значения входных данных с выходным результатом и без результатов. Такой метод обучения применяют, если из набора данных трудно извлечь общие признаки или набор данных настолько велик, что дать значения выхода каждому набору входных параметров трудоемкая задача.

Подобный метод машинного обучения получил широкое применение в медицине, в анализе рентгенов и МРТ.

Обучение с подкреплением.

В отличие от машинного обучения с учителем, при данном типе обучения на вход нейронной сети не подается обучающая выборка. Вместо этого сеть обучается “методом проб и ошибок”. При таком подходе обучения нейросет пытается найти оптимальный путь для достижения цели. Если нейросеть ошибается, то она получает “штрафные балы” и начинает обработку данных с начала. Если нейросеть предпринимает действие, которое приведет ее к искомо верному результату, то она получает “награду” и продолжает работу. В конечном итоге нейросеть обучается предсказывать свое следующие действие для получения максимально “награды”.

Каноничным примером обучения с подкреплением является обучение нейросети управлять машинкой, чтобы пройти заданный маршрут как можно быстрее.

Для осуществления поставленной задачи из машинки испускается от шести до восьми лучей, как показано на рисунке 4, которые являются в свою очередь входами для нейронной сети. Эти лучи передают в нейронную сеть расстояние, до какого либо препятствия. По полученным данным НС корректирует движение машины и избегает столкновение с объектом. По схожему принципу работает автопилот в автомобилях компании Tesla, только в данном случае лучи передают не только расстояние до предмет, а так же способны различать этот предмет и в зависимости от его типа принимать то или иное решение.

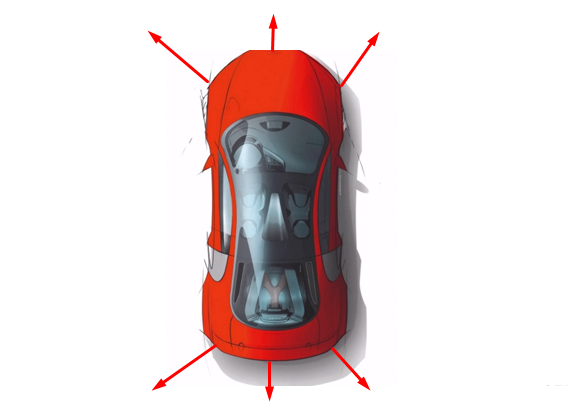


Рисунок 4 – Обучение нейронной сети с подкреплением

Во время обучения машина методом проб и ошибок учится не врезается в препятствия и проходить трассу как можно быстрее.

Такое метод обучения чаще всего используется в компьютерных играх для управления искусственным интеллектом, и в обучение роботов, которые управляют автономными транспортными средствами без участия человека.

Для решения задачи поставленной в рамках данной диссертации больше всего подойдет метод обучения с учителем по нескольким причинам:

Во-первых, в обучающей выборке будет заранее известно какое значение по входным параметрам должна получить нейросеть, для того чтобы с имитировать работу ПИД регулятора.

Во-вторых, обучающая выборка будет получена в ходе работы реального регулятора на объект. Поэтому в разметке обучающих данных не будет проблем.

В-третьих, данный тип обучения решает проблемы регрессии и постоянного потока данных, что будет определяющим фактором при имитации работы ПИД регулятора.

2. Специализированные программы для работы с нейросетями

Из-за большой популярности нейронных сетей существует огромное количество программ позволяющих работать с ними. Все эти программы имеют свои плюсы и свои минусы.

## 2.1 Нейронные сети в Matlab

Matlab – это мощный пакет прикладных программ для решения задач технических вычислений. Он имеет огромное количество библиотек позволяющих смоделировать любую ситуацию. Одна из таких библиотек под названием “Deep Learning Toolbox” позволяет работать с нейросетями. В своем арсенале она имеет уже готовые примеры контроллеров на основе нейросетей, которые можно обучить и настроить на работу.

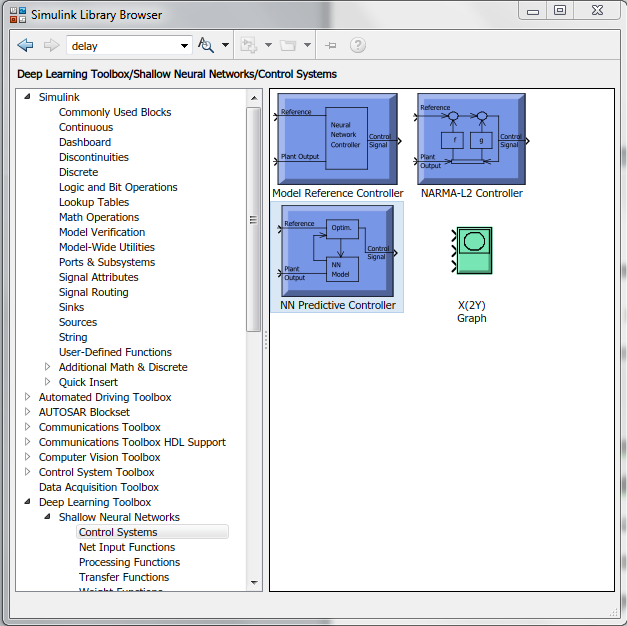


Рисунок 5 – Библиотека Deep Lerning Toolbox

Так же в эту библиотеку входят блоки, с помощью которых можно составить свою нейросеть, смоделировать функции активации

Так же в Matlab существует приложение **Deep Network Designer,** которое визуализирует создание глубоких нейронных сетей.

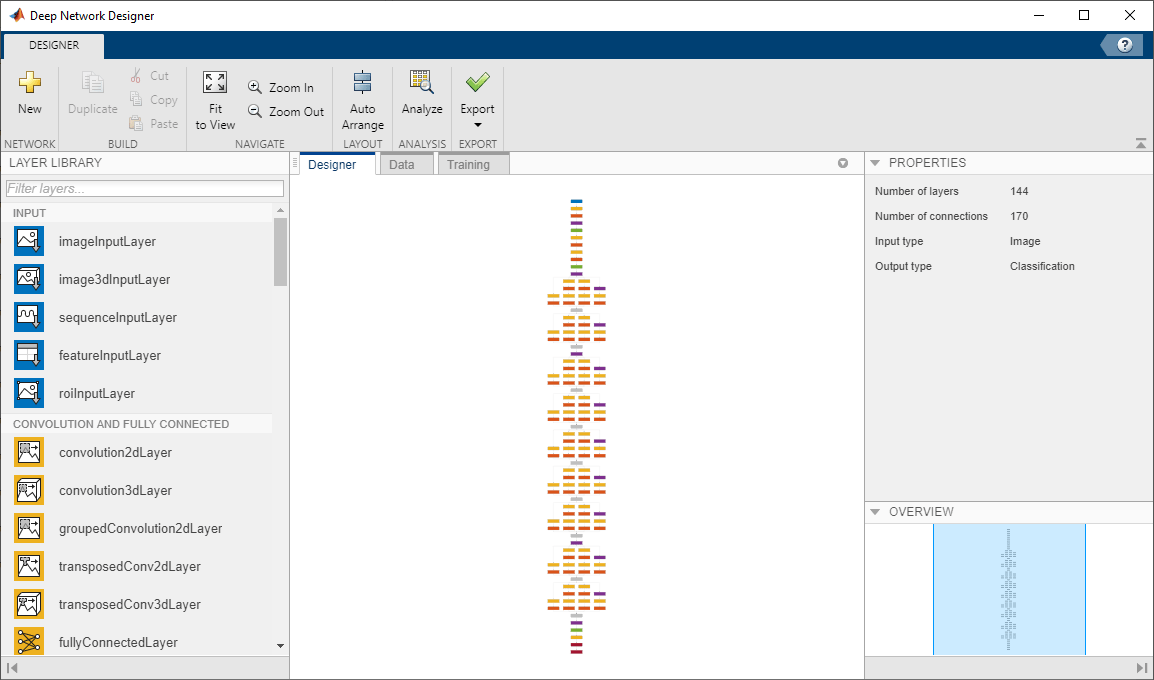


Рисунок 6 – Интерфейс приложения Deep Network Designer

Приложение интуитивно понятно из доступных блоков строится любая топология сети. Все блоки можно подробно настроить под свои нужды. Так же matlab предоставляет большой архив с уже готовыми нейросетями, что помогает пользователю научится правильно выбирать и строить топологию сетей и смотреть уже готовые решения для некоторых проблем.

## 2.2 Python для работы с нейронными сетями

Большую популярность для работы с нейросетями получил язык программирования Python. Python – это высокоуровневый язык программирования с динамической типизации и автоматически управлением памяти. Основной идеей языка является читаемость кода и повышение производительности разработчика, за счет того что многий функционал язык делает за разработчика, например работа с памятью. Python является интерпретируемым языком, что уменьшает скорость работы написанных на нем программ и большим потреблением памяти. Из-за своей популярности и открытости кода python имеет огромное количество библиотек, в том числе и для работы с нейросетями. Одной из таких библиотек является Keras.

Keras является открытой нейросетевой библиотекой, которая представляет из себя надстройку на фреймворками TensorFlow и Theano. Основными преимуществами этой библиотеки перед другими является ее компактность, модульность и предоставление интуитивно более понятных функций, которые помогают, без особых усилий создавать нейронные сети.

В данной работе работа с нейросетями будет вестись с помощью языка программирования Python его библиотеки Keras,потому что “Deep Learning Toolbox” хоть и имеет уже готовые котроллеры на основе нейросетей, но они сложно настраиваемые и не позволят изменить топологию сети и имеют сложный интерфейс и методы работы. Приложение **Deep Network Designer** удобно в использование, имеет обширные возможности для построения нейросетей, но оно в основном заточено на глубокие нейросети и работу с изображениями.

3 Создание и обучение нейронной сети

## 3.1 Работа с нейронной сетью в Keras

Для создания и обучения нейронной сети как говорилось в предыдущей главе будет использоваться язык Python и его библиотека Keras.

В первую очередь необходимо выбрать архитектуру нейронной сети и передать это значение в модель. Выполняется это с помощью следующей команды.

model = keras.Sequential()

Sequential – означает, что будет использоваться линейный стек слоев.

Дальше необходимо добавить слои в модель с помощью функции add().

model.add(Dense(10, input\_shape=(2,), activation='relu',use\_bias=False))

Первое число говорит о том, сколько нейронов будет находиться в слое. Так как вначале создается входной слой то необходимо в параметр input\_shape=(2,) передать число, которое скажет программе какое количество нейронов будут входными. Параметр activation отвечает за то, какая функция активации будет использоваться в данном слое. Функции активации описаны в первой таблице. Параметр use\_bias говорит системе хочет ли разработчик использовать смещение или нет.

С помощью функции add можно добавить в модель сколько угодно слоев.

После того как все слои добавлены в модель необходимо данную модель скомпилировать. Выполняется это с помощью функции

model.compile(loss='mean\_squared\_error', optimizer=keras.optimizers.Adam(0.1)).

Параметр loss - . Это значение, которое модель пытается минимизировать. В данном примере выбрана функция средне квадратичной ошибки

После компиляции модель готова к обучению. Для того чтобы приступить к обучению необходимо воспользоваться функцией fit.

history = model.fit(train, rezults, epochs=10)

В переменную History будет записана история обучения сети.Ее ошибки и точность после каждой эпохи. С помощью переменной history можно визуализировать обучение сети и получить графики по ошибкам и точности сети за весь период обучения.

Переменная train является массивом векторов, которые будут подаваться в сеть для ее обучения. Размер входного вектора должен быть равен количеству входных нейронов.

Переменная results это тоже массив векторов только на основание данных в этом массиве и значений на выходе нейронной сети будет происходить обучение модели и изменение весов синопсисов для уменьшения ошибки.

Переменная epochs принимает количество эпох. Количество эпох отвечает за то сколько раз будет проходить обучения нейросети

С помощью функции predict можно в уже обученную нейронную сеть подать значения на вход и узнать какой результат выдаст нейросеть. Весь программный код Приложение А.

## 3.2 Создание модели нейронной сети в Matlab Simulink

Такие характеристики как: значения подающиеся на входных слой, веса синопсисов, значения на выходном слое, можно представить в виде матриц. Поэтому работа всей сети сводится к умножению матрицы входа на матрицу весов синопсиса. Полученную матрицу необходимо провести через функцию активации.

где,

 – Матрица весов синопсисов;

 - Матрица входных сигналов;

 – результирующая матрица.

На рисунке ниже представлена нейросеть собранная с помощью библиотечных элементов в matlab Simulink.

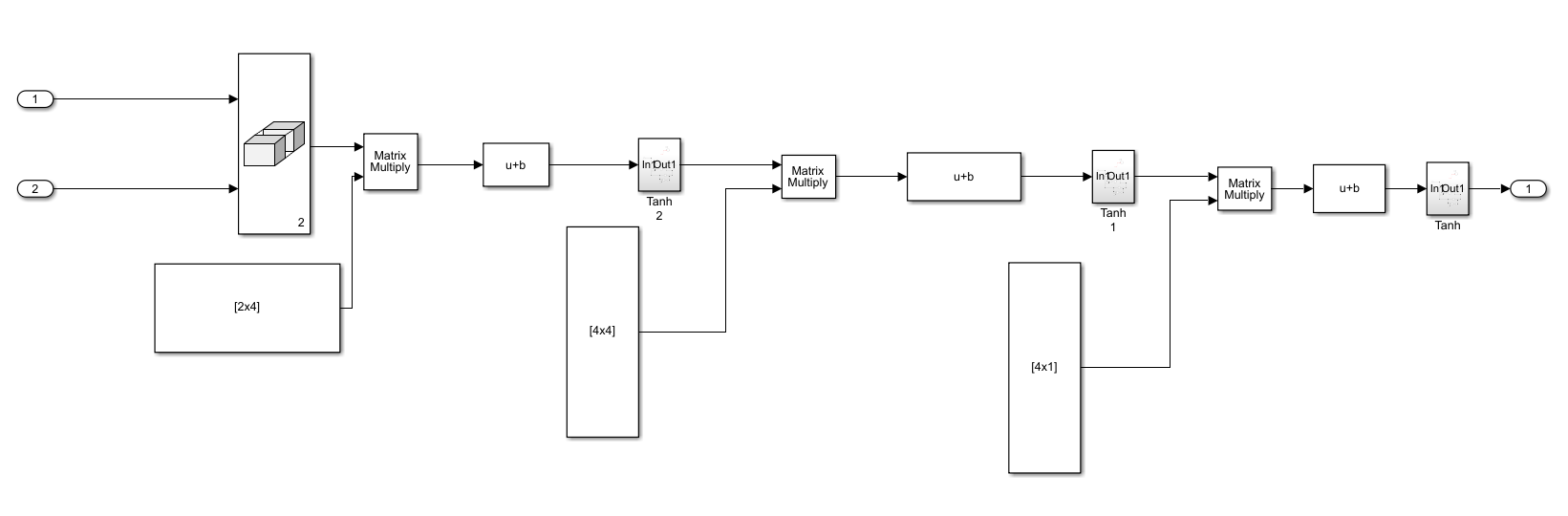


Рисунок 7 – Нейронная сеть, созданная в Matlab Simulink

В модель подаются значения. Далее они преобразуются в матрицу размером 1х2 и умножаются на весовые коэффициенты синопсисов. Полученная матрица пропускается через функцию активации и умножается на веса синопсисов следующего слоя. И так продолжается до тех пор, пока не получится результат на выходе.

Для того что бы проверить правильность собранной модели в Python имеющий два входа, два скрытых слоя по четыре нейрона в каждом и одним выходом, была обучена на логическую операцию «исключающее или» представленную на рисунке ниже.

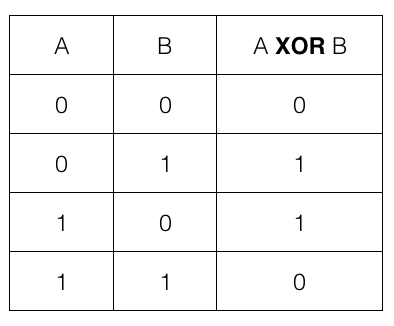


Рисунок 8 – Исключающее или

На графике ниже можно увидеть как ошибка между выходом нейронной сети и обучающим значением менялась с количеством пройденных эпох.

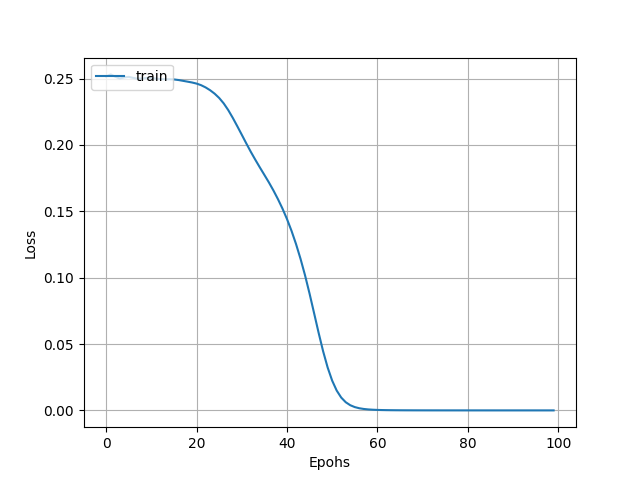


Рисунок 9 – График изменения ошибки при обучении

Значение ошибки на сотой эпохе составляет 0.000005. По графику видно, что модель обучилась к 60 эпохе, а дальше уже не обучалась. Если модель учить слишком много эпох, то возможен момент переобучения, когда ошибка начнет беспричинно расти.

Воспользовавшись функцией Predict и подав на вход нейронной сети значения 0 и 1 функция, показала значение 0.9959903, что по рисунку 8 очень близко к верному значению.

Теперь необходимо выгрузить значения весов и смещений из Python и занести в модель matlab. На рисунке ниже представлен результат работы нейросети из матлаб.

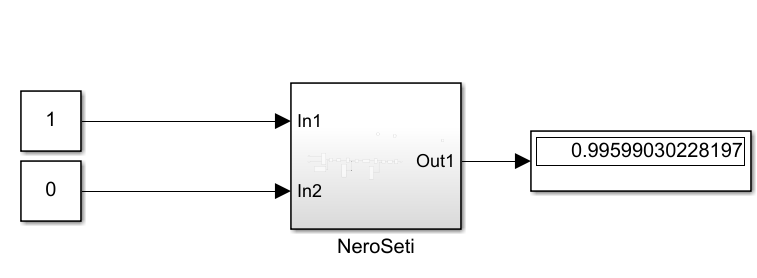


Рисунок 10 – Пример работы обученной сети в Matlab

Результат получился такой же как и при использование функции predict в Python, а это значит, что модель созданная в matlab Simulink является правильной и отвечает всем параметрам.

4. Обучение нейронной сети на однофазной модели

## 4.1 Построение модели в Matlab

Обучение нейронной сети должно происходить по схеме показанной ниже.

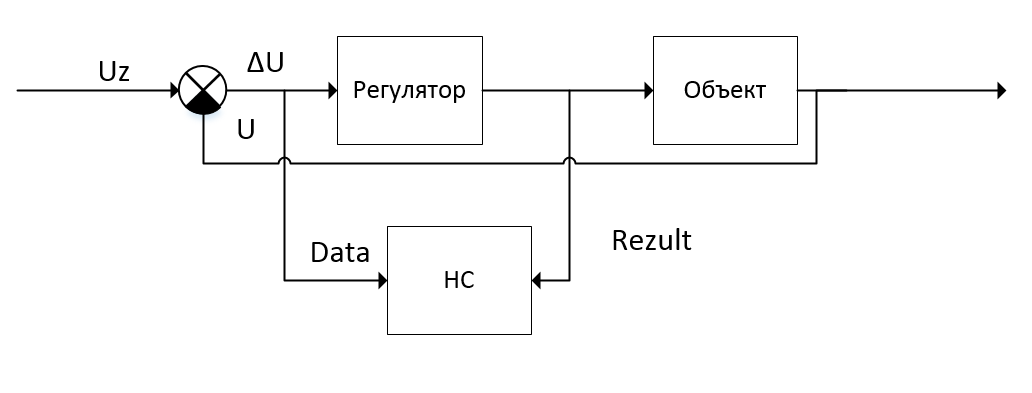


Рисунок 11 – Схема обучения нейронной сети на реальном объекте

Data – это параметр в котором в нейросеть будут передаваться данные на основе которых она будет обучаться. Этот параметр очень важен, потому что от входных данных зависит обучение и работа способность всей сети.

Rezult – результат работы исходного регулятора, который подается на выход нейросети, где она подсчитывает ошибку и меняет свои весовые коэффициенты для уменьшения ошибки.

Прежде чем обучать нейронную сеть на трехфазную систему. Необходимо обучить нейросеть на однофазную систему, что бы выбрать следующие параметры: архитектуру сети, какие данные для обучения нужно использовать. На рисунке ниже представлена однофазная система построенная в Matlab.

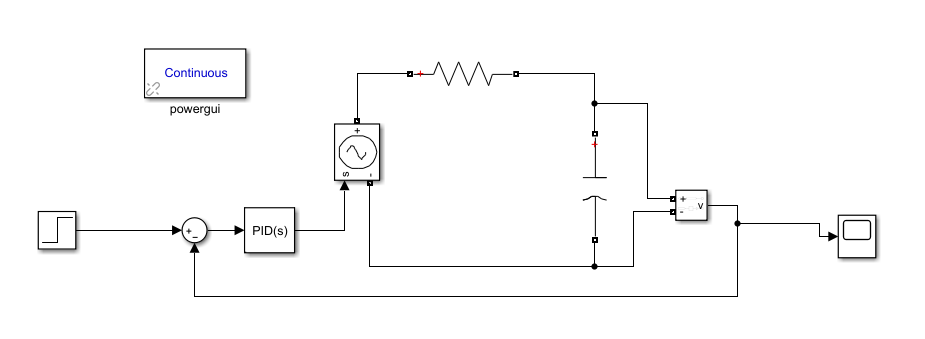


Рисунок 12 – Однофазная модель в Matlab

На вход ПИД регулятора подается ступенькой единица. Настройки регулятора представлены ниже. Регулятор подает сигнал управления на источник. Резистор имеет сопротивление 0.1 Ом.

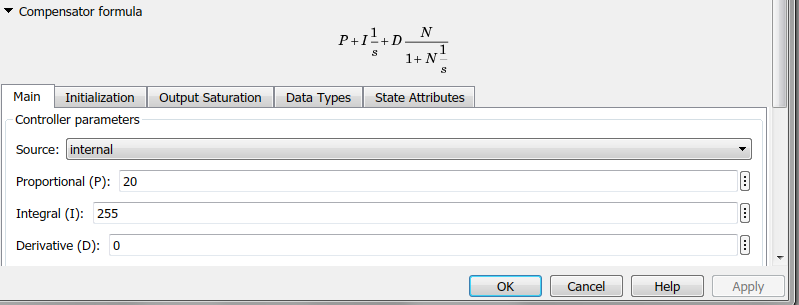


Рисунок 13 – Настройка ПИД регулятора

На вход регулятора ступенькой подается значение от нуля до единицы. Скачок происходит в момент времени равном 0.1 секунде.

Выход регулятора представлен на графике ниже. По графику можно увидеть корректную отработку регулятора

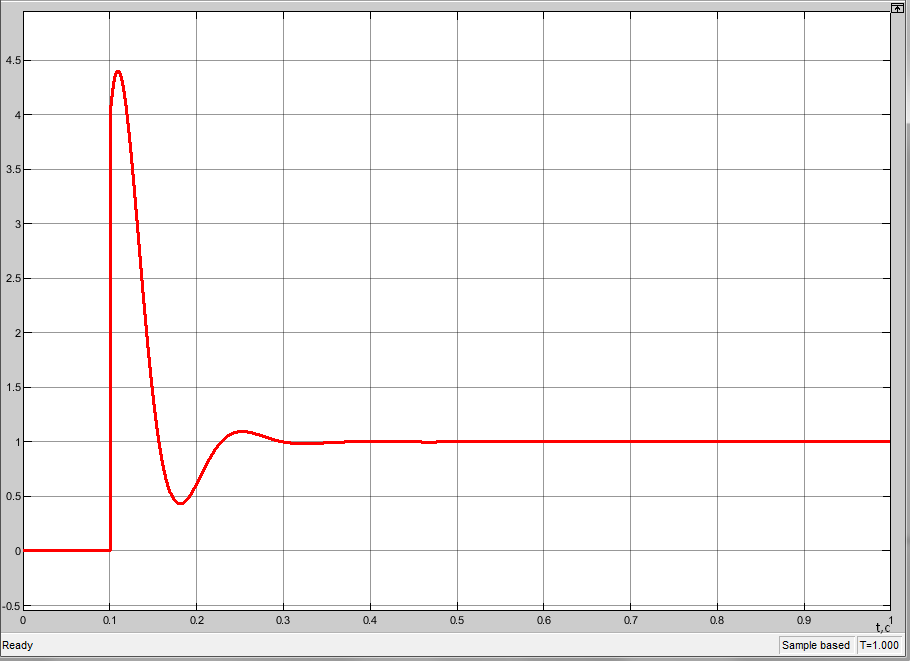


Рисунок 14 – График ПИД регулятора

По графику можно увидеть корректную отработку регулятора. На рисунке ниже представлен график напряжения на конденсаторе.

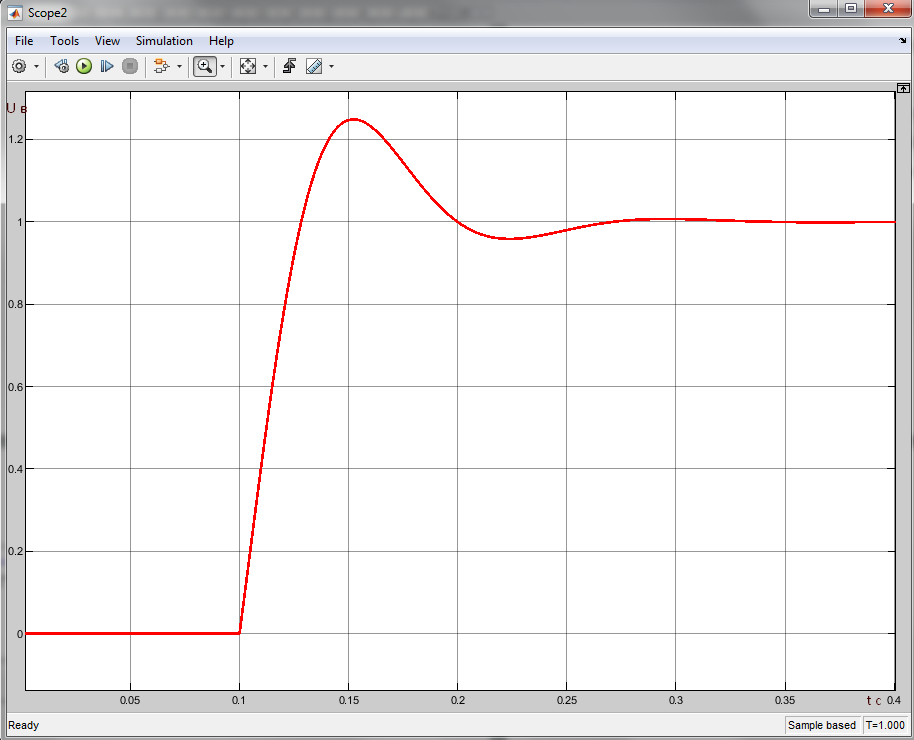


Рисунок 15 – Напряжение на конденсаторе

Эти данные будут приняты за эталон. Нейронная сеть должна будет обучится повторять эти значени, что значит вести себя как ПИД регулятор.

## 4.2 Обучение нейронной сети

Необходимо выбрать данные, которыми будет обучаться нейросеть. В начале архитектура сети буде один нейрон на входном слое, два скрытых слоя по два нейрона и один нейрон на выходном слое. Функциями активации между слоями будет функция линейного выпрямителя. Для выходного слоя гиперболический тангенс. Обучаться нейронная сеть будет только на Uz. График ошибки и график напряжения показаны ниже.

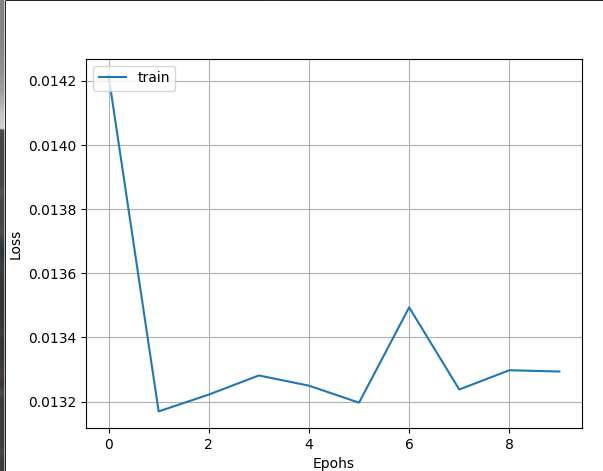


Рисунок 16 – График ошибки при обучение на UZ

Хоть ошибка и близка к нулю, но для НС это слишком большой показатель, что бы уменьшить ошибку необходимо, либо изменить входные данные, либо изменить архитектуру сети.

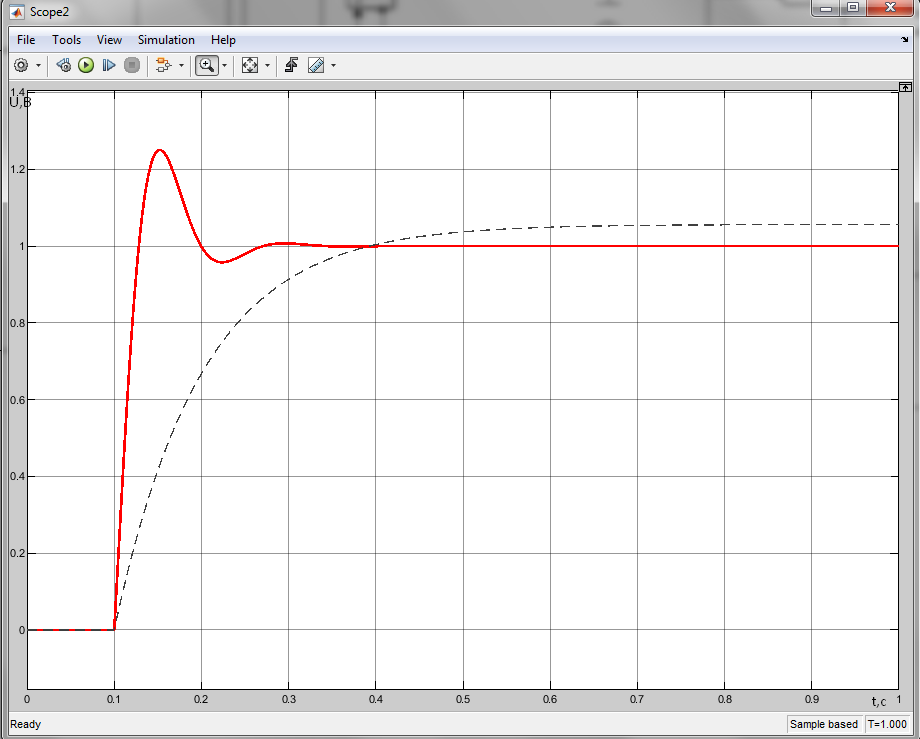


Рисунок 17 – График напряжения при обучение Uz

Сплошной линей показано напряжение на объекте с идеальным регулятором, пунктирной на объекте управляемым нейронной сетью. По графику можно увидеть, что напряжение от НС далеко от идеального, потому что не повторяет режим пере модуляции и не выходит на установившееся значение.

Для того чтобы улучшить показатели модели в качестве обучающих данных на модель будут подаваться Uz и U.Графики после обучения нейронной сети на Uz и U представлены ниже.

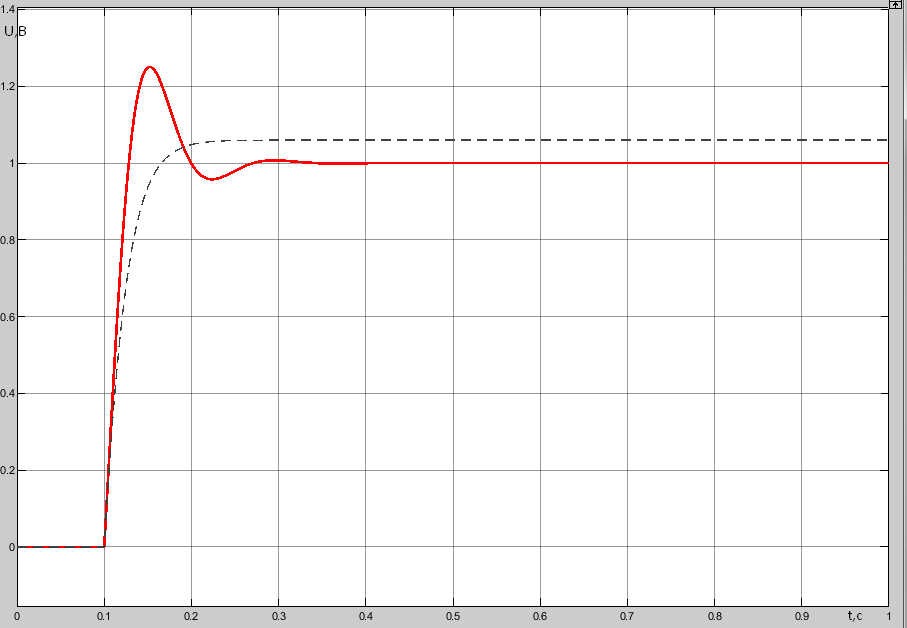
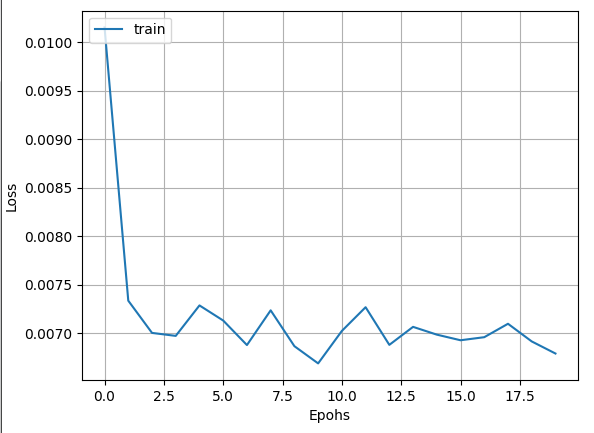


Рисунок 18 Функция ошибки и напряжения при обучение Uz и U

Ошибка обучения не может прийти к какому-то единому минимуму она постоянно меняется, и напряжение не повторяет эталонное, но уже лучше, чем в предыдущем опыте. Для улучшения показателей нейронной сети можно изменить ее архитектуру, увеличив количество нейроном в скрытых слоях на 10 и 5. График ошибки обучения представлен ниже.

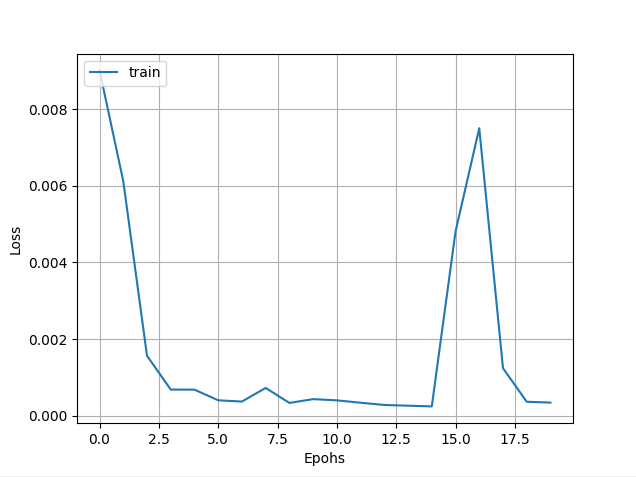


Рисунок 19 – График ошибки при изменение архитектуры сети

Хоть в моменте обучения и был аномальный всплеск ошибки, но за три эпохи ошибка снова опустилась в минимум и стала меньше чем при топологи сети, где только по два нейрона были в скрытых слоях.Из полученного графика можно сделать вывод, что изменение топологии сети привело к лучшей обучаемости сети а значит и повлияло на работоспособность сети.Ниже представлен график напряжения после изменения архитектуры сети.

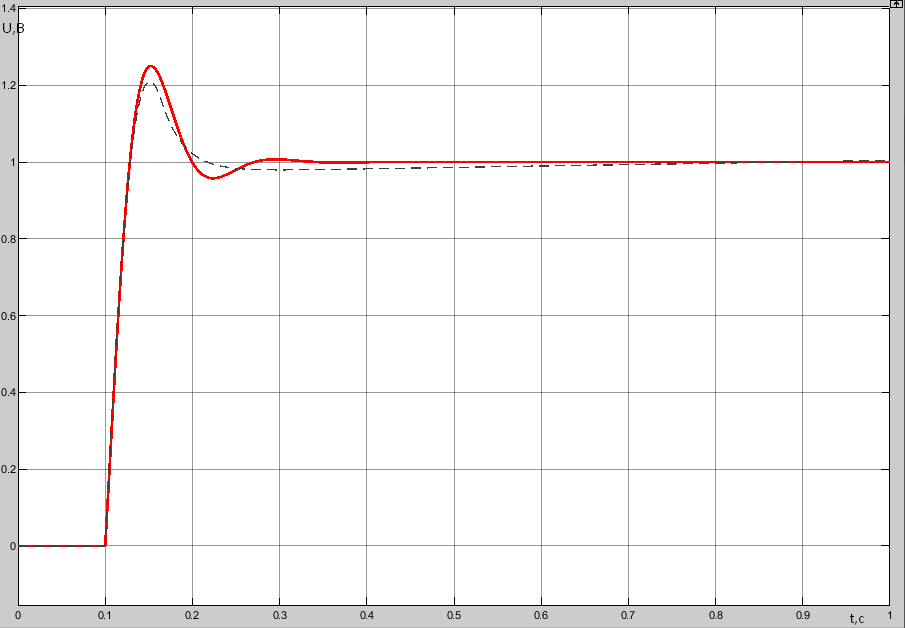


Рисунок 20 – График напряжения

По графику видно, что изменение топологии сети помогло нейронное сети воспроизвести момент перерегулирования напряжения, что тоже является хорошим показателем. Теперь нейронную сеть необходимо обучить на Uz и ΔU, что бы выбрать при каких параметрах сеть обучается лучше. Ниже представлен график ошибки при заданных параметрах.

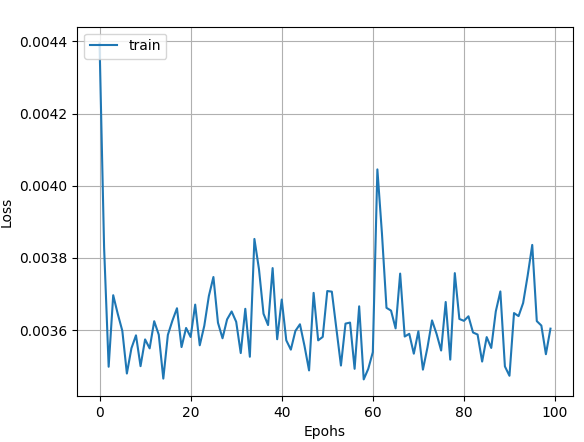


Рисунок 21 – График ошибки при обучение Uz, ΔU

Даже при увеличение количества эпох ошибка обучения не вышла на свой минимум, ее показание постоянно меняется, что является показателем того, что топология нейронной сети или входные параметры выбраны не удачно. Такие параметры обучения нейронной сети не являются сложными поэтому в первую очередь необходимо проверить ошибку в топологии нейронной сети. Известно, что при данном количестве скрытых слоев и нейронов в них нейронная сеть способна давать удовлетворяющий результат. Необходимо подумать о изменение функций активации в скрытых слоях. На данном этапе функциями активации является линейный выпрямитель. Его необходимо заменить на более конечную функцию например сигмоиду. График ошибки и напряжений после изменения функции активаций на сигмоиду представлен ниже.

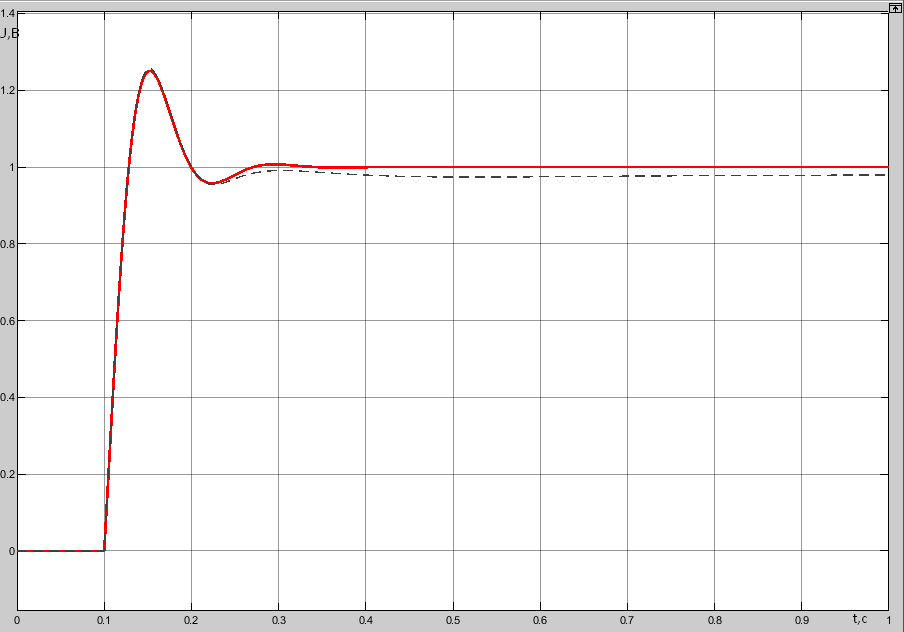
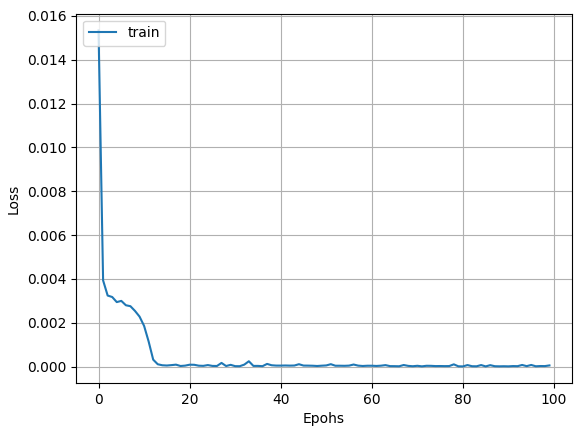


Рисунок 22 – Графики ошибки и напряжения после изменения функции активации

Данная топология сети и выходные параметры для обучения Uz и ΔU, лучше повторяют пререгулирование, но хуже удерживают систему при выходе на заданное значение. Последним эксперементом для однофазной сети, будет обучение сети на все три параметра: U, Uz, ΔU. Для того, что бы можно сделать вывод при каких параметрах обучения нейросеть лучше себя показывает.Ниже представлены графики ошибки и напряжений при обучение на U, Uz, ΔU.

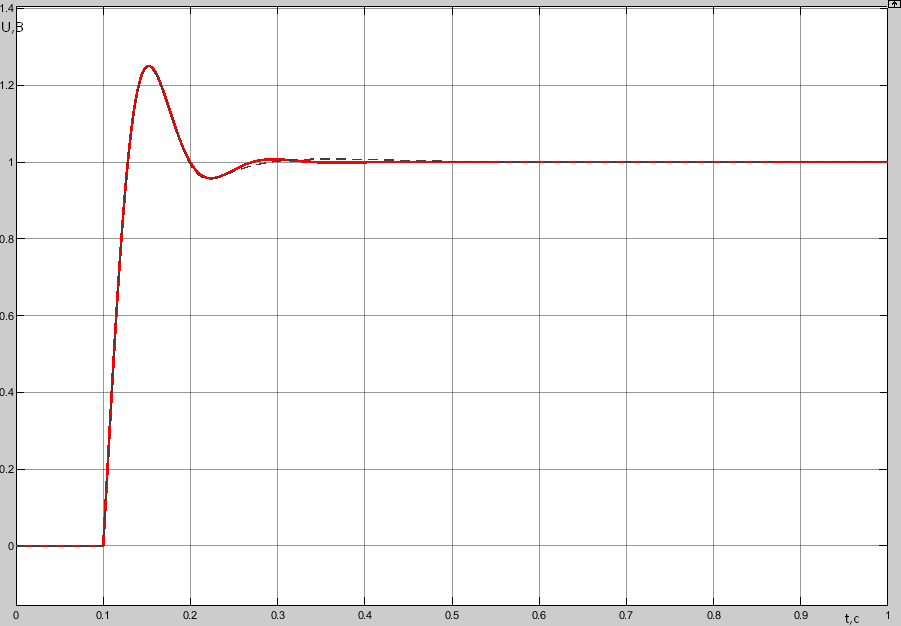
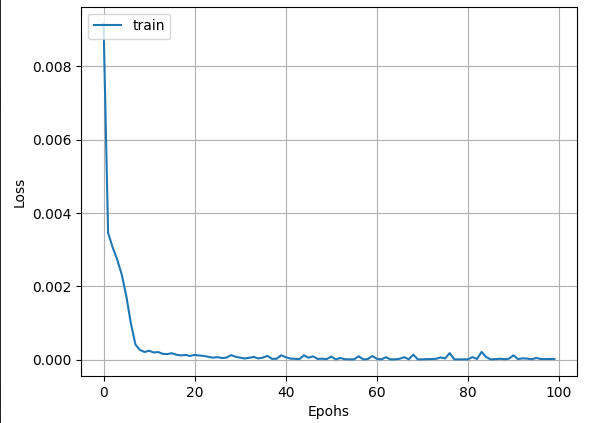


Рисунок 23 – Графики ошибки и напряжения при обучение нейронной сети на U, Uz, ΔU

На данных графиках видно, что добавление третего параметра в обучающую выборку на функции ошибки никак не отразилось, но сильно повлияло на работу нейронной сети с объектом. Теперь нейронная сеть лучше повторяет момент перемодуляции и лучше удерживает систему в заданном значени.

После проведенных эксперементов на однофазной системы, можно сделать выводы, что для обучения трехфазной системы в обучающих данных необходимо использовать заданное значение, значение на объекте и их разницу. Для того, что бы нейросеть правильно отрабытывала она должна содержать как минимум два скрытых слоя и в каждом должно быть как минимум пять нейронов. Во время обучения и работы лучше всего себя показало совместное использование функций активации сигмоида и гиперболический тангенс.