МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**Национальный исследовательский ядерный университет «МИФИ»**

# 

**Факультет кибернетики и информационной безопасности**

**КАФЕДРА КИБЕРНЕТИКИ**

### БДЗ

**по курсу "Теория нейронных сетей"**

**студента группы Б15-506**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Богатыревой А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

#### Вариант № \_\_7\_\_\_

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

##### Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2018 г.

ОТЧЕТ № 1

по теме «Многослойные нейронные сети»

Вариант №\_\_\_7\_\_\_\_

ФИО студента \_\_\_Богатырева А.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Группа \_\_Б15-506\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Результаты обучения многослойной нейронной сети:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод обучения | Параметры метода обучения | Число эпох обучения | Ошибка на обучающей выборке, *E*обуч | Ошибка на тестовой выборке, *Eтест* |
| GD | lr=0.001 | 6300 | 0.0094 | 0.009 |
| GDM | lr=0.001  momentum = 0.9 | 50 | 0.4422 | 0.4155 |
| NAG | lr=0.001  momentum = 0.9 | 50 | 0.4277 | 0.4017 |
| SGD (Neupy) |  | 663 | 0.00356 | 0.00603 |
| Fletcher-Reeeves (Neupy) |  | 100 | 0.003 | 0.0062 |
| Polak-Ribiere (Neupy) |  | 100 | 0.003 | 0.0062 |
| AdaGrad | lr0=0.001 | 100 | 0.4866 | 0.4573 |
| RMSProp | lr0 =0.001; ρ = 0.99 | 100 | 0.1361 | 0.1237 |
| AdaDelta | lr0 =0.001; ρ = 0.99 | 100 | 0.1443 | 0.1469 |
| RProp | η- = 1.2; η+ = 0.5 | 150 | 0.0031 | 0.0054 |
| LM (Neupy) |  | 20 | 0.0048 | 0.0093 |
| BFGS (Neupy) |  | 20 | 0.0048 | 0.0078 |
| Stochastic GD | lr = 0.001 | 50 | 0.5075 | 0.4777 |

Выводы:

|  |
| --- |
| Реализация на PyTorch: наиболее качественно прошло обучение при использовании метода RProp с параметрами: η- = 1.2; η+ = 0.5.  Реализация на Neupy: наилучшие результаты показали методы FR и PR.  Однако в силу реализации среди этих методов наиболее точные предсказания выдавала сеть, обученная с методом RProp. |

ОТЧЕТ № 2

по теме «Решение прикладных задач обработки данных на нейронных сетях»

Вариант №\_\_\_7\_\_\_\_

ФИО студента \_\_\_\_Богатырева А.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Группа \_\_Б15-506\_\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Показатели качества обученной нейросетевой модели:

**Для задач регрессии**: привести диаграммы рассеяния в пространстве «выход модели – желаемый выход» для обучающей и тестовой выборок, изобразить линейные регрессии выхода модели на желаемый выход, указать коэффициенты детерминации линейных регрессионных моделей для обучающей и тестовой выборок.

**Для задач классификации**: привести матрицы ошибок (confusion matrix) нейросетевого классификатора и ошибки классификации на обучающей и тестовой выборках.

|  |
| --- |
| В данной работе решала задача регрессии.  Диаграммы рассеяния в пространстве «выход модели – желаемый выход» для обучающей и тестовой выборок:    Линейные регрессии выхода модели на желаемый выход и коэффициенты детерминации линейных регрессионных моделей для обучающей и тестовой выборок:    Гистограммы распределения ошибок сети на примерах обучающей и тестовой выборок: |

Выводы:

|  |
| --- |
| Построенная нейросетевая модель предсказывает значения выходов с минимальной ошибкой, как это показано на диаграммах. Добавление шума во входные данные не влияет на результаты предсказания.  Задача регрессии решается качественно. |

I. Исходные данные

*1.1.Описание исходных данных*

* Набор данных содержит 9568 точек, представляющих собой данные о среднечасовых переменных окружающего воздуха: температуре (T), атмосферного давления (AP), относительной влажности (RH), вытяжном вакууме (V) и почасовой выработки электроэнергии (EP) станции - собранные за 6 лет с электростанции комбинированного цикла. Признаками являются: T, AP, RH, V. Сети необходимо предсказать почасовой выход энергии станции (EP).
* Ссылка на источник: <https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant>
* Описание данных:

Число признаков: 4

Типы признаков: вещественные

Объём выборки: 9568

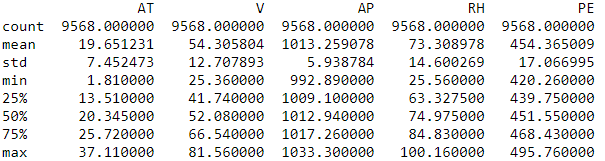
Особенности данных: предварительной информации о наличии дубликатов и пропусков не имеется. Были обнаружены противоречия в данных о RH: максимальное значение равно 100.16, однако больше 100% быть не может

Решаемая задача: на основании данных признаков определить значение выходной часовой энергии.

Тип задачи: регрессия.

Входные переменные: T, AP, RH, V

Выходные переменные: EP



*1.2.Визуальный анализ исходных данных*

*а) Гистограммы распределения и диаграммы Box-and-Whisker*

Результаты представлены на рис.1-2 ниже.

Выводы:

(о характере распределений признаков, наличии выбросов и т.п.)

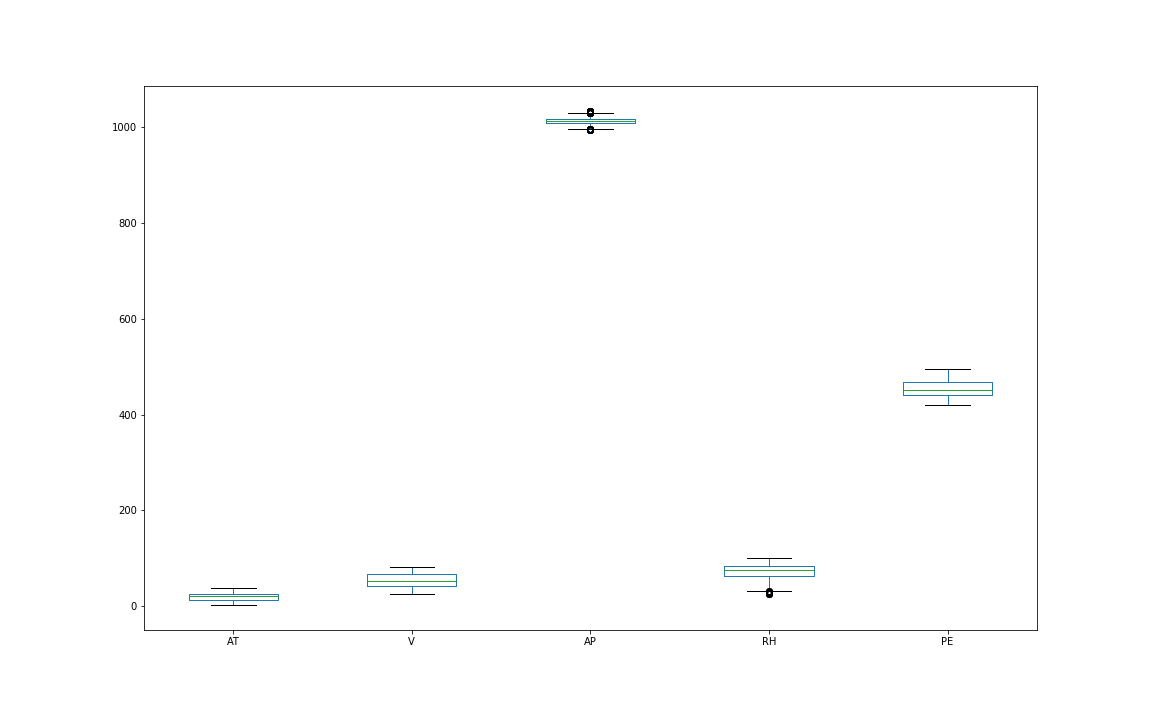


Рис. 1. Box-and-Whisker для всех признаков и выхода

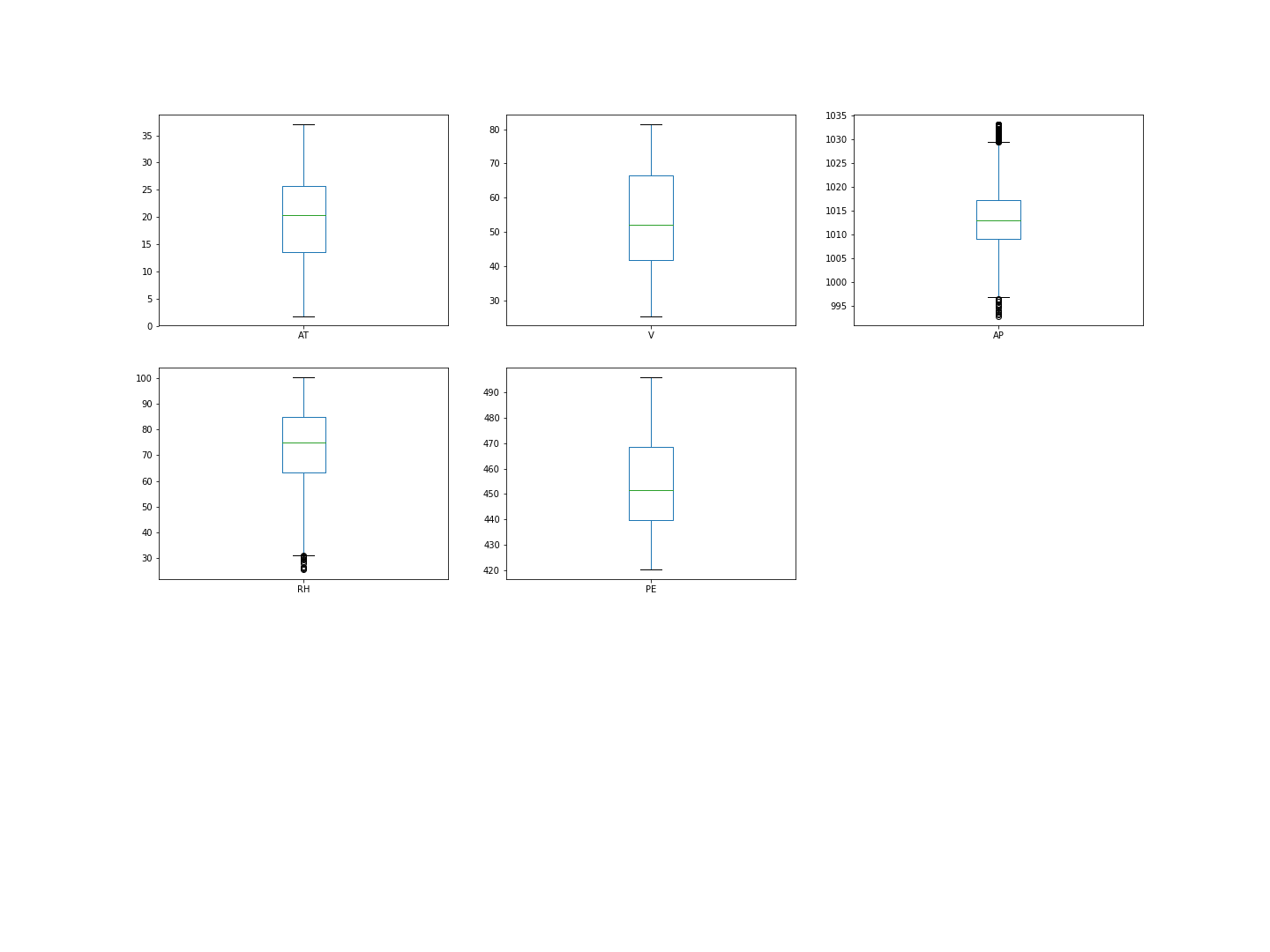
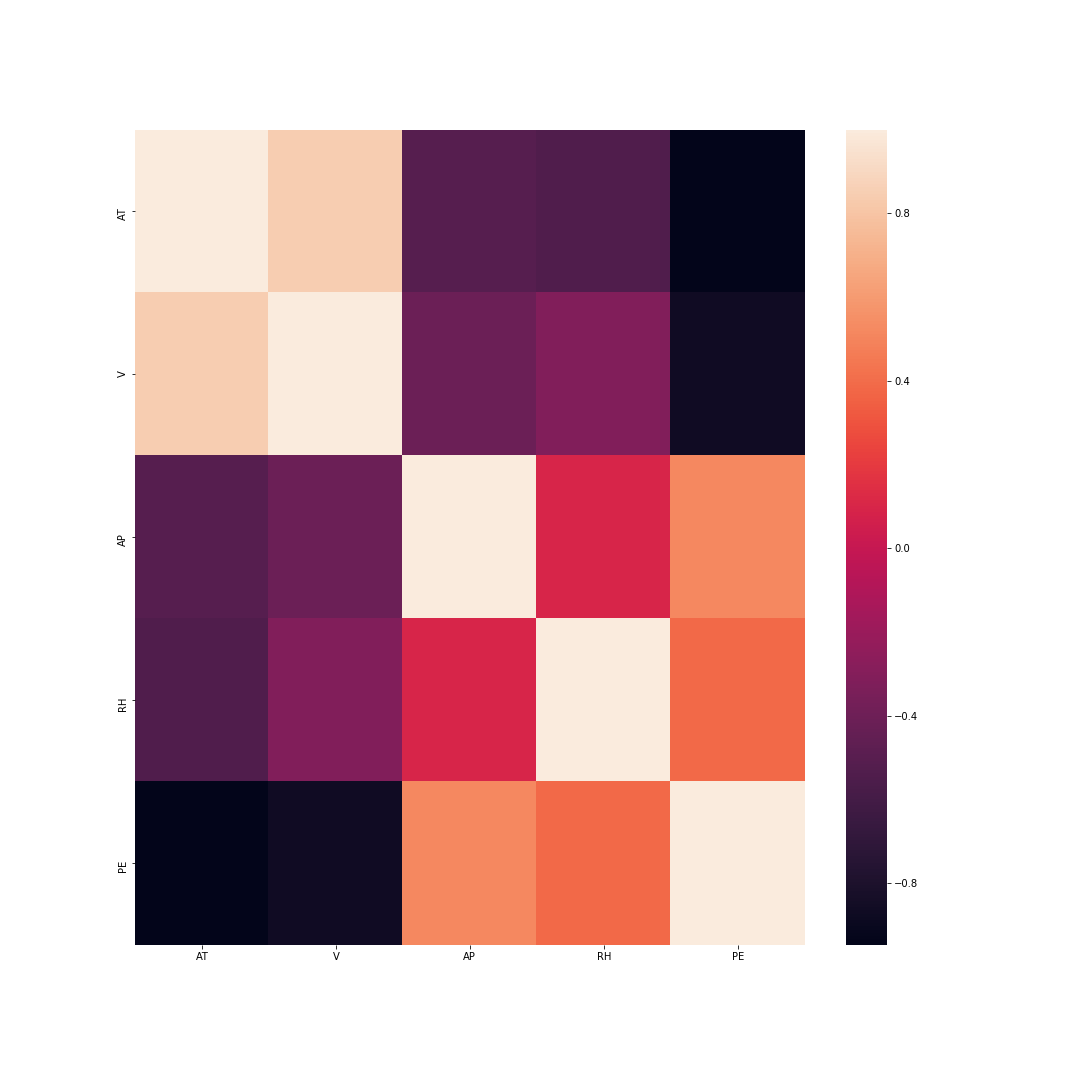
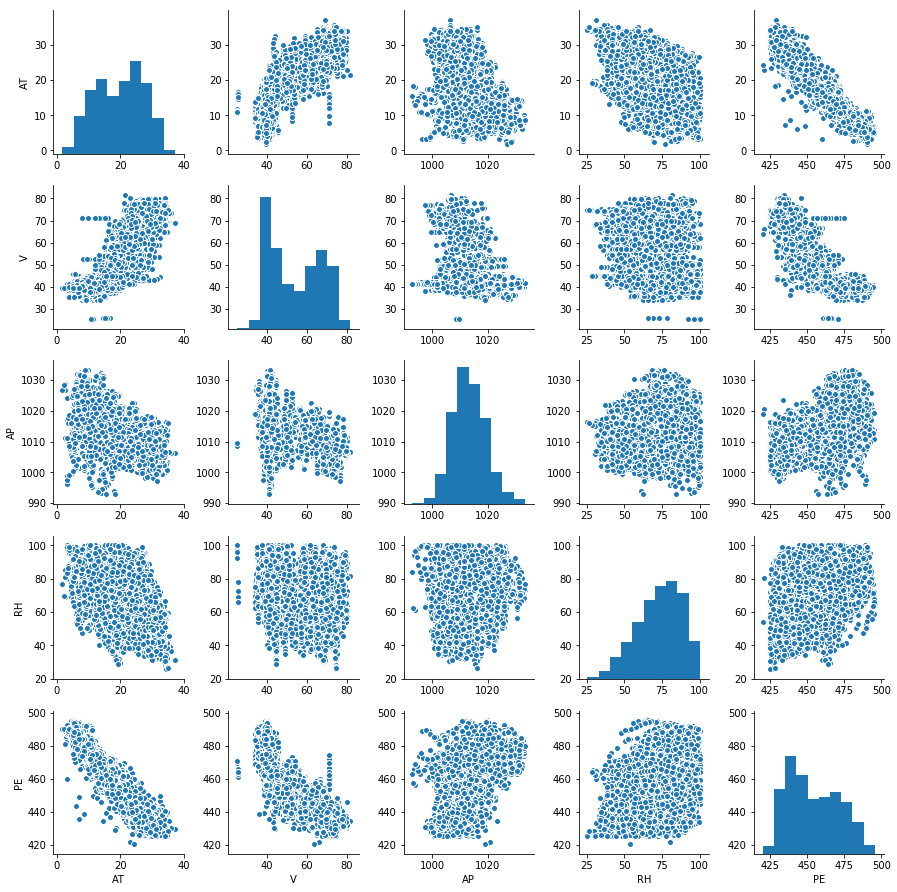


Рис. 2.Box-and-Whisker для каждого по отдельности

*б) Корреляционная матрица признаков*



*в) Диаграммы рассеяния*



Выводы: сильно скоррелированных данных нет. Распределения признаков AT, V и выхода PE совсем далеки от нормального.

*1.3.Выводы*

Сделать выводы по результатам визуального анализа исходных данных.

II. Предобработка данных

*2.1. Очистка данных*

*а) Обнаружение и устранение дубликатов*

Был использован метод из библиотеки Pandas: X\_data.drop\_duplicates(keep='first'). Были удалены все дубликаты за исключение первого вхождения. Результат: была удалена 41 запись. Осталось 9527 элементов.

*б) Обнаружение и устранение выбросов*

1)были устранены все недостоверные значения для столбца: были записи, где RH принимало значения выше 100%, чего в реальности быть не может.

2) для устранения выбросов по каждому признаку и выходу была использована след.эвристика:

Пусть q1 – квантиль уровня 0.25, q3 – квантиль уровня 0.75, iqr – интерквантильный размах. Тогда выбросы находятся в областях: x > q3+1.5\*iqr and x < q1-1.5\*iqr.

Для поиска квантилей были применены методы из Pandas.

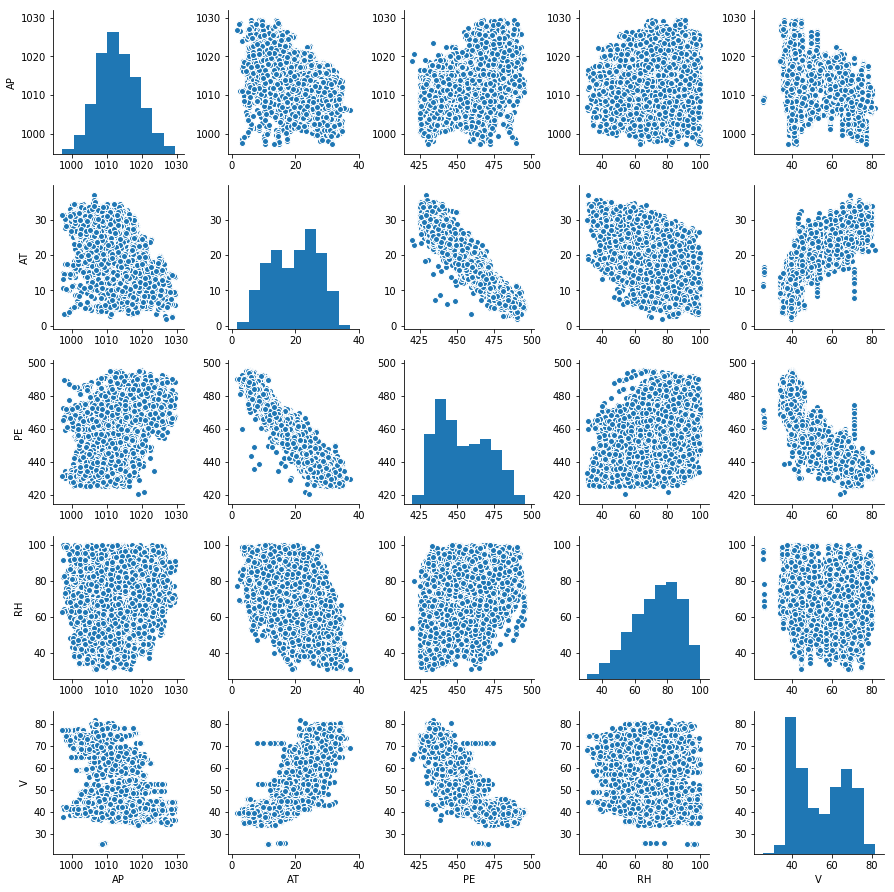
Результат: осталось 9372 элемента.

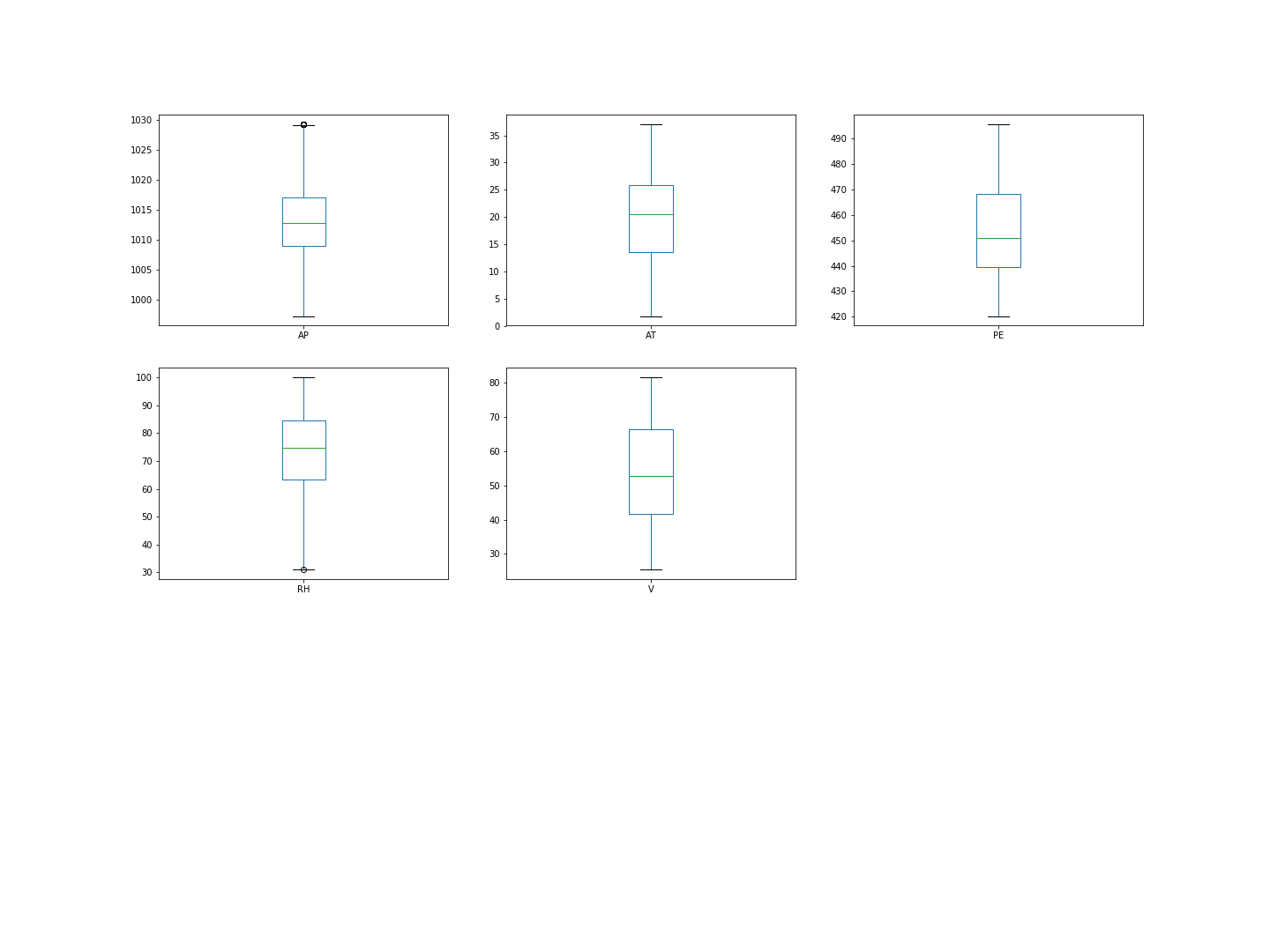
*в) Пропущенные значения*

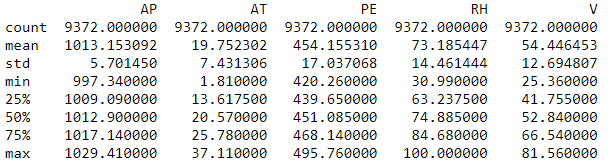
Был использован класс Imputer из sklearn.preprocessing для того, чтобы заменить все значения NaN в каждом признаке и выходе на медиану по признаку/выходу. Была выбрана медиана, т.к. она более устойчива к выбросам по сравнению с другими методами.

Результат: число элементов не изменилось: 9372.

*г) Визуальный анализ очищенных данных*







Сравнение с гистограммами из п.1.2: диаграммы распределения и рассеяния практически не изменились (замечание: в этом пункте случайно изменен порядок представления признаков). На диаграммах Box-and-Whisker и в таблице видно, что статистические параметры признаков также изменились незначительно.

*д) Выводы*

Данные были очищены от нежелательных элементов:

* Были убраны дубликаты некоторых элементов
* Были устранены выбросы, но не полностью: оставшиеся элементы исключать не стоит, чтобы в данные не были слишком идеальны
* Все пропущенные значения были заполнены значениями медиан. Таким образом, число элементов в выборке не уменьшилось и новых «шумов» не появилось.

*2.2. Преобразование данных*

Перед преобразованиями данные были разбиты на обучающую и тестовую выборки. Далее они обрабатывались отдельно.

*а) Преобразование входов*

Был использован класс StandardScaler из sklearn.preprocessing для стандартизации с параметрами (0, 1) и масштабирования каждого из признаков в отдельности.

Стандартизация – это общее требование для многих оценок машинного обучения: они могут вести себя плохо, если отдельные функции не выглядят более или менее как стандартные нормально распределенные данные. Также некоторые алгоритмы предполагают, что все объекты центрированы вокруг 0 и имеют дисперсию в том же порядке. Если объект имеет дисперсию, которая на порядок больше, чем другие, он может доминировать над целевой функцией и сделать оценщика неспособным учиться у других объектов правильно.

Масштабирование необходимо для того, чтобы все данные имели значения из более-менее одинакового промежутка. Это помогает избежать ситуации, когда, например, один признак принимает значения из [-3, 2], а другой – из [-100, 100].

Стоит заметить, что StandartScaler не делает распределение нормальным в строгом смысле слова, но в какой-то мере защищает от выбросов.

*б) Преобразование выходов*

Был использован класс MinMaxScaler из sklearn.preprocessing для масштабирования выходов на отрезок [0, 1].

*в) Визуальный анализ преобразованных данных*

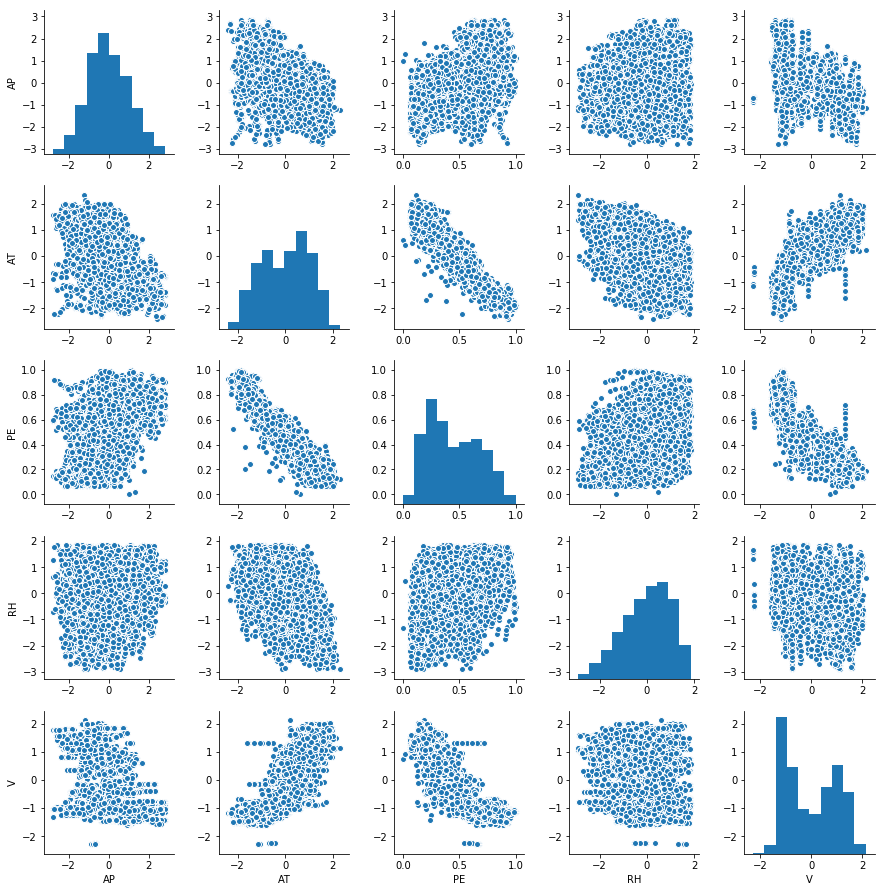


Рис. 3. Гистограммы распределения и рассеяния для обучающей выборки

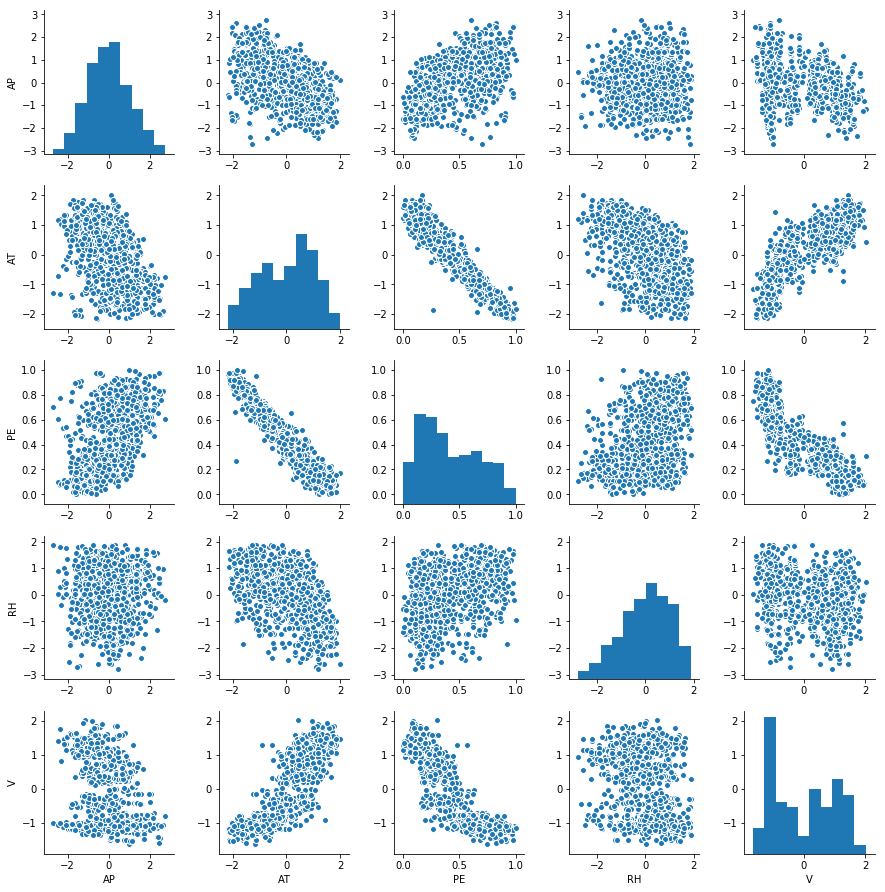


Рис. 4. Гистограммы распределения и рассеяния для тестовой выборки

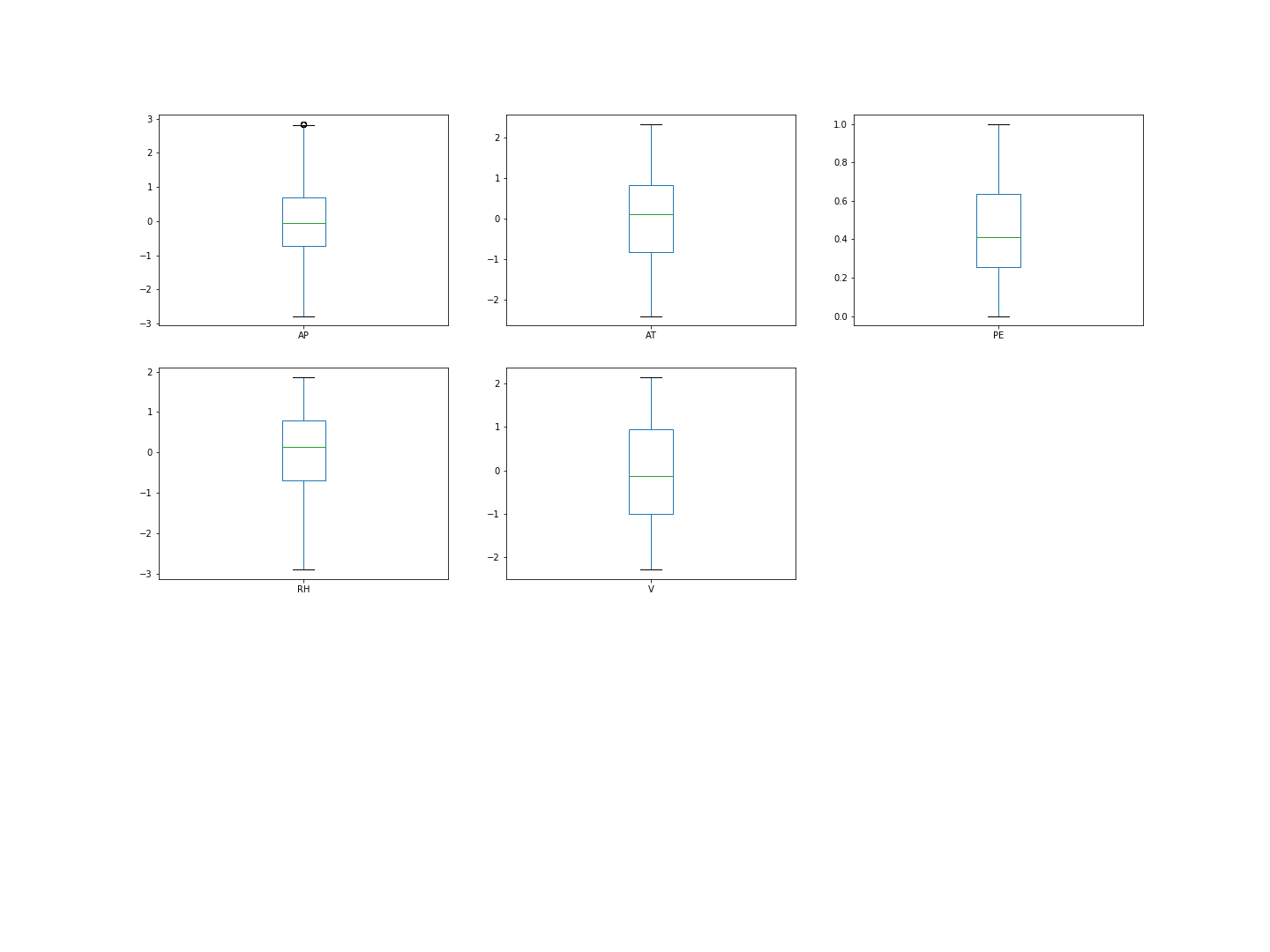


Рис. 5. Box-and-Whisker для обучающей выборки

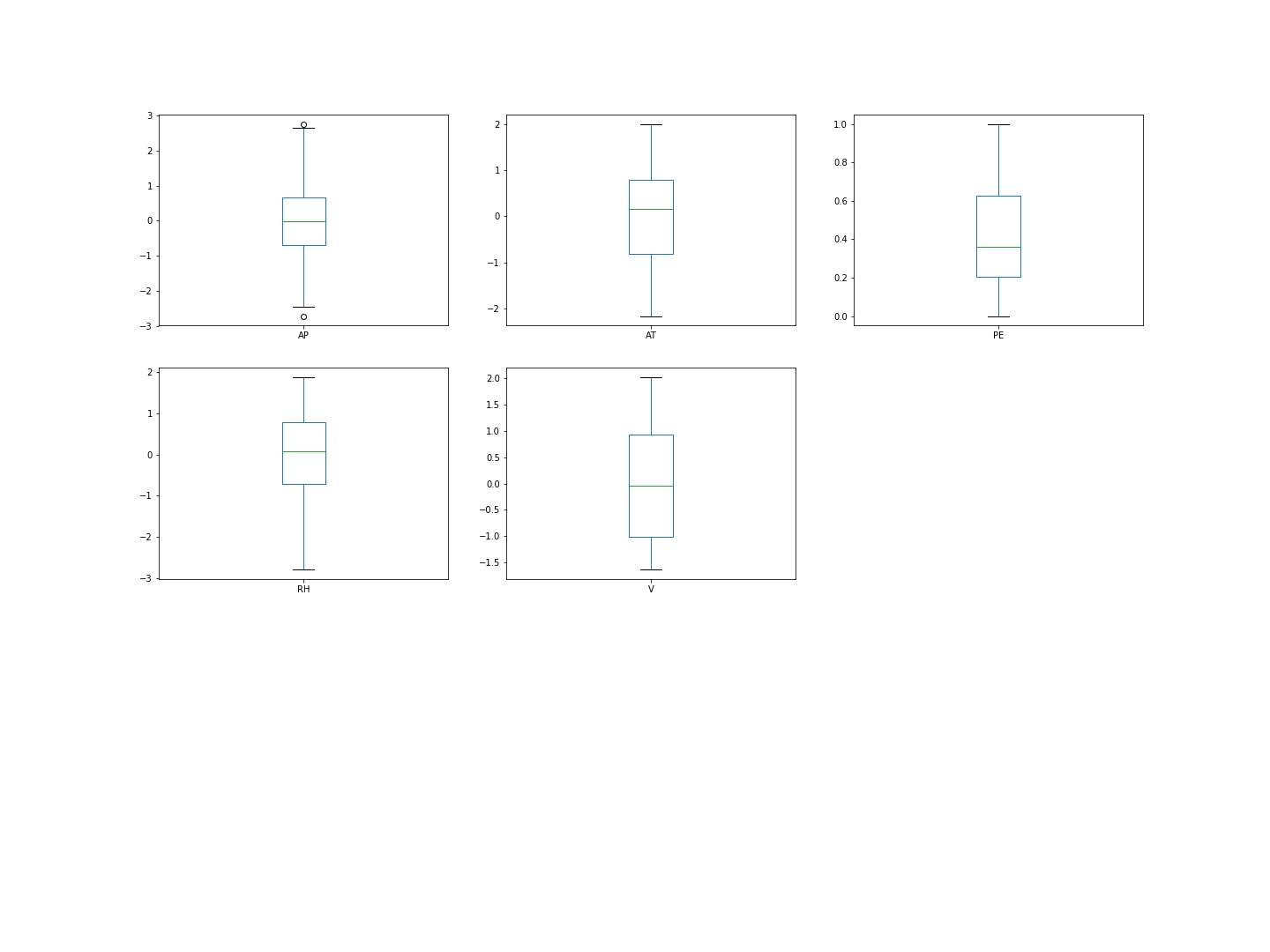


Рис. 6. Box-and-Whisker для тестовой выборки

Сравнение с диаграммами из п. 2.1 г): связи между признаками не изменились (судя по диаграммам рассеяния), распределения – тоже. Данные приведены к небольшим значениям.

*2.3.Выводы*

Была проведена z-score нормализация, в результате чего признаки приведены к необходимому для обучения виду. Т.к. значения выходов много больше 0 и была выбрана функция активации, принимающая значения на [-1, 1] выходы были отмасштабированы.

III. Формирование признаков

*3.1. Сокращение числа признаков*

В текущем наборе данных всего 4 признака и каждый из них несет в себе какую-то информацию. Поэтому сокращать их число нет необходимости.

*3.2. Конструирование новых признаков*

В данной работе нет необходимости включать новые признаки.

*3.3. Выводы*

Выборка не изменилась.

IV. Построение и исследование нейросетевых моделей

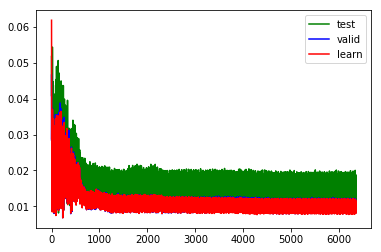
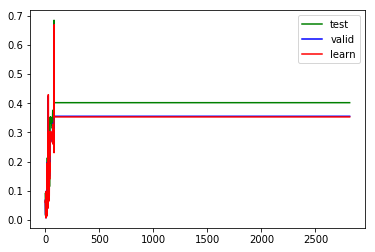
*4.1. Параметры архитектуры и обучения многослойной нейронной сети*

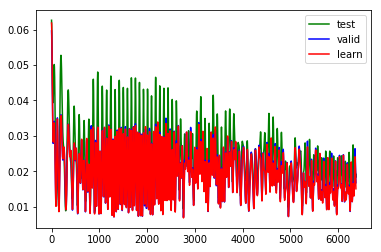
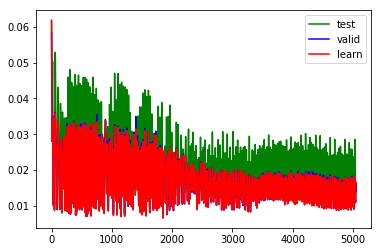
|  |  |
| --- | --- |
| Параметр | Значение |
| Функция потерь | Quadratic |
| Число входов сети | 4 |
| Число выходов сети | 1 |
| Число скрытых слоев сети\* | 3 |
| Число и АХ нейронов 1-го скрытого слоя\* | 10; tanh |
| Число и АХ нейронов 2-го скрытого слоя\* | 10; tanh |
| Число и АХ нейронов 3-го скрытого слоя\* | 5; tanh |
| АХ нейронов выходного слоя | tanh |
| Кросс-валидация | Holdout (60/30/10) |
| Объёмы обучающей / валидационной / тестовой выборок | 5622 / 2812 / 938 |
| Режим обучения\* | batch |
| Метод инициализации весов | метод Хавьера |
| Критерий останова | Число эпох |
| Ранний останов | да |

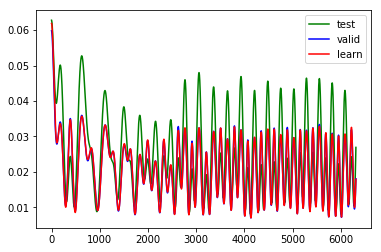
\* Определяется вариантом задания.

*4.2. Исследование простого градиентного метода обучения*

*а) Исследование влияния параметра скорости обучения на качество обучения*







*б) Заполнить таблицу по результатам обучения*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Скорость обучения, α | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | 1.0 | 2800 | 0.3533 | 0.402 |
| 2 | 0.1 | 6300 | 0.0086 | 0.0184 |
| 3 | 0.01 | 5000 | 0.0159 | 0.0137 |
| 4 | 0.001 | 6300 | 0.0094 | 0.009 |
| 5 | 0.0001 | 6300 | 0.0179 | 0.0269 |

**Указание**: все ошибки указываются для обученной сети.

*в) Выводы*

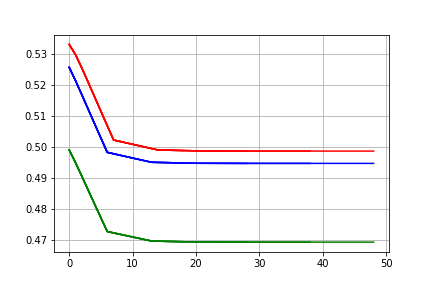
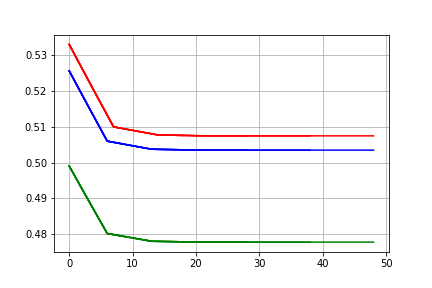
При обучении ошибка все время колебалась в некоторые пределах, но при lr < 1, в общем-то, уменьшалась. До значения 0.001: чем меньше скорость обучения, тем процесс обучения протекает лучше, а после 0.001: при дальнейшем уменьшении процесс обучения ухудшается.

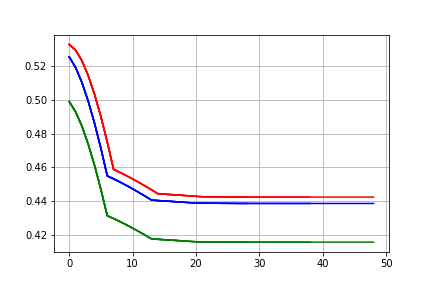
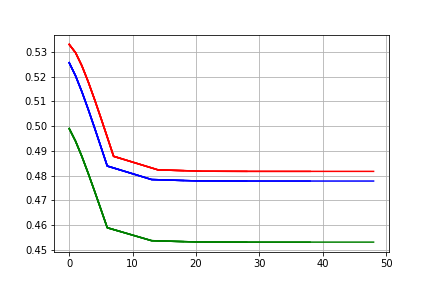
Лучше всего модель обучалась при lr=0.001.

*4.3. Исследование методов GDM и NAG*

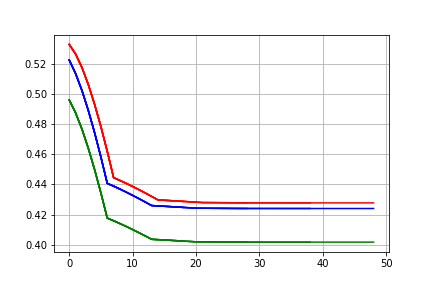
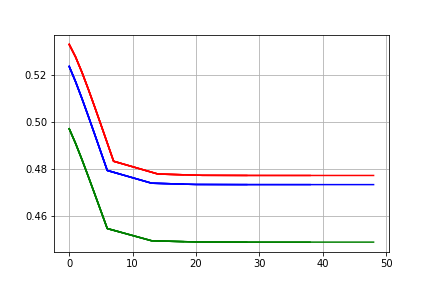
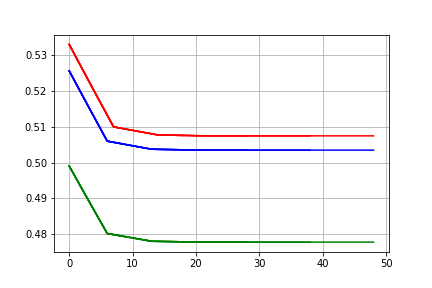
*а) Исследование влияния параметра момента на качество обучения*

Для GDM:





Для NAG:



*б) Заполнить таблицу по результатам обучения*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Метод | Момент, μ | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | GDM | 0 | 50 | 0.5075 | 0.4778 |
| 2 | NAG | 50 | 0.5075 | 0.4778 |
| 3 | GDM | 0.3 | 50 | 0.4987 | 0.4693 |
| 4 | NAG | 50 | 0.4973 | 0.4681 |
| 5 | GDM | 0.6 | 50 | 0.4818 | 0.4532 |
| 6 | NAG | 50 | 0.4772 | 0.4489 |
| 7 | GDM | 0.9 | 50 | 0.4422 | 0.4155 |
| 8 | NAG | 50 | 0.4277 | 0.4017 |

*в) Выводы*

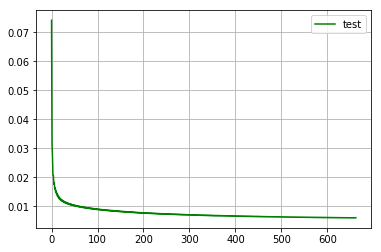
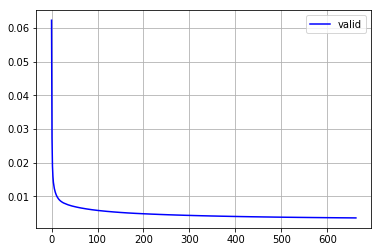
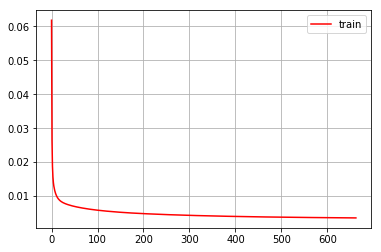
В обоих методах при стремлении параметра к 1 качество обучения улучшалось.

Используя метод NAG, обучение проходило лучше, чем при GDM.

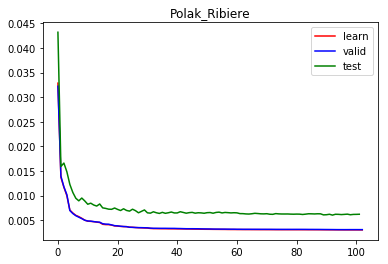
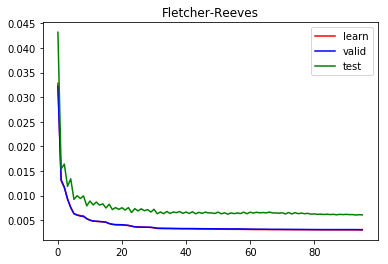
*4.4. Исследование методов наискорейшего спуска и сопряжённых градиентов*

*а) Сравнение кривых обучения*

Графики ошибок для Steepest GD:



Графики ошибок для Fletcher-Reeves и Polak-Ribiere:



*б) Заполнить таблицу по результатам обучения*

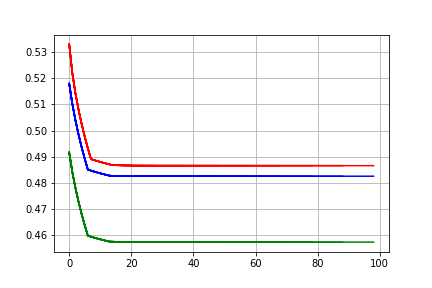
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Метод | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | SGD | 663 | 0.00356 | 0.00603 |
| 2 | Fletcher-Reeeves | 100 | 0.003 | 0.0062 |
| 3 | Polak-Ribiere | 100 | 0.003 | 0.0062 |

*в) Выводы*

Методы Fletcher-Reeves (FR) или Polak-Ribiere (PR) дают одинаковые результаты обучения. Таких же результатов можно добиться и с методом Steepest GD, но с бо́льшими временными затратами.

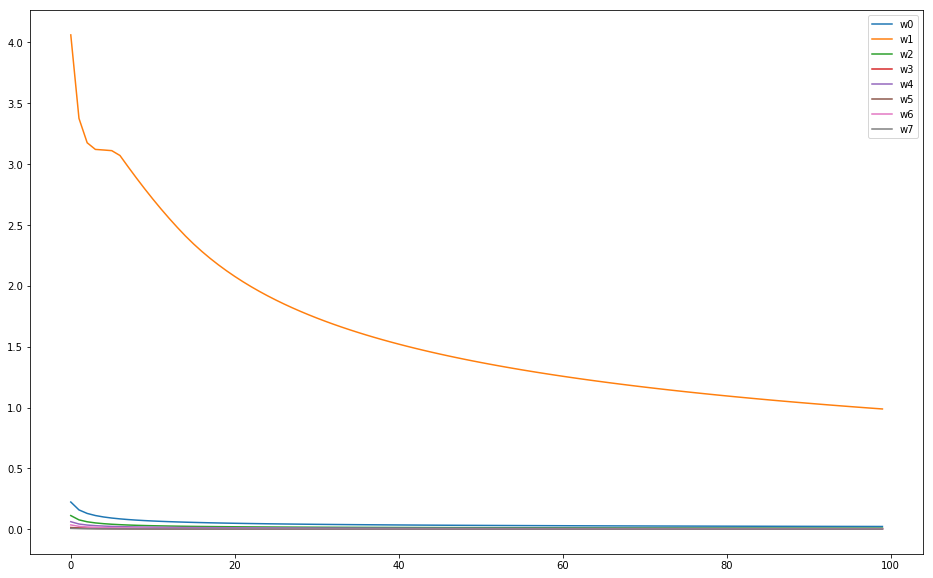
*4.5. Исследование метода AdaGrad*

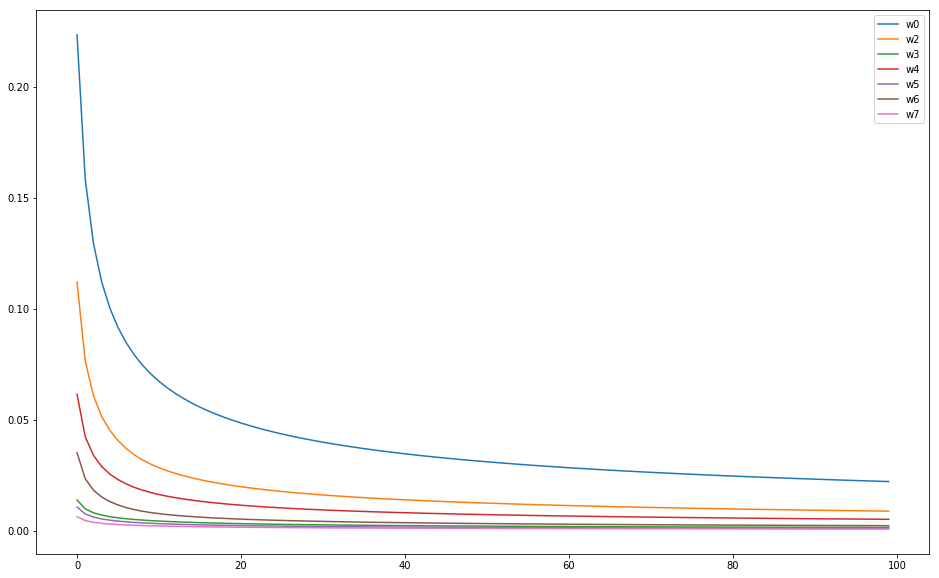
*а) Кривые обучения*



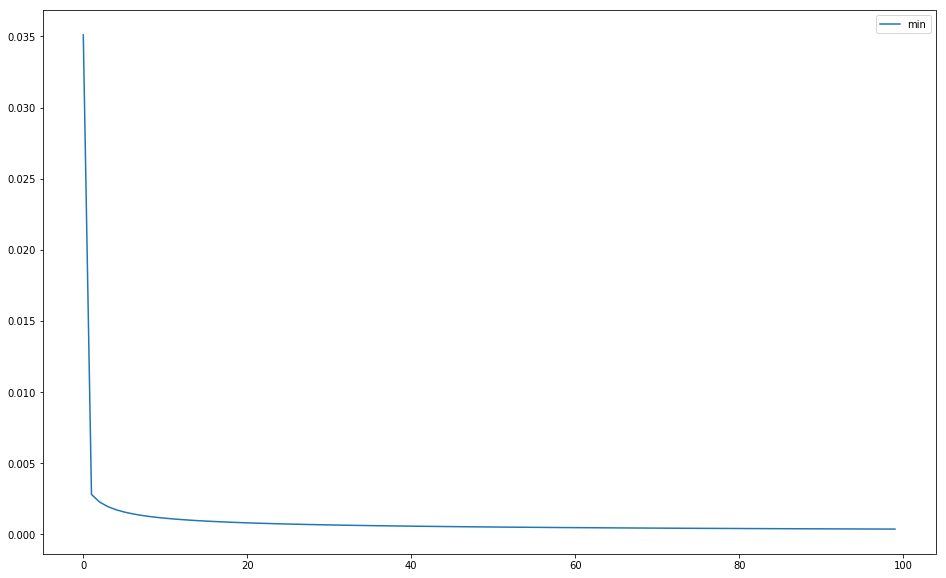
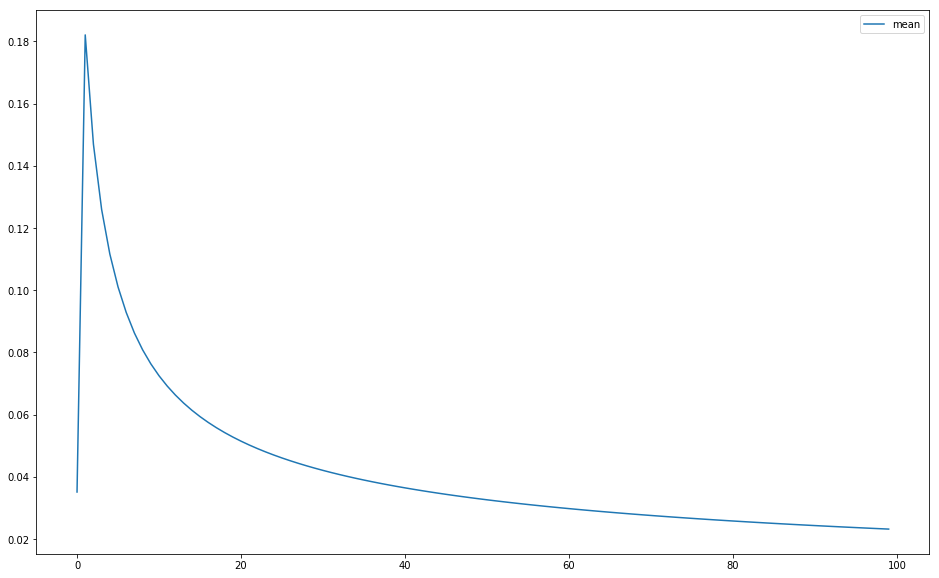
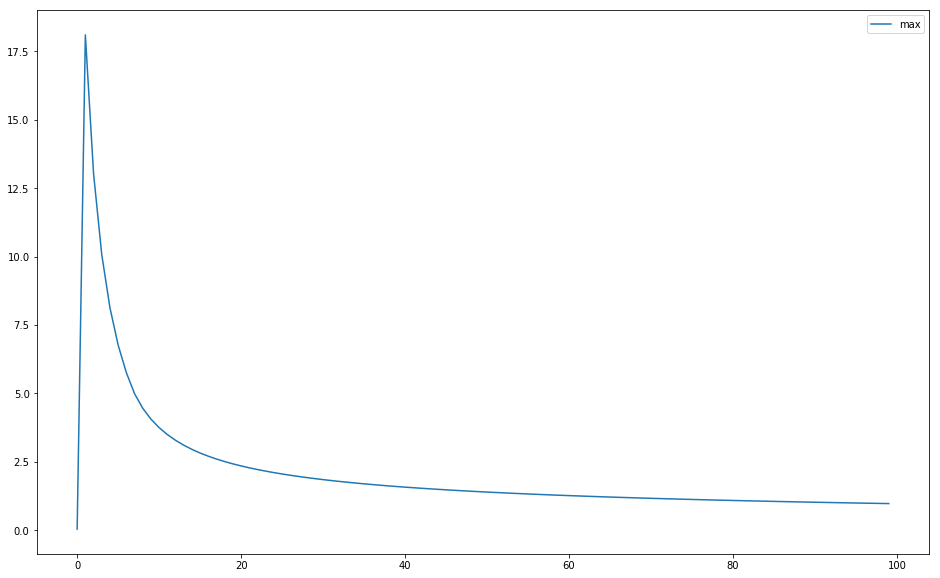
*б) Исследование динамики скорости обучения*

Графики зависимости скорости обучения отдельных синаптических коэффициентов сети:





Графики зависимости скоростей обучения:



*в) Заполнить таблицу по результатам обучения*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Метод | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | AdaGrad | 100 | 0.4866 | 0.4573 |

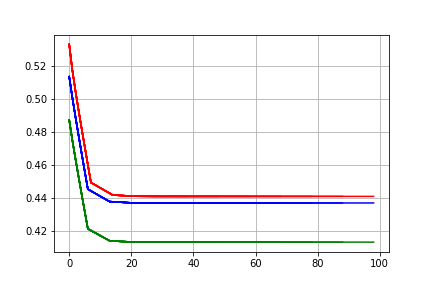
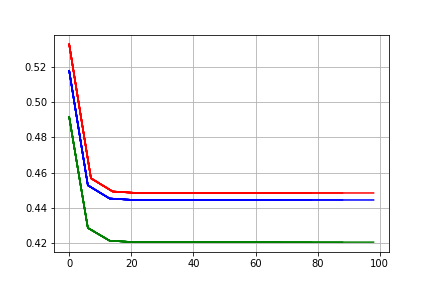
*г) Выводы*

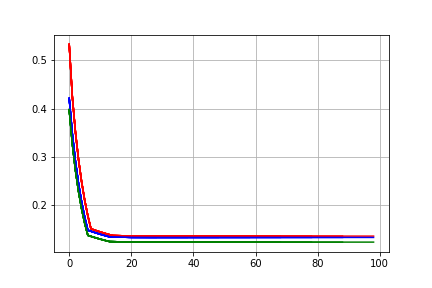
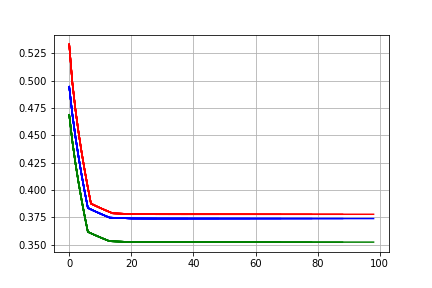
AdaGrad дает приемлемый, но менее точный результат (например, в сравнении с методами Fletcher-Reeves или Polak-Ribiere). Однако продолжать обучение не имело смысла, т.к. из графиков видно, что ошибки перестали меняться, а скорости обучения стали настолько малы, что не влияют на процесс обучения.

*4.6. Исследование методов RMSProp и AdaDelta*

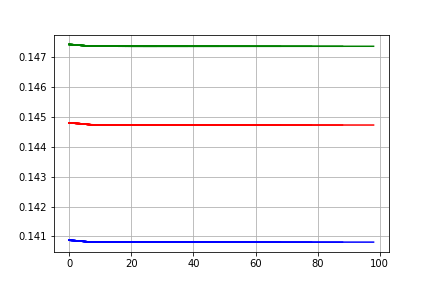
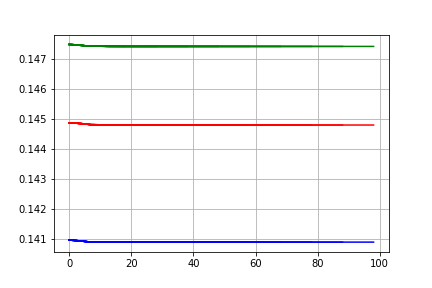
*а) Сравнение кривых обучения*

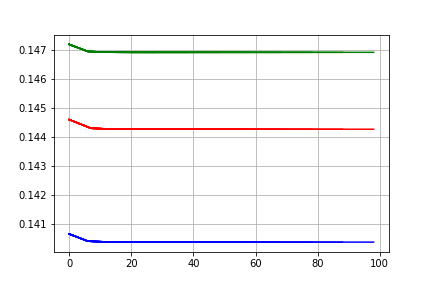
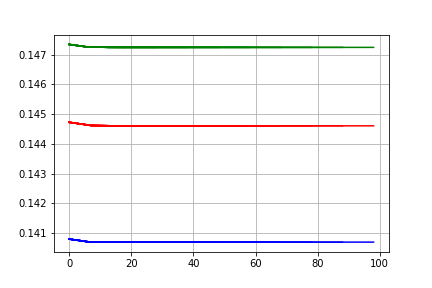
RMSProp:





AdaDelta:

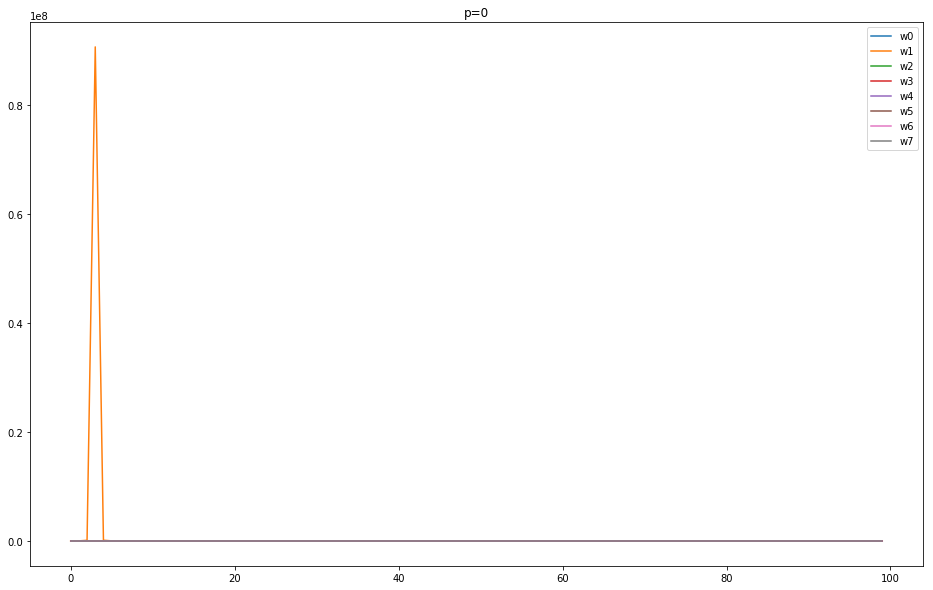


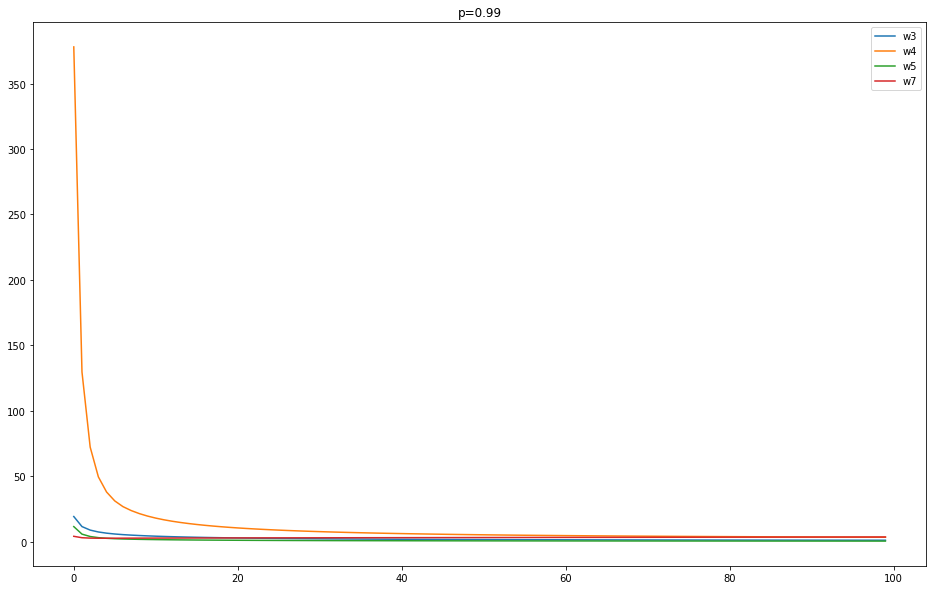
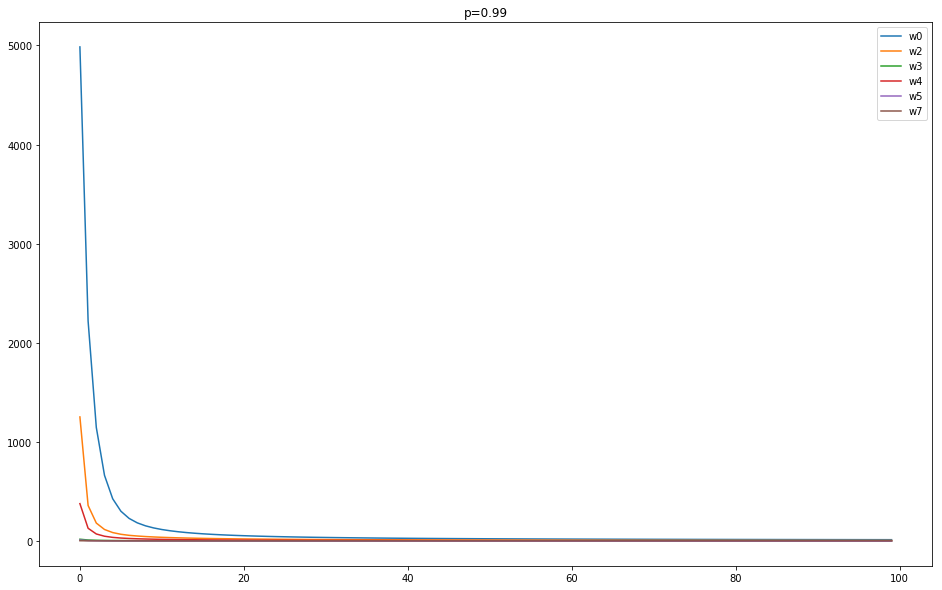
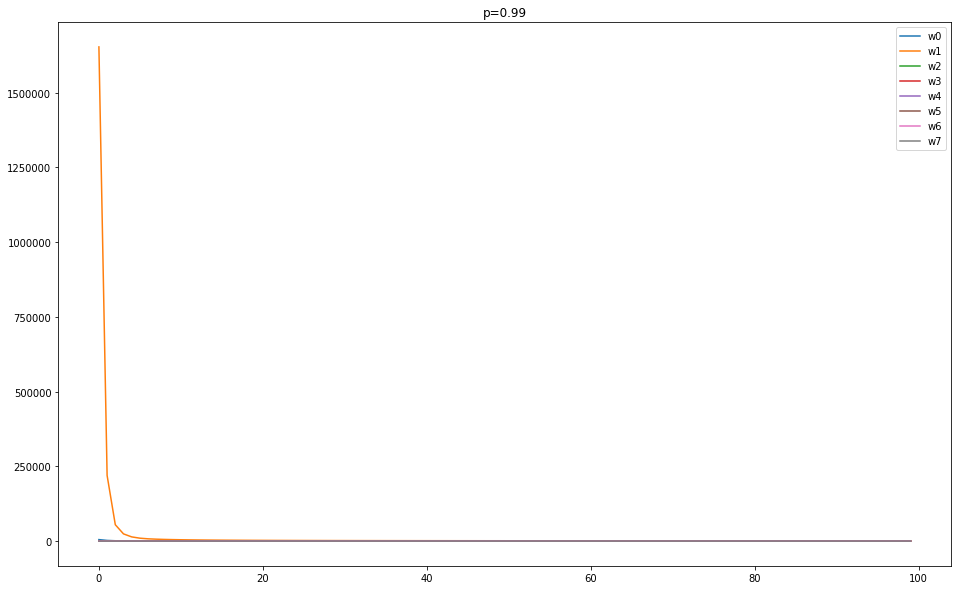
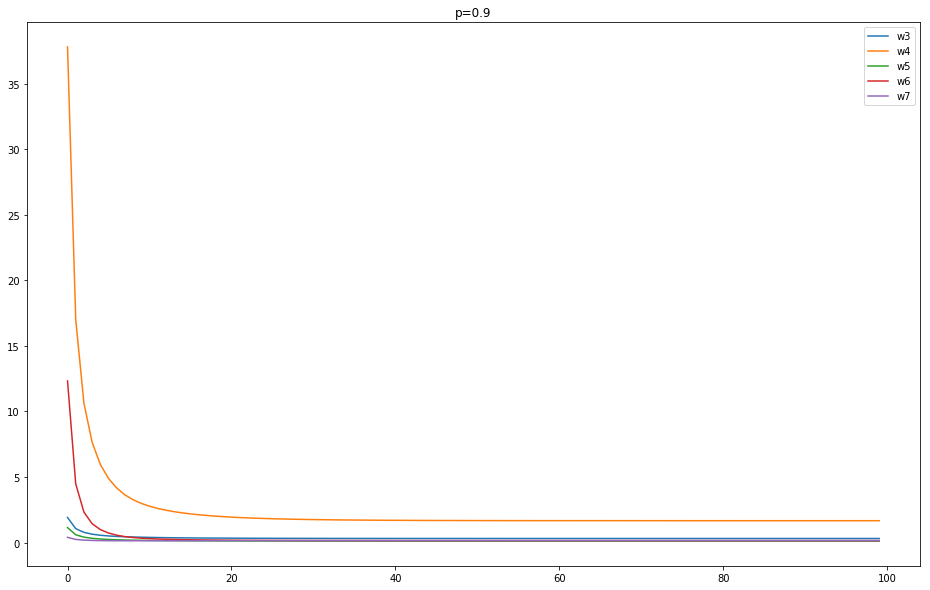
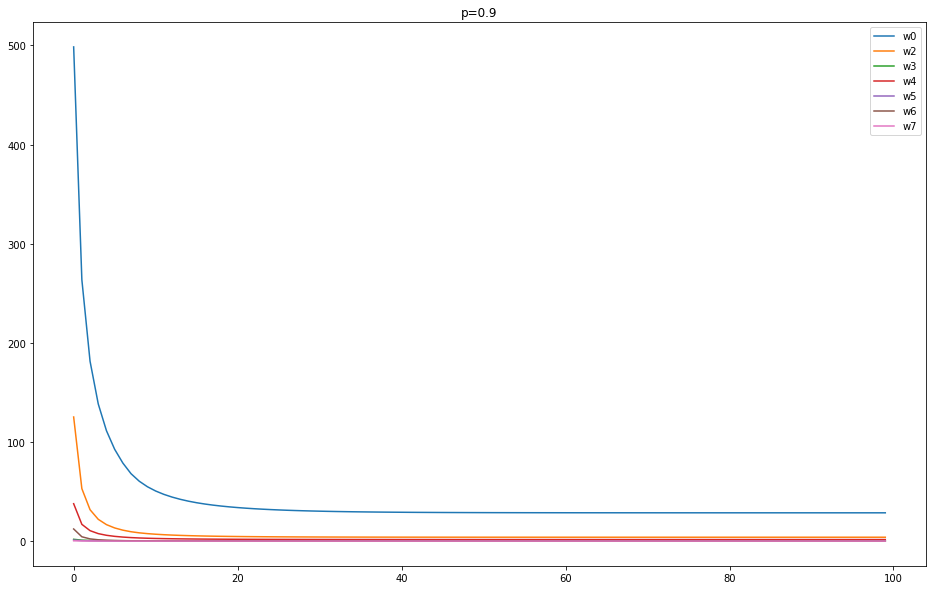
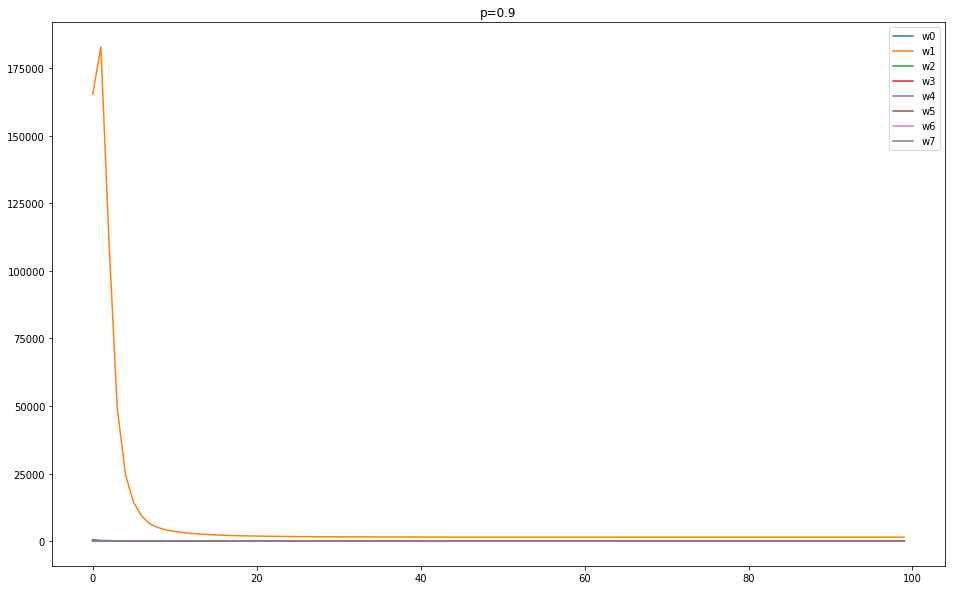
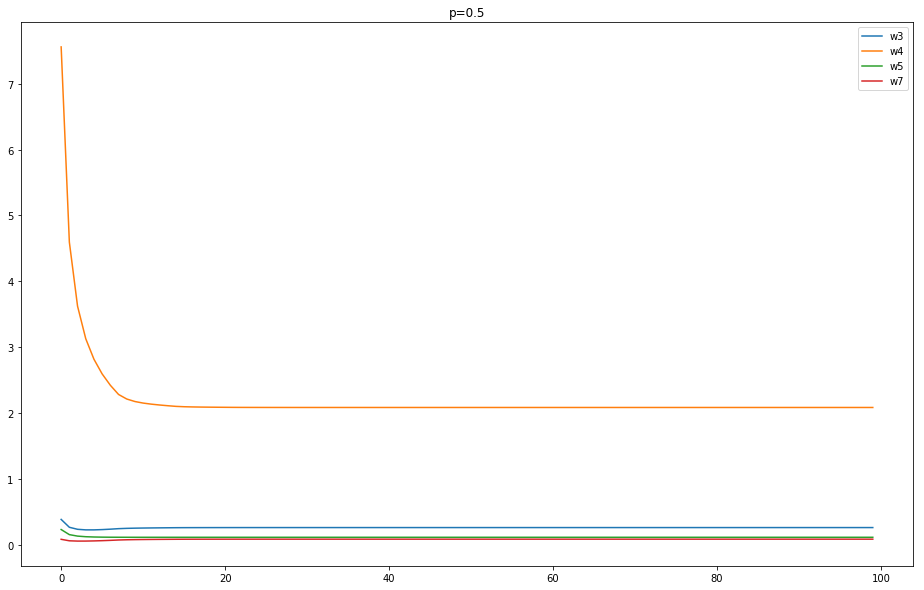
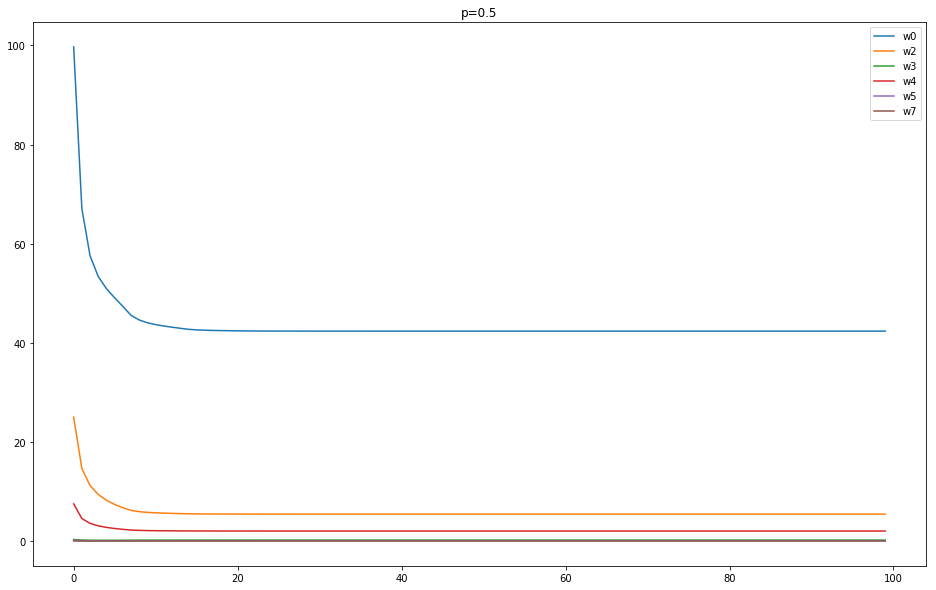
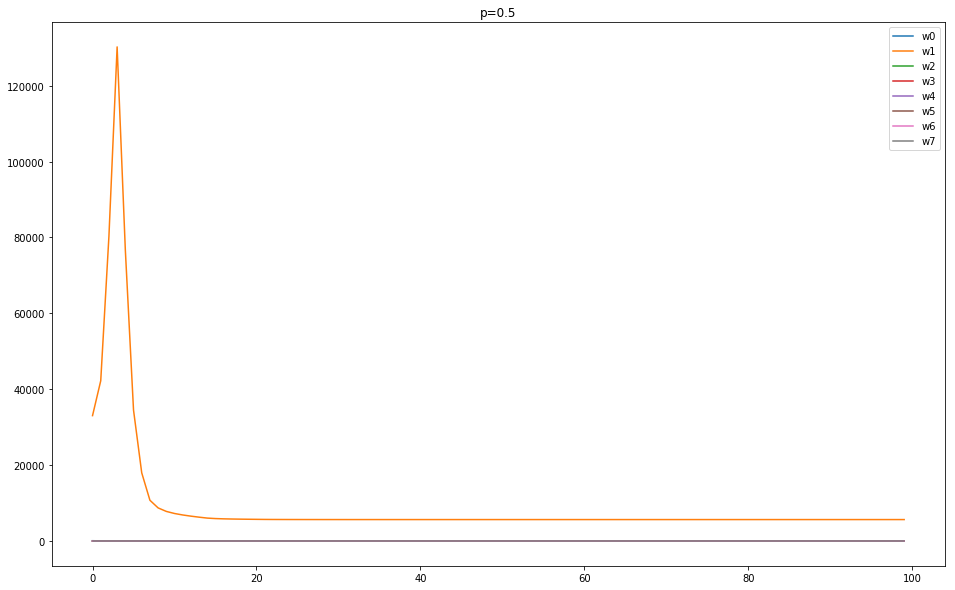
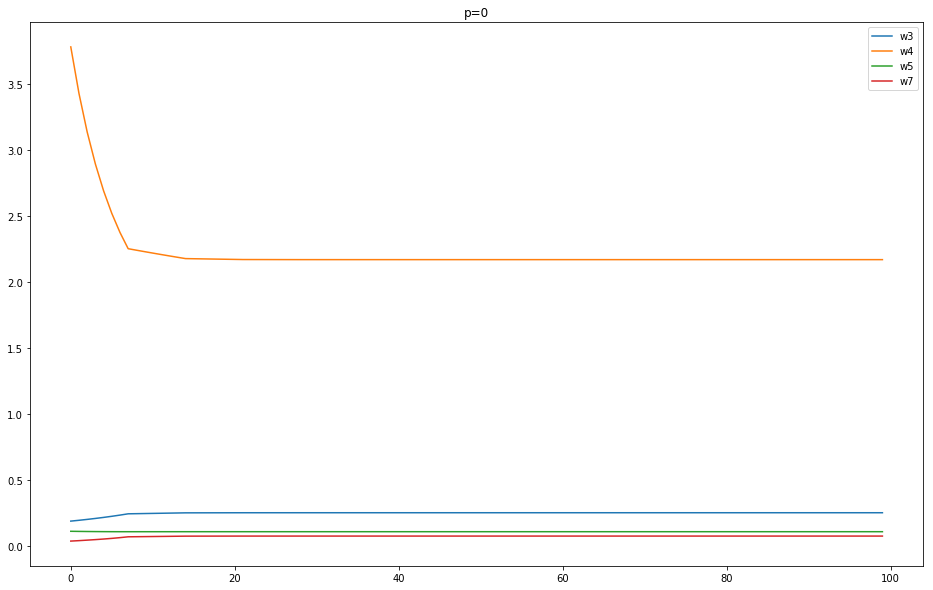
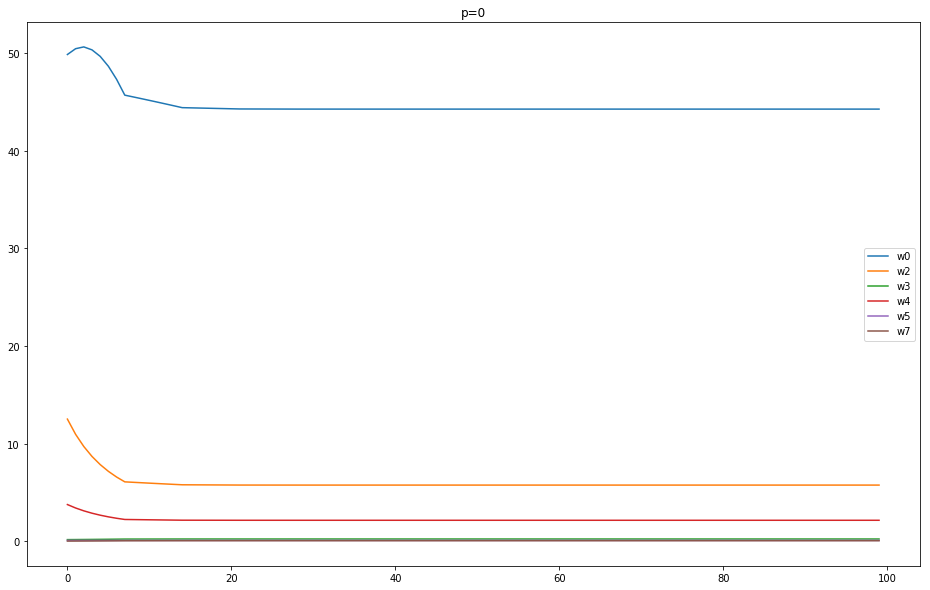


*б) Исследование динамики скорости обучения*

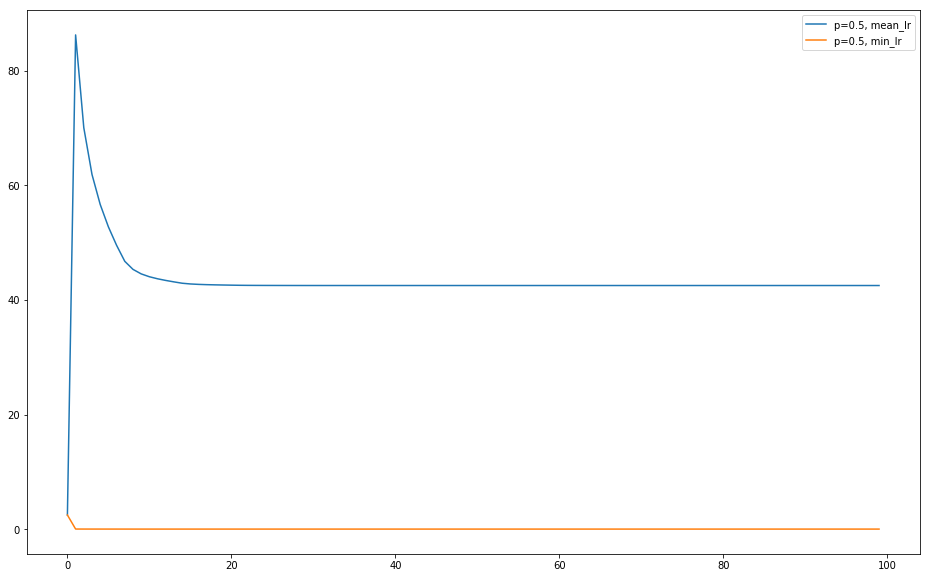
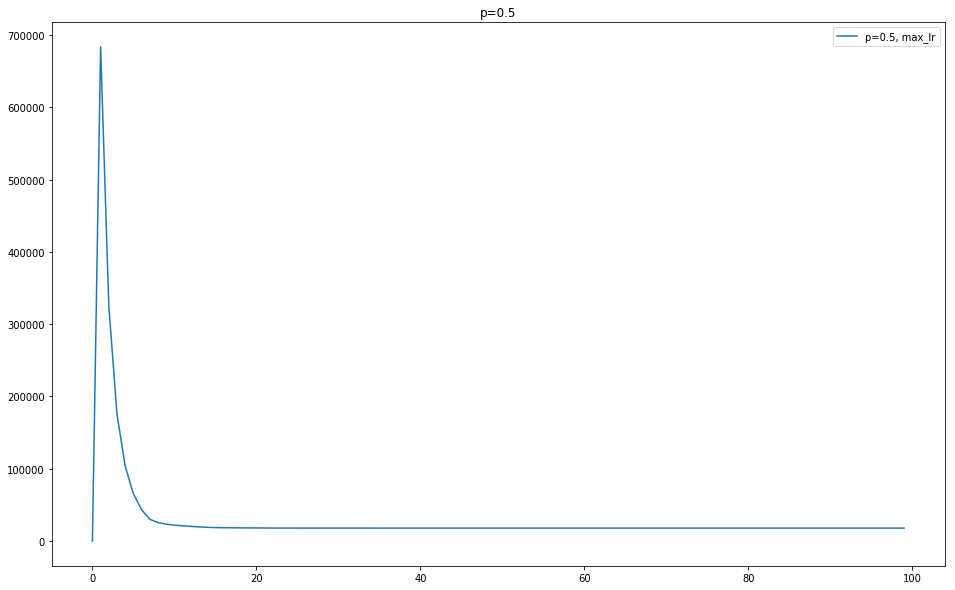
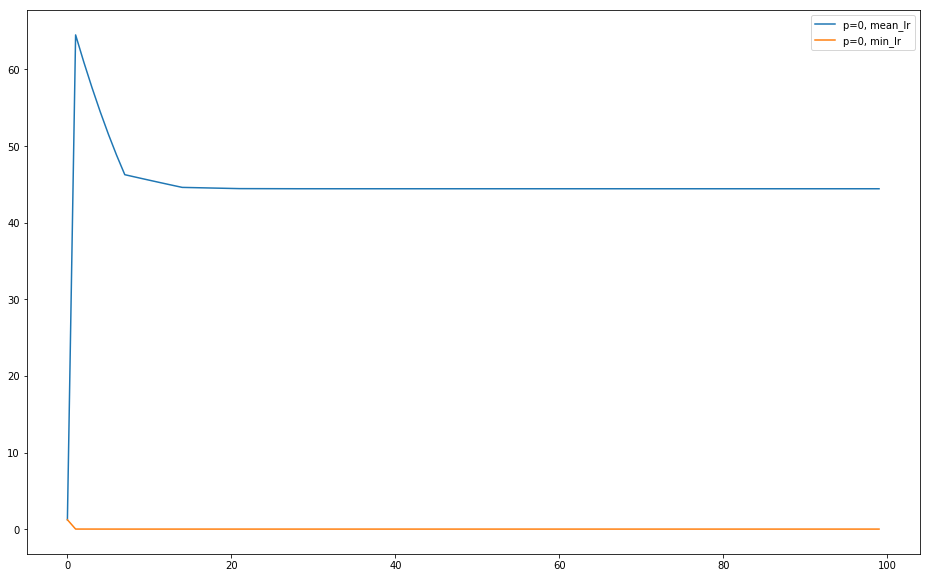
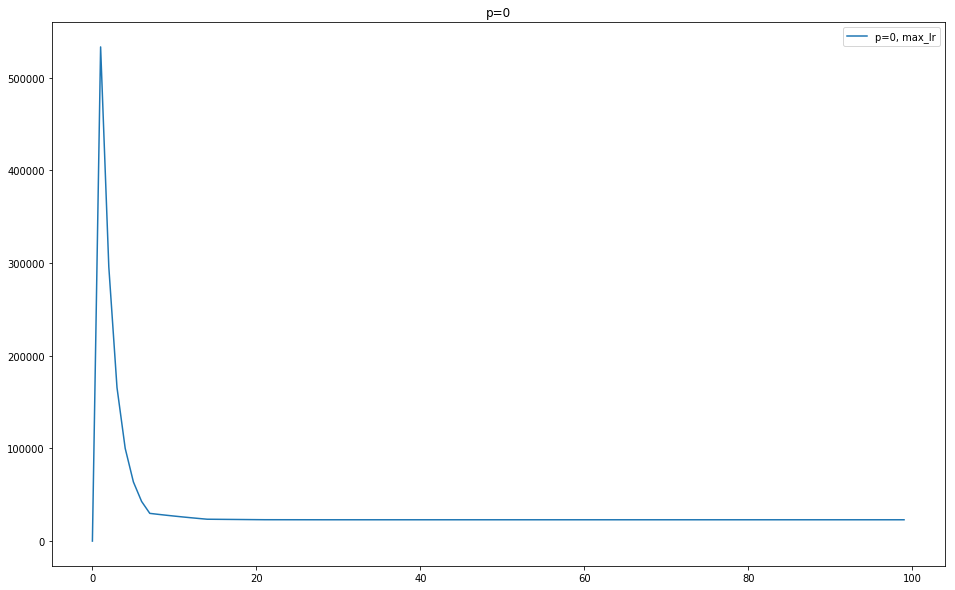
RMSProp:

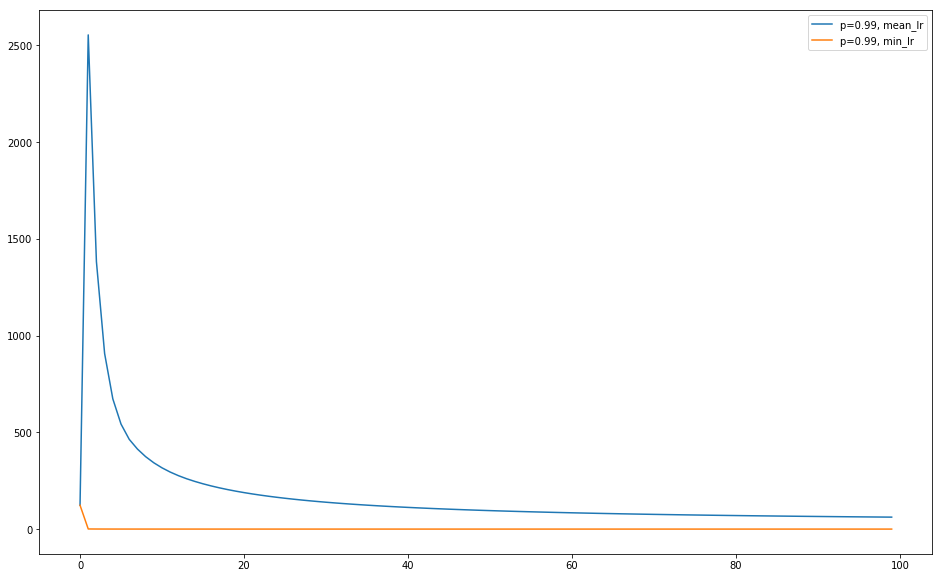
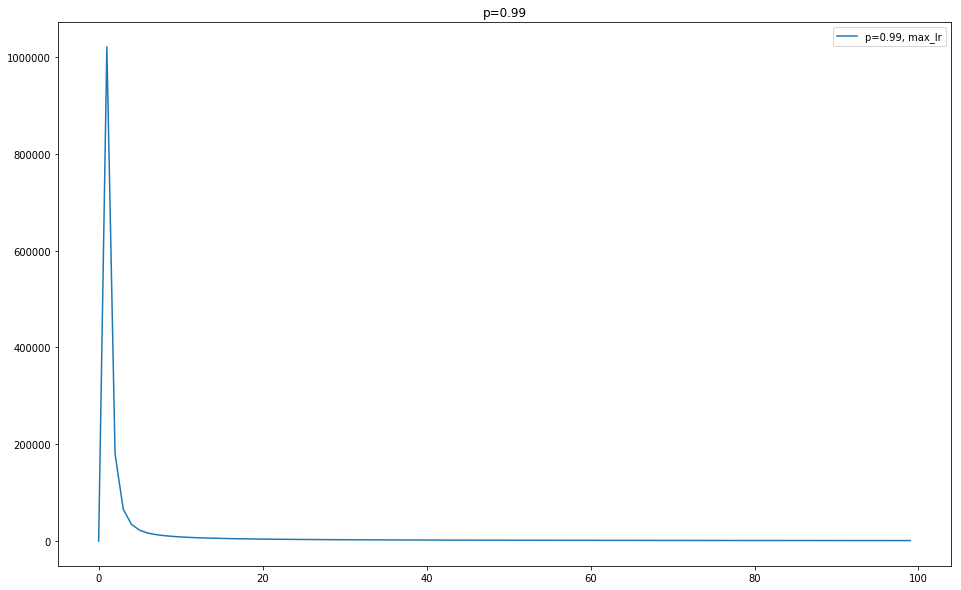
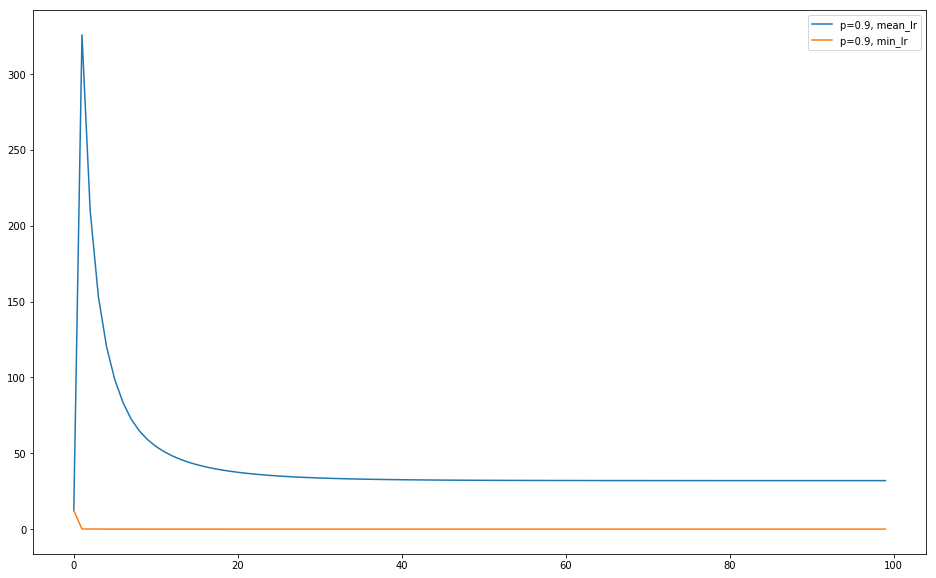
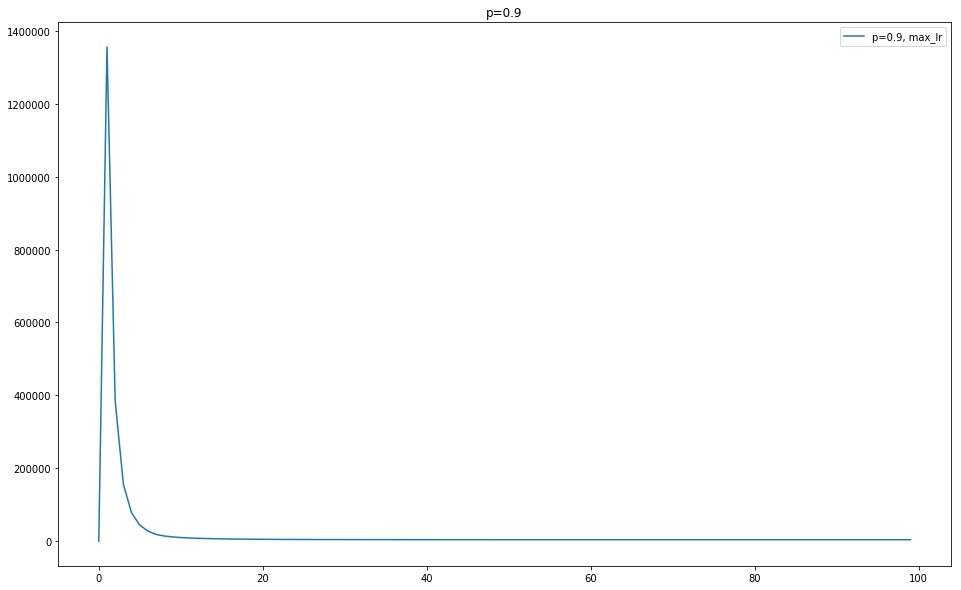
Синаптические коэффициенты:





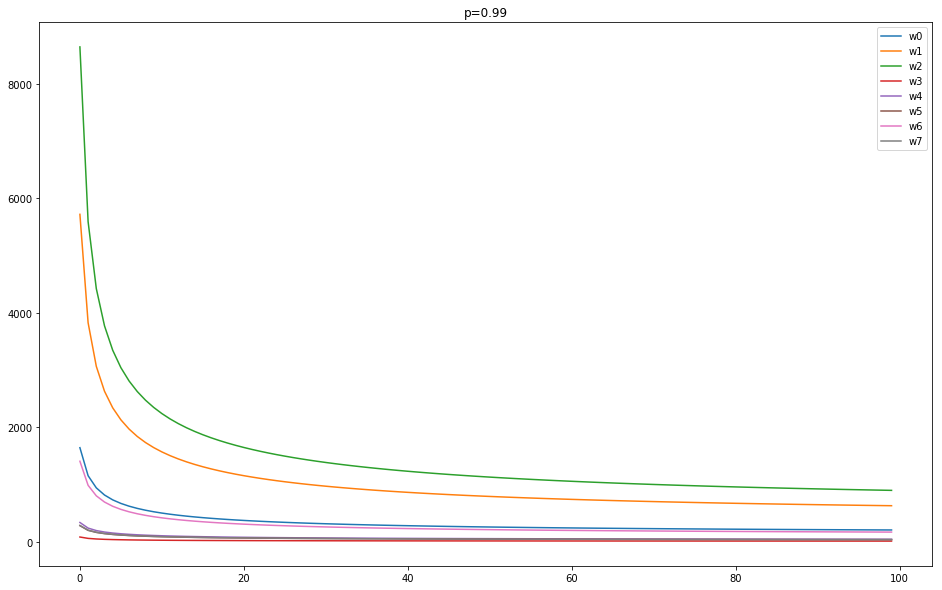
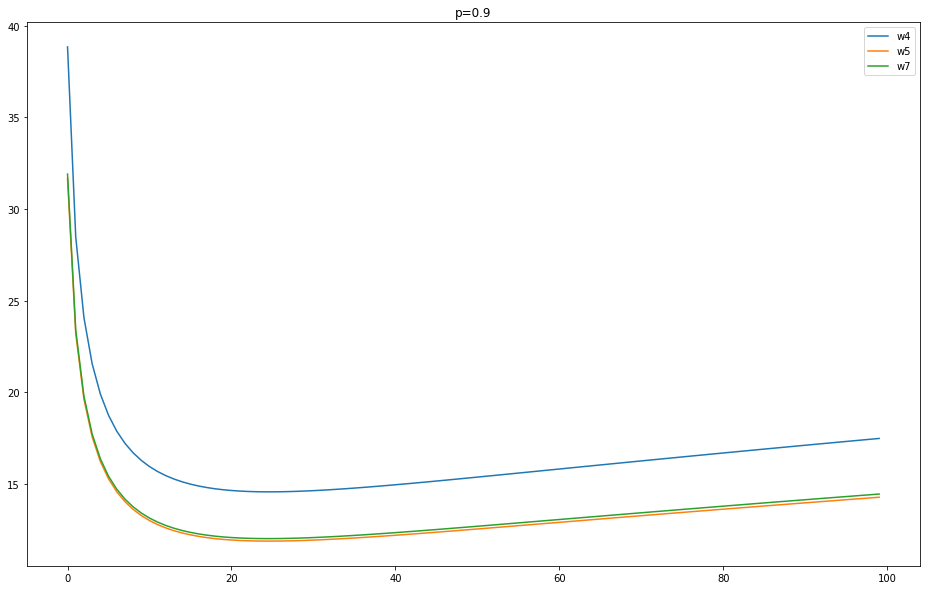
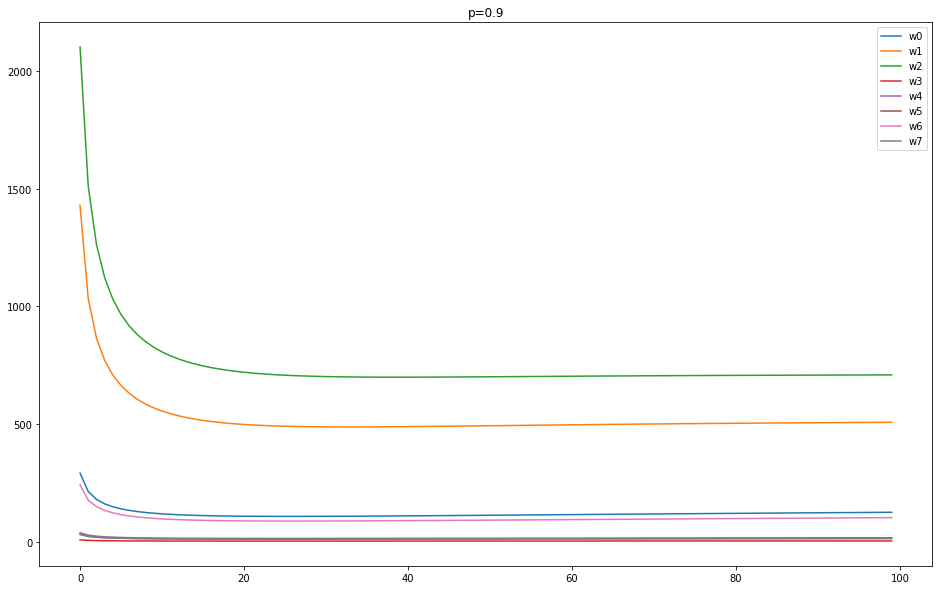
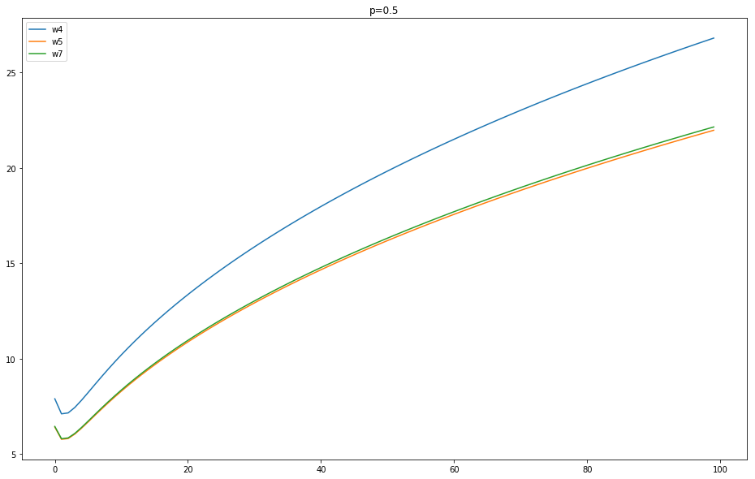
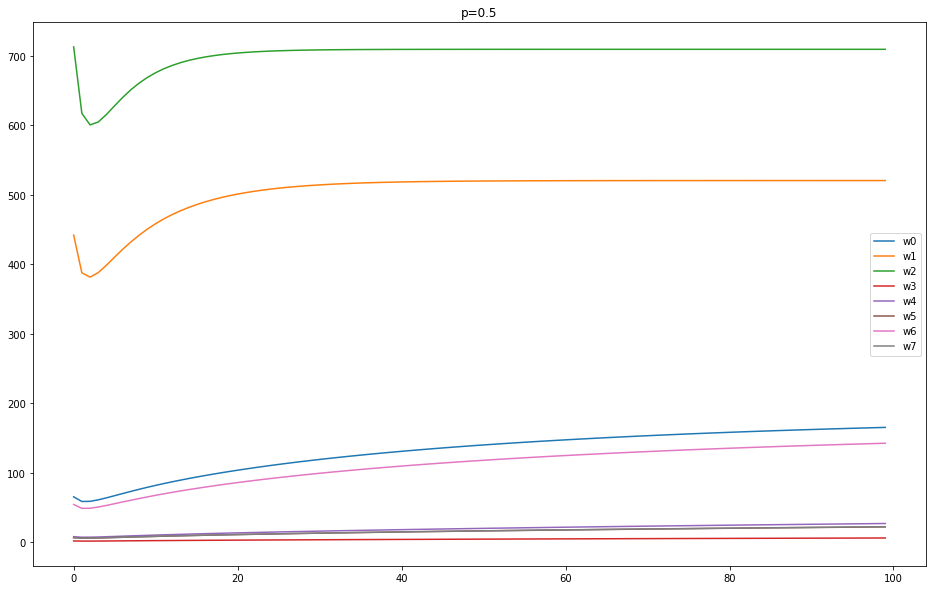
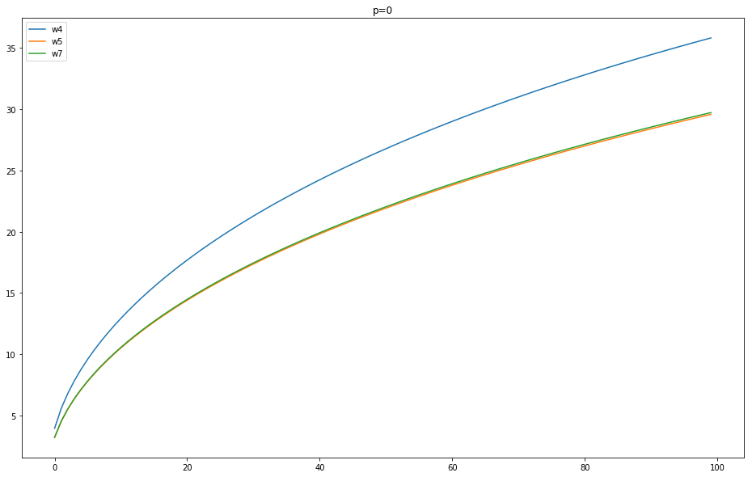
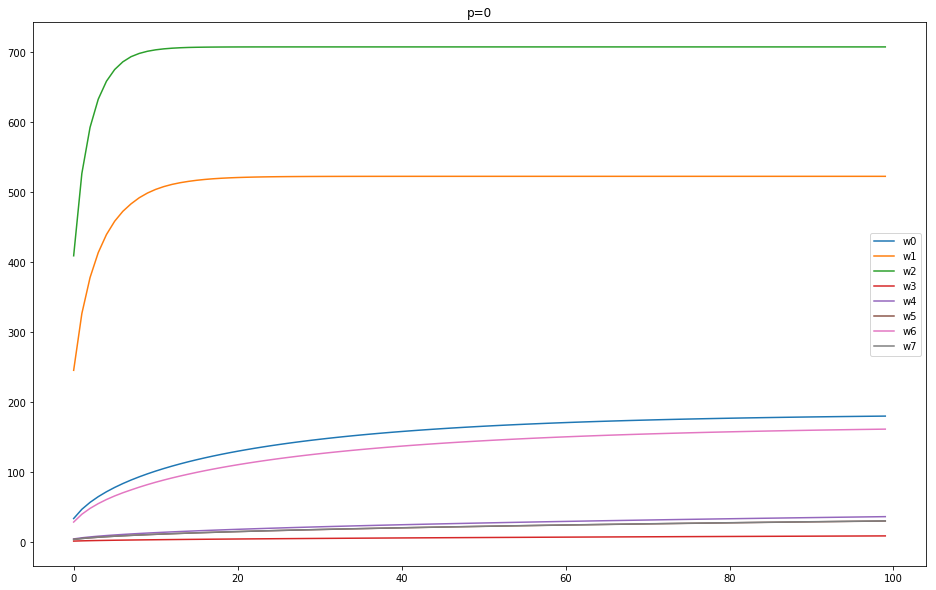
Скорости обучения:



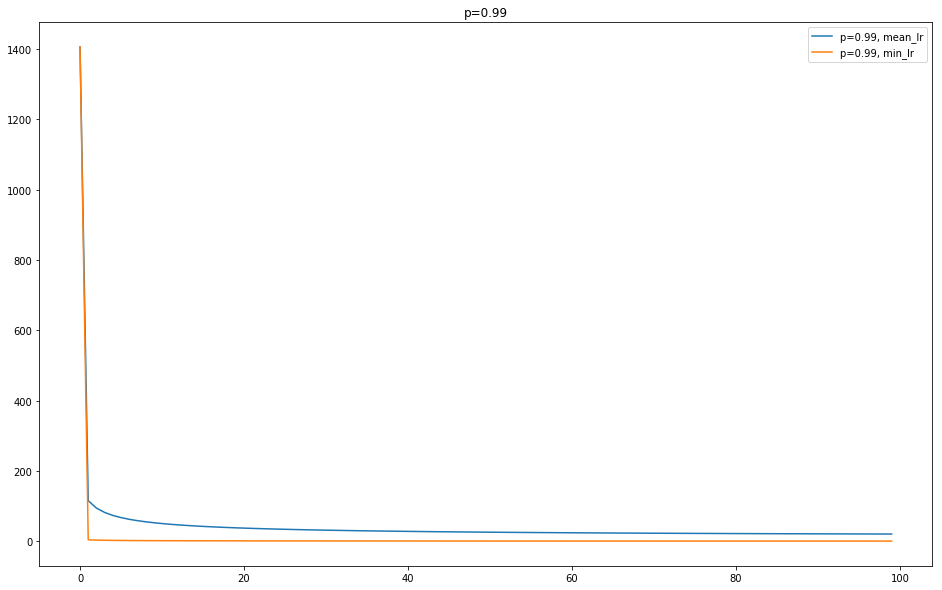
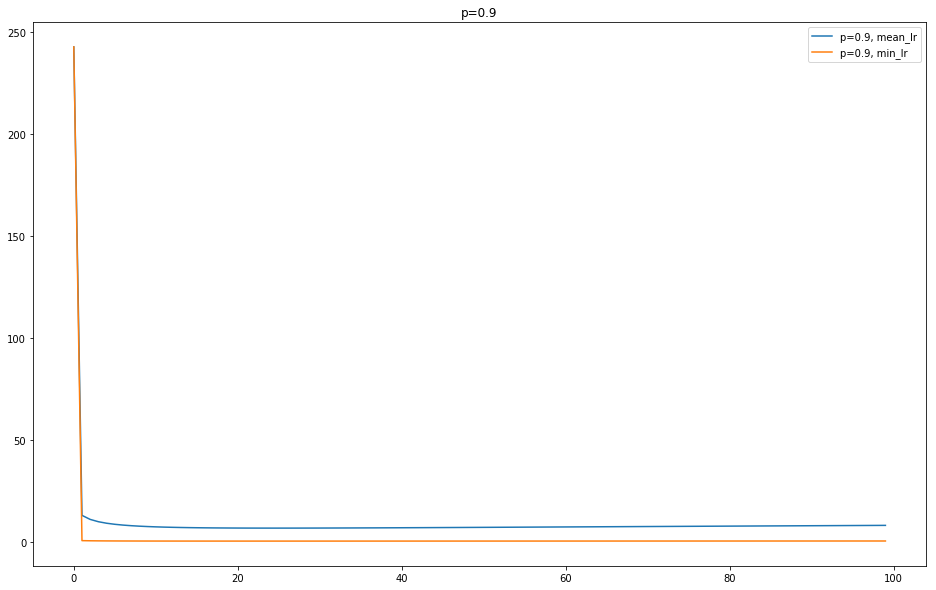
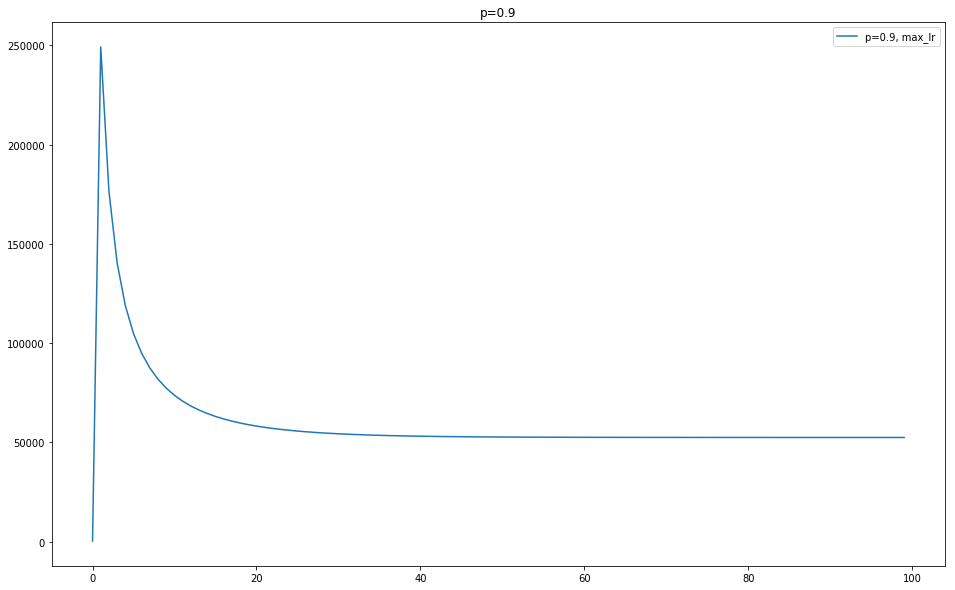
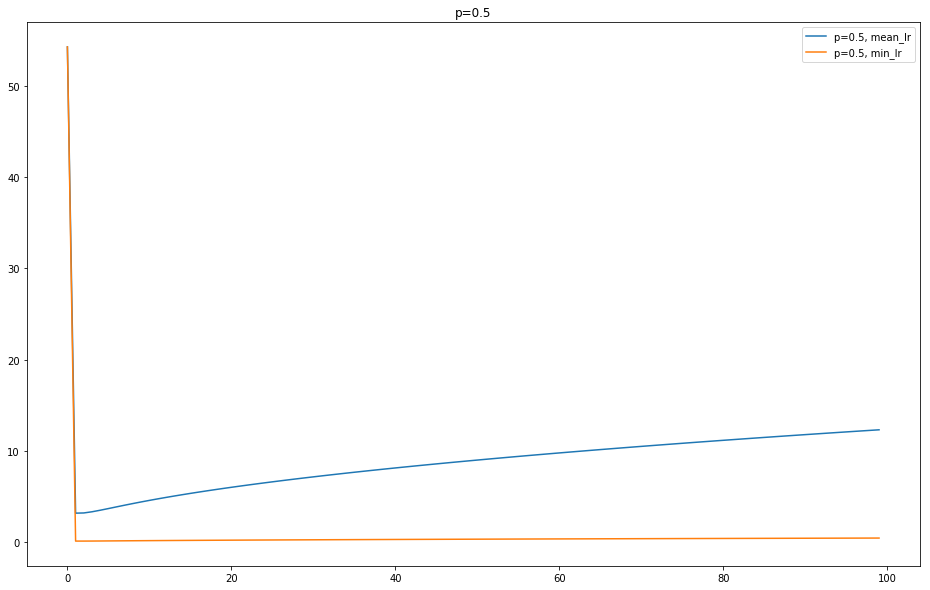
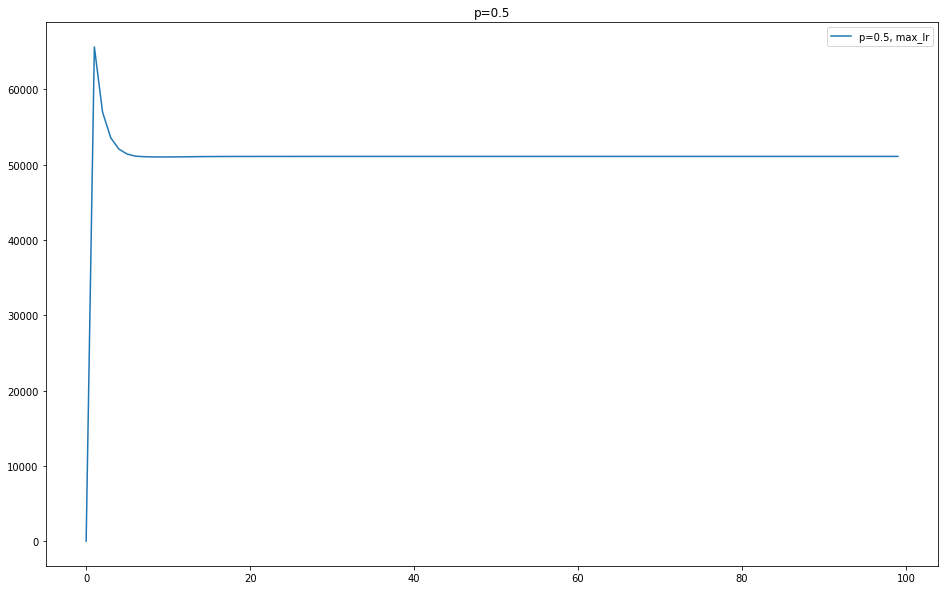
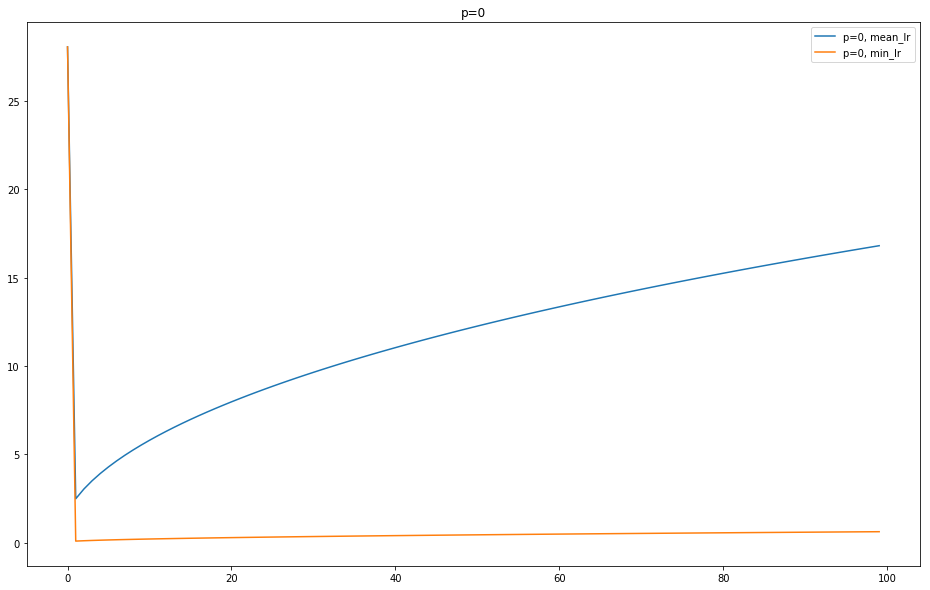
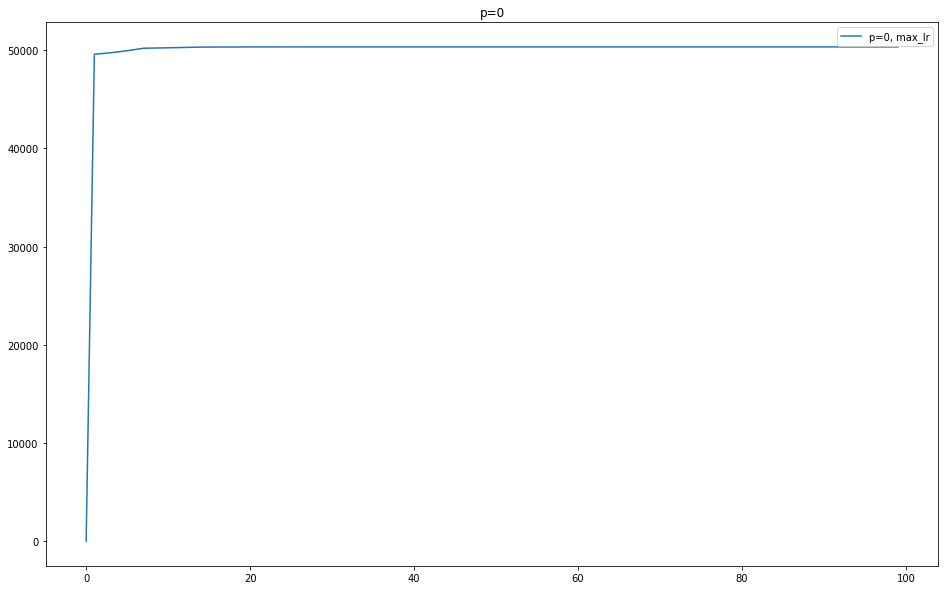


AdaDelta:

Синаптические коэффициенты:



Скорости обучения:



*в) Заполнить таблицу по результатам обучения*

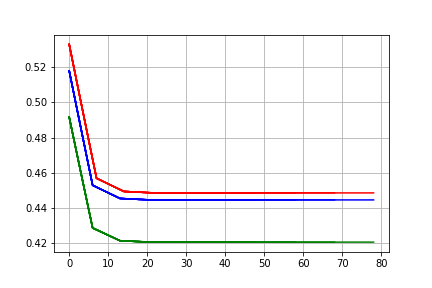
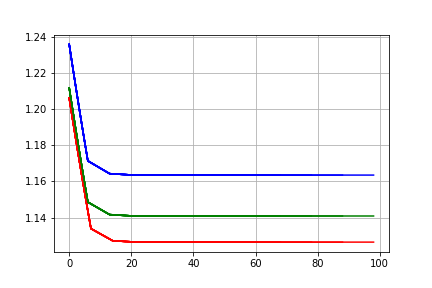
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Метод | Параметр сглаживания, ρ | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | RMSProp | 0 | 100 | 0.4484 | 0.4204 |
| 2 | AdaDelta | 100 | 0.1448 | 0.1474 |
| 3 | RMSProp | 0.5 | 100 | 0.4408 | 0.4131 |
| 4 | AdaDelta | 100 | 0.1447 | 0.1474 |
| 5 | RMSProp | 0.9 | 100 | 0.3779 | 0.3524 |
| 6 | AdaDelta | 100 | 0.1446 | 0.1472 |
| 7 | RMSProp | 0.99 | 100 | 0.1361 | 0.1237 |
| 8 | AdaDelta | 100 | 0.1443 | 0.1469 |

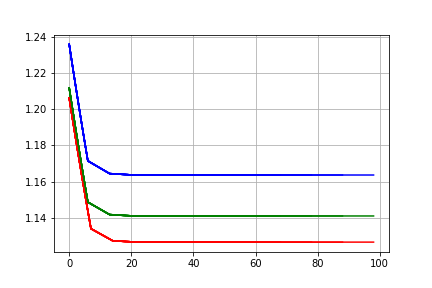
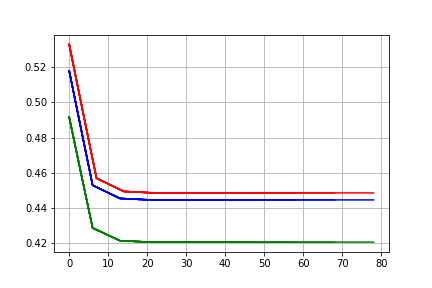
*г) Выводы*

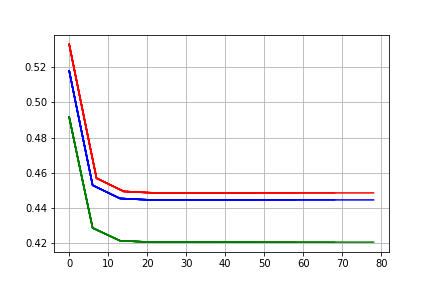
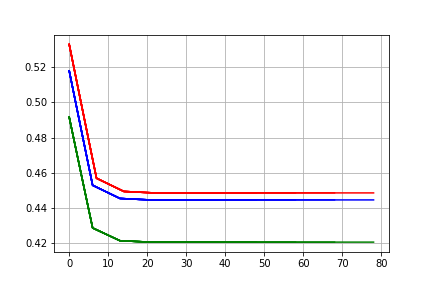
При приближении параметра сглаживания к единице результаты у обоих методов становятся лучше. При значениях параметра сглаживания до 0.9 включительно наблюдаются более качественные результаты у AdaDelta, но при значении 0.99 RMSProp достиг наилучшего в данном пункте результата.

*4.7. Исследование метода Adam*

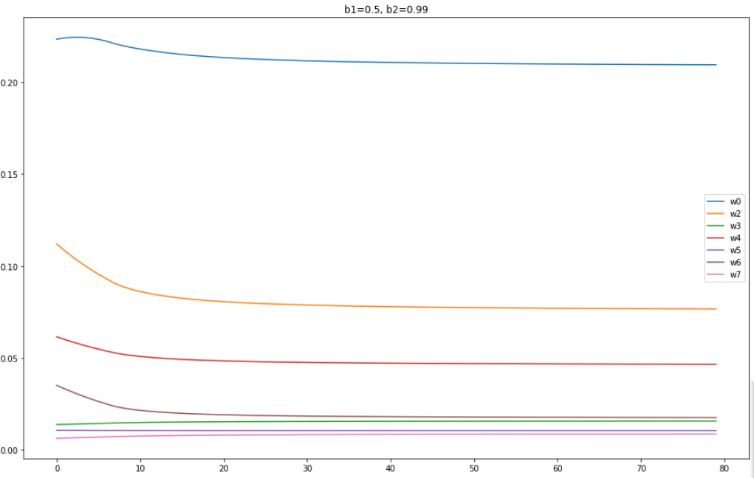
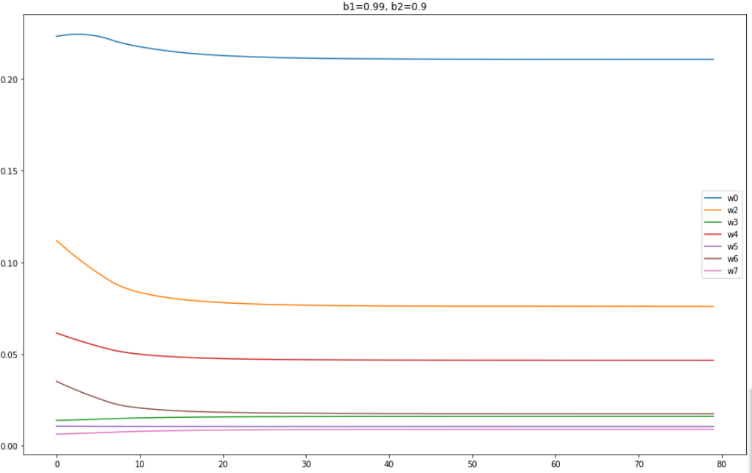
*а) Сравнение кривых обучения*

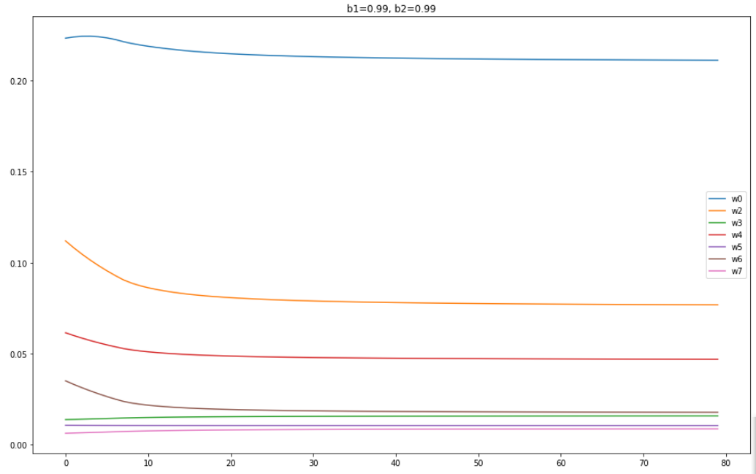
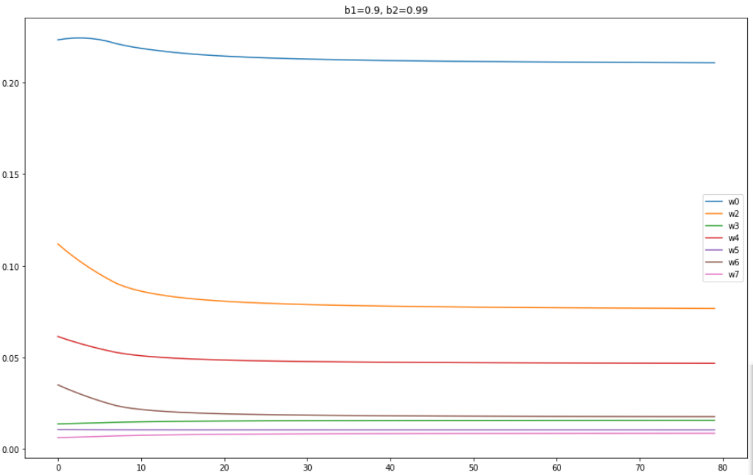




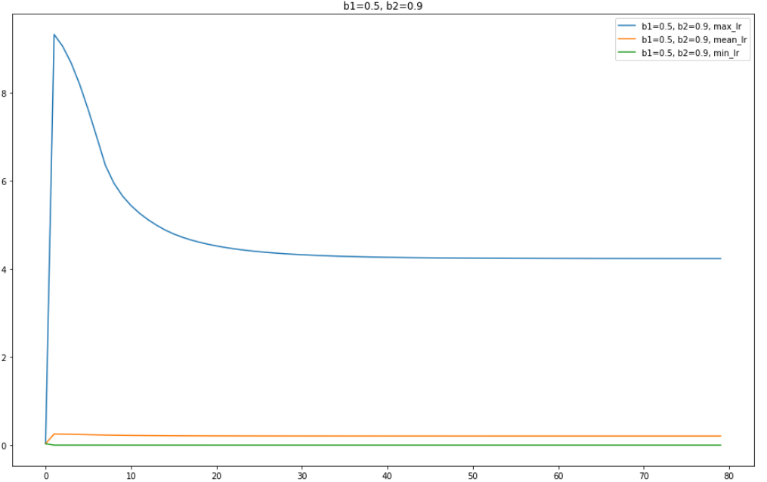
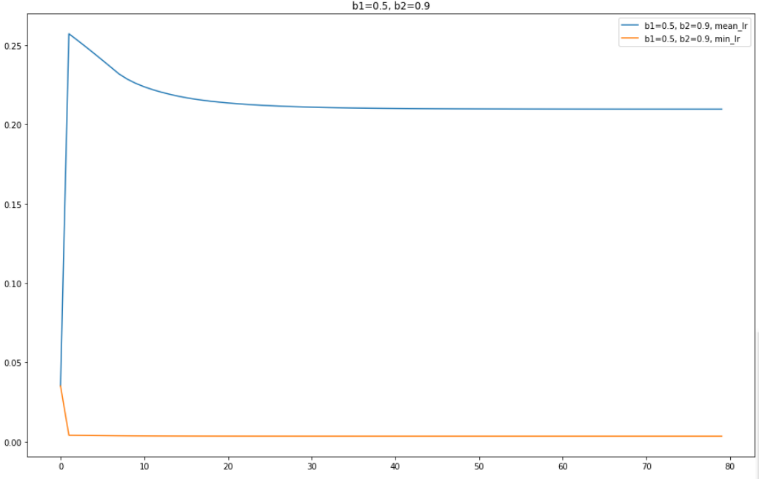


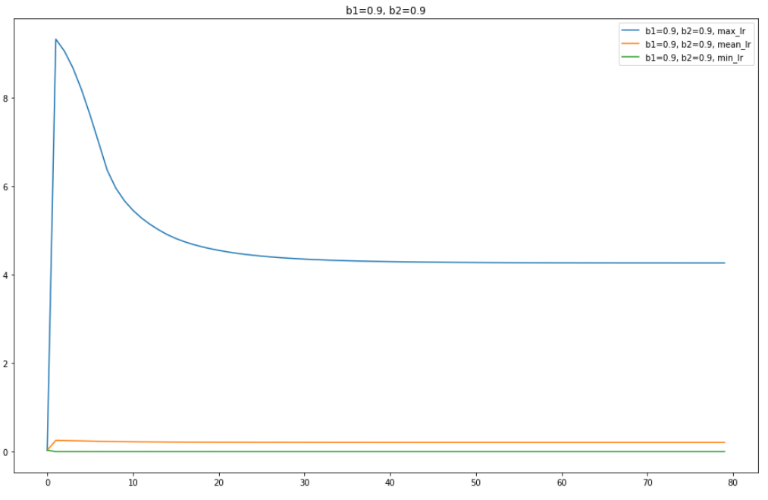
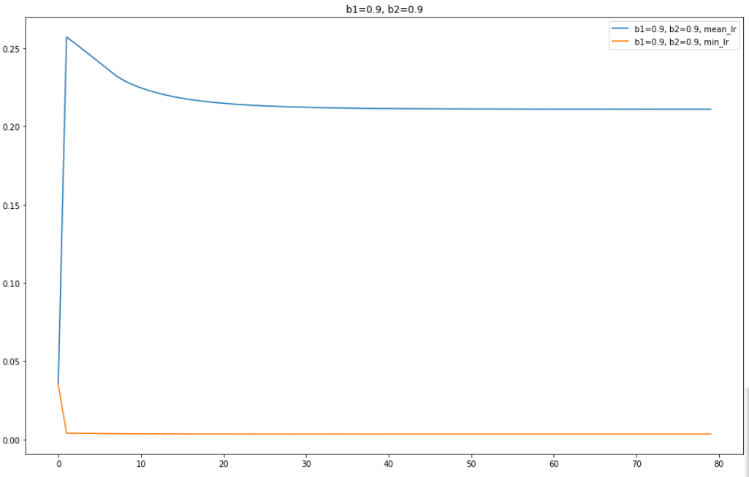
*б) Исследование динамики скорости обучения*

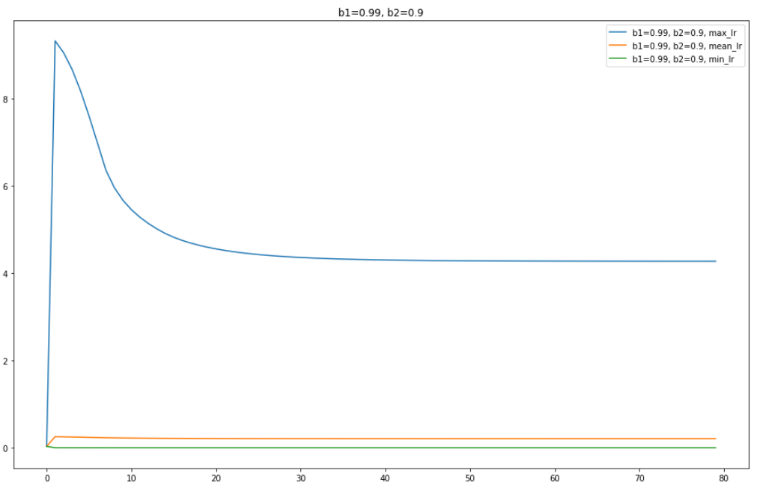
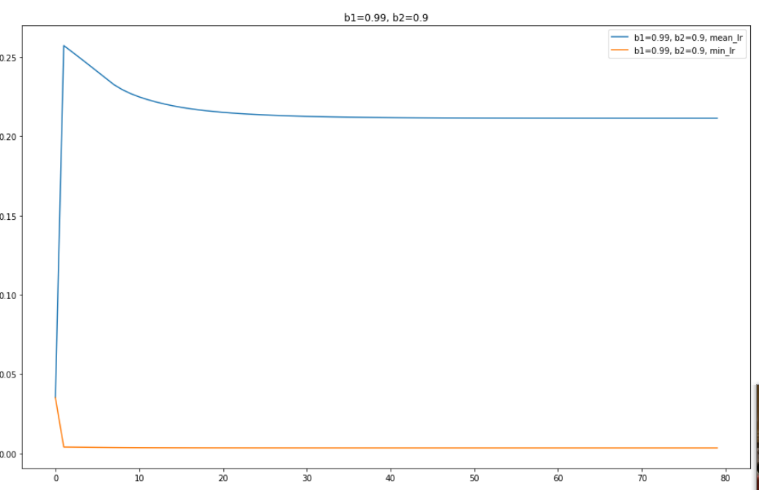
Графики зависимости скорости обучения отдельных синаптических коэффициентов сети: 

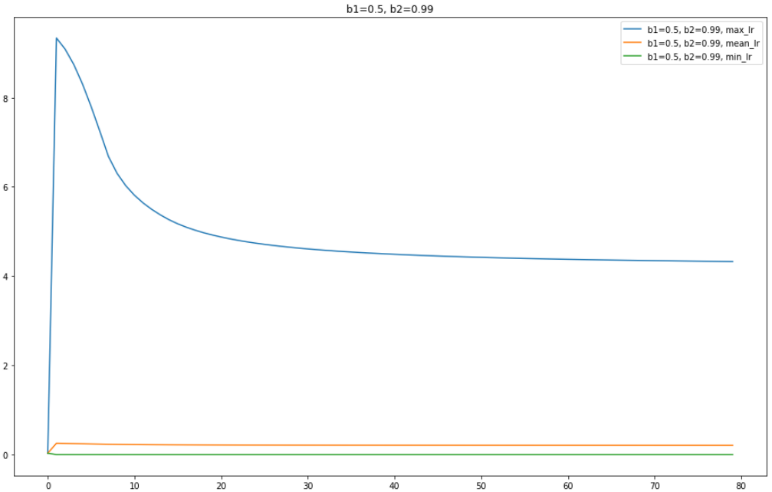
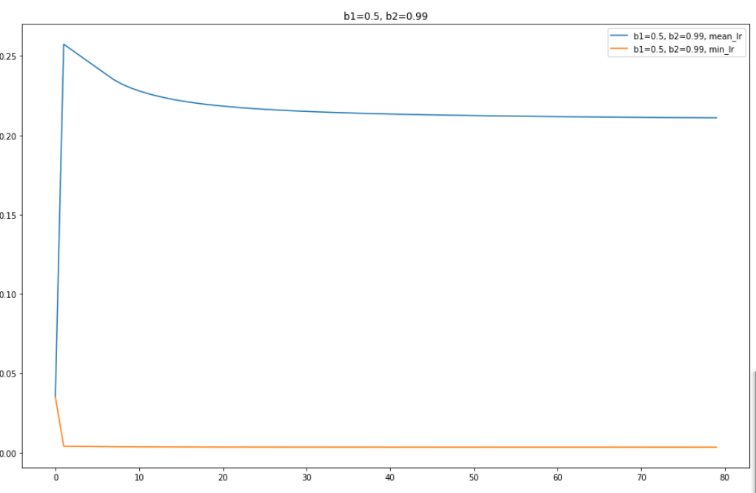


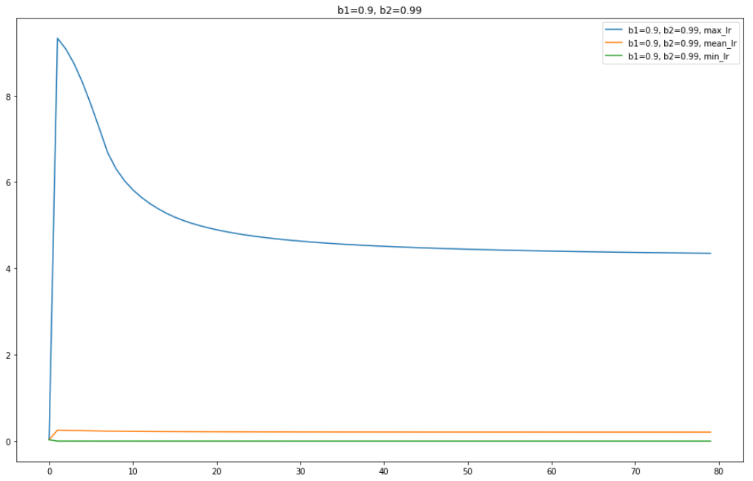
Графики зависимости скоростей обучения:

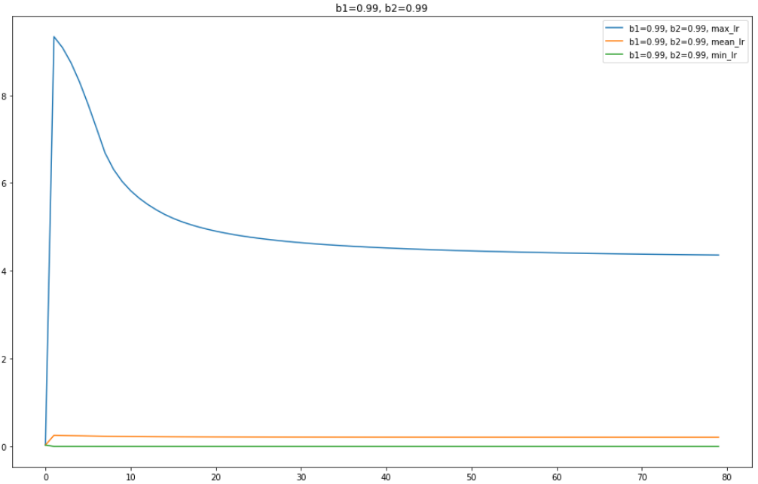
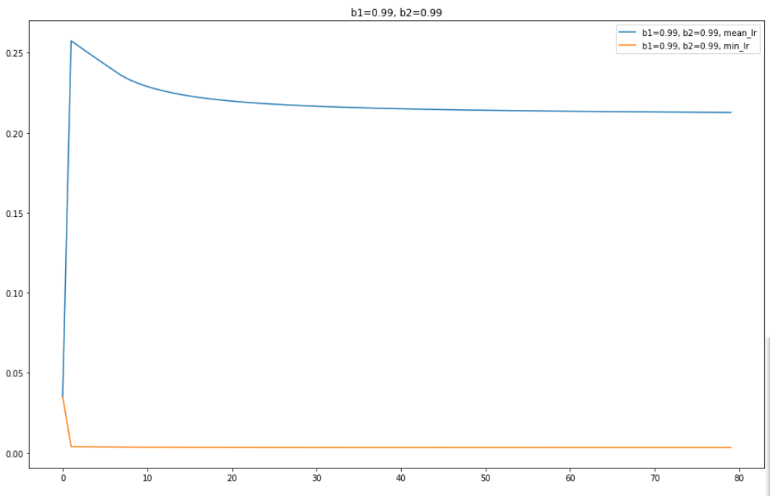












*в) Заполнить таблицу по результатам обучения*

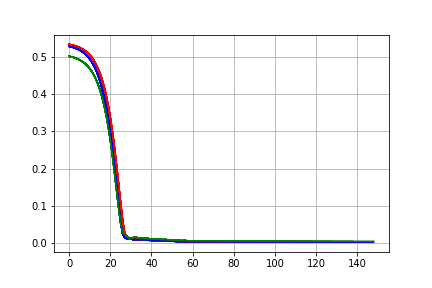
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | β1 | β2 | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | 0.5 | 0.9 | 80 | 0.4484 | 0.4204 |
| 2 | 0.9 | 0.9 | 80 | 0.4485 | 0.4205 |
| 3 | 0.99 | 0.9 | 80 | 0.4485 | 0.4205 |
| 4 | 0.5 | 0.99 | 80 | 0.4484 | 0.4204 |
| 5 | 0.9 | 0.99 | 80 | 0.4485 | 0.4205 |
| 6 | 0.99 | 0.99 | 80 | 0.4485 | 0.4205 |

*г) Выводы*

При увеличении значений параметров ошибка тоже растет, но незначительно. Можно сделать вывод, что незначительное увеличение одного или обоих из параметров не приводит к изменению качества, однако при значительном увеличении параметров качество немного ухудшилось.

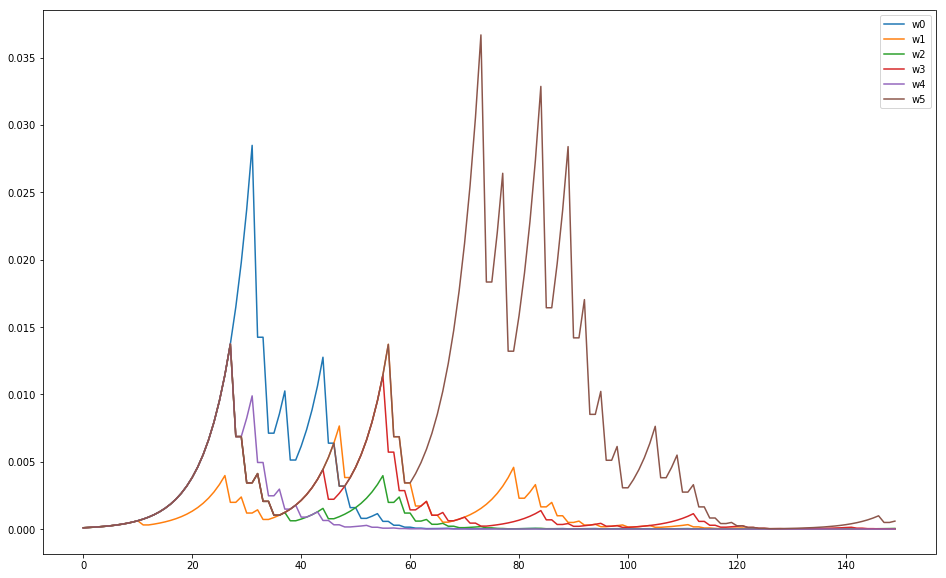
*4.8. Исследование метода RProp*

*а) Кривые обучения*

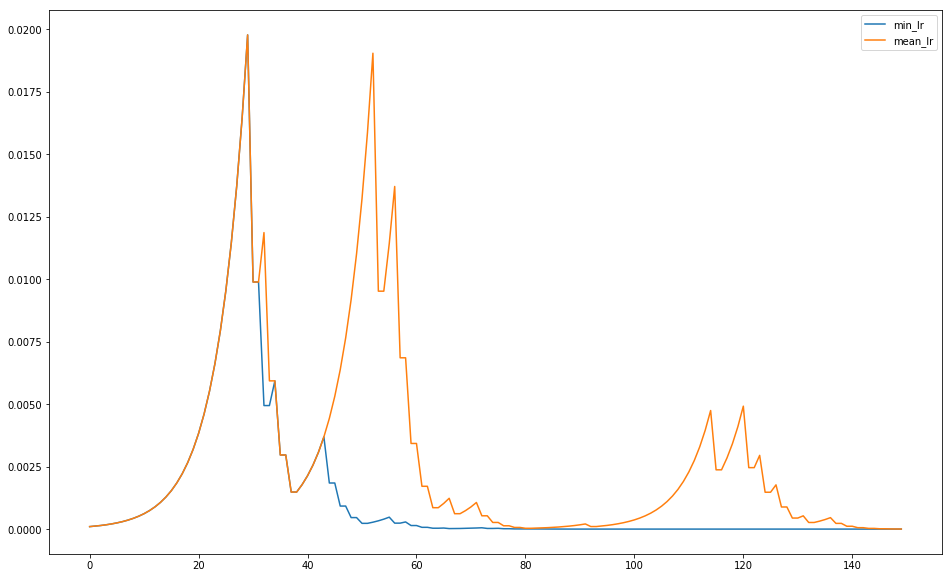
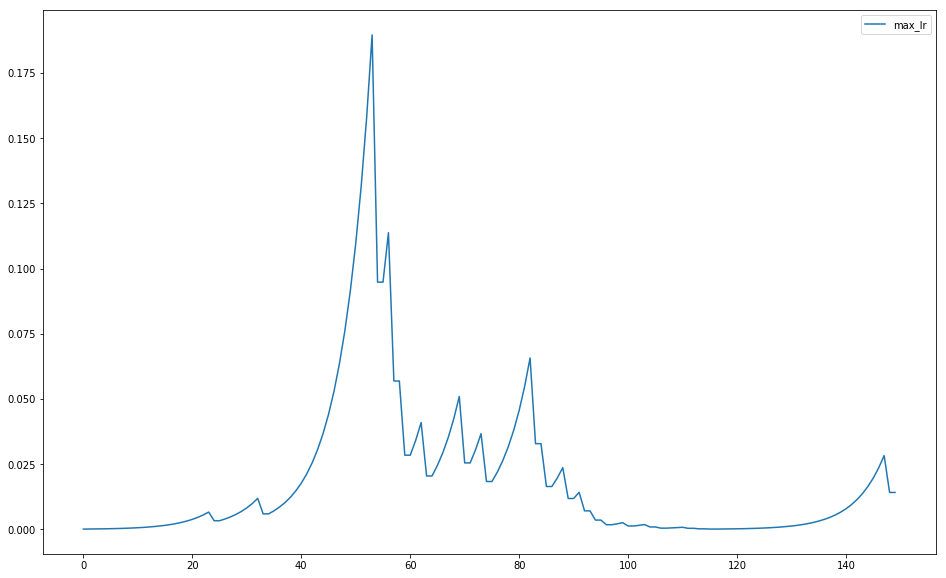


*б) Исследование динамики приращений весов*

Графики зависимости приращений отдельных синаптических коэффициентов сети:



Графики зависимости приращения:



*в) Заполнить таблицу по результатам обучения*

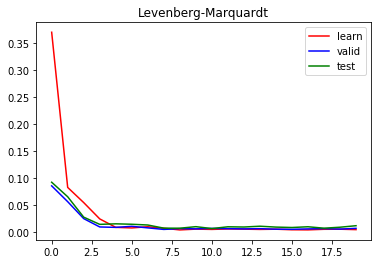
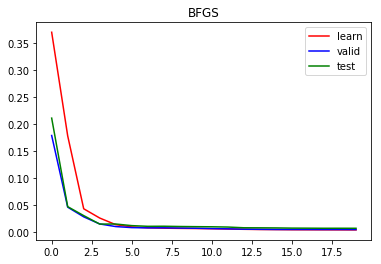
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Метод | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | RProp | 150 | 0.0031 | 0.0054 |

*г) Выводы*

Качество обучения – одно из лучших в сравнении с рассмотренными оптимизаторами. Ошибки на обучающей и тестовой выборках практически одинаковые.

*4.9. Исследование методов Левенберга-Маркардта и BFGS*

*а) Кривые обучения*

*б) Заполнить таблицу по результатам обучения*

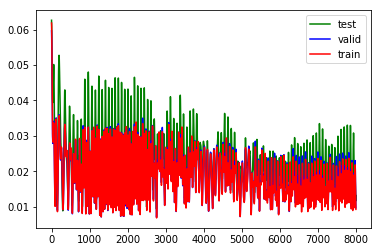
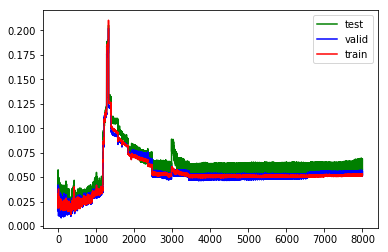
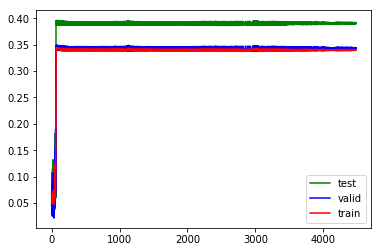
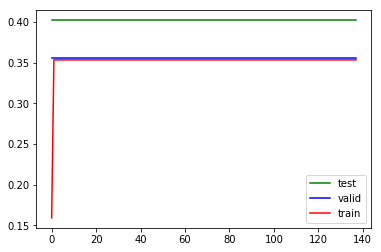
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Метод | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | LM | 20 | 0.0048 | 0.0093 |
| 2 | BFGS | 20 | 0.0048 | 0.0078 |

*в) Выводы*

У метода LM ошибка на тестовой выборке оказалась больше, чем у BFGS. Стоит также отметить, что реализация данных методов была такой, что метод LM работал намного медленнее (по времени), чем BFGS.

*4.10. Исследование метода стохастического градиента*

*а) Сравнение кривых обучения*

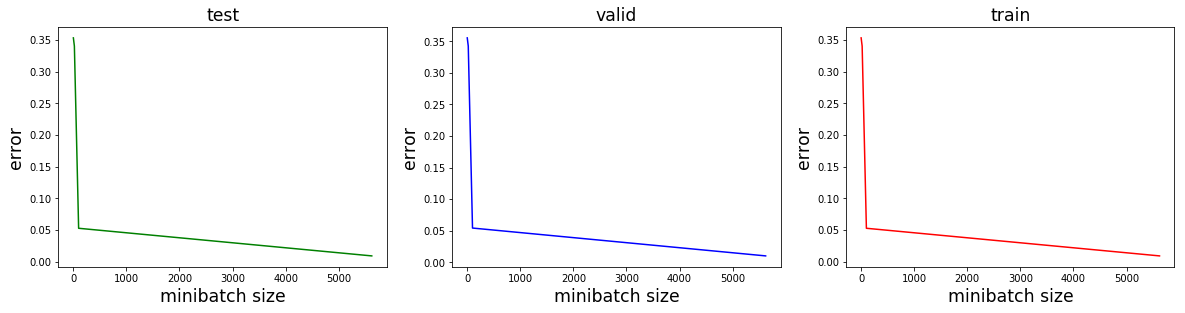


*б) Заполнить таблицу по результатам обучения*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Метод | Размер mini-batch’а | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | GD | 1 | 138 | 0.3533 | 0.402 |
| 2 | GD | 20 | 4485 | 0.3404 | 0.3915 |
| 3 | GD | 100 | 8000 | 0.0528 | 0.0576 |
| 4 | GD | равен объёму выборки=5622 | 8000 | 0.0092 | 0.0131 |

*в) Исследование влияния размера mini-batch’а на качество обучения*

Графики зависимости ошибок обученной сети от размера mini-batch’а:

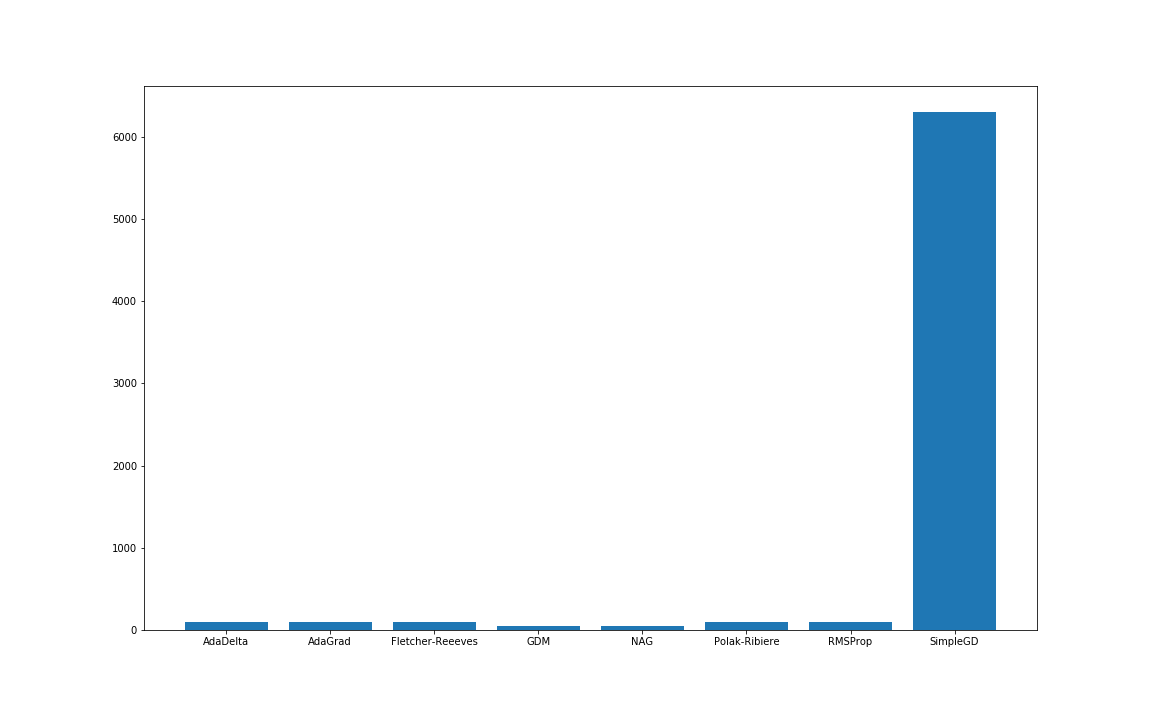


*г) Выводы*

При увеличении размера mini-batch’а улучшается и качество обучения. Также уменьшаются временные завтра.

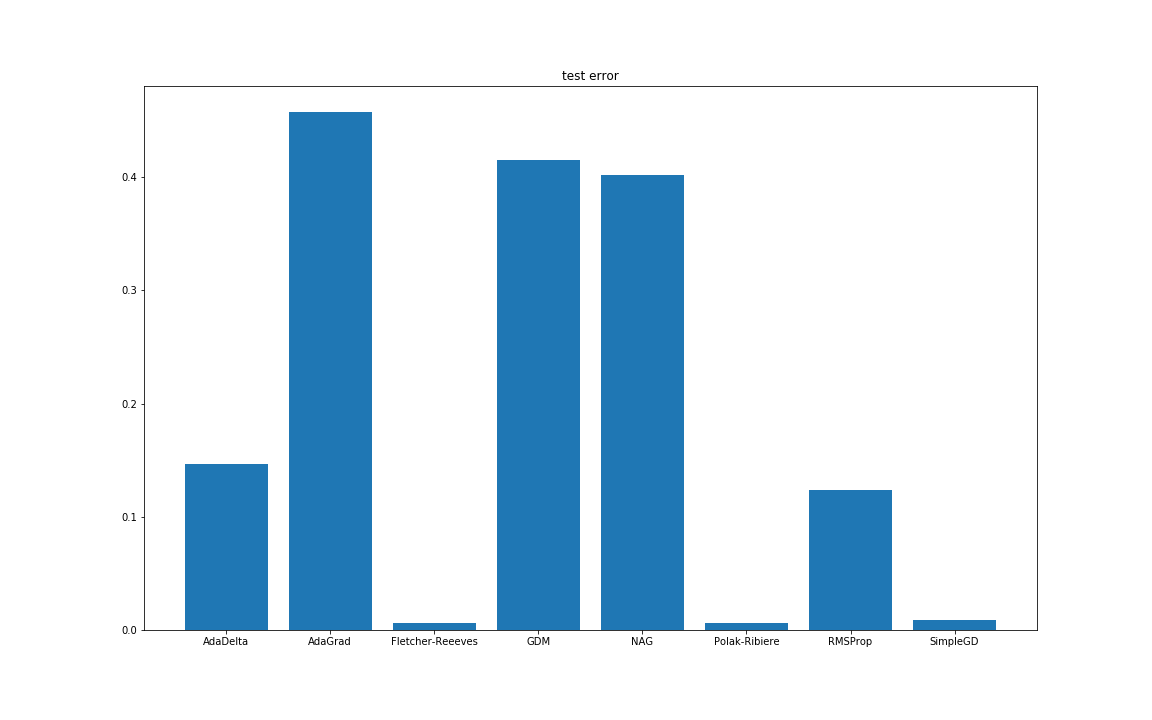
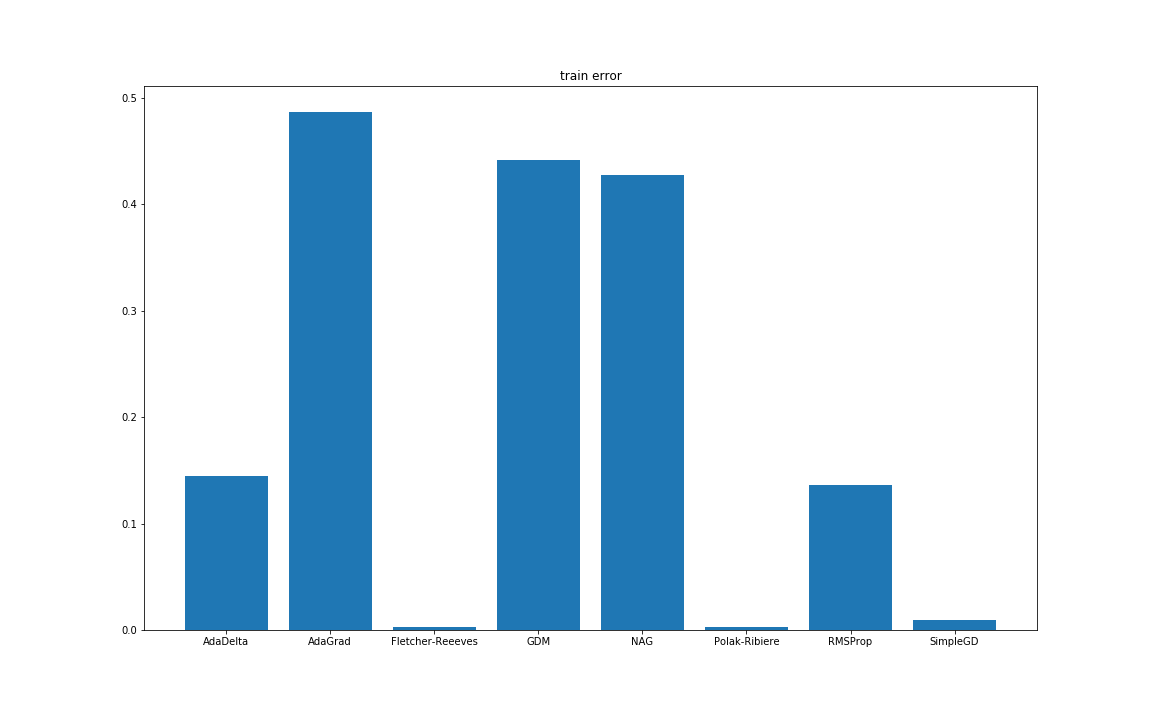
*4.11. Сравнение методов обучения*

*а) Сравнение числа эпох обучения*





*б) Сравнение качества обученных нейросетевых моделей*



*в) Выводы*

Сделать выводы по результатам сравнения методов обучения (GD, GDM, NAG, SGD, Fletcher-Reeeves, Polak-Ribiere, AdaGrad, RMSProp, AdaDelta, RProp, LM, BFGS, Stochastic GD).

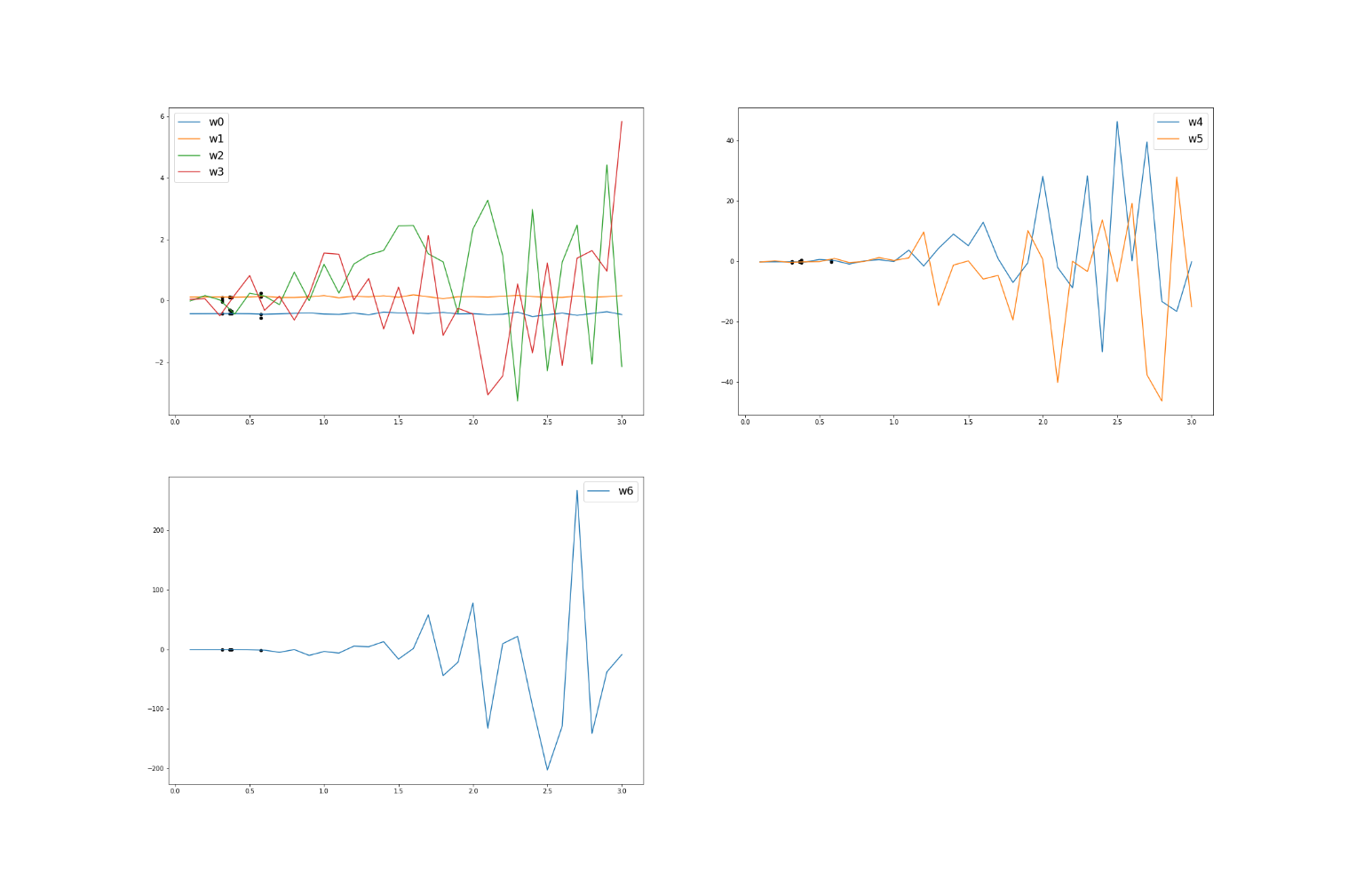
*4.12. Исследование влияния способа инициализации весов на качество обучения*

*а) Исследование прямого информационного потока в сети*

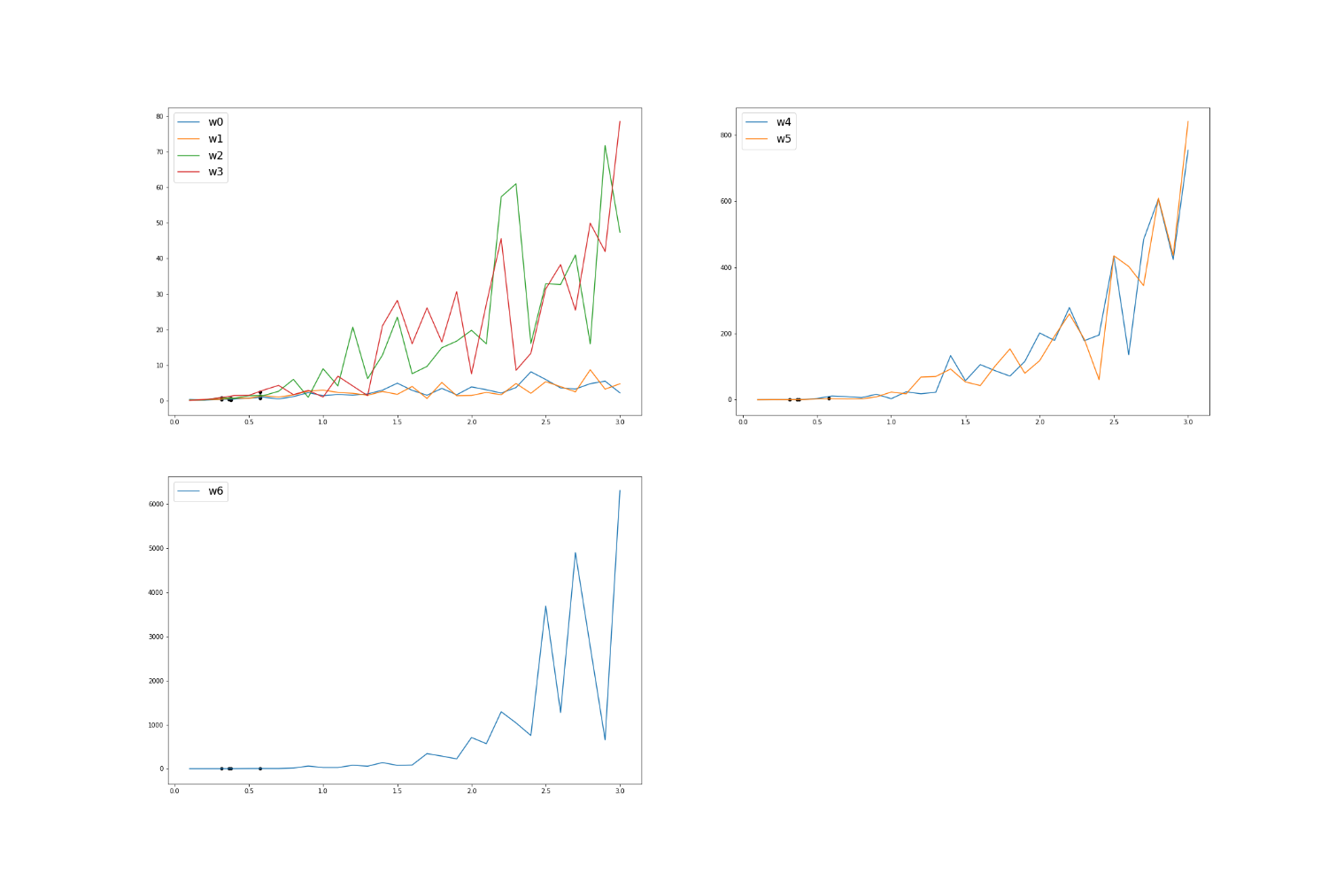
Были взяты следующие нейроны:

1. 1 слой, 4й нейрон
2. 1 слой, 6й нейрон
3. 2 слой, 1й нейрон
4. 2 слой 10й нейрон
5. 3 слой, 3й нейрон
6. 3 слой, 5й нейрон
7. 4 слой, 1й нейрон

Графики зависимости среднего значения выходов при иницилизации весов случ.числами, распределенными по закону *N*(0; σ) от значения σ (черными точками отмечены значения σ, рассчитанные по методу Хавьера для различных слоёв сети):



Графики зависимости с.к.о. выходов при иницилизации весов случ.числами, распределенными по закону *N*(0; σ) от значения σ (черными точками отмечены значения σ, рассчитанные по методу Хавьера для различных слоёв сети):



*б) Исследование обратного информационного потока в сети*

Для тех же нейронов построить графики зависимости среднего значения и с.к.о. (по примерам обучающей выборки) двойственных потенциалов при инициализации весов сети случайными числами, распределёнными по нормальному закону *N*(0; σ), от значения с.к.о. σ. Отметить на графике значения σ, рассчитанные по методу Хавьера для различных слоёв сети.

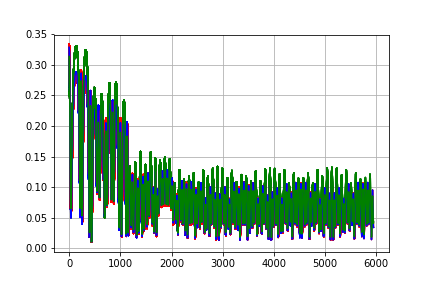
*в) Исследование распределений выходов и двойственных потенциалов нейронов*

Построить гистограммы распределения выходов и двойственных потенциалов (по примерам обучающей выборки) тех же нейронов, получаемые: 1) при слишком малых значениях σ; 2) при значениях σ, рассчитанных по методу Хавьера; 3) при слишком больших значениях σ.

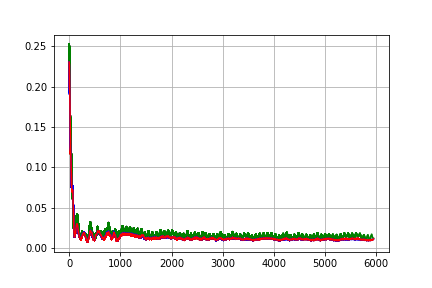
*г) Кривые обучения*

Графики зависимости ошибки сети на обучающей и тестовой выборках от времени обучения (кривые обучения) при инициализации весов сети случайными числами, распределёнными по нормальному закону *N*(0; σ):

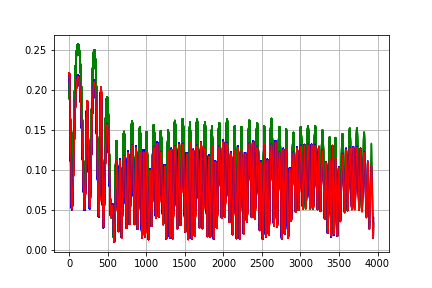
1) при слишком малых значениях σ (σ = 0.1):



2) при значениях σ, рассчитанных по методу Хавьера:



3) при слишком больших значениях σ (σ = 2.0):



**Указание**: lr = 0.001

*д) Заполнить таблицу по результатам обучения*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Распределение начальных весов | С.к.о. начальных весов | Средняя ошибка на обучающей выборке ± с.к.о., | Средняя ошибка на тестовой выборке ± с.к.о., |
| 1 | Равномерное | 0.1 | 0.0468 ± 0.0322 | 0.0496 ± 0.0328 |
| 2 | Нормальное | 0.0486 ± 0.0435 | 0.0526 ± 0.0376 |
| 3 | Усеченное нормальное | 0.0358 ± 0.0204 | 0.0443 ± 0.0235 |
| 4 | Равномерное | Рассчитано по методу Хавьера | 0.0313 ± 0.0184 | 0.035 ± 0.0144 |
| 5 | Нормальное | 0.0291 ± 0.0217 | 0.035 ± 0.026 |
| 6 | Усеченное нормальное | 0.036 ± 0.0259 | 0.0437 ± 0.0274 |
| 7 | Равномерное | 1.0 | 0.2094 ± 0.1335 | 0.2356 ± 0.1538 |
| 8 | Нормальное | 0.1074 ± 0.1298 | 0.127 ± 0.1456 |
| 9 | Усеченное нормальное | 0.0439 ± 0.0339 | 0.0489 ± 0.0298 |

**Указание 1**: для обучения использован простой градиентный метод, lr = 0.001

**Указание 2**: для расчета среднего значения и с.к.о. ошибки обученной сети 10-кратное обучение из различных начальных точек.

*е) Выводы*

При инициализации со слишком большими или слишком маленькими значениями σ процесс обучения протекает сложно – ошибка долгое время не может стабилизироваться, варьируя в большом промежутке. При инициализации весов с параметром σ по методу Хавьера получены наиболее приемлемые результаты.

В пределах каждого распределения с увеличением значения σ улучшалось и качество обучения (снижадись значения ошибок). Наилучшие результаты были получены у нормального распределения с параметром σ по методу Хавьера.

*4.13. Методы кросс-валидации*

*а) Заполнить таблицу по результатам кросс-валидации различными методами*

|  |  |  |  |
| --- | --- | --- | --- |
| Метод кросс-валидации | Число запусков обучения | Средняя ошибка на обучающей выборке ± с.к.о., | Средняя ошибка на тестовой выборке ± с.к.о., |
| Монте-Карло | 10 | 3,42\*10-5 ± 1,31\*10-5 | 8,06\*10-5 ± 3,06\*10-5 |
| Holdout 60/30/10 | 1 | 0,0291 ± 0,0302 | 0,0337 ± 0,0348 |
| 10-fold | 10 | 0,0172 ± 0,0346 | 0,0173 ± 0,0343 |
| LOOCV | 10 |  |  |

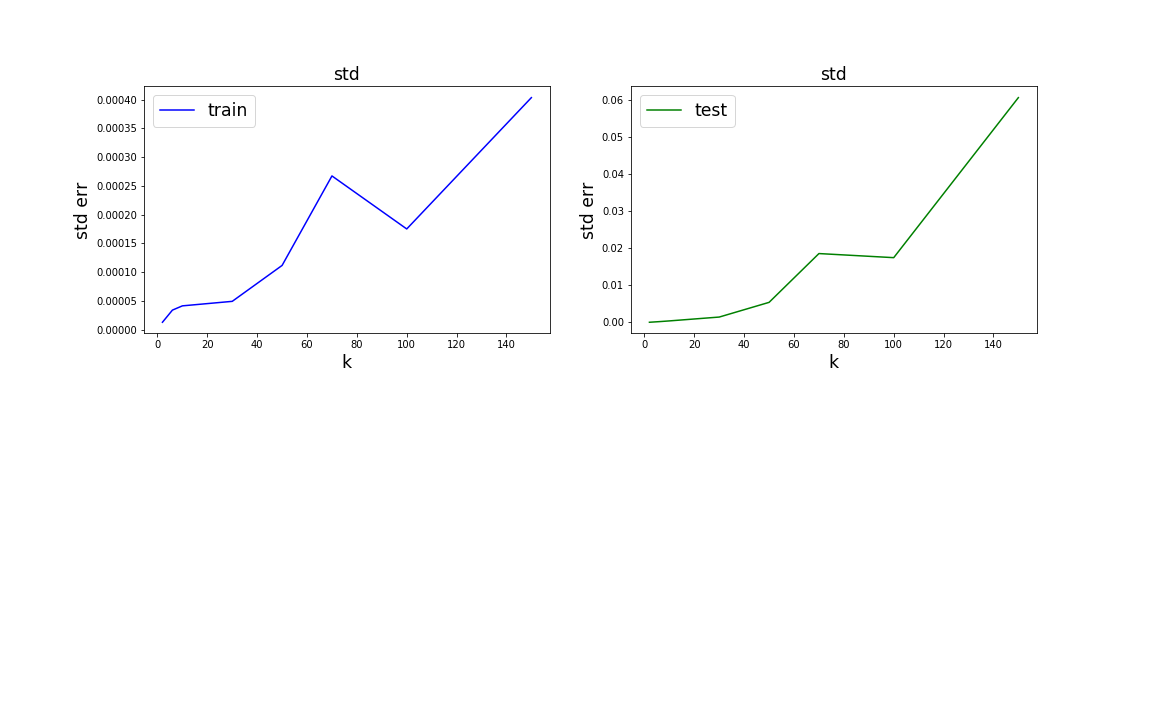
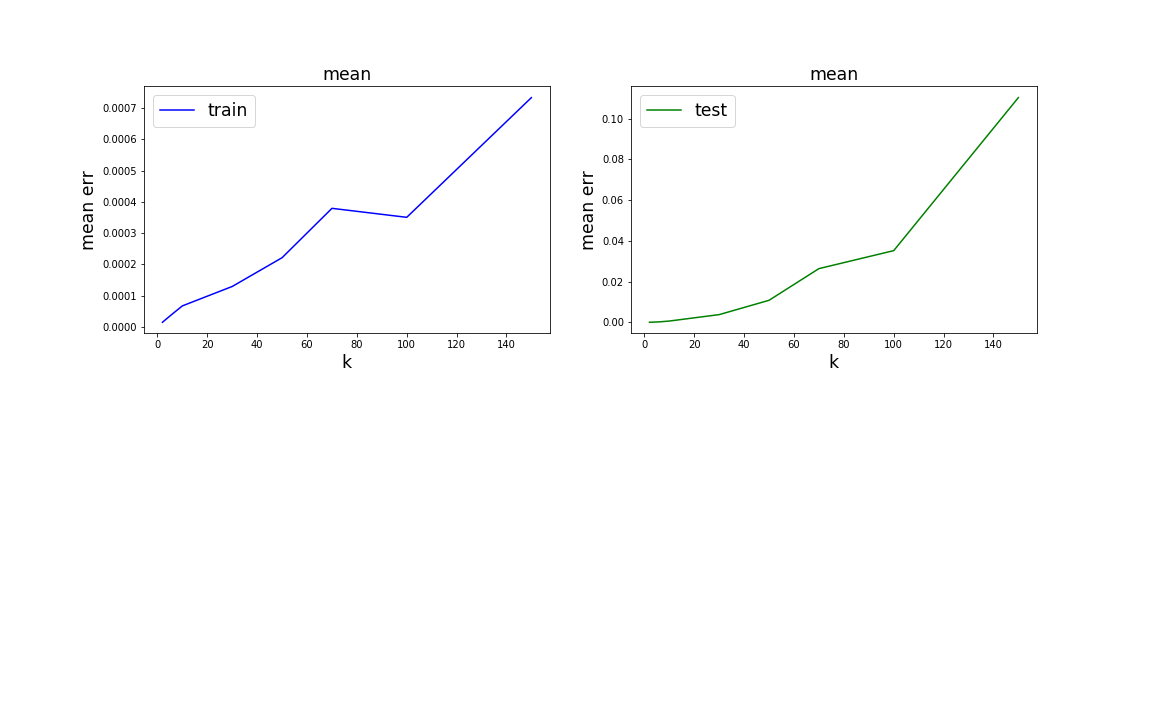
**Указание 1**: для обучения использован простой градиентный метод, lr = 0.001.

**Указание 2**: инициализация сети проведена один раз по закону *N*(0; σ), σ рассчитана по методу Хавьера. Запуски процедуры обучения каждый раз были из одной и той же начальной точки.

**Указание 3**: Внутри каждого фолда кросс-валидации обучающие данные на обучающую и валидационную выборки разбивались в отношении 70/30.

*б) Исследование k-fold кросс-валидации*

Графики зависимости среднего значения и с.к.о. ошибки обученной сети на обучающей и тестовой выборках от числа фолдов *k*:



*в) Выводы*

Используя более статичные методы, качество обучения относительно плохое. Чем больше вносится элемент случайности в выборки данных (при размерах выборок >> 1), тем качественнее будет обучение. Например, с методом Монте-Карло работает лучше остальных из-за элемента случайности элементов в обучающей и тестовой выборке каждую эпоху.

*4.14. Исследование различных архитектур нейронных сетей*

*а) Исследование зависимости качества обучения от числа нейронов в скрытых слоях*

Провести обучение нейронных сетей с различным числом нейронов в скрытых слоях. По результатам обучения заполнить таблицу.

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Число нейронов в скрытых слоях | Средняя ошибка на обучающей выборке ± с.к.о., | Средняя ошибка на тестовой выборке ± с.к.о., |
| 1 | 20; 10; 5 | 0,8683\*10-5 ± 1.0596\*10-5 | 7,3096\*10-5 ± 6.6902\*10-5 |
| 2 | 10; 30; 5 | 0,8533\*10-5 ± 1.0184\*10-5 | 6,5086\*10-5 ± 4.9673\*10-5 |
| 3 | 10; 20; 5 | 0,5569 \*10-5 ± 0,4106 \*10-5 | 5,3982\*10-5 ± 2,6042\*10-5 |
| 4 | 10; 5; 5 | 0,5556\*10-5 ± 0,4035\*10-5 | 5.1404\*10-5 ± 2.0873\*10-5 |
| 5 | 10; 10; 3 | 0,5143\*10-5 ± 0,3242\*10-5 | 4,7476\*10-5 ± 1.3086\*10-5 |
| 6 | 10; 7; 3 | 0,4706\*10-5 ± 0,2357\*10-5 | 4,6635\*10-5 ± 1.127\*10-5 |

**Указание 1**: обучения использован метод BFGS.

**Указание 2**: среднее значение и с.к.о. ошибок на обучающей и тестовой выборках считалось по результатам 10-fold кросс-валидации, внутри каждого фолда обучающие данные разбиты на обучающую и валидационную выборки в отношении 70/30.

*б) Исследование зависимости качества обучения от активационных характеристик нейронов*

Для архитектуры 10-7-3 (из п. а), провести обучение сети при различных активационных характеристиках нейронов скрытых слоёв. Результаты:

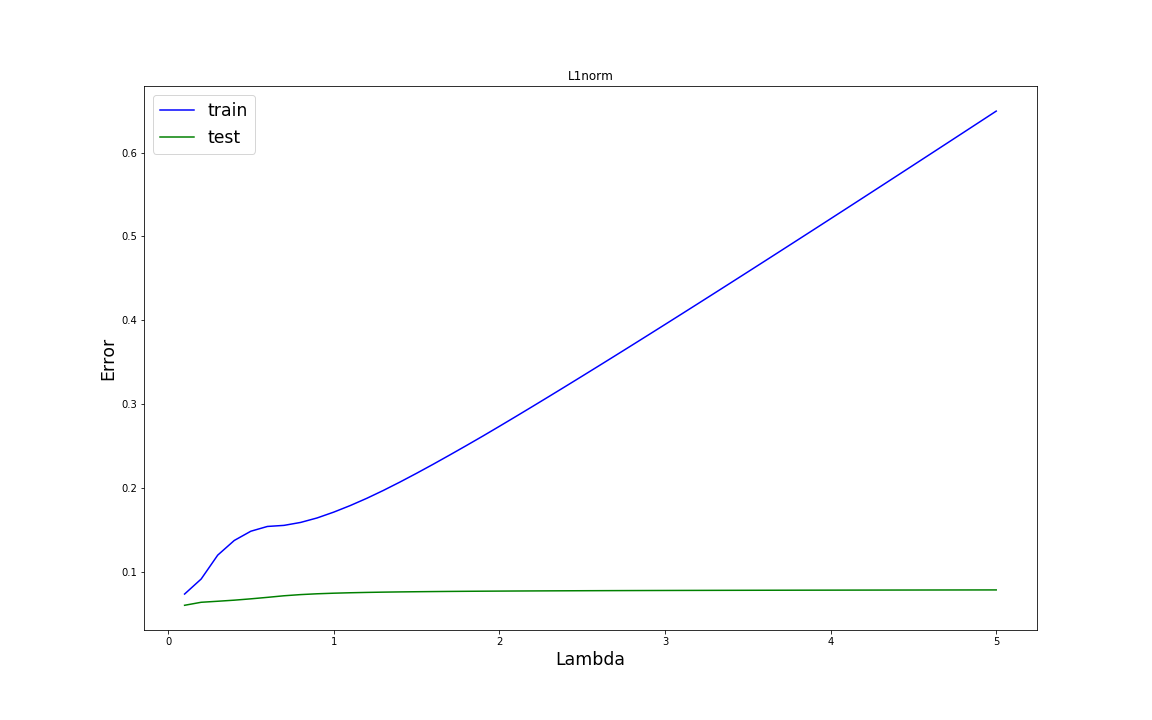
|  |  |  |
| --- | --- | --- |
| АХ нейронов  скрытых слоёв | Средняя ошибка на обучающей выборке ± с.к.о., | Средняя ошибка на тестовой выборке ± с.к.о., |
| logistic | 0,554\*10-5 ± 0,3662\*10-5 | 5,2684\*10-5 ± 2,1182\*10-5 |
| tanh | 0,4706\*10-5 ± 0,2357\*10-5 | 4,6635\*10-5 ± 1.127\*10-5 |
| linear | 0,7344\*10-5 ± 0,2745\*10-5 | 6,5596\*10-5 ± 0,5981\*10-5 |
| softsign |  |  |
| softplus | 0,415\*10-5 ± 0,0101\*10-5 | 43903\*10-5 ± 0,4446\*10-5 |

*4.15. L1 и L2 регуляризация весов*

*а) Исследование зависимости качества обученной сети от параметра регуляризации*

График зависимости ошибки обученной сети на обучающей и тестовой выборках от значения параметра λ:

1) при *L*1-регуляризации весов 2) при *L*2-регуляризации весов

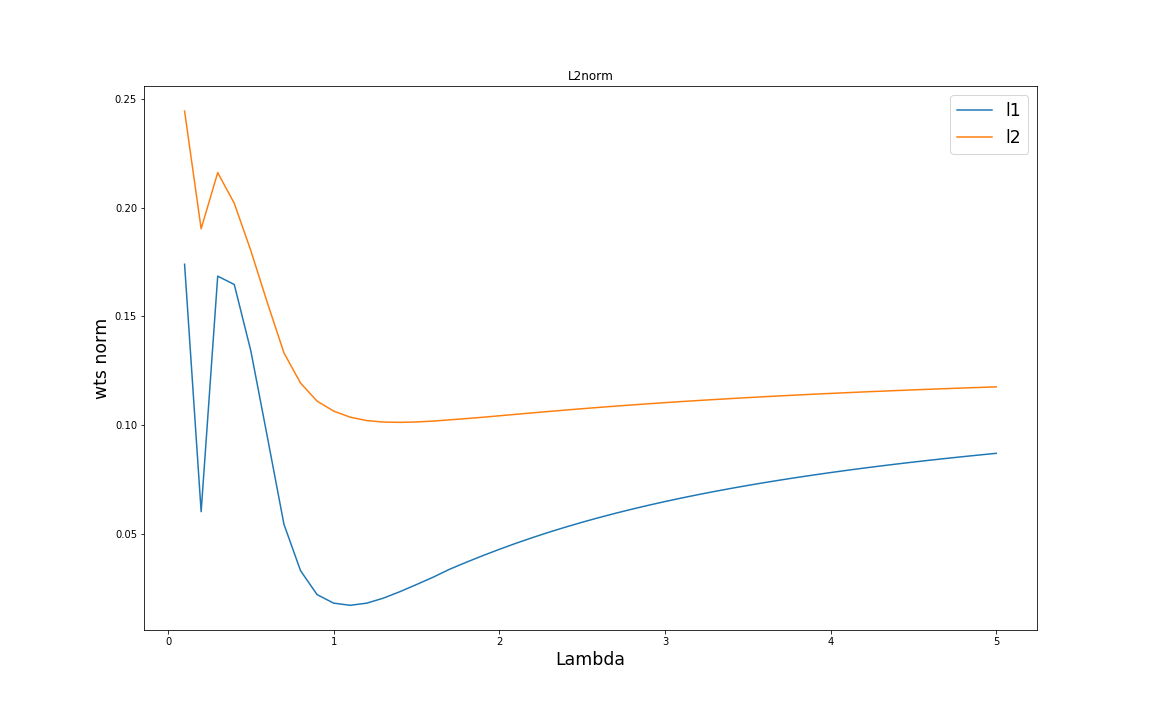
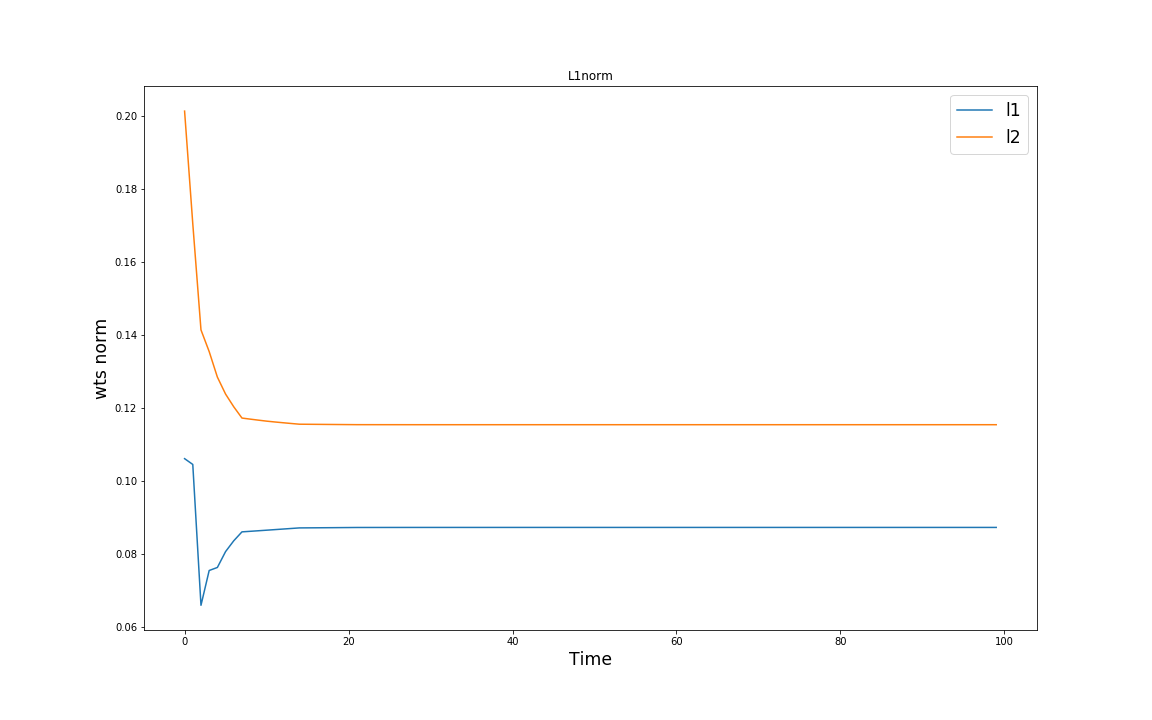


**Указание**: исследования в пп. а)–в) проводятся для архитектуры 10-7-3. Для обучения использовать метод RMSprop с параметрами: lr0 =0.001, ρ = 0.99.

*б) Исследование зависимости нормы вектора синаптических коэффициентов от параметра регуляризации*

Графики зависимости *L*1 и *L*2 нормы вектора синаптических коэффициентов обученной сети от параметра λ:

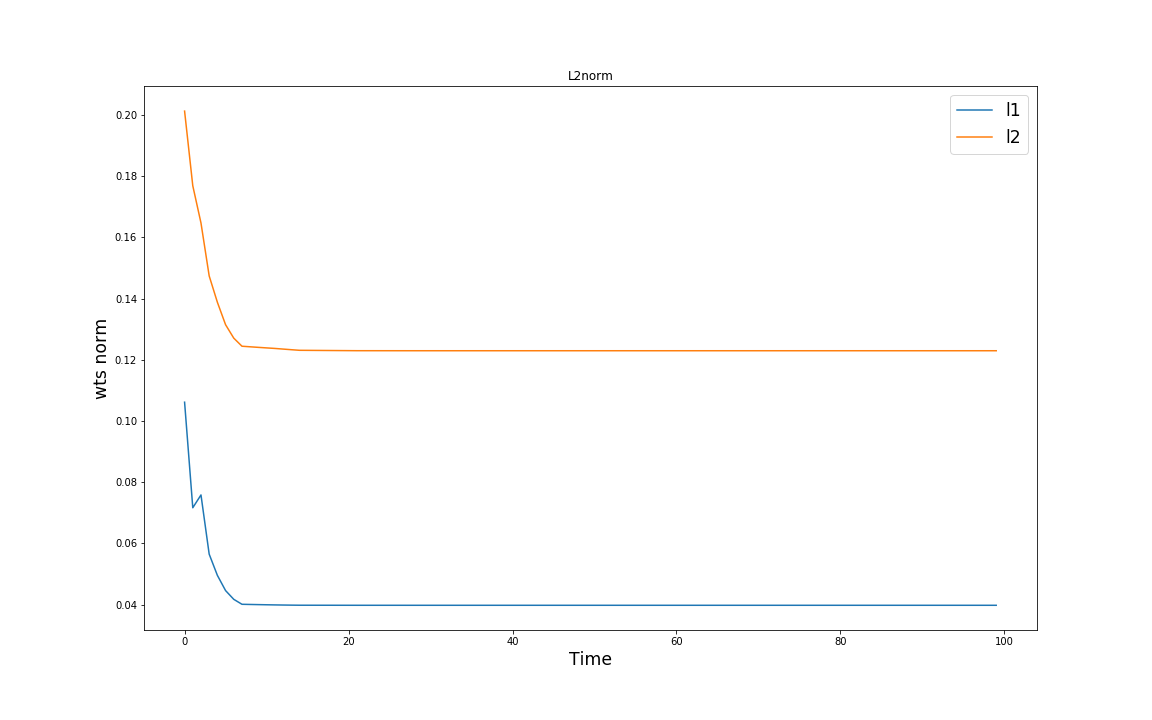
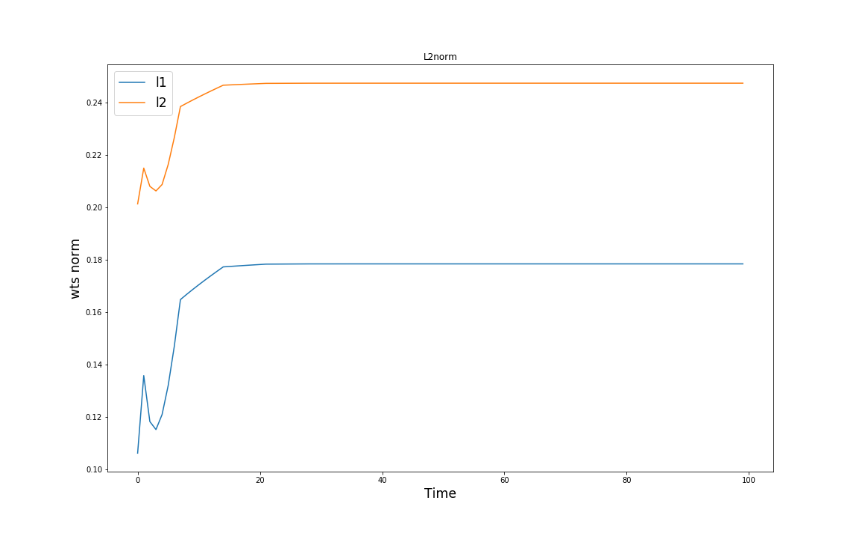
1) при *L*1‑регуляризации 2) при *L*2-регуляризации:



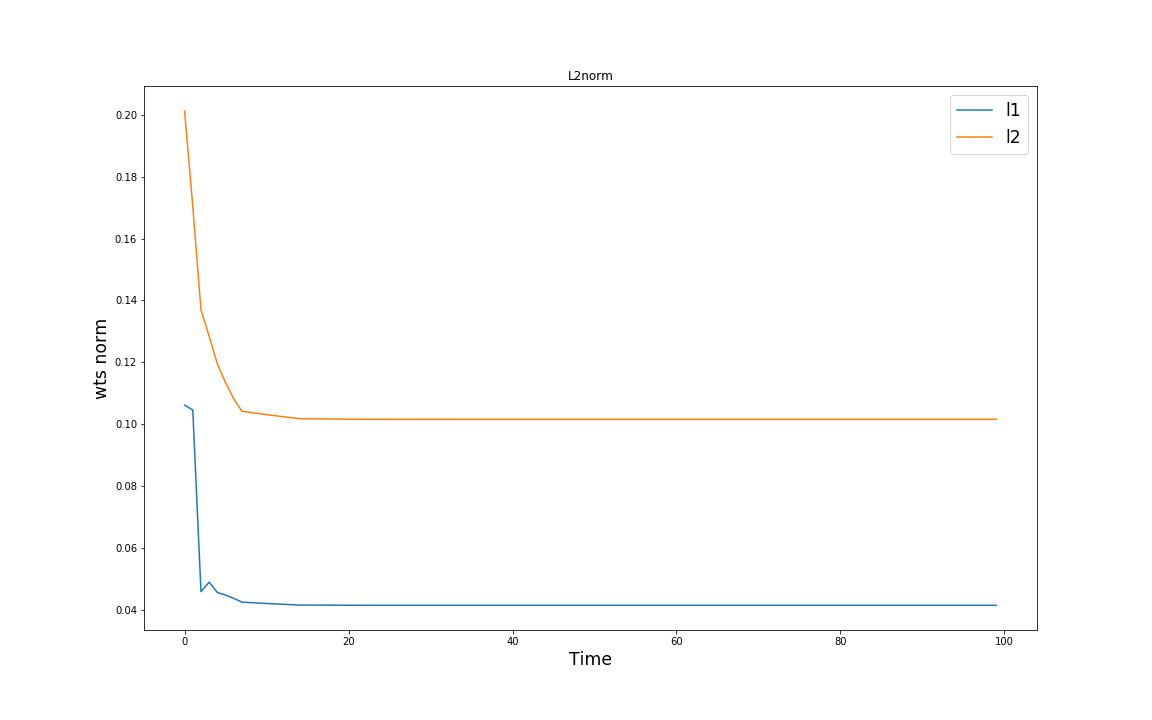
*в) Сравнение кривых обучения при различных значениях параметра регуляризации*

При фиксированных значениях параметра λ (λ ≈ 0, λ > 0, λ >> 0) построить графики зависимости *L*1 и *L*2 нормы вектора синаптических коэффициентов сети от времени обучения.

λ ≈ 0: λ > 0:



λ >> 0:



*г) Выводы*

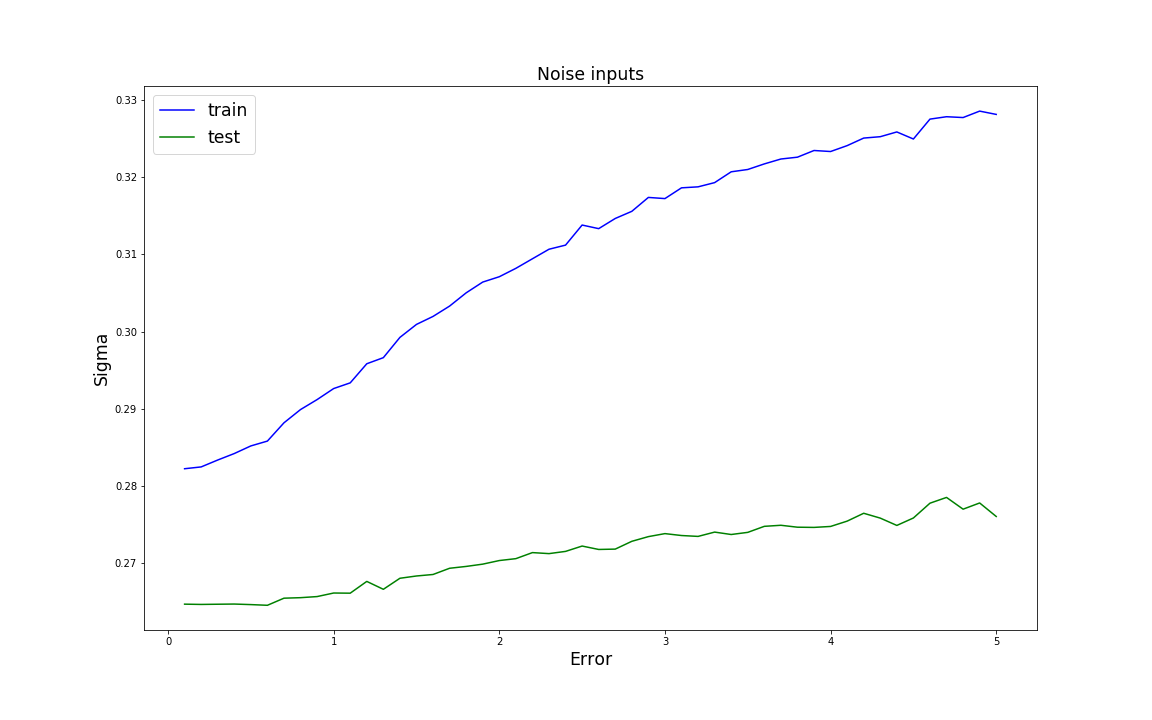
При увеличении λ  ухудшается качество обучения сети. Стоит отметить, что нормы весов стабилизируются. Причем при увеличении λ нормы синаптических коэффициентов уменьшаются.

*4.16. Инъекция шума*

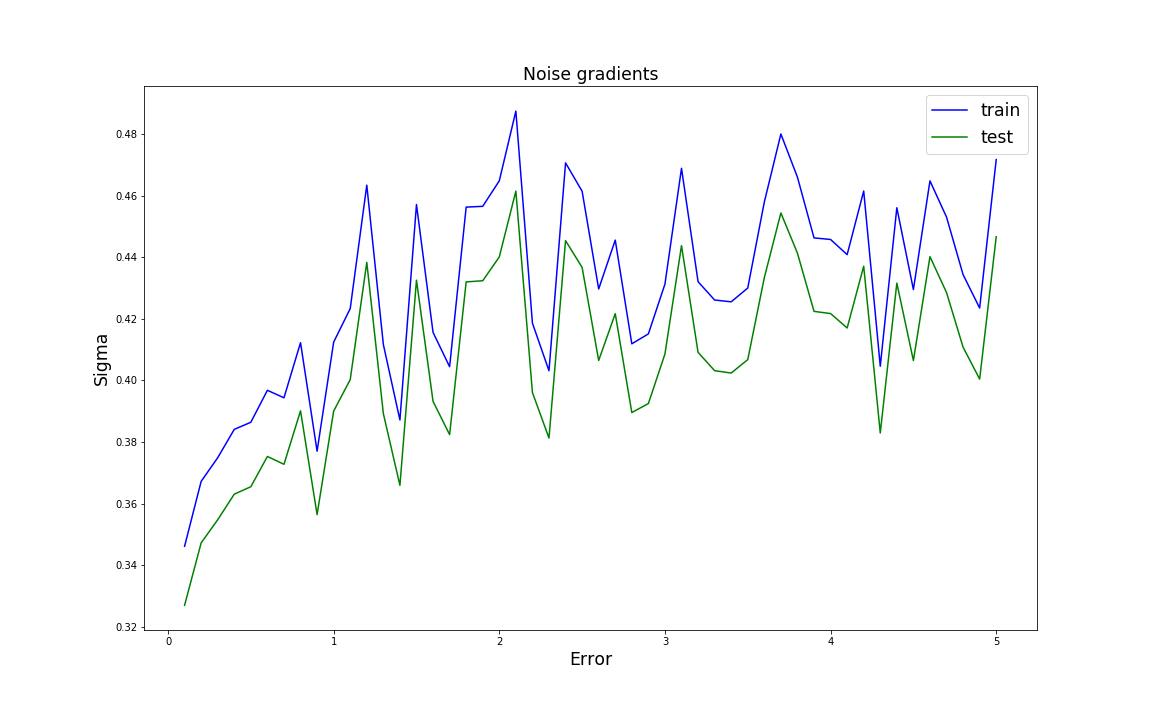
*а) Исследование зависимости качества обученной сети от параметра зашумления*

Графики зависимости ошибки обученной сети на обучающей и тестовой выборках от с.к.о. σ шума при:

1) зашумлении входов



2) зашумлении градиентов



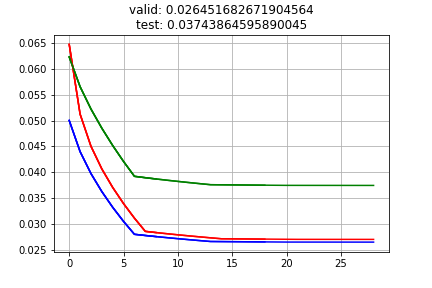
**Указание 1**: шум генерировать из нормального распределения *N*(0; σ).

**Указание 2**: исследования проводились для архитектуры 10-7-3. Для обучения использовать метод RMSProp с параметрами: lr0 =0.001; ρ = 0.99.

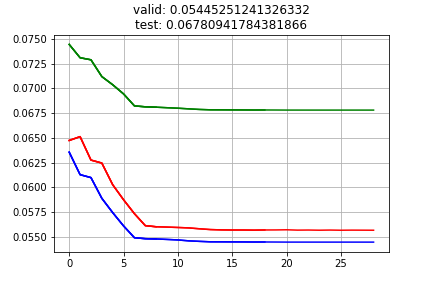
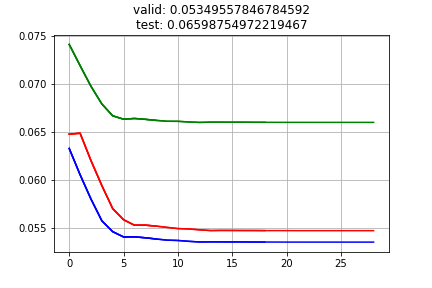
*б) Сравнение кривых обучения*

Кривые обучения на обучающей, валидационной и тестовой выборках при обучении:

1) без зашумления 2) с зашумлением входов



3) с зашумлениемградиентов 4) с зашумлениемвходов иградиентов



**Указание**: шум генерировать из нормального распределения *N*(0; σ), с.к.о. шума σ=0.1.

*в) Выводы*

При увеличении параметра σ в процессе зашумления качество обучения сети ухудшается – растет ошибка. Наилучшим значения является значение σ=0.1.

На незашумленных данных и при незашумленных градиентах были показаны наилучшие результаты. Зашумление данных на качество не повлияло, однако зашумление градиентов его ухудшило.

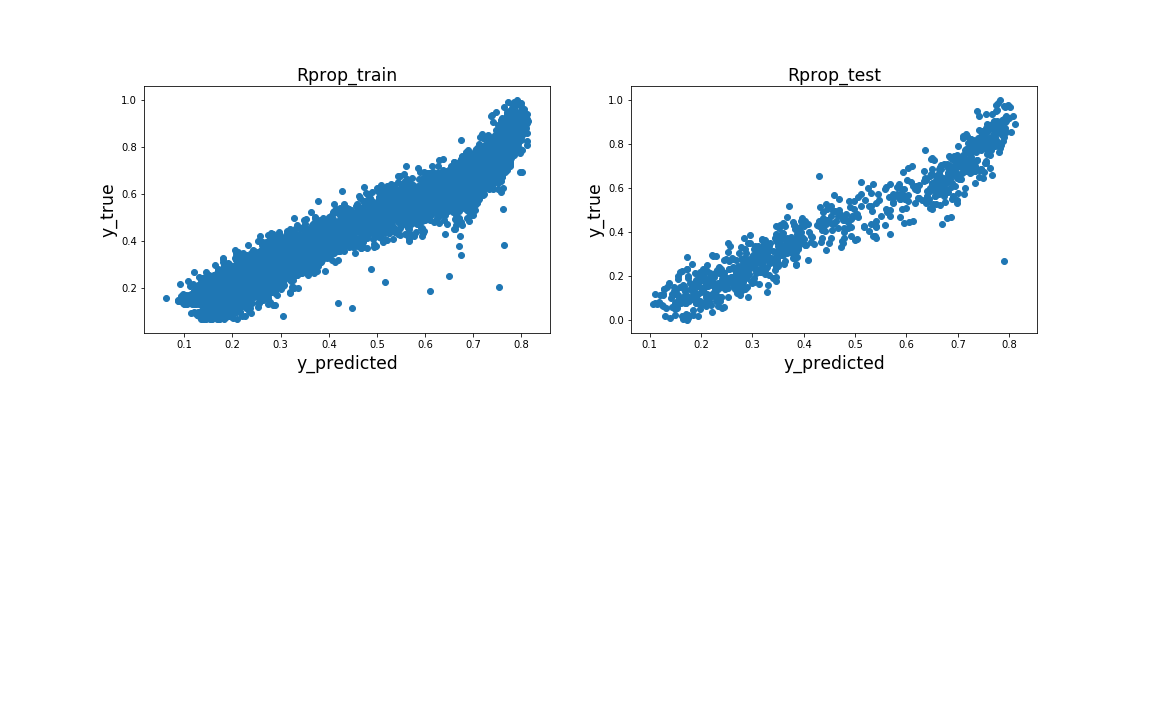
V. Исследование обученной нейросетевой модели

**Указание:** модель с наименьшими значениями ошибок на обучающей и валидационной выборках: RProp с параметрами: η- = 1.2; η+ = 0.5.

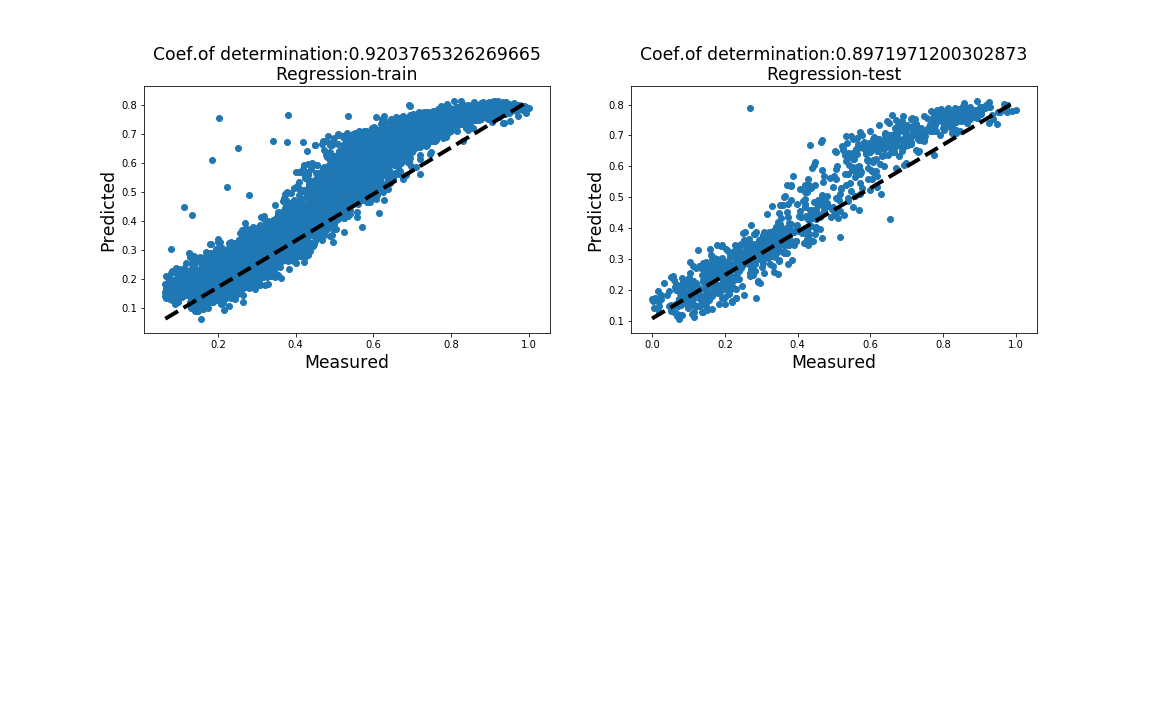
*5.1.Исследование качества обученной модели*

**В процессе работы решалась задача регрессии.**

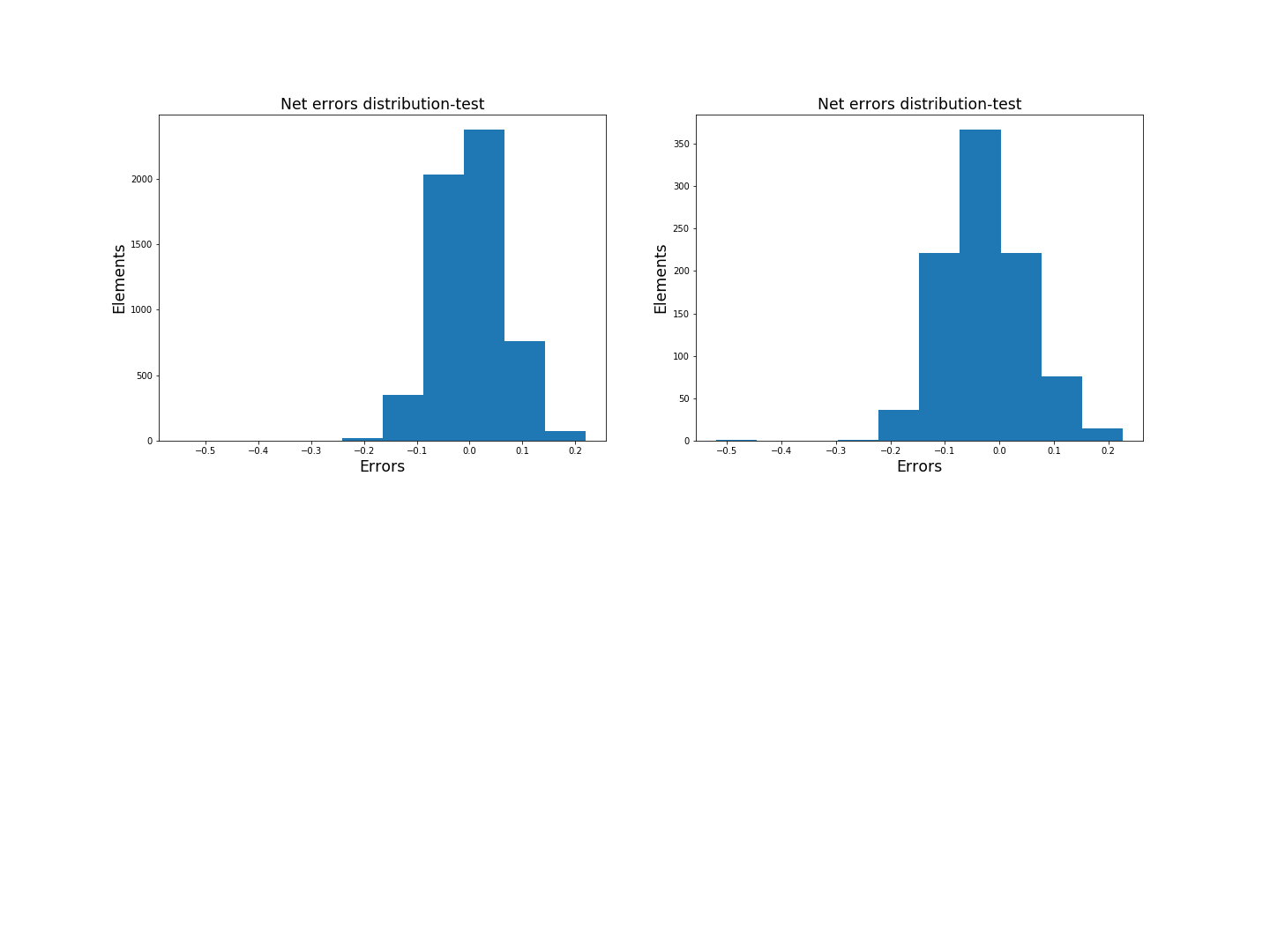
Диаграммы рассеяния в пространстве «выход модели – желаемый выход» для обучающей и тестовой выборок:



Линейные регрессии выхода модели на желаемый выход и коэффициенты детерминации линейных регрессионных моделей для обучающей и тестовой выборок:

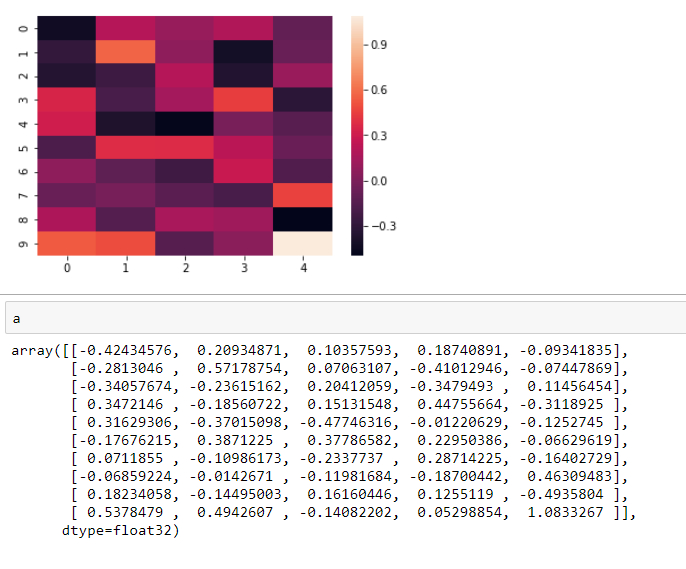
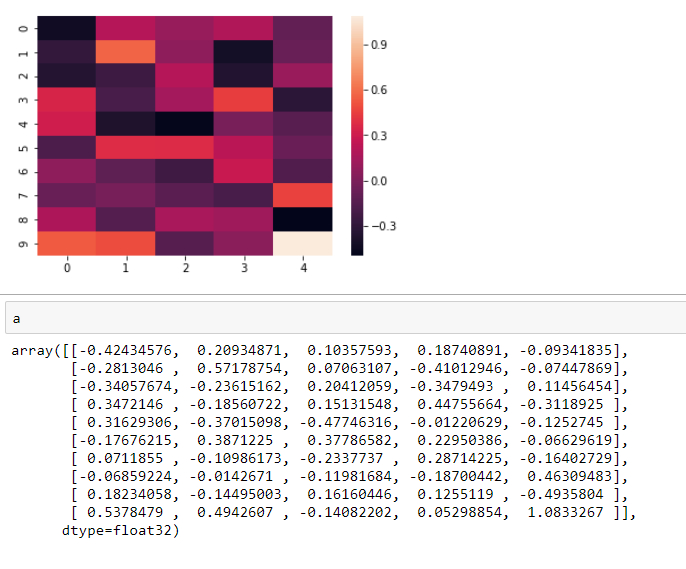


Гистограммы распределения ошибок сети на примерах обучающей и тестовой выборок:

**

*5.2. Оценка важности признаков*

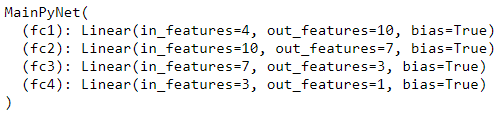
Матрица синаптических коэффициентов 1-го слоя обученной сети (heatmap и сама матрица):



Из диаграммы видно, что все признаки имеют важность для решения поставленной задачи. Признаки 0 и 1 имеют большую важность, чем 2 и 3, т.к. в них присутствует больше значений, сильно отличающихся от 0.

*5.3. Выводы*

Была построена многослойная нейросетевая модель с архитектурой:



В качестве оптимизатора использовался RProp с параметрами: η- = 1.2; η+ = 0.5.

Построенная нейросетевая модель предсказывает значения выходов качественно, с минимальной ошибкой, даже при добавлении шума во входные данные.