|  |  |
| --- | --- |
| **Изображение выглядит как эмблема, герб, символ, нашивка  Контент, сгенерированный ИИ, может содержать ошибки.** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное автономное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления

**Лабораторная работа №4**

**По курсу**

«**Разработка нейронных сетей**»

**По теме «Введение в DL»**

Выполнил:  
Студент группы ИУ5-22МБибиков П.А.  
24.04.2025

Проверил:

**Канев А.И.**

2025 г.

# Задание

1. Проведите обучение модели по вашему варианту с наилучшими параметрами аугментации из предыдущей лабораторной.

2. Проанализируйте результаты обучения вашей модели. Как изменилась точность на обучающей и тестовой выборке по сравнению с предыдущими моделями?

3. Сравните обучение модели с заморозкой и без заморозки весов.

4. Измените гиперпараметры обучения для повышения точности модели: количество эпох, размер батча, скорость обучения

5. Укажите, какие действия помогли повысить точность вашей модели и объясните почему.

Вы должны использовать следующие классы из CIFAR100:

1. Номер группы + 15
2. Номер варианта + 56
3. ИУ5 (Номер варианта + 21); ГУИМЦ (80); Иностранцы (90)

Полученные варианты:

1. 22 + 15 = 37
2. 3 + 56 = 59
3. 3 + 21 = 24

**Часть 1.  Обучение исходной нейронной сети.**

Возьмем НС resnet20 в качестве исходной. Так как она обучена на 100 классов из CIFAR 100, разморозим последний полносвязный слой и укажем, что необходимо три выхода под наши требуемые классы. Ниже на рисунке 1 представлена, конечная часть конфигурации сети.

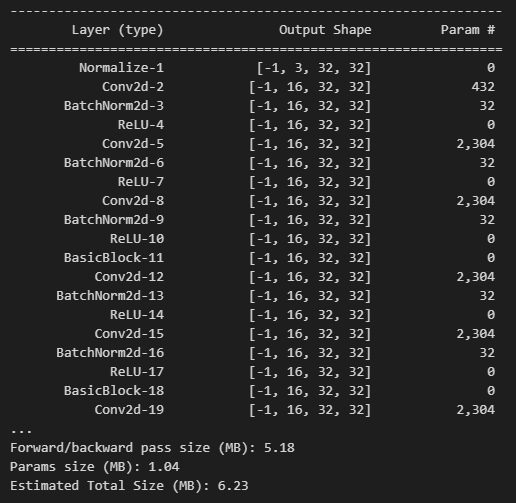


Рисунок 1 - Конфигурация сети с изменениями в полносвязном слое.

Исходная модель предназначена для классификации 100 классов, поэтому имеет 100 выходных. В рамках выполнения лабораторных работ, нам необходимо, чтобы на выходе у модели было 3 нейрона. Для этого разморозим последний полносвязный слой, и вместо него зададим новый – такой, у которого будет 3 выхода.

Точность модели на тестовой выборке составила 98 %, что лучше результатов, полученных в лабораторных работах ранее. Это связано с тем, что исходная модель содержит большее количество слоёв, обучена и создана специалистами в области нейронных сетей.

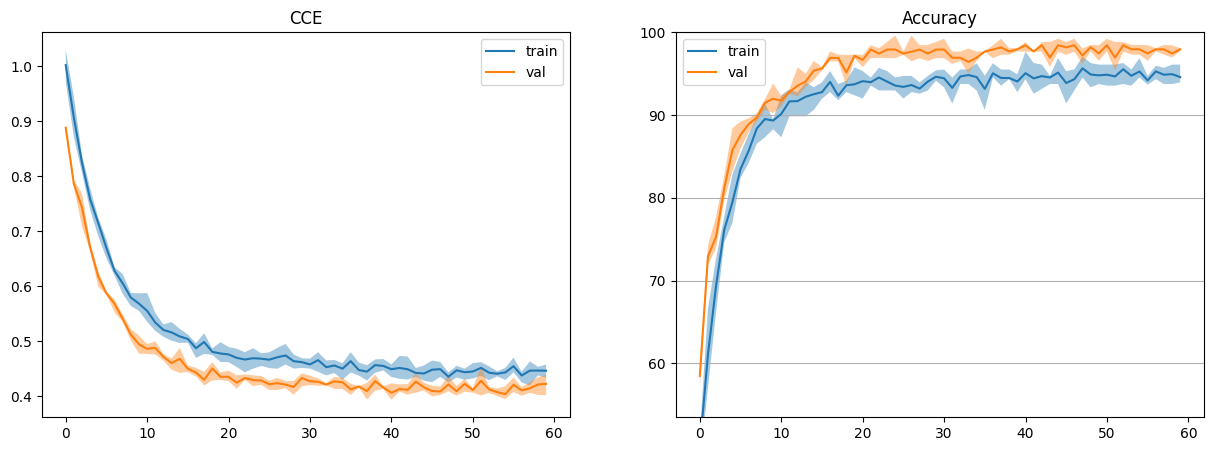


Рисунок 2. Результаты работы исходной модели.

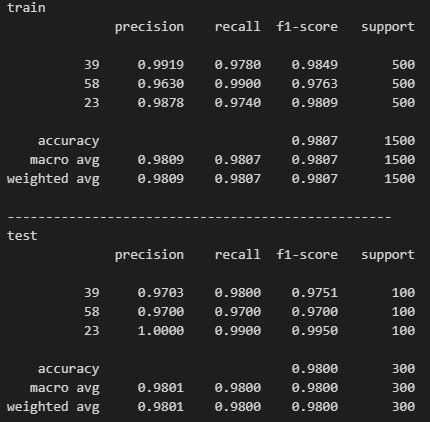


Рисунок 3 – Результаты.

Попробуем разморозить больше параметров для изменения, поставим значение параметра keep\_last = 5. Предполагаем, что разморозка улучшит точность модели, так как модель станет более «специфично» обучена под наши классы изображений. Точность модели на тестовой выборке понизилась до 95,7%.

Теперь обучим модель, используя метод Fine Tuning. Обучать будем сначала на 30 эпохах с заморозкой (keep\_last = 2), а затем разморозим модель и дообучим еще 30 эпох (keep\_last = 65). Точность на тестовой выборке составила 96,3%, на обучающей – 98,5%.

Перейдем к полной разморозке весов и выставим значение keep\_last = 65. Точность практически не изменилась, по сравнению с Fine Tuning.

Для повышения точности модели попробуем увеличить размер батча и количество эпох в два раза. Точность модели слегка увеличилась и составляет 98%, при этом затрачивается больше времени на обучение модели.

Результаты лабораторной работы приведены в таблице 2.

Таблица 1. Результаты.

| **Конфигурация сети** | **Гиперпараметры** | **Точность** | **Комментарии** |
| --- | --- | --- | --- |
| Conv2d(3, 64, kernel\_size=(5, 5), stride=(1, 1), padding=(2, 2))  MaxPool2d(kernel\_size=4, stride=4, padding=0, dilation=1, ceil\_mode=False)  Dropout2d(p=0.2, inplace=False)  Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))  MaxPool2d (kernel\_size=4, stride=4, padding=0)  Dropout2d(p=0.3, inplace=False) | lr = 0.001, batch\_size = 128, epochs = 750,  momentum = 0.9,  weight\_decay = 1e-3, degrees= 5, translate=(0.1, 0.1), scale=(0.9, 1.1),shear=3 | test = 93.3%,  train = 98.5% | Вариант из ЛР3 |
| Resnet  Keep\_last = 2 | lr = 0.0003,  batch\_size = 128,  epochs = 60  weight\_decay=1e-5  degrees= 25, translate=(0.2, 0.2), scale=(0.8, 1.2), shear=7 | test = 98%,  train = 98% | Базовый вариант |
| Resnet  Keep\_last = 5 | lr = 0.0003,  batch\_size = 128,  epochs = 60  weight\_decay=1e-5  degrees= 25, translate=(0.2, 0.2), scale=(0.8, 1.2), shear=7 | test = 95,7%,  train = 97,1% | Разморозили больше слоёв |
| Resnet  keep\_last = 2 (30 эпох)  keep\_last = 65 (30 эпох) | lr = 0.0003,  batch\_size = 128,  epochs = 60  weight\_decay=1e-5  degrees= 25, translate=(0.2, 0.2), scale=(0.8, 1.2), shear=7 | test = 96,3%, train = 98,5% | Применили метод Fine Tuning |
| Resnet  Keep\_last = 65 | lr = 0.0003,  batch\_size = 128,  epochs = 60  weight\_decay=1e-5  degrees= 25, translate=(0.2, 0.2), scale=(0.8, 1.2), shear=7 | test = 97,3%,  train = 98,9% | Разморозили всю модель |
| Resnet  Keep\_last = 65 | lr = 0.0003,  batch\_size = 256,  epochs = 120  weight\_decay=1e-5  degrees= 25, translate=(0.2, 0.2), scale=(0.8, 1.2), shear=7 | test = 98%, train = 99,1% | Увеличили размер батча и количество эпох в 2 раза |
| Resnet  keep\_last = 2 (10 эпох)  keep\_last = 65 (50 эпох) | lr = 0.0003,  batch\_size = 128,  epochs = 60  weight\_decay=1e-5  degrees= 25, translate=(0.2, 0.2), scale=(0.8, 1.2), shear=7 | test = 96,3%,  train = 98,4% | 10 с заморозкой, 50 без заморозки |

# Вывод

В ходе выполнения лабораторной работы, было подтверждено, что перенос обучения может значительно улучшить классификацию изображений, когда предобученная модель имеет высокую точность на задачах подобного рода.

Была проведена разморозка разного количества весов предобученной модели. Этот процесс позволил продолжить обучение модели для классификации изображений по трем классам с сохранением информации, полученной при предобучении. В результате разморозки весов удалось добиться повышения точности при распознавании каждого из классов по варианту.

Использование предобученной модели позволило нам сэкономить время и ресурсы, необходимые для обучения модели с нуля, значительно повысив, при этом, качество решения задачи машинного обучения. На практике мы убедились, что модель с разморозкой работает лучше, так как все ее параметры участвуют в обучении и модель может более эффективно использовать информацию из предобученных параметров.

Используя Fine Tuning, мы смогли получить результаты лучше, чем при обучении с заморозкой, но хуже, чем при обучении с разморозкой, однако, мы потратили сравнимо похожее время на обучение модели, как в случае с заморозкой.