МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Лабораторная работа №\_\_3\_\_**

по дисциплине«Разработка нейросетевых систем»

Тема: «Регуляризация и аугментация»

ИСПОЛНИТЕЛЬ: \_\_Журавлев Н.В.\_\_\_

ФИО

группа ИУ5-24М \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"10"\_\_03\_\_\_\_\_2024 г.

ПРЕПОДАВАТЕЛЬ: \_\_\_Канев А.И.\_\_\_\_\_\_

ФИО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

"\_\_"\_\_\_\_\_\_\_\_\_202\_ г.

Москва - 2024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Задание**

1. По заданию выбрать свои классы и обучить сверточную нейронную сеть c использованием техник дропаут и аугментации данных. Сравнить три варианта дропаута и три варианта аугментации данных.
2. Проанализировать результаты обучения в Tensorboard.
3. Проанализируйте результаты обучения вашей модели. Как изменилась точность на обучающей и тестовой выборке по сравнению со сверточной моделью?
4. Попробуйте применить регуляризацию и аугментацию к модели из 2-ой лабораторной. Возникает ли переобучение вашей модели?
5. Измените вашу модель - сравните три варианта параметра дропаута по заданию. Сравните результаты обучения этих 3-ех вариантов
6. Измените 3 варианта штрафа за сложность модели. Проанализируйте результаты обучения.
7. Измените гиперпараметры обучения для повышения точности модели: количество эпох, размер батча, скорость обучения
8. Сравните три варианта аугментации данных: с маленьким коэффициентом, средним и большим. Проанализируйте результаты обучения.
9. Укажите, какие действия помогли повысить точность вашей модели и объясните почему.

**Часть 1. Проанализируйте результаты обучения вашей модели. Как изменилась точность на обучающей и тестовой выборке по сравнению со сверточной моделью?**

Точность модели получилась 0.8067, что ниже свёрточной, которая равна 0.84, за счёт того, что в данная модель недообучена. Результат обучения модели представлен на рис. 1.

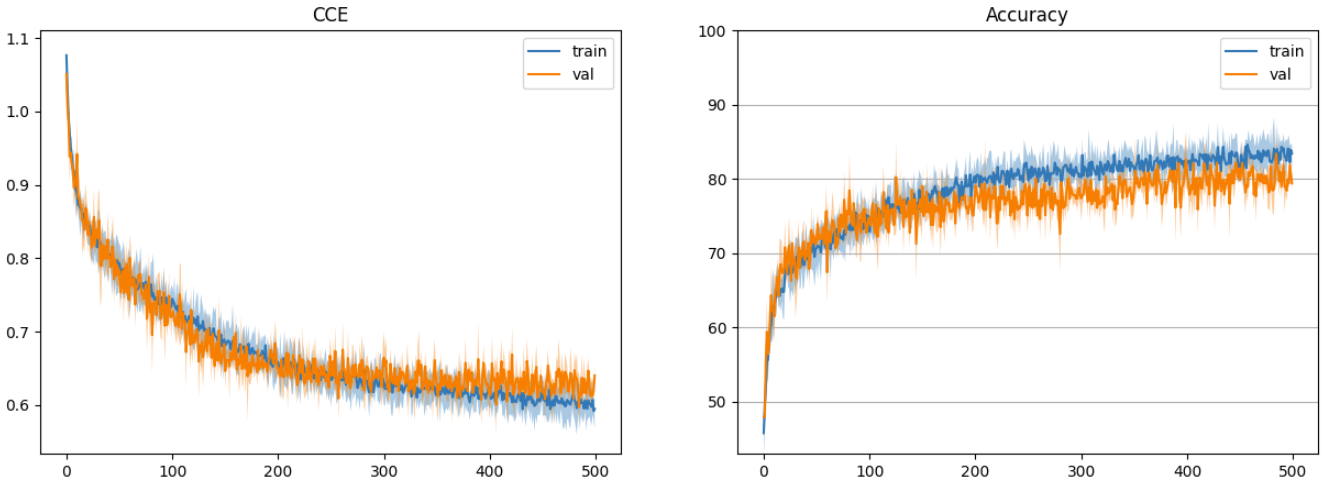


Рисунок 1 – Результат обучения модели базовой версии

Распределение точности по классам, представлена в табл. 1.

Таблица 1 - Точность для каждого классы при обучении модели базового варианта

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.873 | 0.894 | 0.8834 |
| 61 | 0.8471 | 0.82 | 0.8333 |
| 26 | 0.9206 | 0.9280 | 0.9243 |
| **Accuracy** |  | | 0.8807 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8621 | 0.75 | 0.8021 |
| 61 | 0.027 | 0.78 | 0.7393 |
| 26 | 0.8725 | 0.89 | 0.8812 |
| **Accuracy** |  | | 0.8067 |

**Часть 2. Попробуйте применить регуляризацию и аугментацию к модели из 2-ой лабораторной. Возникает ли переобучение вашей модели?**

После применения регуляризации и аугментации уменьшилась и составляет 0.8233. Переобучение не возникает, т.к. точность на тестовой выборке и на тренировочной не имеют существенную разницу, однако, возможно точность упала из-за недообучения. Результат представлен на рис. 2.

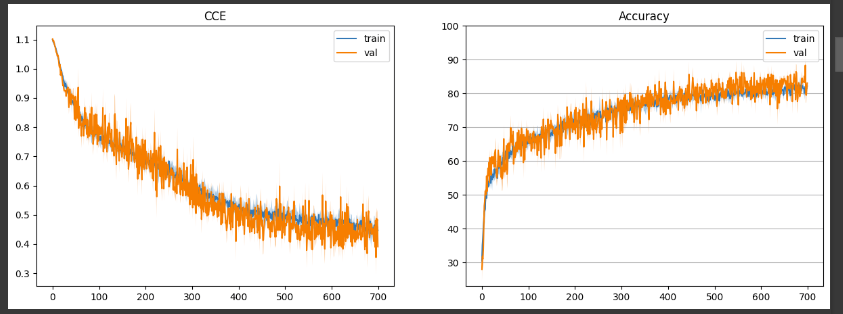


Рисунок 2 – результаты обучения модели из лаб. 2 с применение регуляризации и аугментации

Распределение точности по классам, представлена в табл. 2.

Таблица 2 - Точность для каждого классы при обучении модели из лаб.2 с применение регуляризации и аугментации

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8664 | 0.856 | 0.8612 |
| 61 | 0.7961 | 0.812 | 0.804 |
| 26 | 0.9153 | 0.908 | 0.9116 |
| **Accuracy** |  | | 0.8587 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8265 | 0.81 | 0.8182 |
| 61 | 0.7573 | 0.78 | 0.7685 |
| 26 | 0.8889 | 0.88 | 0.8844 |
| **Accuracy** |  | | 0.8233 |

**Часть 3. Измените вашу модель - сравните три варианта параметра дропаута по заданию. Сравните результаты обучения этих 3-ех вариантов**

Для второго дропаута изменим на значение 0.4; представлен на рис. 3 и точность составляет 0,80.

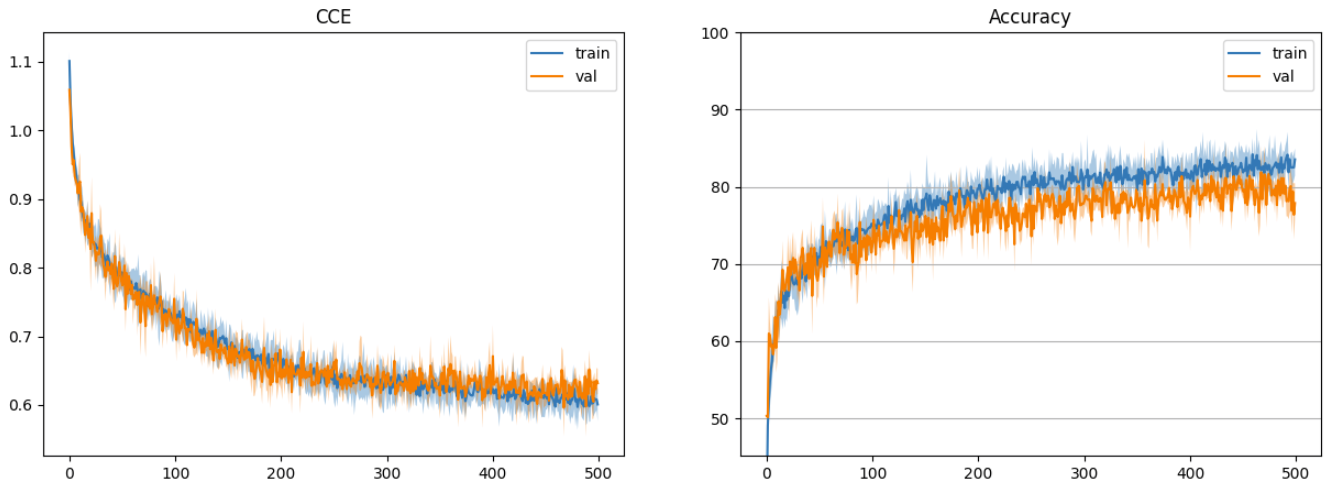


Рисунок 3 – Обучение модели, при изменении второго дропаута на 0.4

Распределение точности по классам, представлена в табл. 3.

Таблица 3 - Точность для каждого классы при обучении модели при изменении второго дропаута на 0.4

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.872 | 0.886 | 0.879 |
| 61 | 0.831 | 0.816 | 0.8234 |
| 26 | 0.9242 | 0.926 | 0.9251 |
| **Accuracy** |  | | 0.876 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8523 | 0.75 | 0.7979 |
| 61 | 0.687 | 0.79 | 0.9349 |
| 26 | 0.8866 | 0.86 | 0.8731 |
| **Accuracy** |  | | 0.80 |

Точность немного уменьшилась, т.к. такое большое значение дропаута может привести к потере слишком большого количества информации в процессе обучения.

Для первого дропаута изменим на значение 0.1; результат представлен на рис. 4 и точность составляет 0,8033.

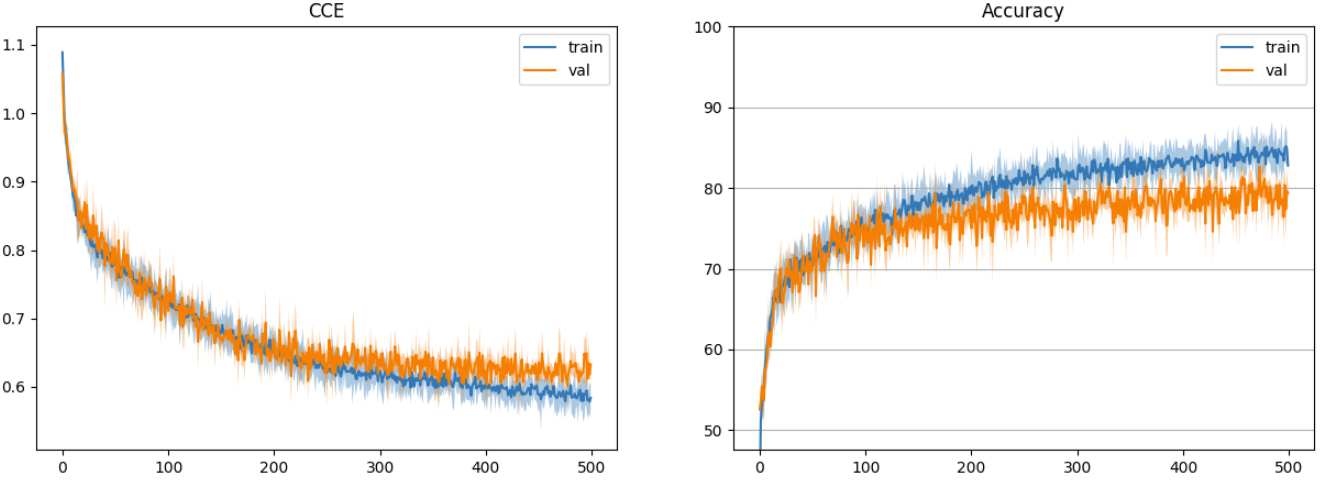


Рисунок 4 - Обучение модели, при изменении первого дропаута на 0.1

Распределение точности по классам, представлена в табл. 4.

Таблица 4 - Точность для каждого классы при обучении модели при изменении первого дропаута на 0.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8943 | 0.88 | 0.8871 |
| 61 | 0.8416 | 0.85 | 0.8458 |
| 26 | 0.9245 | 0.93 | 0.9272 |
| **Accuracy** |  | | 0.8867 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8824 | 0.75 | 0.8108 |
| 61 | 0.6957 | 0.8 | 0.7442 |
| 26 | 0.86 | 0.86 | 0.86 |
| **Accuracy** |  | | 0.8033 |

Точность незначительно уменьшилась, т.к. возникло переобучение.

Для первого дропаута изменим на значение 0.1, а для второго на 0.4; результат представлен на рис. 5 и точность составляет 0.8067.

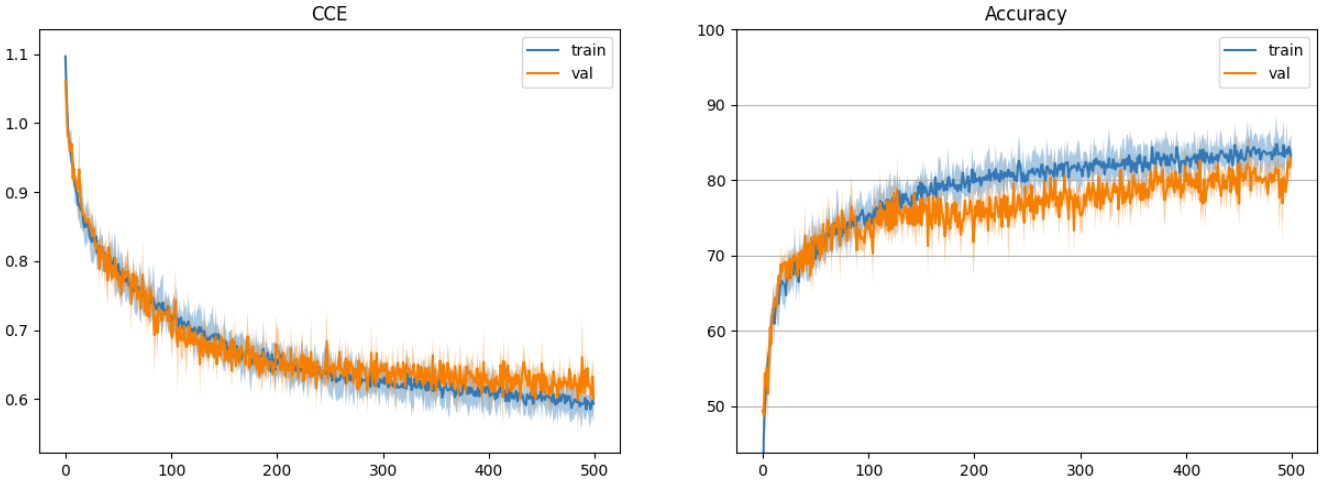


Рисунок 5 - Обучение модели, при изменении первого дропаута на 0.1 и второго на 0.4

Распределение точности по классам, представлена в табл. 5.

Таблица 5 - Точность для каждого классы при обучении модели при изменении первого дропаута на 0.1 и второго на 0.4

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8927 | 0.882 | 0.8873 |
| 61 | 0839 | 0.848 | 0.8429 |
| 26 | 0.928 | 0.928 | 0.928 |
| **Accuracy** |  | | 0.886 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8736 | 0.76 | 0.8128 |
| 61 | 0.7018 | 0.8 | 0.7477 |
| 26 | 0.8687 | 0.86 | 0.8643 |
| **Accuracy** |  | | 0.8067 |

Точность не изменилась, однако по графику видно, что в среднем стало хуже, т.к. выбрали два самый крайних случая, которые оба делают модель менее точной.

**Часть 4. Измените 3 варианта штрафа за сложность модели. Проанализируйте результаты обучения.**

Возьмём 3 значения штрафа: 1e-3, 1e-1, 1e-7.

Для значения 1e-3 результат представлен на рис. 6 и точность составляет 0.79.

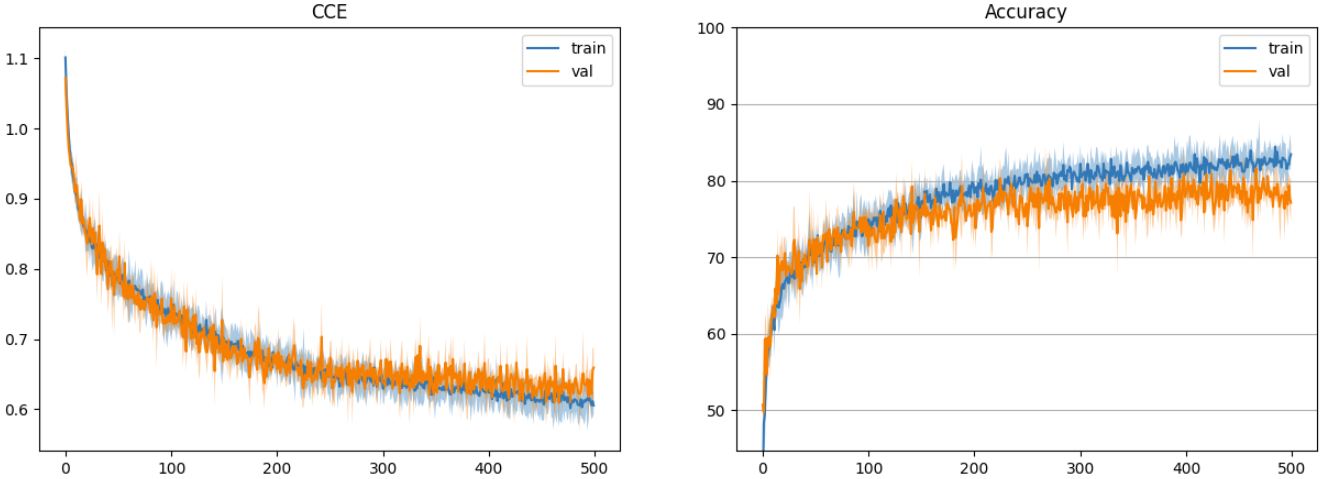


Рисунок 6 - Обучение модели, при штрафе 1е-3

Распределение точности по классам, представлена в табл. 6.

Таблица 6 - Точность для каждого классы при обучении модели при штрафе 1е-3

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8798 | 0.878 | 0.8789 |
| 61 | 0.8277 | 0.826 | 0.8268 |
| 26 | 0.9163 | 0.92 | 0.9182 |
| **Accuracy** |  | | 0.8746 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8554 | 0.71 | 0.776 |
| 61 | 0.675 | 0.81 | 0.7364 |
| 26 | 0.8763 | 0.85 | 0.8329 |
| **Accuracy** |  | | 0.79 |

Точность уменьшилась, т.к. возникло переобучение.

Для значения 1e-1 результат представлен на рис. 7 и точность составляет 0.65.

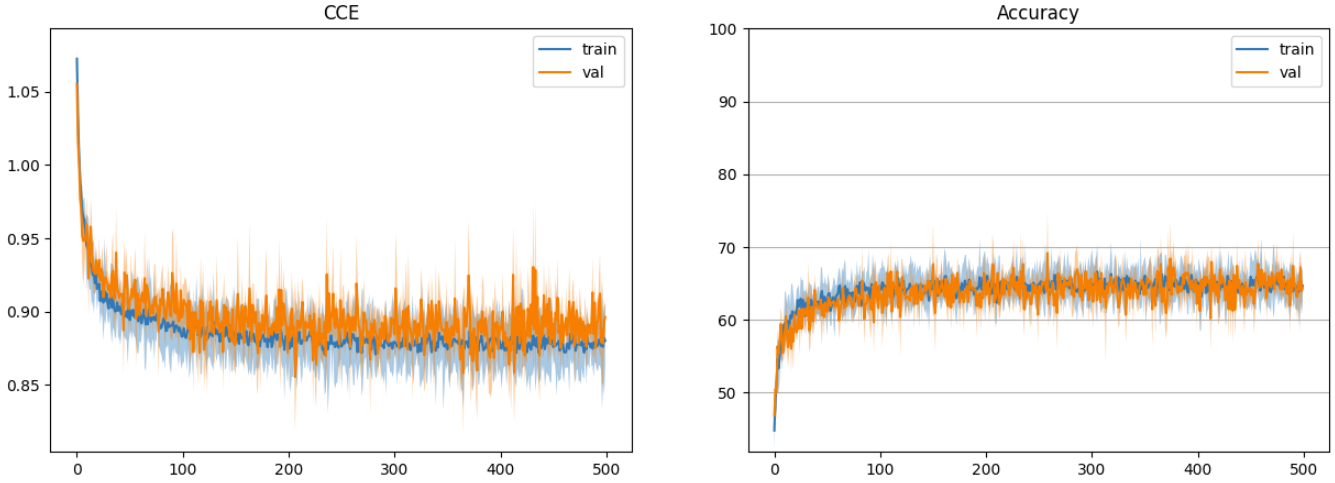


Рисунок 7 - Обучение модели, при штрафе 1е-1

Распределение точности по классам, представлена в табл. 7.

Таблица 7 - Точность для каждого классы при обучении модели при штрафе 1е-1

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.6364 | 0.7 | 0.6667 |
| 61 | 0.6119 | 0.536 | 0.5714 |
| 26 | 0.7539 | 0.772 | 0.7628 |
| **Accuracy** |  | | 0.6693 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.6404 | 0.73 | 0.6822 |
| 61 | 0.6145 | 0.51 | 0.5574 |
| 26 | 0.6893 | 0.71 | 0.6995 |
| **Accuracy** |  | | 0.65 |

Точность уменьшилась, т.к. возникло недобучение.

Для значения 1e-7 результат представлен на рис. 8 и точность составляет 0.7967.

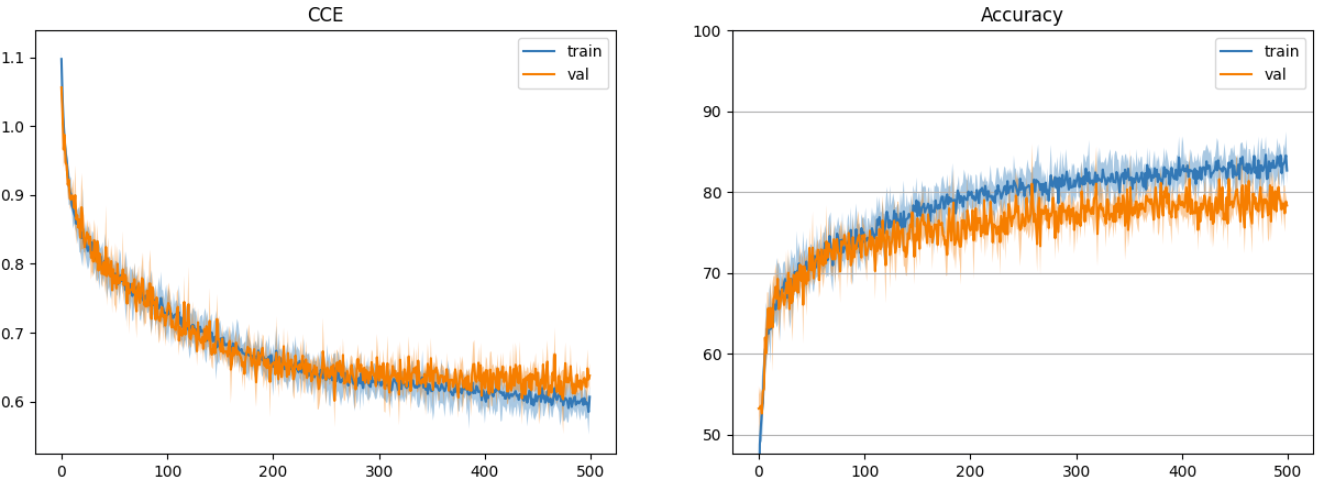


Рисунок 8 - Обучение модели, при штрафе 1е-7

Распределение точности по классам, представлена в табл. 8.

Таблица 8 - Точность для каждого классы при обучении модели при штрафе 1е-7

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8887 | 0.862 | 0.8751 |
| 61 | 0.8166 | 0.828 | 0.8222 |
| 26 | 0.9075 | 0.922 | 0.9147 |
| **Accuracy** |  | | 0.8707 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.878 | 0.72 | 0.7912 |
| 61 | 0.681 | 0.79 | 0.7315 |
| 26 | 0.8627 | 0.88 | 0.8713 |
| **Accuracy** |  | | 0.7967 |

Точность уменьшилась, т.к. из-за уменьшения штрафа уменьшается обобщающая способность модели.

**Часть 5. Измените гиперпараметры обучения для повышения точности модели: количество эпох, размер батча, скорость обучения**

После изменения гиперпараметров был выявлен наиболее лучший вариант с точностью 0.86, что представлено на рис. 9.

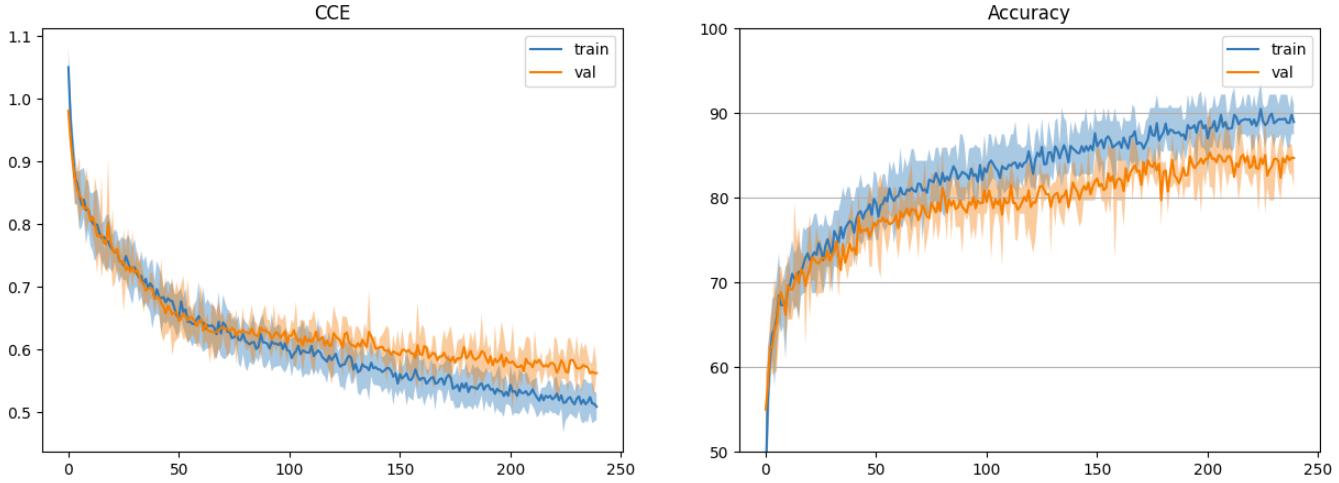


Рисунок 9 – Обучение модели с оптимальными параметрами

Распределение точности по классам, представлена в табл. 9.

Таблица 9 - Точность для каждого классы при обучении модели с оптимальными параметрами

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.9738 | 0.968 | 0.9709 |
| 61 | 0.9556 | 0.946 | 0.9508 |
| 26 | 0.9965 | 0.982 | 0.9742 |
| **Accuracy** |  | | 0.953 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8913 | 0.82 | 0.8542 |
| 61 | 0.81 | 0.81 | 0.81 |
| 26 | 0.8796 | 0.95 | 0.9135 |
| **Accuracy** |  | | 0.86 |

**Часть 6. Сравните три варианта аугментации данных: с маленьким коэффициентом, средним и большим. Проанализируйте результаты обучения.**

Возьмём 3 значения аугментации: 0.3, 0.1, 0.7.

Для значения 0.3 результат представлен на рис. 10 и точность составляет 0.85.

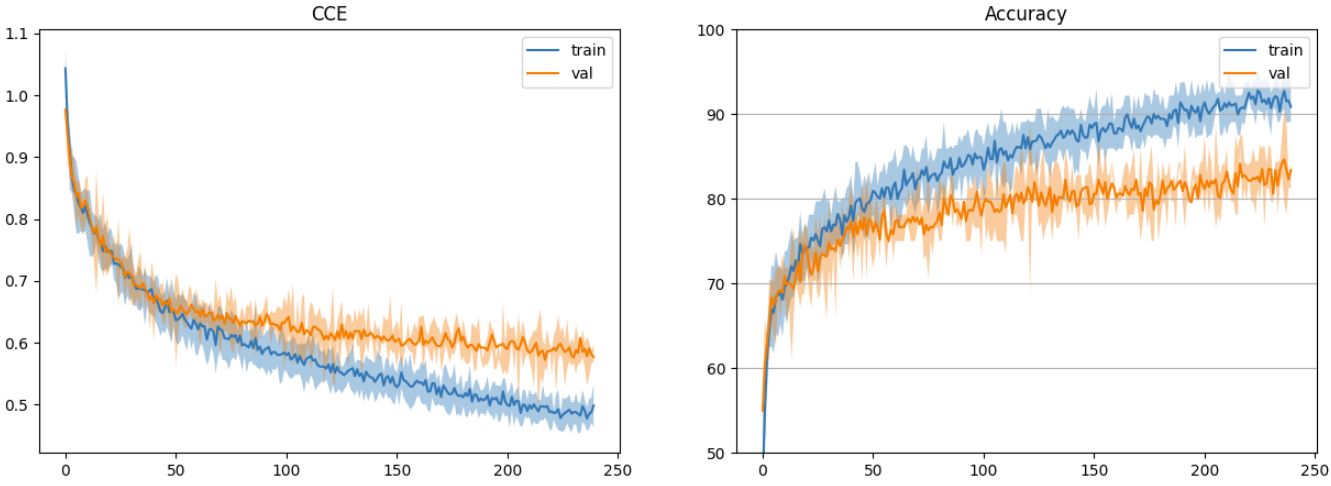


Рисунок 10 - Обучение модели, при p=0.3

Распределение точности по классам, представлена в табл. 10.

Таблица 10 - Точность для каждого классы при обучении модели при p=0.3

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.9725 | 0.99 | 0.9812 |
| 61 | 0.9854 | 0.948 | 0.9664 |
| 26 | 0.9725 | 0.992 | 0.9822 |
| **Accuracy** |  | | 0.9767 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8333 | 0.85 | 0.8416 |
| 61 | 0.8427 | 0.75 | 0.7937 |
| 26 | 0.8716 | 0.95 | 0.9091 |
| **Accuracy** |  | | 0.85 |

Точность уменьшилась, т.к. уменьшилось количество поступающих данных для обучения.

Для значения 0.1 результат представлен на рис. 11 и точность составляет 0.8233.

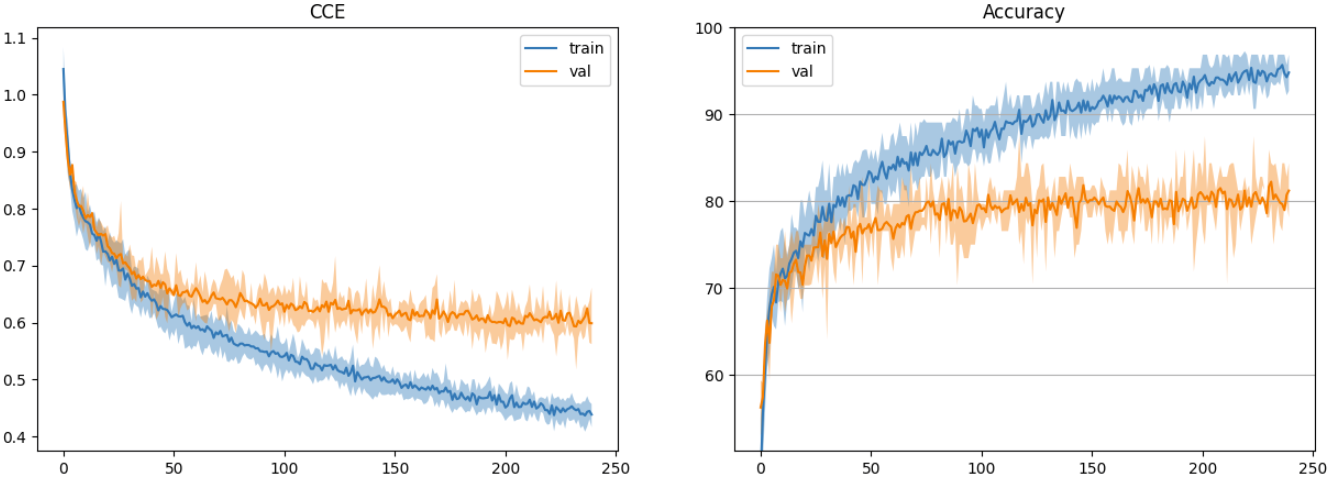


Рисунок 11 - Обучение модели, при p=0.1

Распределение точности по классам, представлена в табл. 11.

Таблица 11 - Точность для каждого классы при обучении модели при p=0.1

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.996 | 0.998 | 0.997 |
| 61 | 0.9959 | 0.982 | 0.9889 |
| 26 | 0.9842 | 0.996 | 0.901 |
| **Accuracy** |  | | 0.992 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.8602 | 0.8 | 0.829 |
| 61 | 0.7624 | 0.77 | 0.7662 |
| 26 | 0.8491 | 0.9 | 0.8738 |
| **Accuracy** |  | | 0.8233 |

Точность уменьшилась, т.к. уменьшилось количество поступающих данных для обучения.

Для значения 0.7 результат представлен на рис. 12 и точность составляет 0.8367.

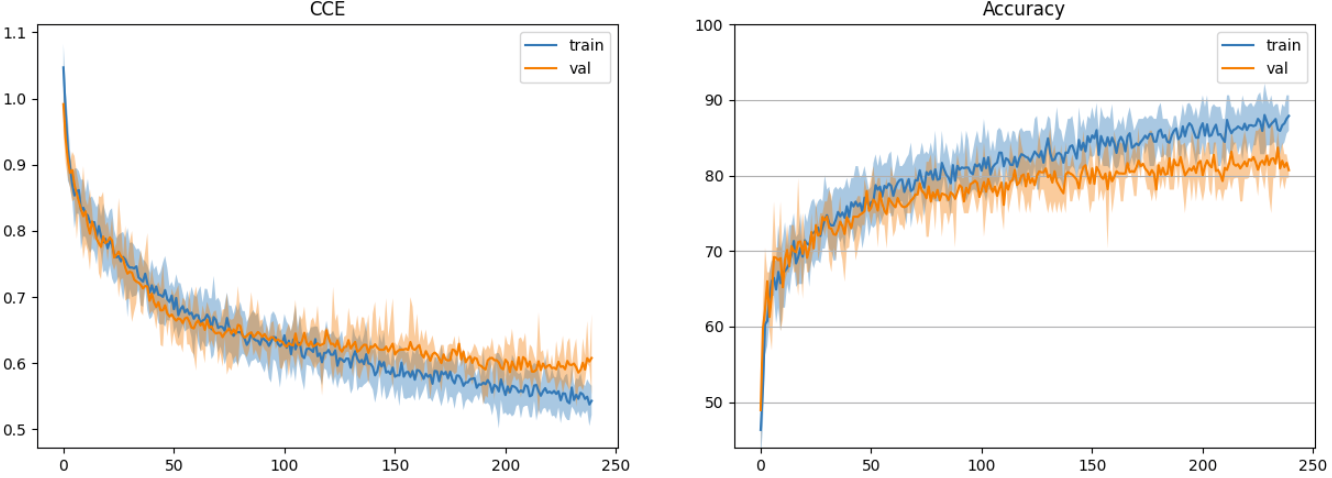


Рисунок 12- Обучение модели, при p=0.7

Распределение точности по классам, представлена в табл. 12.

Таблица 12 - Точность для каждого классы при обучении модели при p=0.7

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.9649 | 0.936 | 0.9503 |
| 61 | 0.9206 | 0.928 | 0.9243 |
| 26 | 0.9472 | 0.968 | 0.9575 |
| **Accuracy** |  | | 0.944 |
| **Train** |  | | |
| **Классы** | **Precision** | **Recall** | **F1-score** |
| 19 | 0.907 | 0.78 | 0.8387 |
| 61 | 0.97664 | 0.82 | 0.7923 |
| 26 | 0.8505 | 0.91 | 0.8792 |
| **Accuracy** |  | | 0.8367 |

Точность уменьшилась, т.к. увеличилась сложность модели.

**Часть 7. Укажите, какие действия помогли повысить точность вашей модели и объясните почему.**

Повысить точность позволили следующие действия:

* Изменение гиперпараметров
* Увеличение аугментаци

**Итоговая таблица с результатами для всех вариантов обучения**

На табл. 13 представлены результаты обучения.

Таблица 13 - Итоговая таблица с результатами для всех вариантов обучения

|  |  |  |  |
| --- | --- | --- | --- |
| **Конфигурация нейросети** | **Гиперпараметры** | **Точность** | **Комментарий** |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.003  batch\_size = 128  epoch = 500  weight\_decay=1e-5  p=0.5 | train = 88,07%  test= 80,67% | Базовый вариант |
| CL(32, 5, stride=1, padding=2),  MaxPool(4),  Dropout(0.2),  CL(64, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  CL(64, 3, stride=1, padding=1),  FC(256) | lr=0.001  batch\_size = 256  epoch = 700  weight\_decay=1e-5  p=0.5 | train = 85,67%  test= 82,33% | Применение регуляризации и аугментации к модели из лаб.2 |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.4),  FC(512) | lr=0.003  batch\_size = 128  epoch = 500  weight\_decay=1e-5  p=0.5 | train = 87,90%  test= 80% | Изменение значение второго дропаута на 0.4 |
| CL(64, 3, stride=4, padding=0),  Dropout(0.1),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.003  batch\_size = 128  epoch = 500  weight\_decay=1e-5  p=0.5 | train = 88,67%  test= 80,33% | Изменение значение первого дропаута на 0.1 |

Продолжение таблицы 13

|  |  |  |  |
| --- | --- | --- | --- |
| **Конфигурация нейросети** | **Гиперпараметры** | **Точность** | **Комментарий** |
| CL(64, 3, stride=4, padding=0),  Dropout(0.1),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.4),  FC(512) | lr=0.003  batch\_size = 128  epoch = 500  weight\_decay=1e-5  p=0.5 | train = 88,60%  test= 80,67% | Изменение значение дропаута на 0.1 и 0.4 |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.003  batch\_size = 128  epoch = 500  weight\_decay=1e-3  p=0.5 | train = 87,47%  test= 79,0% | Изменение значение штрафа за сложность на 1e-3 |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.003  batch\_size = 128  epoch = 500  weight\_decay=1e-1  p=0.5 | train = 66,93%  test= 65,00% | Изменение значение штрафа за сложность на 1e-1 |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.003  batch\_size = 128  epoch = 500  weight\_decay=1e-7  p=0.5 | train = 87,07%  test= 79,67% | Изменение значение штрафа за сложность на 1e-7 |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.006  batch\_size = 64  epoch = 250  weight\_decay=1e-5  p=0.5 | train = 96,53%  test= 86,00% | Изменение гиперпараметров |

Продолжение таблицы 13

|  |  |  |  |
| --- | --- | --- | --- |
| **Конфигурация нейросети** | **Гиперпараметры** | **Точность** | **Комментарий** |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.006  batch\_size = 64  epoch = 250  weight\_decay=1e-5  p=0.3 | train = 97,67%  test= 85,00% | Коэффициент аугментации p=0.3 |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.006  batch\_size = 64  epoch = 250  weight\_decay=1e-5  p=0.1 | train = 99,2%  test= 82,33% | Коэффициент аугментации p=0.1 |
| CL(64, 3, stride=4, padding=0),  Dropout(0.2),  CL(128, 3, stride=1, padding=1),  AvgPool(4),  Dropout(0.3),  FC(512) | lr=0.006  batch\_size = 64  epoch = 250  weight\_decay=1e-5  p=0.7 | train = 94,40%  test= 83,67% | Коэффициент аугментации p=0.7 |

**Вывод**

В теории при значениях дропаута от 0.1 до 0.4 точность будет улучшаться, относительного вариантов со значениями 0 и 1, так как при больших значениях точность падает из-за потери слишком большого количества информации в процессе обучения, а при маленьких из-за вероятности возникновения переобучения. В текущей лабораторной работе теория подтверждается.

В теории для значений штрафа за сложность модели точность будет улучшаться в том случае, если значение является средним, то есть не слишком большим или не слишком маленьким. При больших значениях точность падает из-за вероятности возникновения недообучения, а при маленьких из-за вероятности возникновения переобучения. В текущей лабораторной работе теория подтверждается.

В теории при значениях коэффициента аугментации будет улучшаться в том случае, если значение является средним, то есть не слишком большим или не слишком маленьким. При больших значениях точность падает из-за создаваемого в результате этого шума, а при маленьких из-за вероятности возникновения переобучения. В текущей лабораторной работе теория подтверждается.