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 Б15-503**

**Ильичева Алексея Михайловича**

#### Вариант № 3

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

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

2018 г.

ОТЧЕТ № 1

по теме «Многослойные нейронные сети»

Вариант № 3

ФИО студента: Ильичев Алексей Михайлович Группа: Б15-503

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 | a = 1.0 | 10000 | 0.006 | 0.1584 |
| GDM | m = 0.0 | 2500 | 0.01 | 0.2671 |
| NAG | m = 0.0 | 2500 | 0.01 | 0.2634 |
| SGD |  | 5000 | 0.02 | 0.0815 |
| Fletcher-Reeeves |  | 450 | 0.00014 | 0.1094 |
| Polak-Ribiere |  | 350 | 0.00015 | 0.1333 |
| AdaGrad |  | 1000 | 0.013 | 0.2712 |
| RMSProp | p = 0.9 | 3500 | 0.3998 | 0.89 |
| AdaDelta | p = 0.99 | 3500 | 0.03727 | 0.17559 |
| RProp |  | 100 | 0.028 | 0.4166 |
| LM |  | 21 | 0.2469 | 0.2373 |
| BFGS |  | 100 | 0.074 | 0.2614 |
| Stochastic GD |  | 2270 | 0.3277 | 0.6977 |
| Adam | b1, b2 = (0.5 0.9) | 5000 | 0.079 | 0.922 |

Выводы:

|  |
| --- |
| Обычный градиентный метод показал хороший результат, но для его реализации было потрачено наибольшее количество эпох среди всех методов. Методы RMSProp и Adam показали большую ошибку на тестовой выборке, возможно, из-за большой скорости обучения равной 1.0 (следовал указанию в методе). Ошибка на тестовой выборке, меньшая чем у метода GD, у методов SGD, Fletcher-Reeeves, Polak-Ribiere, AdaDelta. Наилучший результат показал метод SGD. |

ОТЧЕТ № 2

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

Вариант № 3

ФИО студента: Ильичев Алексей Михайлович Группа: Б15-503

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

Показатели качества обученной нейросетевой модели:

|  |
| --- |
|  |

Выводы:

|  |
| --- |
| Обученная по методу SGD нейросеть, действительно выполняет классификацию объектов. 4 класса из 7 распознаны со 100% результатом в обучающей и в тестовой выборке. В итоге, общая точность на обучающей выборке составила 99.41%, на тестовой – 94.97%. Разбиение элементов по классам происходило случайно (разное количество элементов в классах, но достаточное для обучения) . |

I. Исходные данные

*1.1.Описание исходных данных*

Исходные данные получены из источника <https://archive.ics.uci.edu/ml/datasets/Image+Segmentation>.

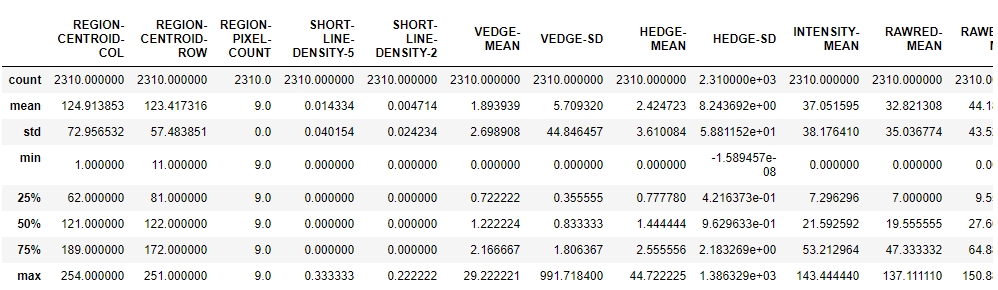
Были взяты экземпляры 7 изображений, которые были сегментированы вручную. Каждый сегмент представляет из себя область размером 3х3 пикселя. В итоге, по каждому сегменту были произведены различные вычисления, результаты которых являются признаками исходных данных. Число признаков – 19. Описание признаков:

1. REGION-CENTROID-COL. Столбец центрального пикселя области (сегмента) (целочисленное значение)
2. REGION-CENTROID-ROW. Строка центрального пикселя области (сегмента) (целочисленное значение)
3. REGION-PIXEL-COUNT. Количество пикселей в области (3х3 = 9 одинаково во всех примерах) (целочисленное значение)
4. SHORT-LINE-DENSITY-5. Результаты алгоритма экстракции линий, который подсчитывает, сколько строк длины 5 (любая ориентация) с низким контрастом, меньше или равным 5, проходит через область (вещественное значение)
5. SHORT-LINE-DENSITY-2. То же, что и п.4, но подсчитывает линии с высокой контрастностью, более 5 (вещественное значение)
6. VEDGE-MEAN. Величина среднего контраста горизонтально примыкающих пикселей в области (вещественное значение)
7. VEDGE-SD. Величина стандартного отклонения контраста горизонтально примыкающих пикселей в области (вещественное значение)
8. HEDGE-MEAN. Величина среднего контраста вертикально примыкающих пикселей в области (вещественное значение)
9. HEDGE-SD. Величина стандартного отклонения контраста вертикально примыкающих пикселей в области (вещественное значение)
10. INTENSITY-MEAN. Средняя интенсивность (вещественное значение)
11. RAWRED-MEAN. Среднее значение красного компонента цвета в области (вещественное значение)
12. RAWBLUE-MEAN. Среднее значение синего компонента цвета в области (вещественное значение)
13. RAWGREEN-MEAN. Среднее значение зеленого компонента цвета в области (вещественное значение)
14. EXRED-MEAN. Средняя мера избытка красного компонента цвета в области (вещественное значение)
15. EXBLUE-MEAN. Средняя мера избытка синего компонента цвета в области (вещественное значение)
16. EXGREEN-MEAN. Средняя мера избытка зеленого компонента цвета в области (вещественное значение)
17. VALUE-MEAN. Среднее значение 3-d нелинейной трансформации RGB (вещественное значение)
18. SATURATION-MEAN. Среднее насыщение 3-d нелинейной трансформации RGB (вещественное значение)
19. HUE-MEAN. Средний оттенок 3-d нелинейной трансформации RGB (вещественное значение)

В принципе, все признаки могут быть рассмотрены как вещественные значения.

Выборка разделена на обучающую и тестовую: 210 примеров для обучающей выборки и 2100 примеров для тестовой. Всего 2310 элементов. Рассматривается задача классификации. Всего 7 классов изображений: кирпич (лицевая сторона), небо, зелень, цемент, окно, тропа, трава. Каждый класс имеет 30 обучающих примеров и 300 тестовых. Требуется составить и обучить нейросеть, которая бы могла определять класс изображения по примеру данных, описанных выше. Вход сети – пример данных (19 признаков). Выход – класс, принадлежность к которому установила сеть.

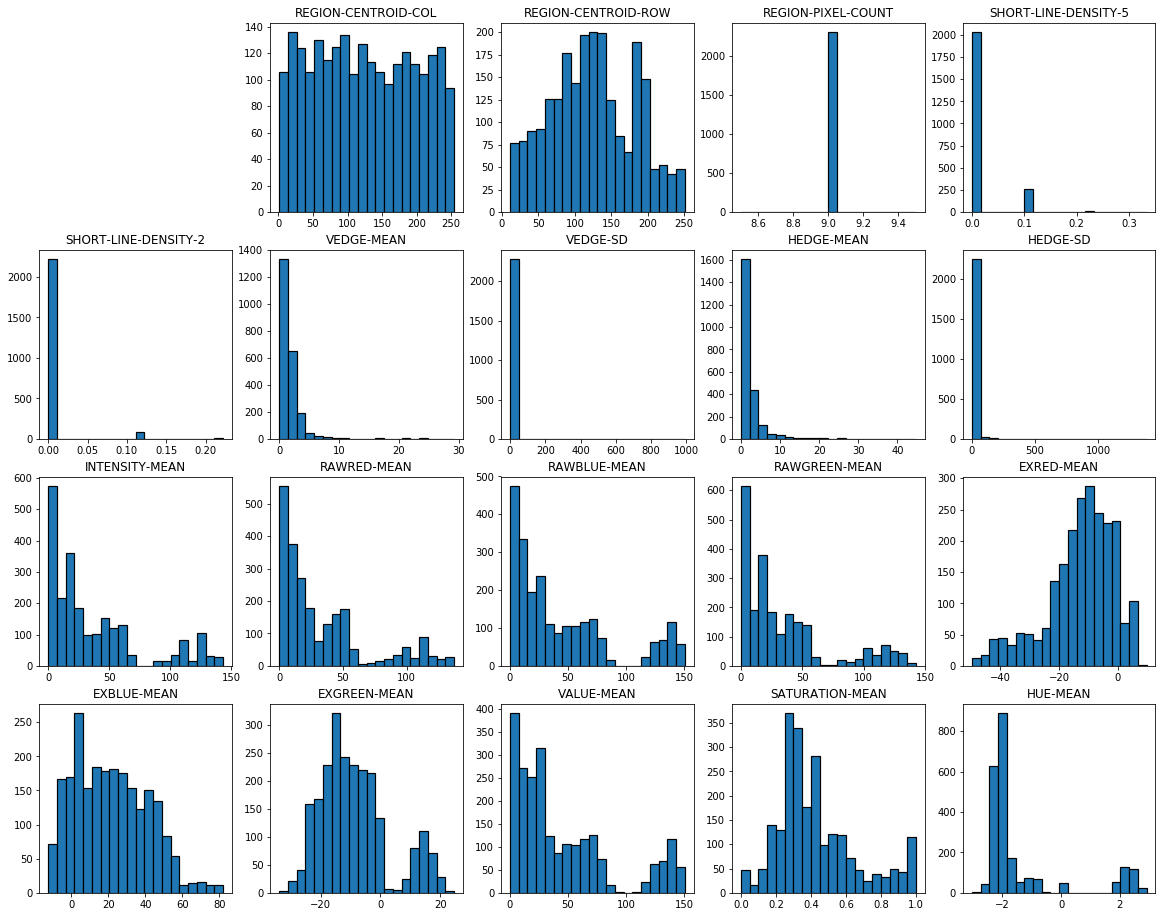
Табл.1.1 – Описание исходных данных с помощью библиотеки pandas



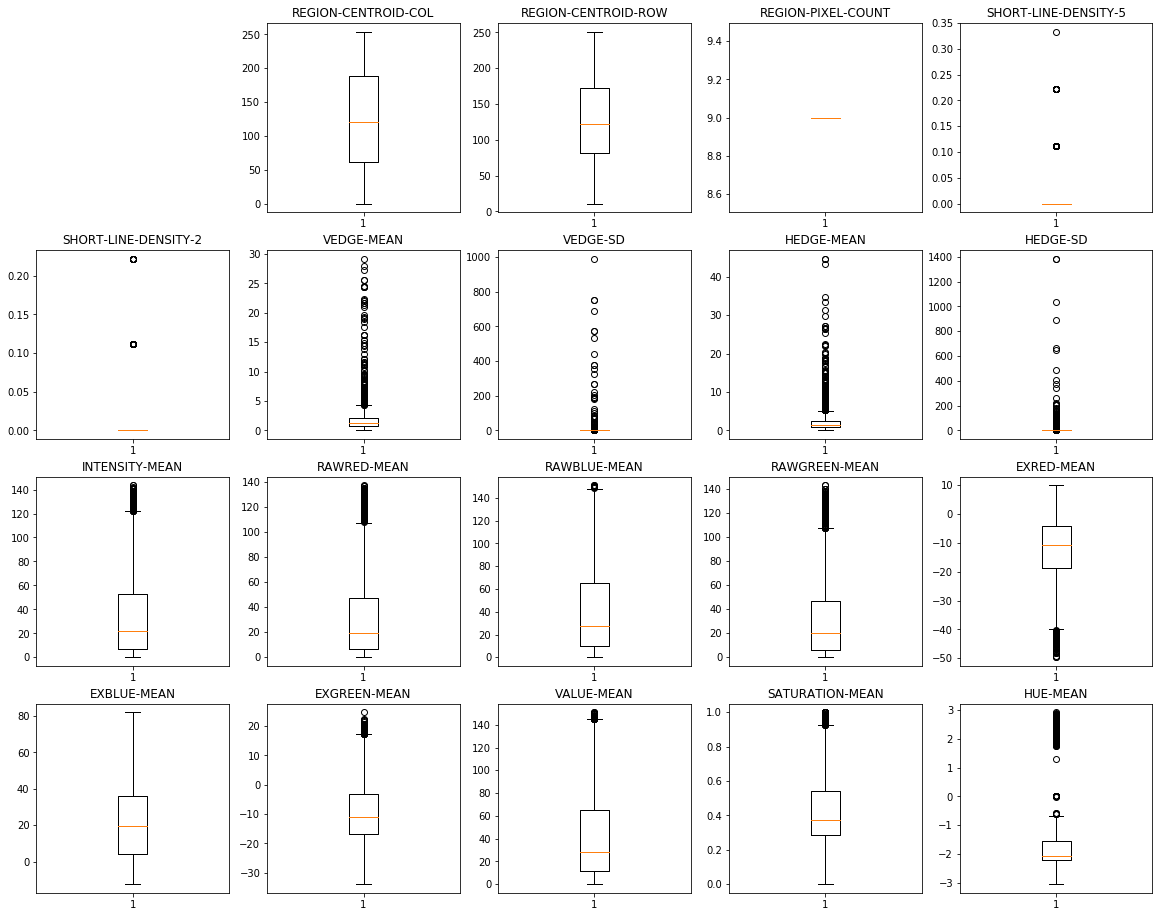
*1.2.Визуальный анализ исходных данных*

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

Гистограммы распределения и диаграммы Box-and-Whisker для всех признаков представлены на рисунке 1.1 и рисунке 1.2 соответственно. На рисунках можно увидеть, что сегменты изображения выбирались равномерно по всему изображению, чему свидетельствуют распределения признаков REGION-CENTROID-COL и REGION-CENTROID-ROW. Признак REGION-PIXEL-COUNT действительно равен 9 для всех примеров, следовательно этот признак абсолютно не информативен. Также не информативны признаки SHORT-LINE-DENSITY-5, SHORT-LINE-DENSITY-2, VEDGE-SD, HEDGE-SD, так как большинство данных равно единственному значению, а остальные данные лишь представляют собой выбросы. Распределение ни одного из признаков не напоминает нормальное распределение.



*Рис. 1.1 – Гистограммы распределения признаков*



*Рис. 1.2 – Диаграммы Box-and-Whisker признаков*

*б) Корреляционная матрица признаков*

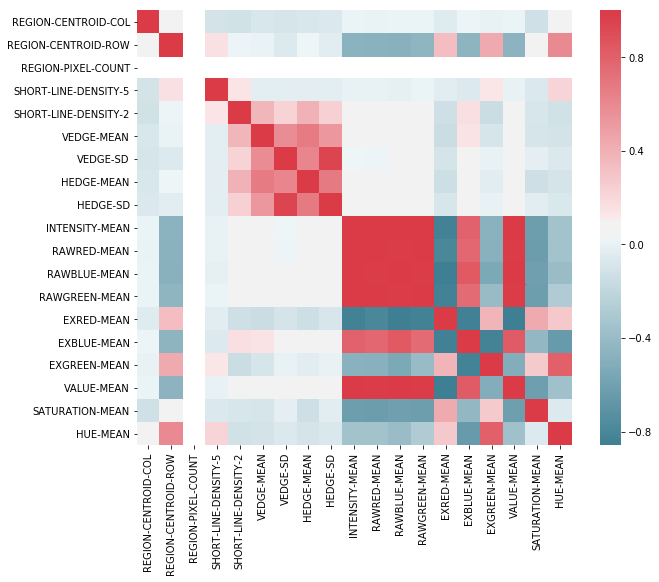


Рис. 1.3 – Корреляционная матрица признаков

Выводы: глядя на полученную корреляционную матрицу, можно сказать, что признаки INTENSITY-MEAN, RAWRED-MEAN, RAWBLUE-MEAN, RAWGREEN-MEAN, VALUE-MEAN очень сильно скоррелированы между собой, коэффициент корреляции равен почти 1. Следовательно, можно заменить эти признаки меньшим количеством признаков, выделив главные компоненты.

*в) Диаграммы рассеяния*

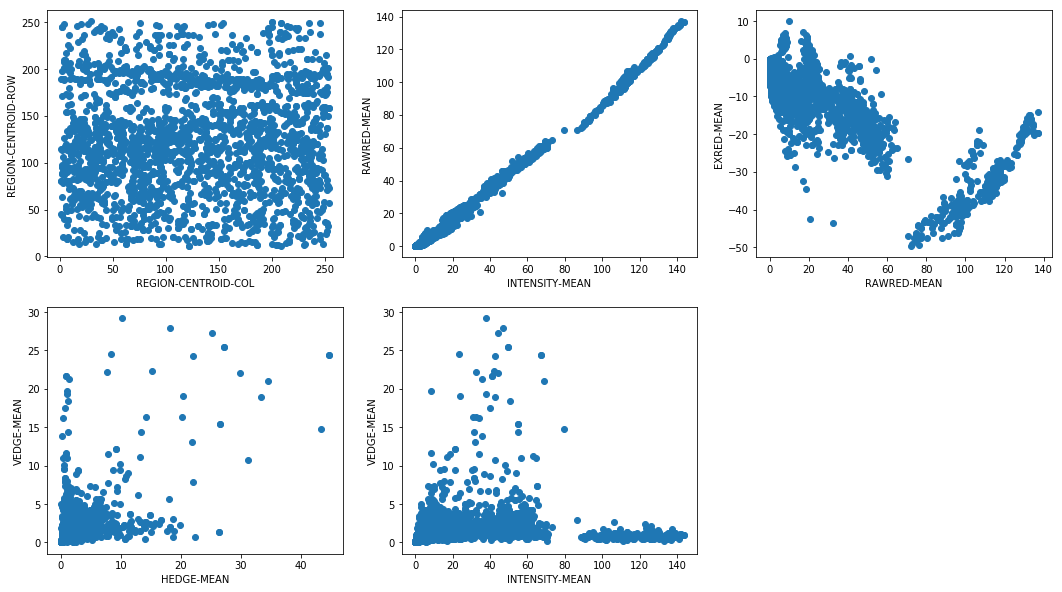


Рис. 1.4 – *Диаграммы рассеяния отдельных признаков*

Выводы: на диаграмме рассеяния признаков REGION-CENTROID-ROW и REGION-CENTROID-COL видно, что сегменты выбирались равномерно по всему изображению. На следующей диаграмме показана сильная скоррелированность признаков INTENSITY-MEAN и RAWRED-MEAN. Признаки RAWRED-MEAN и EXRED-MEAN имеют отрицательный коэффициент корреляции, хотя распределены более-менее равномерно.

*1.3.Выводы*

Исходные данные были детально описаны, построены все необходимые диаграммы.

Были рассмотрены распределения признаков, их выбросы, взаимная корреляция. По полученным свойствам сделаны выводы в каждом подпункте.

II. Предобработка данных

*2.1. Очистка данных*

*а) Обнаружение и устранение дубликатов*

Был использован метод из библиотеки Pandas: drop\_duplicates(keep='first'). Были удалены все дубликаты за исключением первого вхождения. Результат: было удалено 14 записей. Осталось 2296 элементов.

*б) Обнаружение и устранение выбросов*

Для устранения выбросов были выбраны квантили q1i, q2i на уровнях 0.1 и 0.99 соответственно для всех признаков. После чего данные были отсеяны, остались только данные, которые находятся в областях q1i < x < q2i для всех признаков i. Тогда выбросы находятся в областях q1i > x и x > q2i.

Для поиска квантилей были применены методы из Pandas.

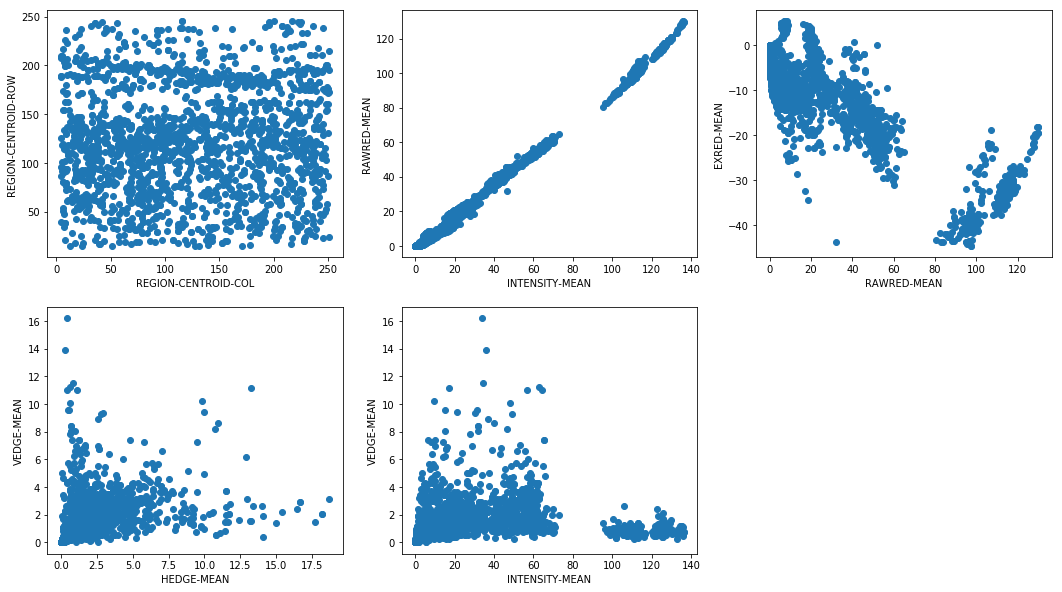
Результат: осталось 1988 элементов.

*в) Пропущенные значения*

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

Результат: число элементов не изменилось: 1988.

*г) Визуальный анализ очищенных данных*



*Рис. 2.1 – Диаграммы рассеяния отдельных признаков очищенных данных*

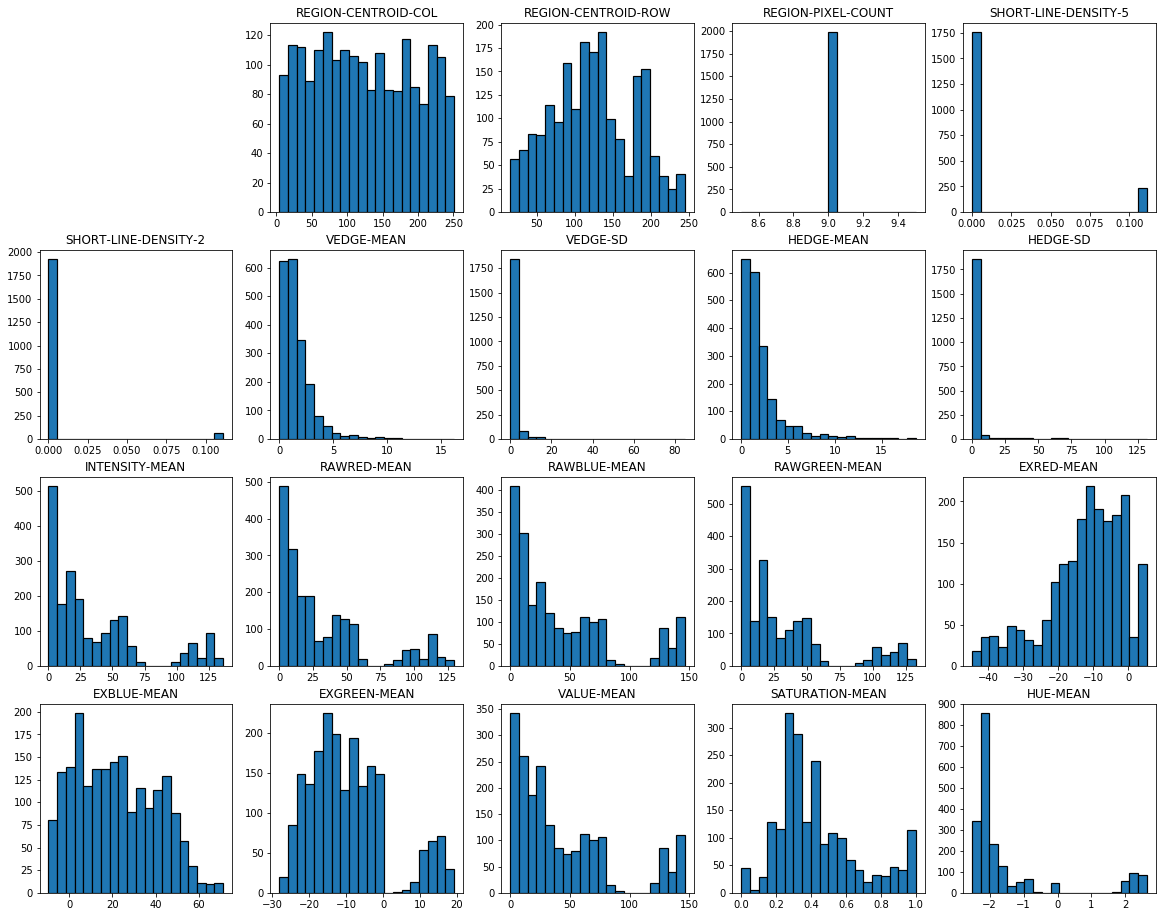


Рис. 2.2 – Гистограммы распределения очищенных данных

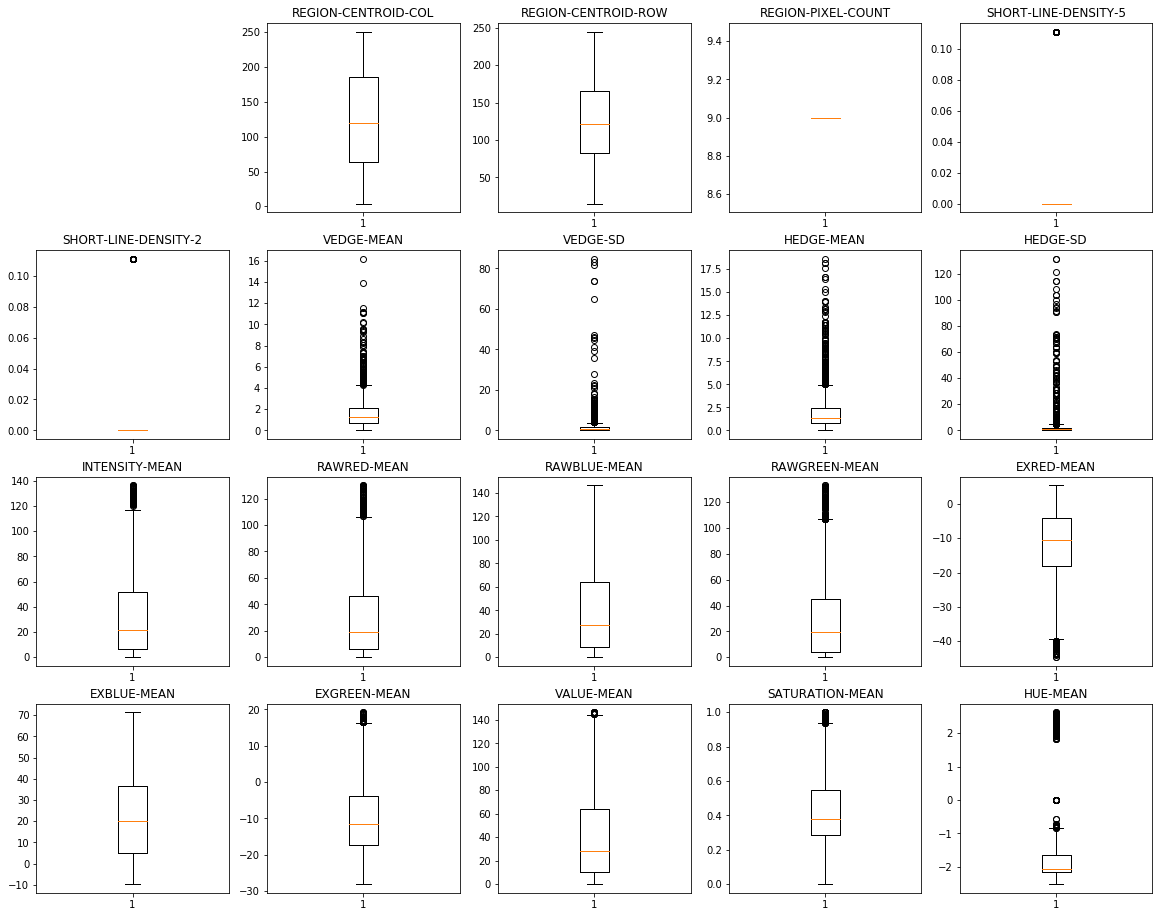
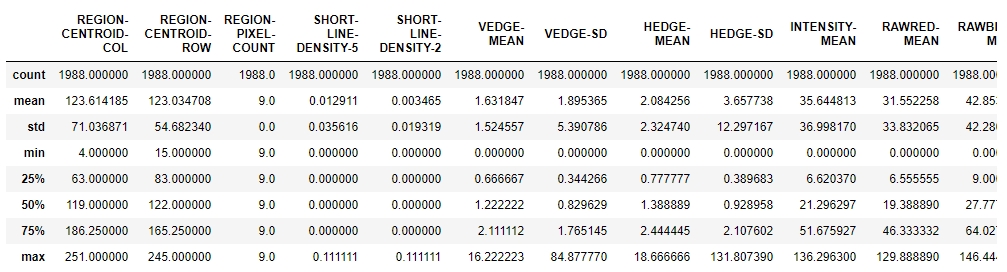


Рис. 2.3 – Диаграммы *Box-and-Whisker очищенных данных*

Табл.2.1 – Описание исходных данных с помощью библиотеки pandas после очистки данных



Сравнение с гистограммами из п.1.2: диаграммы распределения немного изменились для отдельных признаков. Были удалены крайние выбросы для признаков SHORT-LINE-DENSITY-5, SHORT-LINE-DENSITY-2, HEDGE-SD, VEDGE-SD. Все выбросы устранить не удалось, так как пострадает очень много данных. Изменение данных также можно отследить по таблице описания данных, сравнив ее с предыдущей. Главное отличие в том, что значения min и max признаков поменялись.

*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 данные не слишком пострадали и 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*

*а) Преобразование входов*

Был использован класс StandardScaler из sklearn.preprocessing для стандартизации с параметрами (0, 1) и масштабирования каждого из признаков в отдельности. Стандартизация необходима для того, чтобы все признаки были примерно одинаковы по величине и одинаково влияли на обучение сети, иначе признаки с высокими значениями будут влиять гораздо сильнее чем признаки с низкими.

*б) Преобразование выходов*

Выход с наибольшим значением заменяется единицей. Все остальные выходы – нулями.

Считаем, что сеть выбрала класс с наибольшим значением вероятности.

*в) Визуальный анализ преобразованных данных*

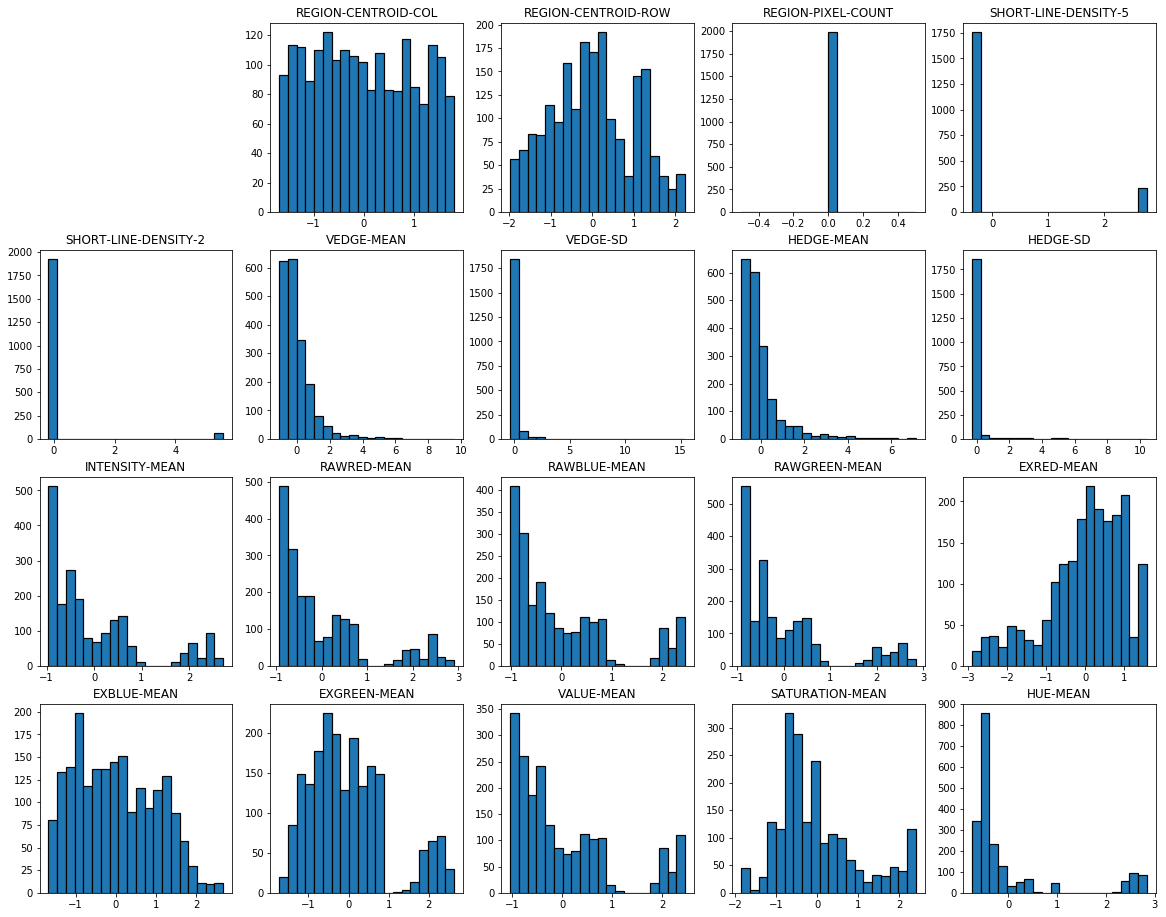


Рис. 2.4 – Гистограммы распределения нормализованных данных

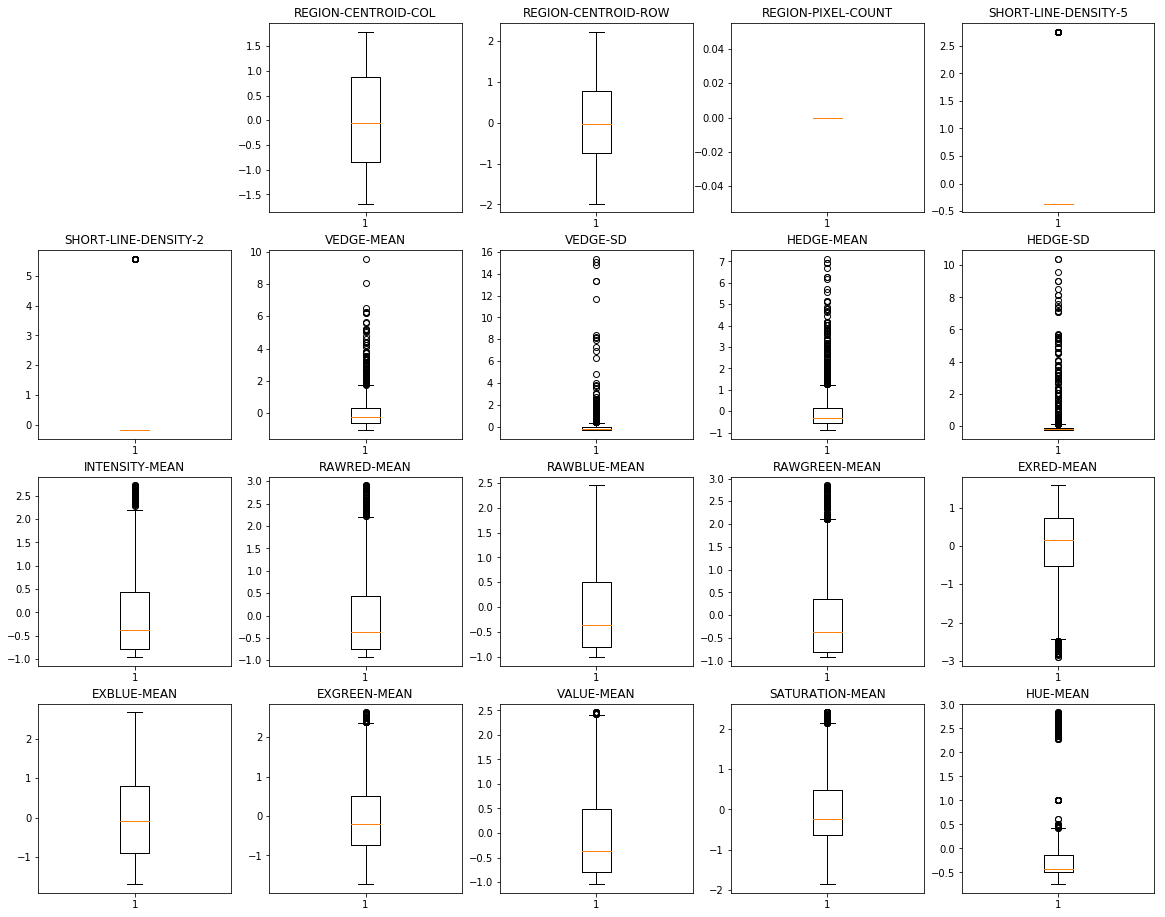
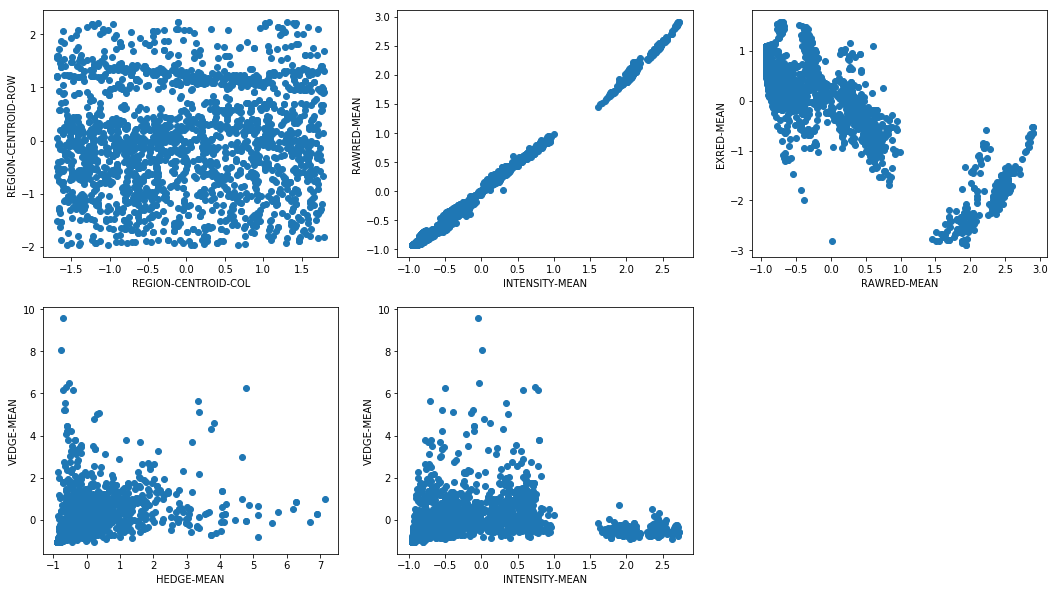


Рис. 2.5 – Диаграммы *Box-and-Whisker нормализованных данных*



*Рис. 2.6 – Диаграммы рассеяния отдельных признаков* *нормализованных данных*

Сравнение с диаграммами из п. 2.1 г): Форма диаграмм не поменялась. Значит, распределения и зависимости между признаками остались такими же. Данные приведены к небольшим значениям.

*2.3.Выводы*

Была проведена нормализация, в результате чего признаки приведены к необходимому для обучения виду.

III. Формирование признаков

*3.1. Сокращение числа признаков*

В текущем наборе данных всего 19 признаков, из которых REGION-PIXEL-COUNT, SHORT-LINE-DENSITY-5, SHORT-LINE-DENSITY-2, HEDGE-SD, VEDGE-SD малоинформативны, судя по их диаграммам, построенным ранее. Данные признаки лучше убрать из выборки.

*Результат: осталось 14 признаков.*

*3.2. Конструирование новых признаков*

Сильно скоррелированные признаки лучше обработать по методу главных компонент. Тем самым мы уменьшим время обучения, уменьшим вероятность переобучения и повысим точность. Был использован метод главных компонент PCA из библиотеки sklearn. Было принято решение преобразовать 5 сильно скоррелированных признаков INTENSITY-MEAN, RAWRED-MEAN, RAWBLUE-MEAN, RAWGREEN-MEAN, VALUE-MEAN в два новых признака. Преобразование в три или более признака оставляет последний признак с очень маленькими значениями данных.

*Результат: осталось 11 признаков.*

*3.3. Выводы*

По результатам проделанной работы количество признаков было сокращено с 19 до 11.

IV. Построение и исследование нейросетевых моделей

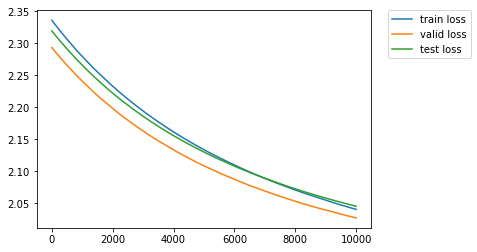
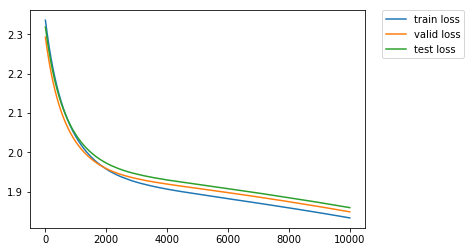
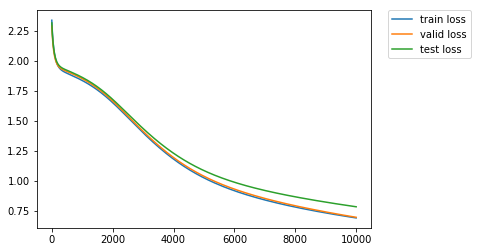
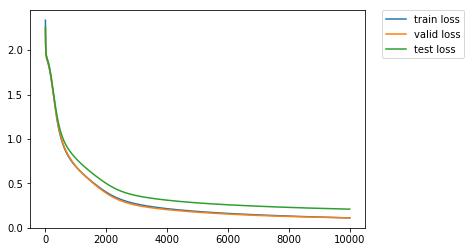
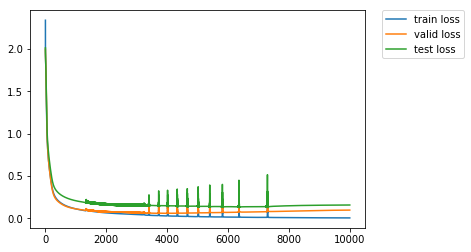
*4.1. Параметры архитектуры и обучения многослойной нейронной сети*

|  |  |
| --- | --- |
| Параметр | Значение |
| Функция потерь | *Categorical Cross Entropy* |
| Число входов сети | 11 |
| Число выходов сети | 7 |
| Число скрытых слоев сети\* | 2 |
| Число и АХ нейронов 1-го скрытого слоя\* | 20, logistic |
| Число и АХ нейронов 2-го скрытого слоя\* | 10, logistic |
| Число и АХ нейронов 3-го скрытого слоя\* | - |
| АХ нейронов выходного слоя | SoftMax |
| Кросс-валидация | Holdout (60/30/10) |
| Объёмы обучающей / валидационной / тестовой выборок | 1193 / 596 / 199 |
| Режим обучения\* | Batch |
| Метод инициализации весов | метод Хавьера |
| Критерий останова | Выполнение определенного количества эпох |
| Ранний останов | да |

\* Определяется вариантом задания.

*4.2. Исследование простого градиентного метода обучения*

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



**Указание**: обучение каждый раз начинать из одной и той же начальной точки.

*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 | 10000 | 0.006 | 0.1584 |
| 2 | 0.1 | 10000 | 0.112 | 0.210 |
| 3 | 0.01 | 10000 | 0.688 | 0.7815 |
| 4 | 0.001 | 10000 | 1.833 | 1.859 |
| 5 | 0.0001 | 10000 | 2.040 | 2.045 |

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

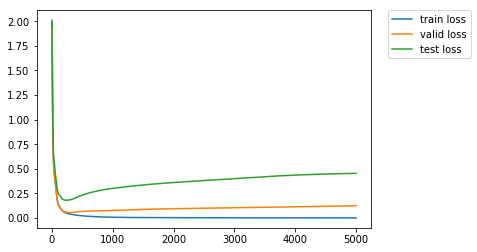
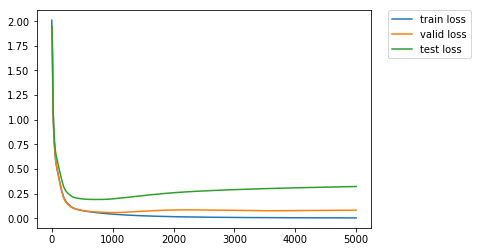
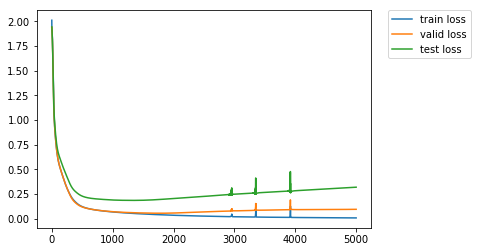
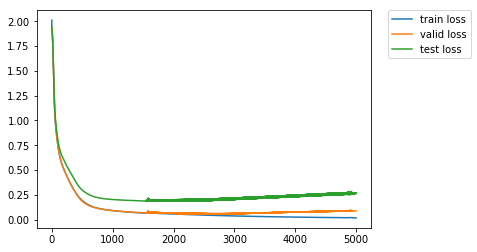
*в) Выводы*

Чем больше скорость обучения, тем сильнее обучалась сеть. Возможно, это связано с гладкостью оптимизируемой функции при градиентном спуске.

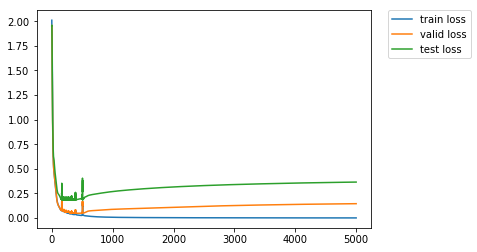
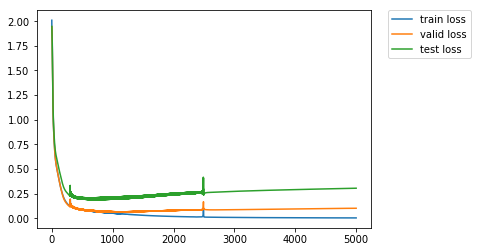
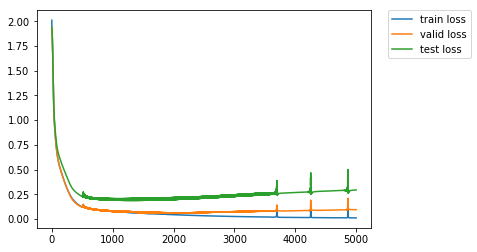
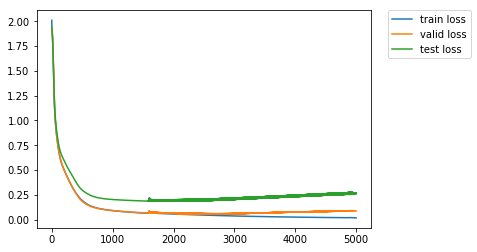
*4.3. Исследование методов GDM и NAG*

*а) Исследование влияния параметра момента на качество обучения*

метод *GDM (по увеличению* μ*):*



метод *NAG (по увеличению* μ*):*



**Указание**: параметр скорости обучения выбрать наилучшим по результатам исследований п. 4.2.

*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 | 2500 | 0.0179 | 0.2671 |
| 2 | NAG | 2500 | 0.0180 | 0.2634 |
| 3 | GDM | 0.3 | 2500 | 0.009 | 0.3203 |
| 4 | NAG | 2500 | 0.0106 | 0.2935 |
| 5 | GDM | 0.6 | 2500 | 0.0041 | 0.3230 |
| 6 | NAG | 2500 | 0.0045 | 0.3057 |
| 7 | GDM | 0.9 | 2500 | 0.0007 | 0.4536 |
| 8 | NAG | 2500 | 0.0006 | 0.3648 |

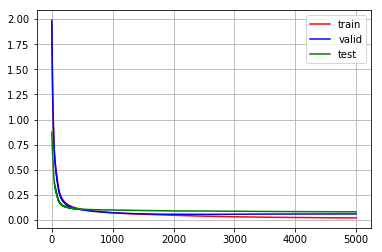
*в) Выводы*

В данном случае, чем больше использовалось значение момента μ, тем быстрее обучалась сеть, тем быстрее обучение доходило до точки переобучения, и тем лучше конечный результат ошибки на обучающей выборке, что не скажешь про тестовую. Методы GDM и NAG дали почти одинаковые результаты при равных μ. Разногласия конечных ошибок можно видеть в таблице, но они не существенны.

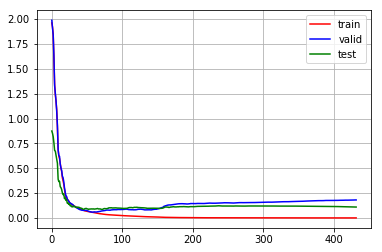
*4.4. Исследование методов наискорейшего спуска и сопряжённых градиентов*

*а) Сравнение кривых обучения*

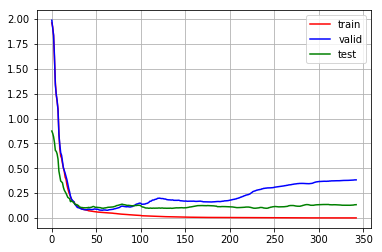
1) метода наискорейшего спуска



2) метода Флетчера-Ривса



3) метода Полака-Райбера.



*б) Заполнить таблицу по результатам обучения*

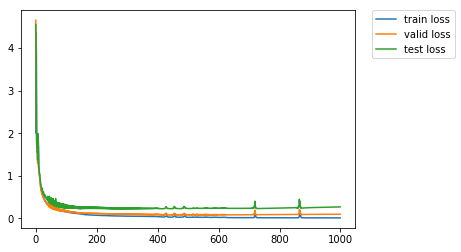
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Метод | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | SGD | 5000 | 0.0206 | 0.0815 |
| 2 | Fletcher-Reeeves | 450 | 0.00014 | 0.1094 |
| 3 | Polak-Ribiere | 350 | 0.00015 | 0.1333 |

*в) Выводы*

При использовании метода SGD мы видим ровную кривую обучения, хорошее значение ошибки на обучающей и тестовой выборке. При использовании методов Fletcher-Reeeves и Polak-Ribiere значение ошибки на обучающей выборке предельно мало, но ошибка на тестовой выборке получилась хуже, чем при использовании метода наискорейшего спуска. Примечательно, что методы Fletcher-Reeeves и Polak-Ribiere обучаются за малое кол-во эпох.

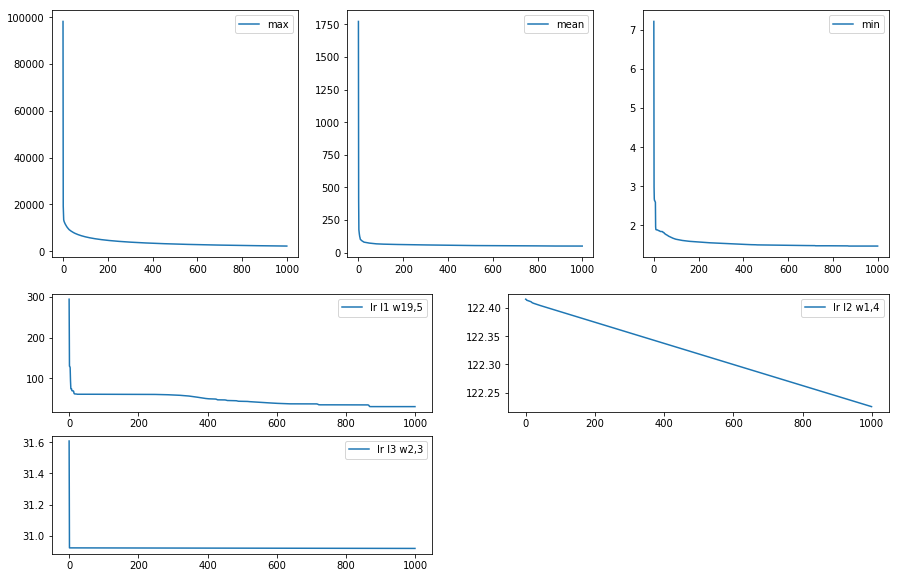
*4.5. Исследование метода AdaGrad*

*а) Кривые обучения*



**Указание**: базовую скорость обучения выбрать наилучшей по результатам исследований п. 4.2.

*б) Исследование динамики скорости 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 | 1000 | 0.0130 | 0.2712 |

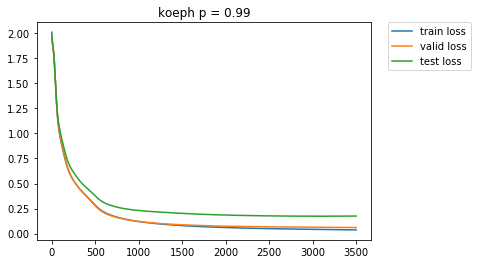
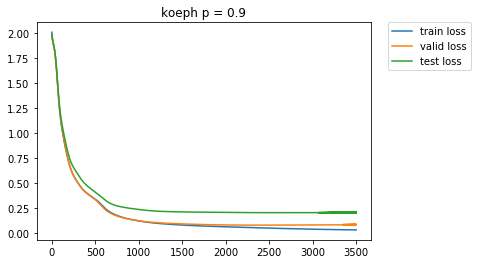
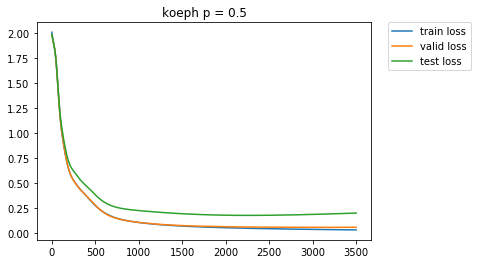
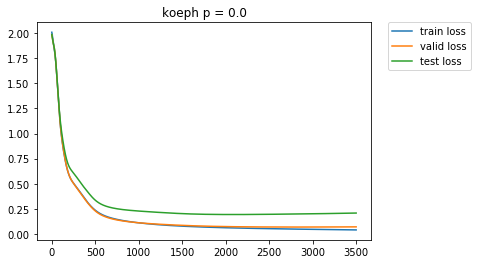
*г) Выводы*

Резкое уменьшение динамики скорости обучения всех весов. Скорость обучения синаптического коэффициента w1,4 в слое l2 линейно падала, держась около значения 122.

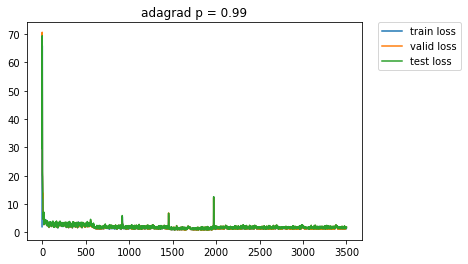
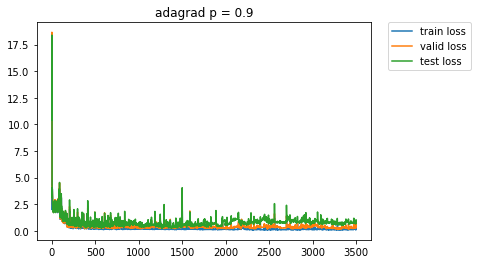
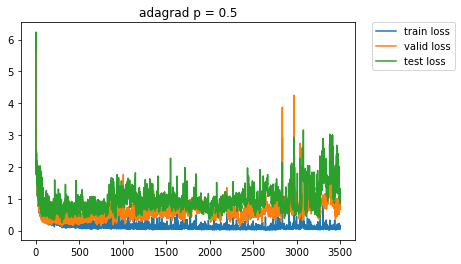
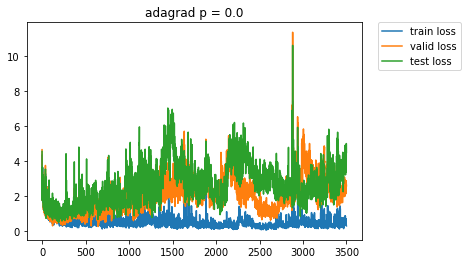
*4.6. Исследование методов RMSProp и AdaDelta*

*а) Сравнение кривых обучения*

*Кривые обучения AdaDelta:*

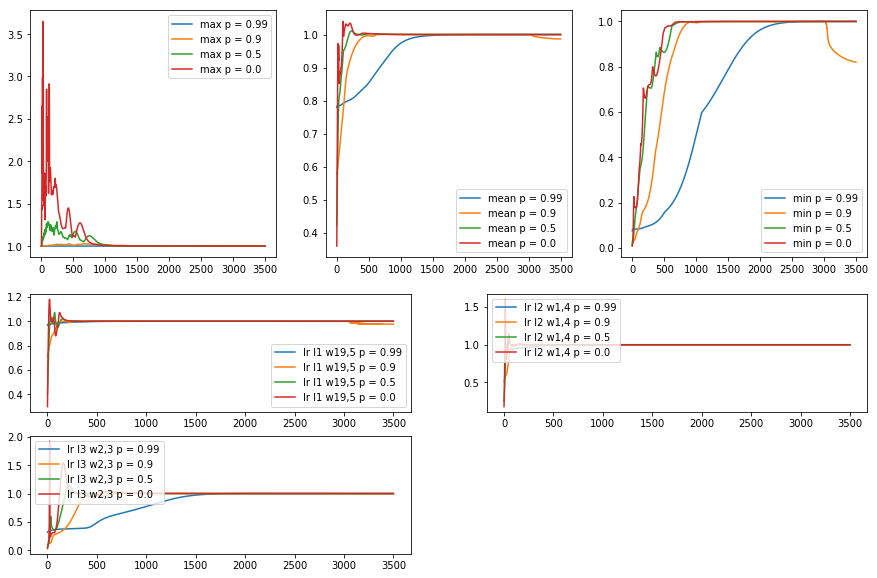


*Кривые обучения RMSProp:*

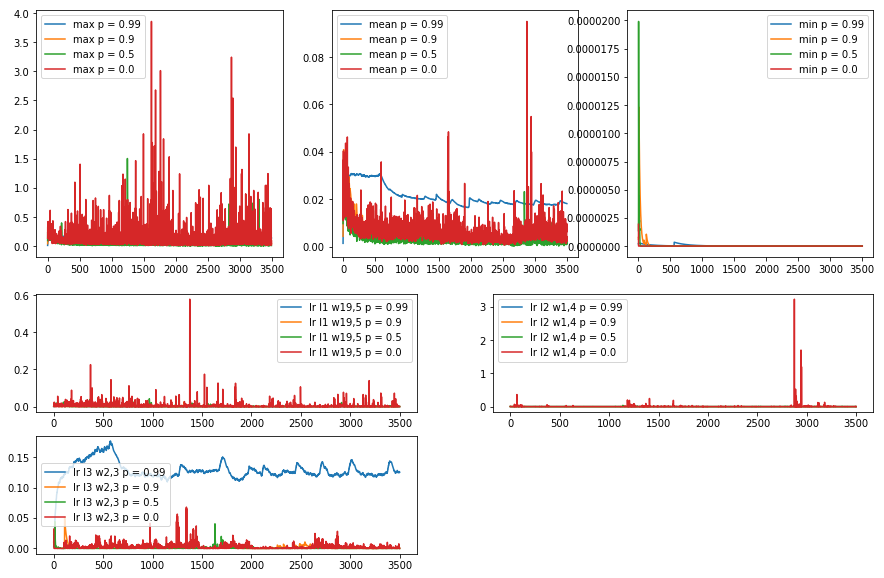


*б) Исследование динамики скорости обучения*

*Скорости обучения* AdaDelta:



*Скорости обучения* RMSProp:



*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 | 0.0 | 3500 | 0.2946 | 4.4938 |
| 2 | AdaDelta | 3500 | 0.03913 | 0.20727 |
| 3 | RMSProp | 0.5 | 3500 | 0.0504 | 0.9884 |
| 4 | AdaDelta | 3500 | 0.03431 | 0.20264 |
| 5 | RMSProp | 0.9 | 3500 | 0.3998 | 0.8900 |
| 6 | AdaDelta | 3500 | 0.03289 | 0.21144 |
| 7 | RMSProp | 0.99 | 3500 | 1.6575 | 1.8490 |
| 8 | AdaDelta | 3500 | 0.03727 | 0.17559 |

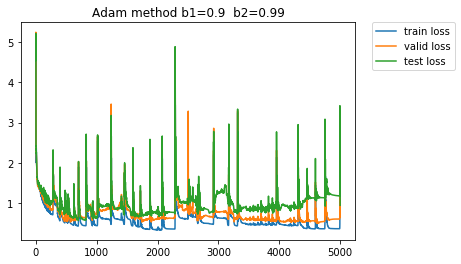
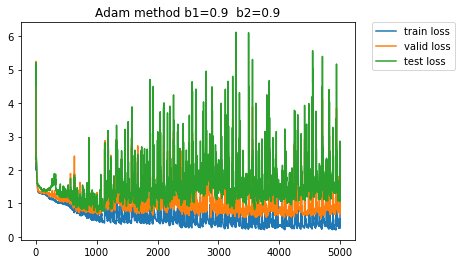
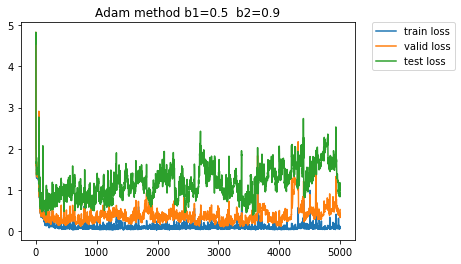
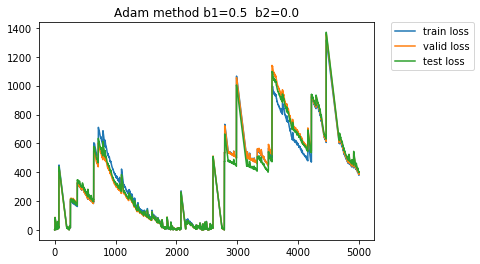
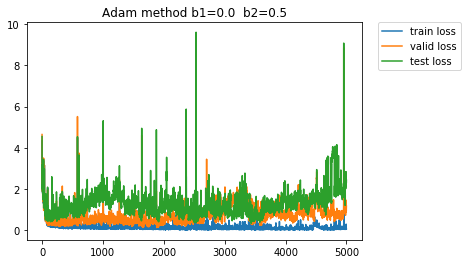
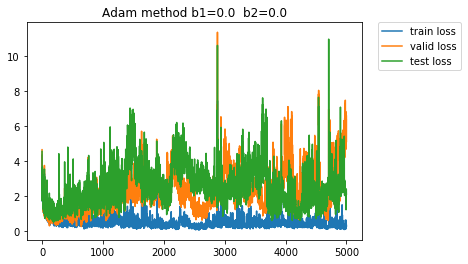
*г) Выводы*

*Кривые обучения по методу* AdaDelta почти не отличаются друг от друга при изменении параметра сглаживания. Однако, можно проследить конечное изменение ошибки на обучающей и тестовой выборке. Наилучшее значение ошибки на обучающей выборке было получено с параметром ρ = 0.9. Наилучшее значение ошибки на тестовой выборке было получено с параметром ρ = 0.99. Стоит отметить, что скорости обучения отдельных синаптических весов почти всегда равны начальной, изредка становясь чуть больше, но чаще чуть меньше единицы.

При обучении по методу RMSProp чем ниже было значение параметра сглаживания, тем хаотичней происходило обучение. Возможно, это связано с высокой начальной скоростью обучения равной 1.0. Наилучшее значение параметра сглаживания для метода RMSProp в данном случае ρ = 0.9. При данном значении параметра, кривая обучения не ведет себя слишком хаотично, а значение ошибки на тестовой выборке минимально.

*4.7. Исследование метода Adam*

*а) Сравнение кривых обучения*

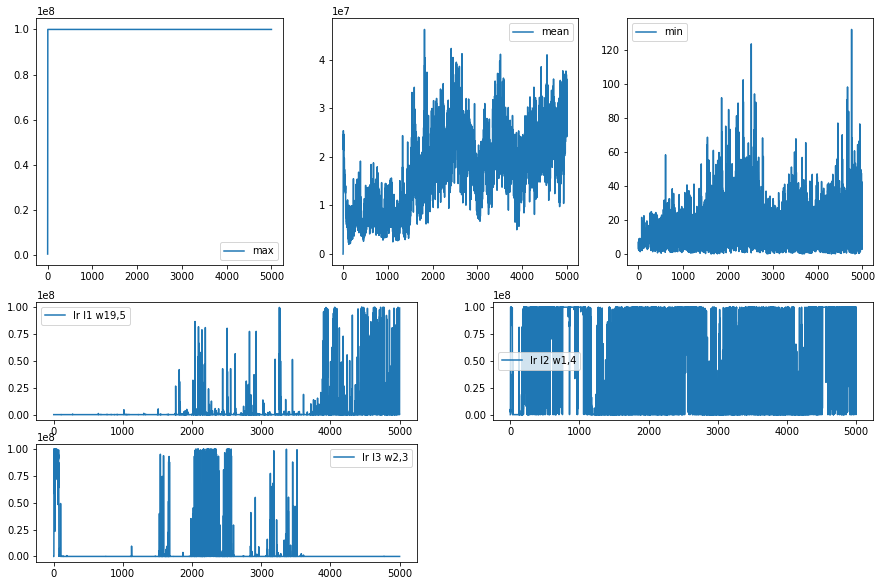


**Указание 1**: базовую скорость обучения выбрать наилучшей по результатам исследований п. 4.2.

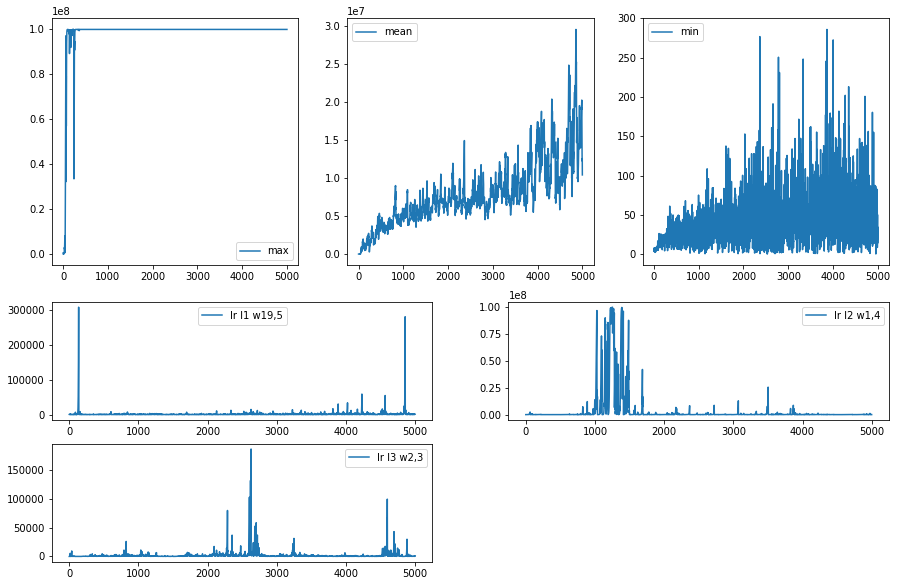
**Указание 2**: в одном из экспериментов выбрать β1 и β2 равными наилучшему значению параметра ρ по результатам исследований п. 4.6.

*б) Исследование динамики скорости обучения*

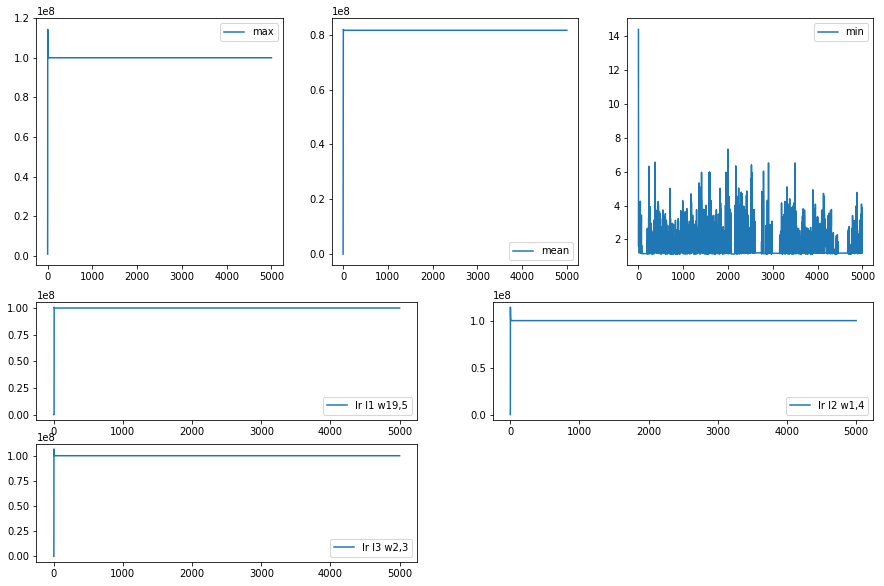
β1 = 0.0 β2 = 0.0:



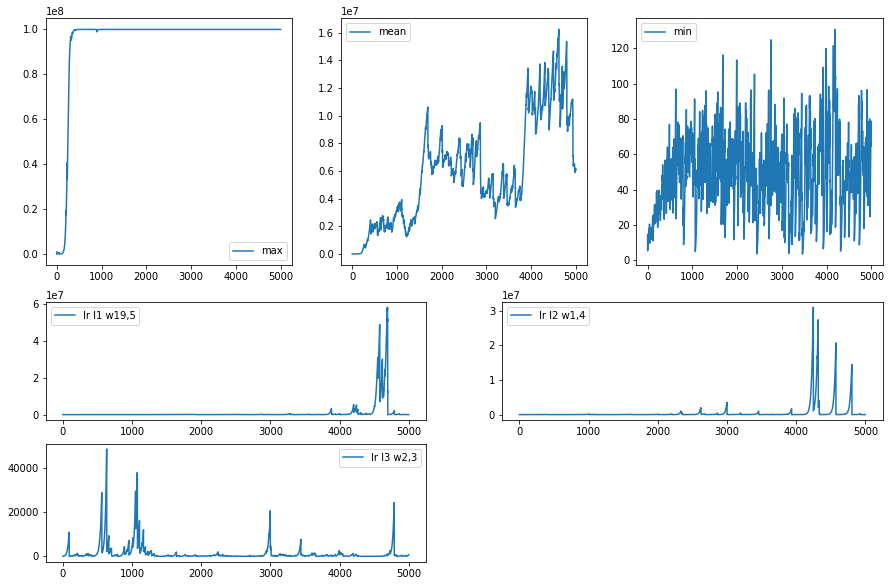
β1 = 0.0 β2 = 0.5:



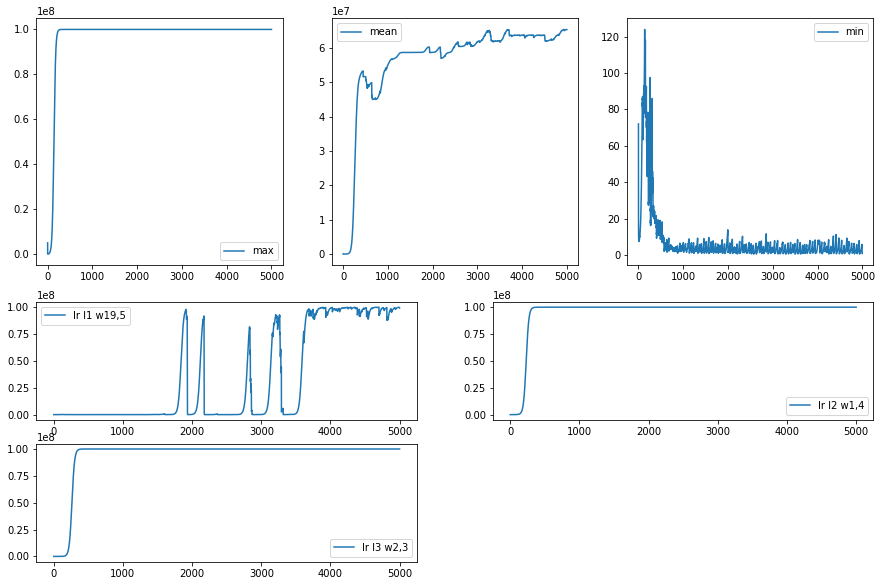
β1 = 0.5 β2 = 0.0:



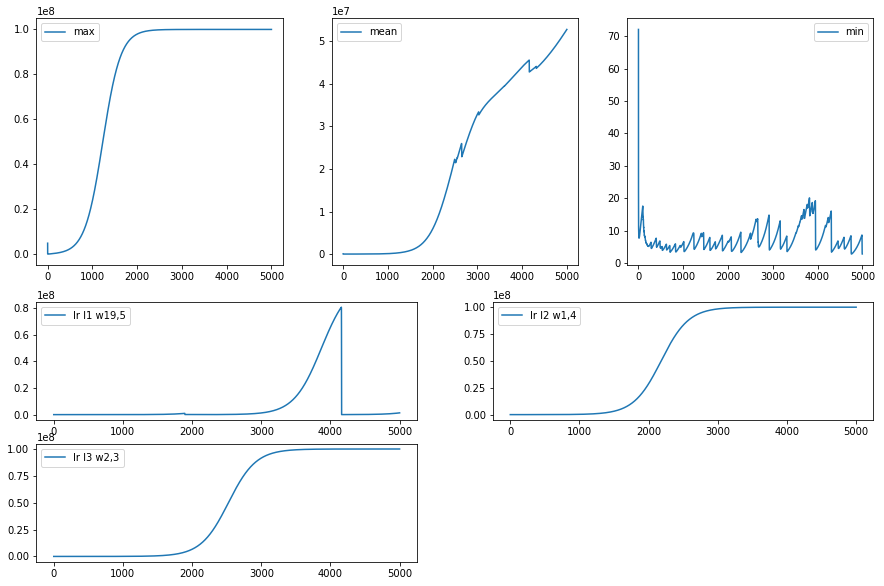
β1 = 0.5 β2 = 0.9:



β1 = 0.9 β2 = 0.9:



β1 = 0.9 β2 = 0.99:



*в) Заполнить таблицу по результатам обучения*

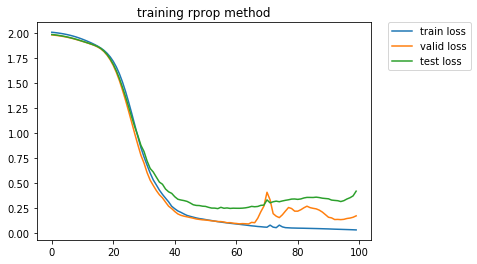
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | β1 | β2 | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | 0.0 | 0.0 | 5000 | 0.276 | 2.113 |
| 2 | 0.0 | 0.5 | 5000 | 0.071 | 2.430 |
| 3 | 0.5 | 0.0 | 5000 | 382.404 | 399.964 |
| 4 | 0.5 | 0.9 | 5000 | 0.079 | 0.922 |
| 5 | 0.9 | 0.9 | 5000 | 1.798 | 1.082 |
| 6 | 0.9 | 0.99 | 5000 | 2.049 | 0.939 |

*г) Выводы*

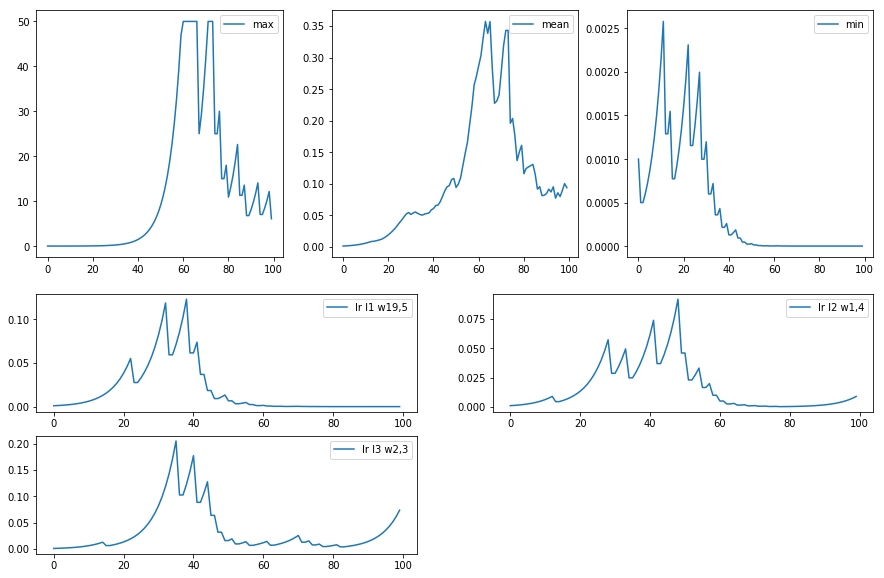
При параметрах сглаживания β1 = 0.5 β2 = 0.0 сеть обучилась неправильно. При остальных параметрах сеть обучалась, но обучалась очень хаотично. Скорости обучения отдельных синаптических весов сильно возрастали в течении обучения, что могло изменить конечное значение ошибки на обучающей и тестовой выборке. Возможно, это связано с высокой начальной скоростью обучения равной 1.0. При параметрах (0.5, 0.9), (0.9, 0.9), (0.9, 0.99) сеть выдала более-менее нормальное значение ошибки на тестовой выборке.

*4.8. Исследование метода RProp*

*а) Кривые обучения*



*б) Исследование динамики приращений весов*



*в) Заполнить таблицу по результатам обучения*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Метод | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | RProp | 100 | 0.0280 | 0.4166 |

*г) Выводы*

После 100 эпох обучения сеть обучилась, далее начинается переобучение сети. Среднее приращение непрерывно менялось в процессе обучения. По факту, переобучение сети начинается еще после 60-й эпохи. Это момент, когда максимальное приращение весов достигло предела, а также минимальное приращение весов достигло минимума.

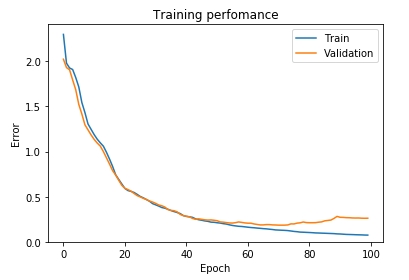
*4.9. Исследование методов Левенберга-Маркардта и BFGS*

*а) Кривые обучения*

1) метода Левенберга-Маркардта



2) метода BFGS.



*б) Заполнить таблицу по результатам обучения*

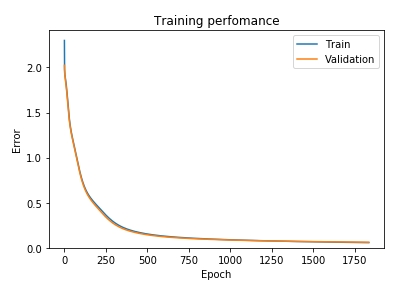
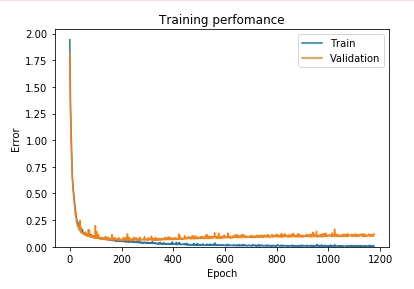
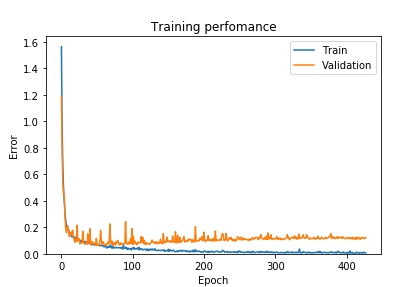
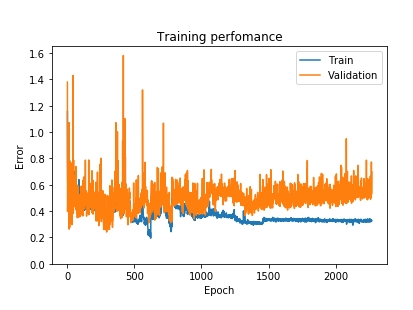
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Метод | Число эпох обучения | Ошибка на обучающей выборке, *Eобуч* | Ошибка на тестовой выборке, *Eтест* |
| 1 | LM | 21 | 0.2469 | 0.2373 |
| 2 | BFGS | 100 | 0.074 | 0.2614 |

*в) Выводы*

В библиотеке NeuPy оказалось невозможным использовать функцию ошибки CategoricalCrossEntropy для метода Левенберга-Маркардта. Возможно, из-за этого сеть плохо (неправильно) обучилась. При использовании метода BFGS происходило переобучение сети после 100 эпохи. Ошибка на тестовой выборке, которую удалось добиться 0.2614.

*4.10. Исследование метода стохастического градиента*

*а) Сравнение кривых обучения*

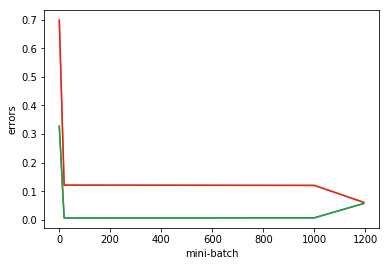


**Указание**: параметр скорости обучения простого градиентного метода выбрать наилучшим по результатам исследований п. 4.2.

*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 | 2270 | 0.3277 | 0.6977 |
| 2 | GD | 20 | 427 | 0.0070 | 0.1216 |
| 3 | GD | 100 | 1177 | 0.0073 | 0.1206 |
| 4 | GD | равен объёму выборки 1193 | 1833 | 0.0574 | 0.0615 |

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

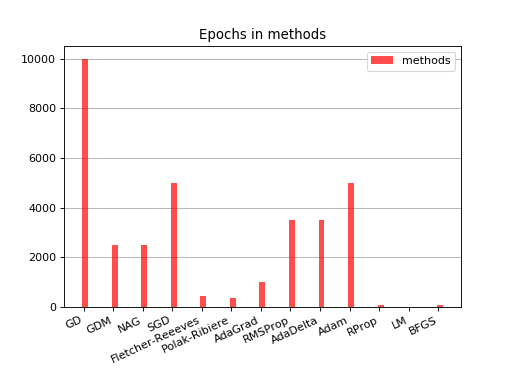


*г) Выводы*

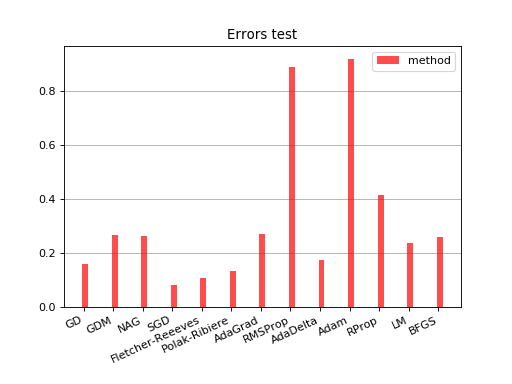
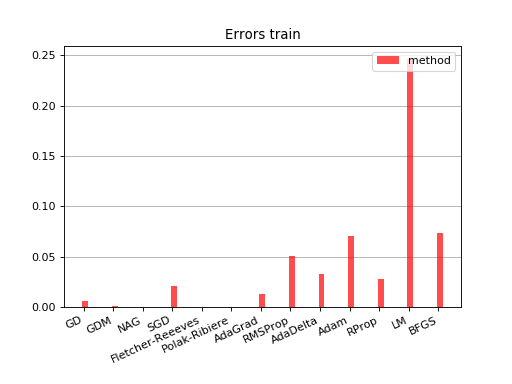
Есть смысл использовать batch, а не стохастический GD, т.к. при размере mini-batch-a равному 1 кривая обучения ведет себя хаотично, гораздо лучше уже при небольшом количестве элементов в batch-е. При размере mini-batch-a равного 20 сеть имеет низкую ошибку на обучающей выборке. Наилучшая ошибка на тестовой выборке была получена при размере mini-batch-a равного объему выборки.

*4.11. Сравнение методов обучения*

*а) Сравнение числа эпох обучения*



*б) Сравнение качества обученных нейросетевых моделей*

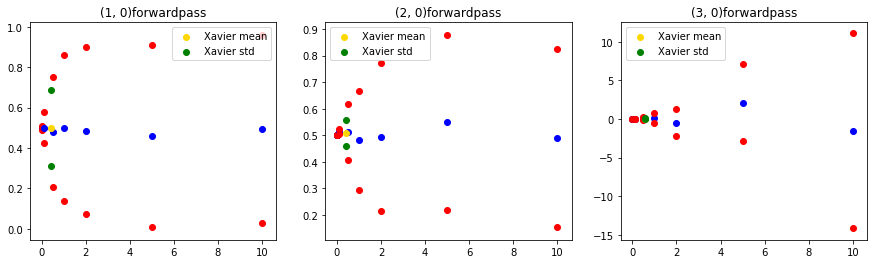


*в) Выводы*

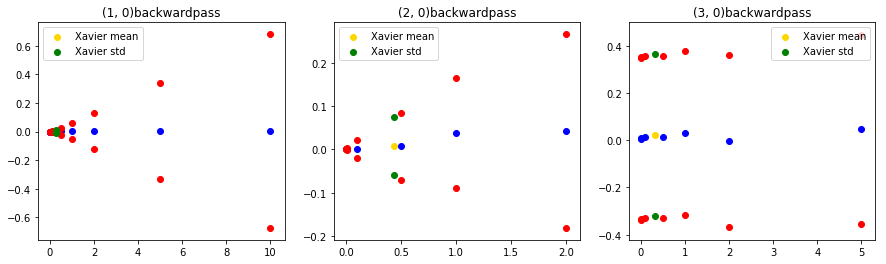
Обычный градиентный метод показал хороший результат, но для его реализации было потрачено наибольшее количество эпох среди всех методов. Методы RMSProp и Adam показали большую ошибку на тестовой выборке, возможно, из-за большой скорости обучения равной 1.0 (следовал указанию в методе). Ошибка на тестовой выборке, меньшая чем у метода GD у методов SGD, Fletcher-Reeeves, Polak-Ribiere, AdaDelta.

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

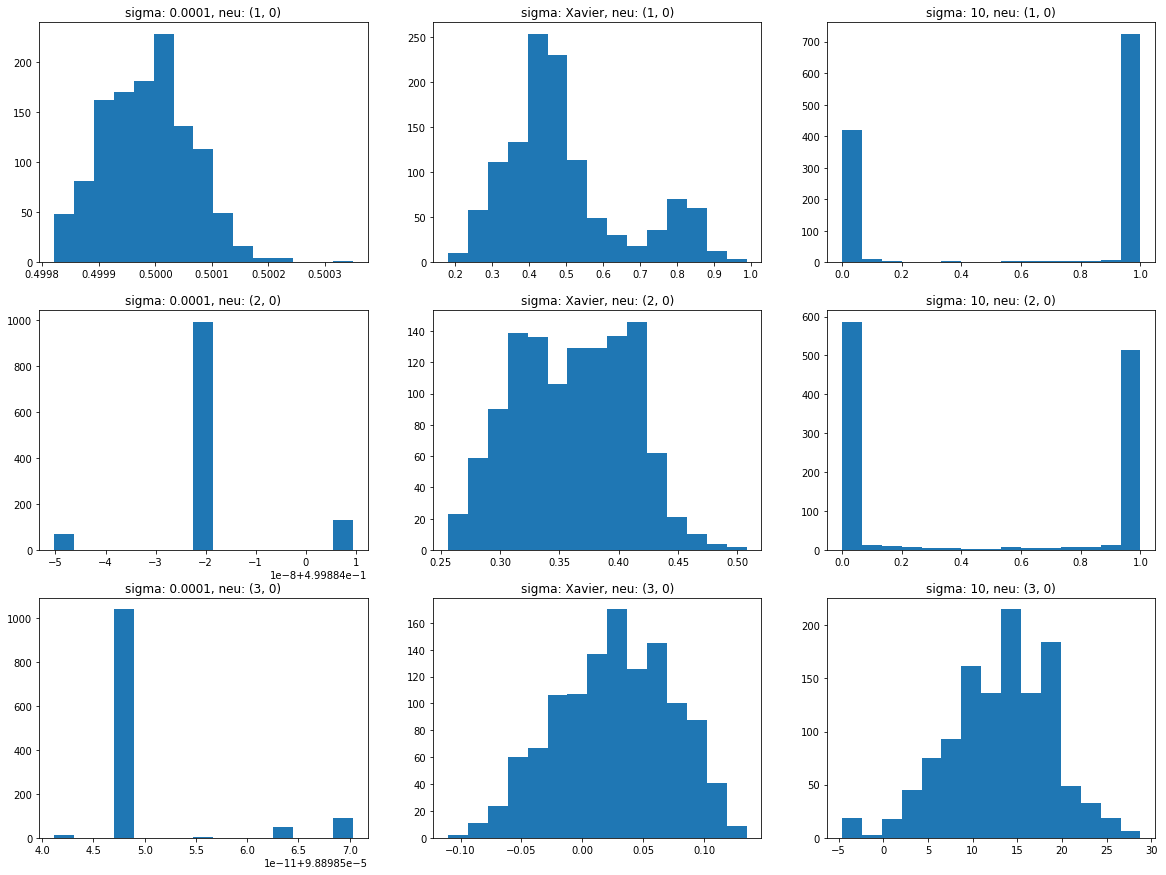
*а) Исследование прямого информационного потока в сети*



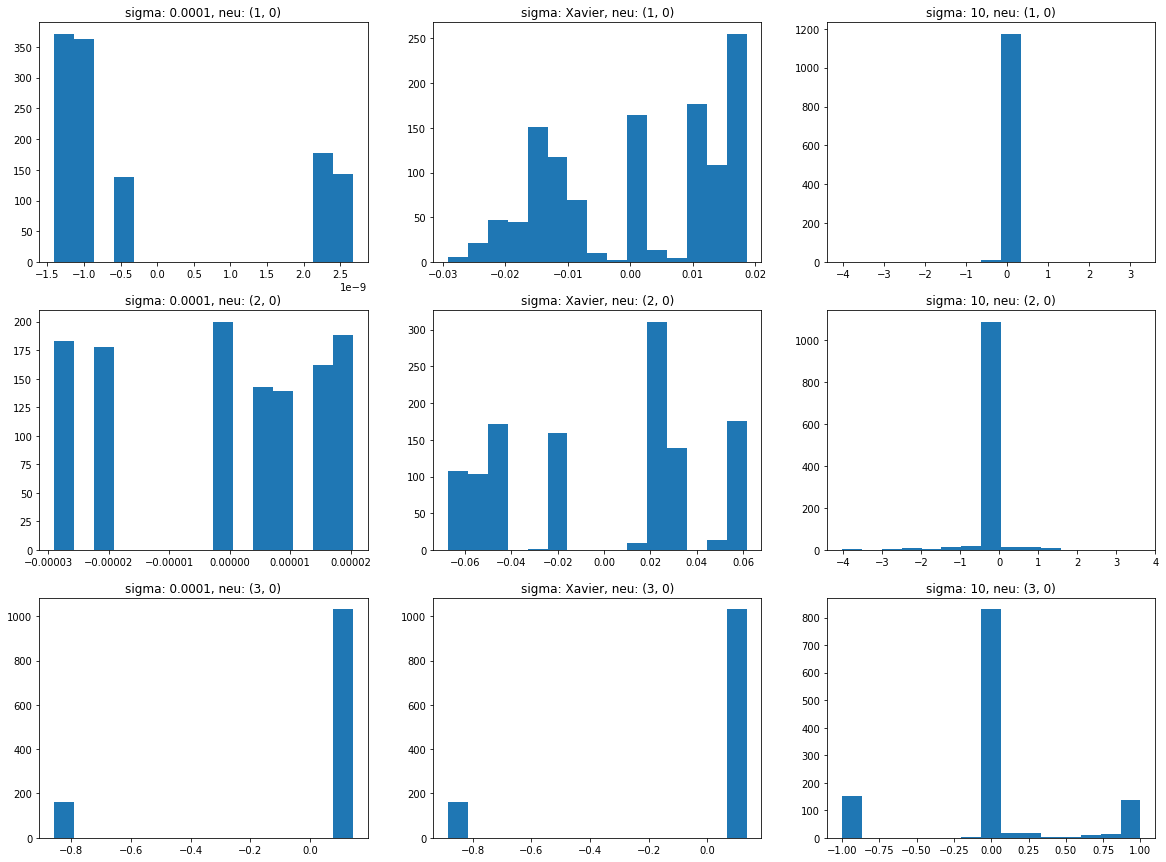
*б) Исследование обратного информационного потока в сети*



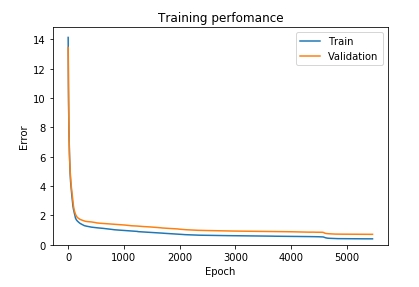
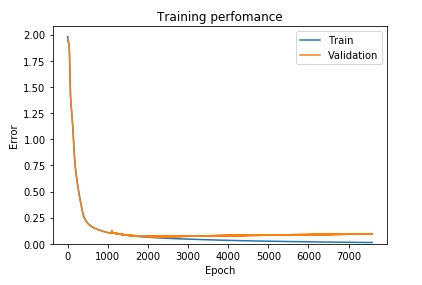
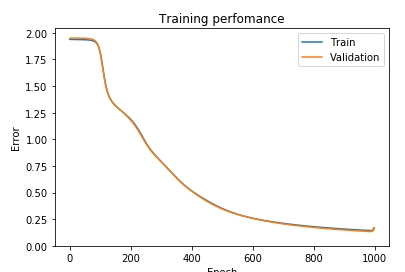
*в) Исследование распределений выходов*



*и двойственных потенциалов нейронов:*



*г) Кривые обучения (мал., Xavier, бол.)*



**Указание**: для обучения использовать простой градиентный метод, параметр скорости обучения выбрать наилучшим по результатам исследований п. 4.2.

*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 | 0.05 | 0.0918±0.0517 | 0.1051±0.0365 |
| 2 | Нормальное | 0.0721±0.0468 | 0.0868±0.0357 |
| 3 | Усеченное нормальное | 0.0802±0.0343 | 0.0929±0.0173 |
| 4 | Равномерное | По методу Ксавьера | 0.0588±0.0165 | 0.0709±0.0088 |
| 5 | Нормальное | 0.0559±0.0366 | 0.0815±0.0211 |
| 6 | Усеченное нормальное | 0.055±0.040 | 0.085±0.016 |
| 7 | Равномерное | 10.0 | 0.911±1.500 | 1.406±1.862 |
| 8 | Нормальное | 0.3142±0.063 | 0.721±0.117 |
| 9 | Усеченное нормальное | 0.0319±0.0027 | 0.0925±0.0160 |

**Указание 1**: для обучения использовать простой градиентный метод, параметр скорости обучения выбрать наилучшим по результатам исследований п. 4.2.

**Указание 2**: для расчета среднего значения и с.к.о. ошибки обученной сети проводить многократное (не менее 10 раз) обучение из различных начальных точек.

*е) Выводы*

При слишком маленьком с.к.о. распределения весов при начальной инициализации сеть не обучается. При значении с.к.о. 0.05 обучение возможно. Усеченное нормальное распределение показывает примерно такие же результаты, как нормальное. Инициализация с.к.о. весов по методу Ксавьера показала наилучшие результаты.

*4.13. Методы кросс-валидации*

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

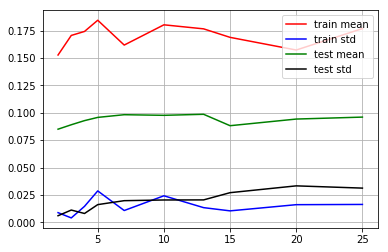
|  |  |  |  |
| --- | --- | --- | --- |
| Метод кросс-валидации | Число запусков обучения | Средняя ошибка на обучающей выборке ± с.к.о., | Средняя ошибка на тестовой выборке ± с.к.о., |
| Монте-Карло | 10 | 0.0815±0.02 | 0.0921±0.01 |
| Holdout 60/30/10 | 1 | 0.0758 | 0.1047 |
| 10-fold | 10 | 0.0659±0.008 | 0.089±0.02 |
| LOOCV | 1988 | 0.0672±0.003 | 0.090±0.05 |

**Указание 1**: для обучения использовать простой градиентный метод, параметр скорости обучения выбрать наилучшим по результатам исследований п. 4.2.

**Указание 2**: инициализацию сети провести один раз наилучшим методом по результатам исследований п. 4.12, запуски процедуры обучения каждый раз проводить из одной и той же начальной точки.

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

*б) Исследование k-fold кросс-валидации*



*в) Выводы*

LOOCV кросс-валидация имеет наибольшее количество запусков обучения, однако конечный результат не сильно отличается от метода 10-fold. Если учитывать погрешность, то все методы дают примерно одинаковый результат, судя по ошибке на тестовой выборке. Стоит отметить, что дисперсия ошибки на тестовой выборке при LOOCV кросс-валидации наибольшая. Это связано с тем, что каждую итерацию обучения тестовая выборка состоит только из 1 элемента.

Исследование k-fold кросс-валидации показывает, что результат не сильно зависит от количества элементов в выборке. Нет какого-либо возрастания или убывания ошибки от значения k.

*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 | 5, 5 | 0.222±0.06 | 0.156±0.05 |
| 2 | 10, 5 | 0.152±0.07 | 0.128±0.04 |
| 3 | 10, 10 | 0.112±0.07 | 0.110±0.05 |
| 4 | 20, 10 | 0.0638±0.03 | 0.110±0.019 |
| 5 | 20, 20 | 0.0134±0.008 | 0.094±0.029 |
| 6 | 25, 25 | 0.011±0.01 | 0.108±0.026 |

**Указание 1**: при выполнении пп. а)–б) для обучения использовать метод сопряженных градиентов, Левенберга-Маркардта или BFGS.

**Указание 2**: при выполнении пп. а)–б) среднее значение и с.к.о. ошибок на обучающей и тестовой выборках считать по результатам 10-fold кросс-валидации, внутри каждого фолда разбивать обучающие данные на обучающую и валидационную выборки в отношении 70/30.

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

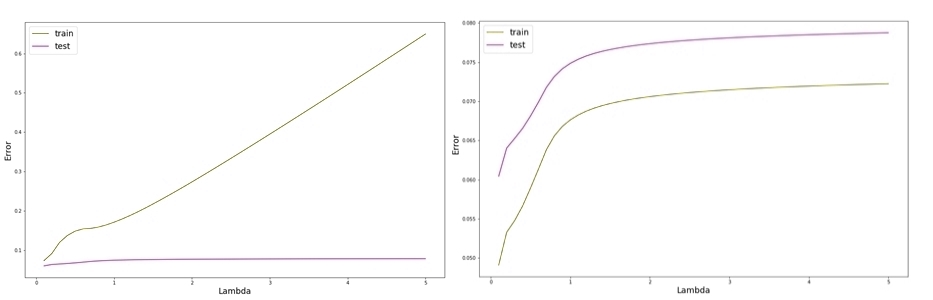
|  |  |  |
| --- | --- | --- |
| АХ нейронов  скрытых слоёв | Средняя ошибка на обучающей выборке ± с.к.о., | Средняя ошибка на тестовой выборке ± с.к.о., |
| logistic | 0.057±0.03 | 0.09±0.02 |
| tanh | 0.0083±0.004 | 0.098±0.029 |
| linear | 0.109±0.011 | 0.083±0.018 |
| softsign | 0.0077±0.005 | 0.111±0.029 |
| softplus | 0.00019±0.0005 | 0.079±0.025 |

*4.15. L1 и L2 регуляризация весов*

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

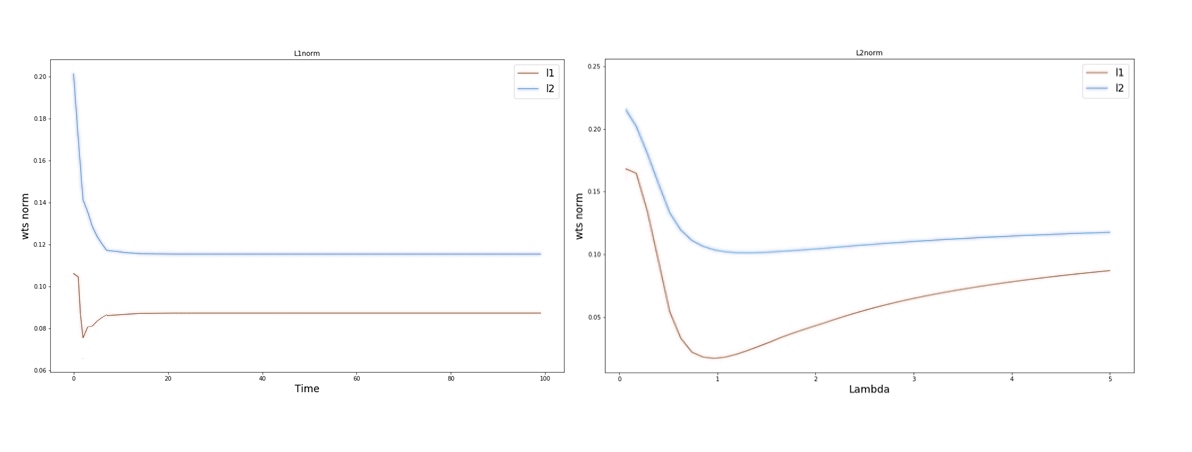
**Указание**: для архитектуры 20-20, метод AdaGrad

Графики зависимостей ошибки обученной сети на обучающей и тестовой выборках от значения параметра λ при L1 и L2 регуляризации соответственно.



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

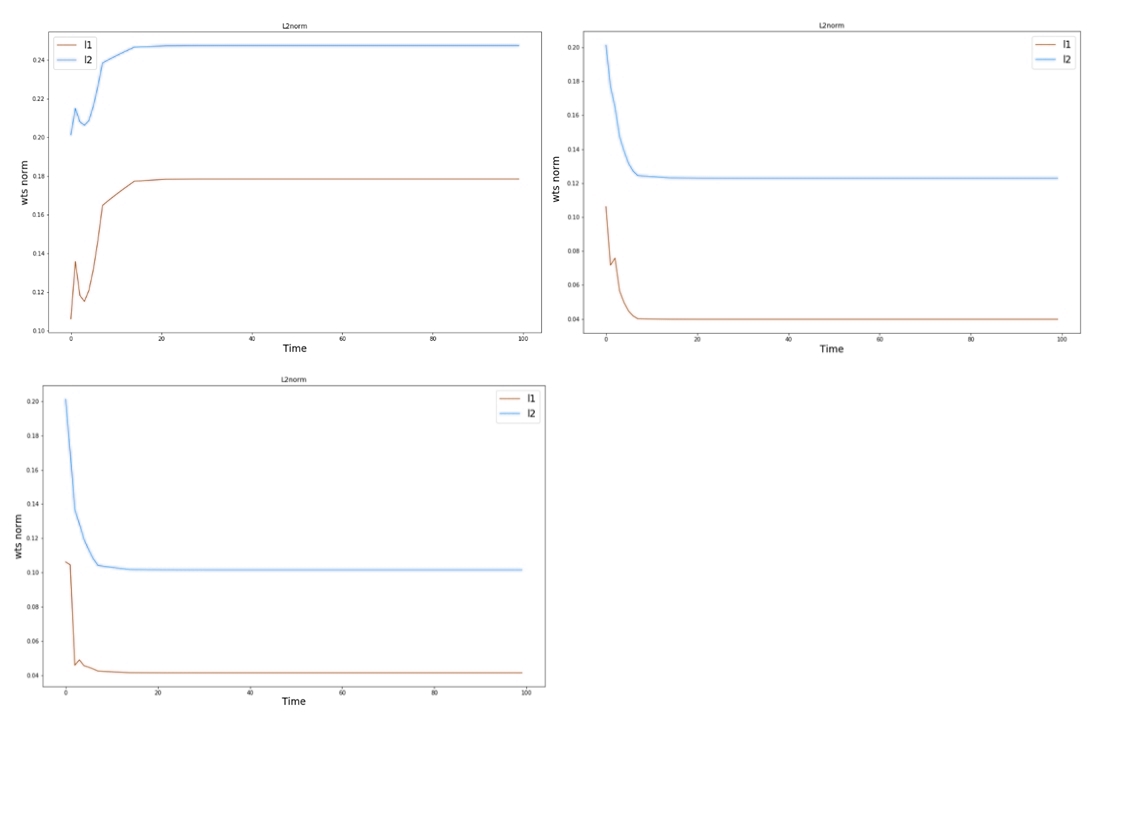
Графики зависимостей *L*1 и *L*2 нормы вектора синаптических коэффициентов обученной сети от параметра λ при L1 и L2 регуляризации соответственно.



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

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

*г) Выводы*

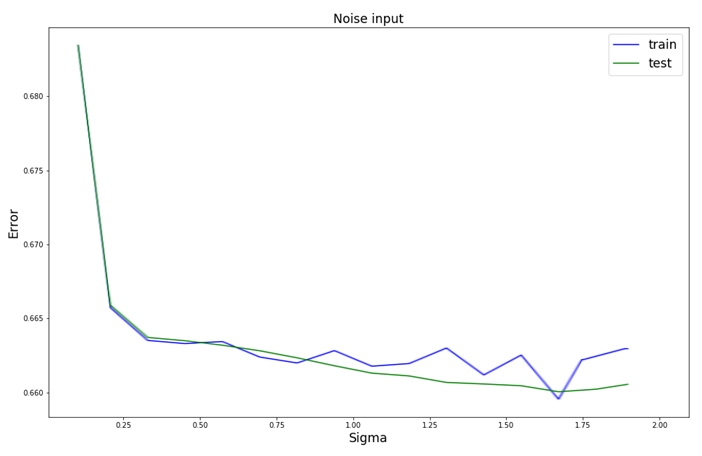
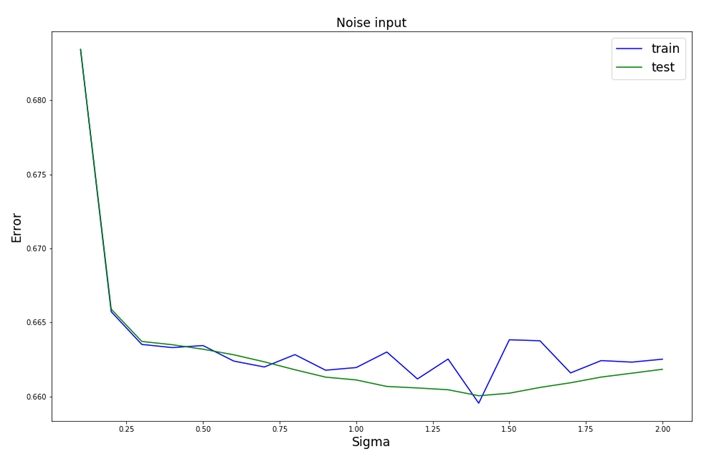


Чем больше λ, тем хуже качество обучения, а также тем нормы синаптических весов меньше в начале обучения.

*4.16. Инъекция шума*

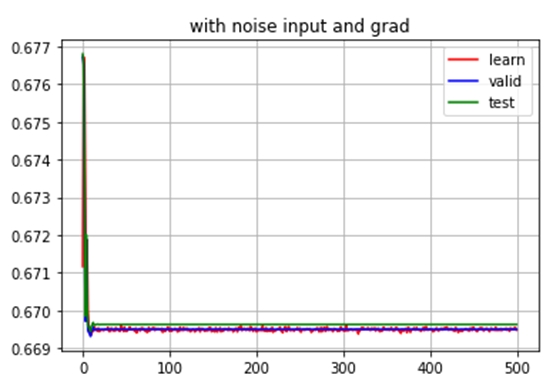
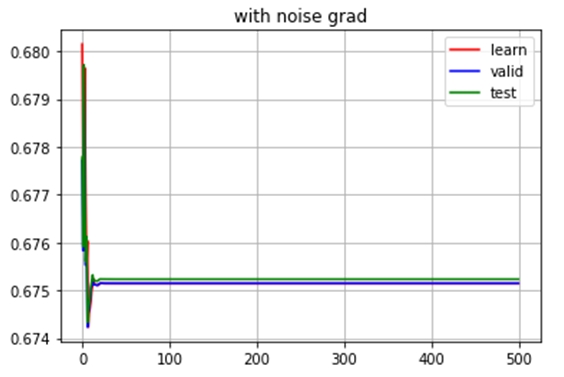
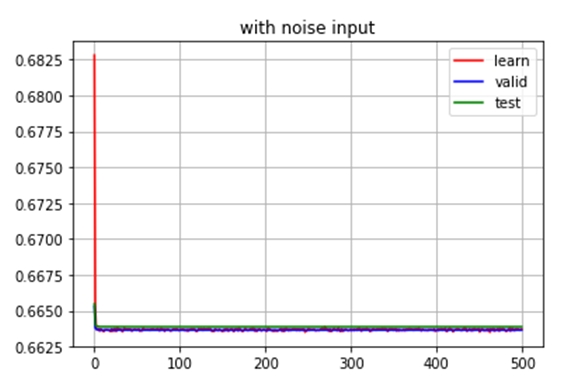
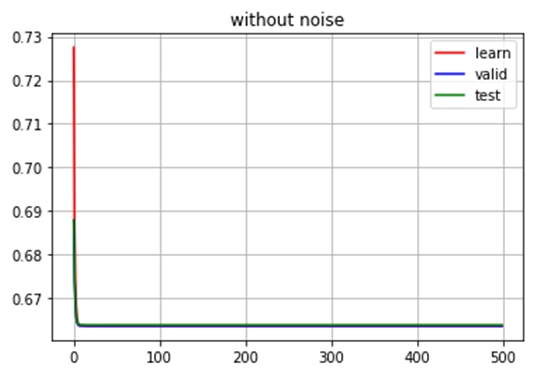
*а) Исследование зависимости качества обученной сети от параметра зашумления*

Построить графики зависимости ошибки обученной сети на обучающей и тестовой выборках от с.к.о. σ шума при: 1) зашумлении входов; 2) зашумлении градиентов (соответственно).



*б) Сравнение кривых обучения*

Построить кривые обучения на обучающей, валидационной и тестовой выборках при обучении: 1) без зашумления; 2) с зашумлением входов; 3) с зашумлениемградиентов; 4) с зашумлениемвходов иградиентов.



*в) Выводы*

При увеличении σ шума ошибка обученной сети уменьшалась. Модель робастна.

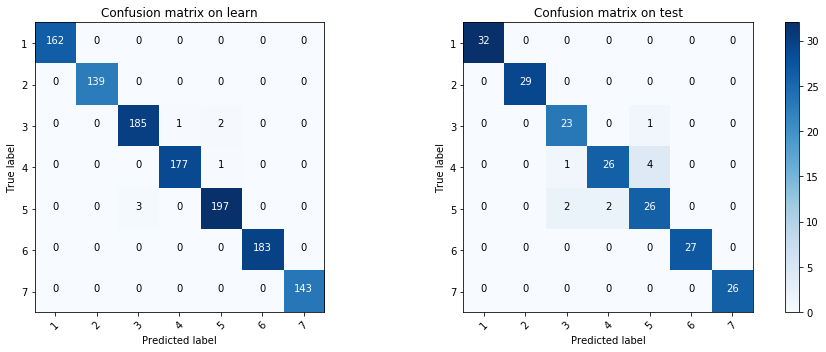
V. Исследование обученной нейросетевой модели

**Указание:** среди всех построенных в пп. 4.2–4.16 нейросетевых моделей выбрать модель с наименьшими значениями ошибок на обучающей и валидационной выборках.

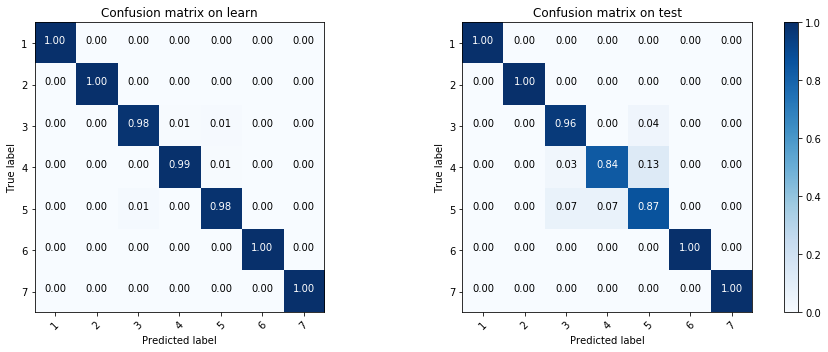
Среди всех построенных моделей была выбрана модель SGD (наискорейшего спуска).

*5.1.Исследование качества обученной модели*

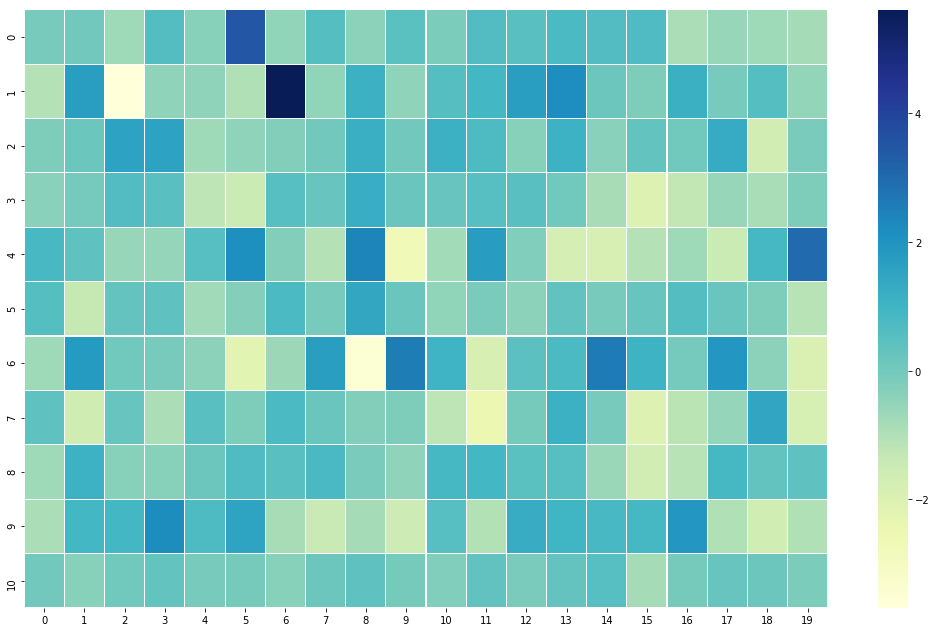
*Матрицы ошибок в абсолютных величинах на обучающей и тестовой выборке:*



*Матрицы ошибок в относительных величинах на обучающей и тестовой выборке:*



*5.2. Оценка важности признаков*



*5.3. Выводы*

По тепловой карте важности признаков можно сказать, что почти все признаки играют роль в классификации, так как синаптические веса в соответствующих строках выделены синим и желтым цветом. Менее важен, чем остальные, признак 10 (последний признак). Этот признак был добавлен в результате выделения главных компонент. В результате этого метода 5 признаков были превращены в 2. Однако по тепловой карте видно, что достаточно было получить всего один признак из 5.

Обученная по методу SGD нейросеть, действительно выполняет классификацию объектов. 4 класса из 7 распознаны со 100% результатом в обучающей и в тестовой выборке. В итоге, общая точность на обучающей выборке составила 99.41%, на тестовой – 94.97%.