**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра вычислительной техники**

Ы

**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «Нейронные сети»**

**Тема: Подбор и анализ медицинских наборов данных**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 8308 |  | Сергеев В.С. |
| Преподаватель |  | Беляев П.Ю. |

Санкт-Петербург

2023

Оглавление

[1. Цель работы 3](#_Toc129211692)

[2. Ход работы 4](#_Toc129211693)

[3. Вывод 12](#_Toc129211694)

# Цель работы

Задание 1: Подготовка данных

В задании необходимо продемонстрировать методы подготовки и предварительной обработки данных, таких как увеличение и нормализация данных.

В рамках лабораторной работы необходимо самостоятельно собрать набор данных. В качестве набора данных может быть выбран любой объект. Для расширения набора данных – можете найти существующий, но большую часть должен содержать уникальные изображения.

Задание 2: Архитектура CNN и трансферное обучение

В этом задании необходимо написать и уметь объяснить как работает разработанная ими архитектура. Что поможет в этом: изучение различных слоенв, используемых в CNN, включая сверточные слои, объединяющие слои, функции активации, полносвязные слои и отсев.

Разработка собственной архитектуры (и последующее обучение с 0) – 3-5 различных архитектур с разной комбинацией слоев/функций активации;

Задание 3: Обучение CNN

В данном задании студентам необходимо продемонстрировать знания, как обучать CNN, используя функции потерь, оптимизаторы и методы регуляризации. Студенты также должны понимать, как предотвратить переобучение и недообучение в CNN.

Задание 4: Настройка гиперпараметров и выбор модели

В этом задании студенты должны объяснить, как настраивать гиперпараметры в CNN и как выполнять выбор модели.

# Ход работы

Задание 1.

Для выполнения лабораторной работы был собран набор данных из 1547 фотографий котов, содержащий следующие классы:

1. кот Ластик (lastik, 773 фотографии);
2. прочие коты (other, 774 фотографии).

Данные для первого класса собраны самостоятельно из 200 фотографий и 500 кадров, вырезанных из 5 видеозаписей. Данные для второго класса взяты из датасета Cat Dataset [1] с сайта Kaggle.

При загрузке к изображениям применятся трансформации: размер изображения подгоняется под 256 на 256 пикселей, применяется случайное аффинное преобразование с поворотом изображения до 90 градусов, изображение переводится в тензор, после чего нормализуется. Применяемые трансформации представленные на рисунке 1.

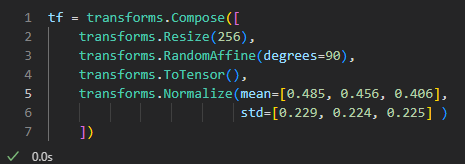


Рисунок 1. Применяемые трансформации

Данные загружаются и разбиваются на тренировочную (70%) и тестовую (30%) выборки.

Задание 2.

В ходе работы созданы 4 CNN архитектуры. В качестве функции активации во всех сетях используется ReLU.

Архитектура Net\_1 представлена на рисунке 2.

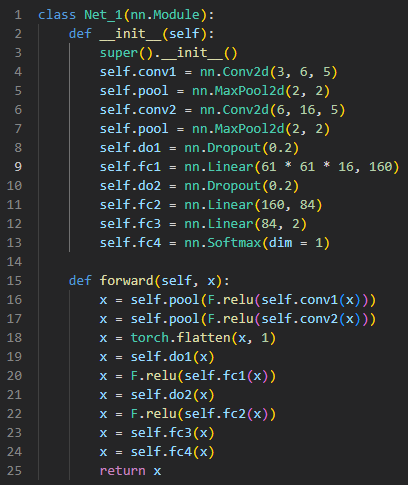


Рисунок 2. Архитектура Net\_1

Сеть состоит из свёртки 5х5 с выходом на 6 каналов, слоя пулинга 2х2, второй свёртки 5х5 с выходом на 16 каналов, слоя пулинга 2х2, за которым идут три линейный слоя. В конце применяется SoftMax.

Архитектура Net\_2 представлена на рисунке 3.

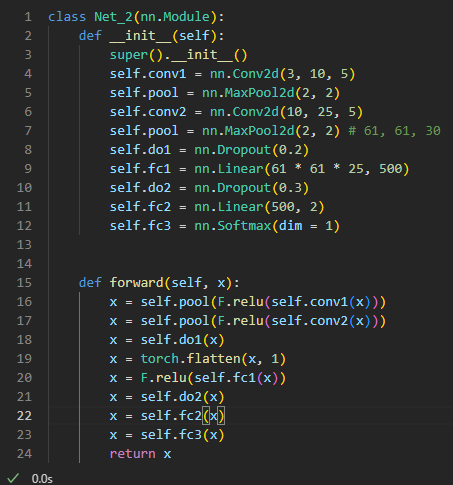


Рисунок 3. Архитектура Net\_1

Сеть состоит из свёртки 5х5 с выходом на 10 каналов, слоя пулинга 2х2, второй свёртки 5х5 с выходом на 25 каналов, слоя пулинга 2х2, за которым идут два линейный слоя. В конце применяется SoftMax.

Архитектура Net\_3 представлена на рисунке 4.

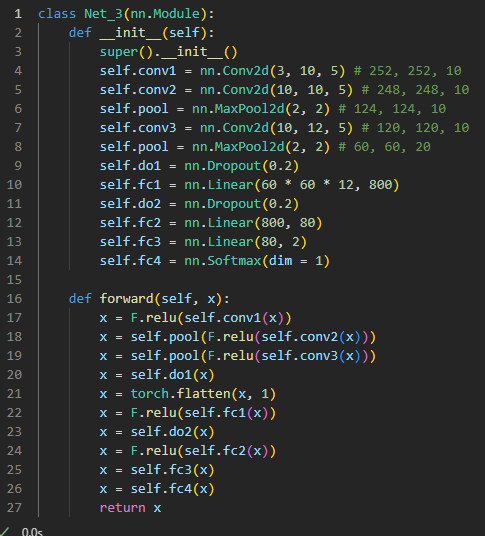


Рисунок 4. Архитектура Net\_3

Сеть состоит из свёртки 5х5 с выходом на 10 каналов, свёртки 5х5 с выходом на 10 каналов, слоя пулинга 2х2, третьей свёртки 5х5 с выходом на 12 каналов, слоя пулинга 2х2, за которым идут три линейный слоя. В конце применяется SoftMax.

Архитектура Net\_4 представлена на рисунке 5.

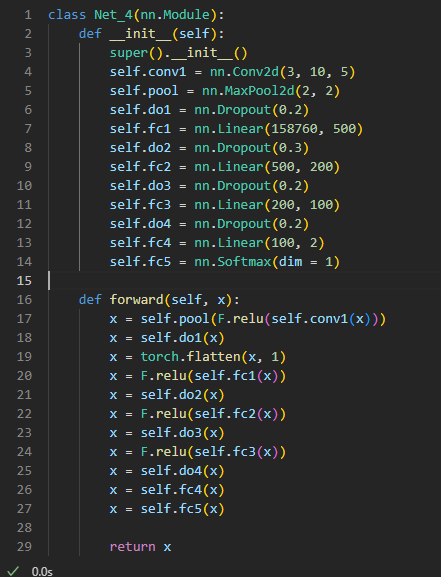


Рисунок 5. Архитектура Net\_4

Сеть состоит из свёртки 5х5 с выходом на 10 каналов, слоя пулинга 2х2, и трёх линейных слоёв. В конце применяется SoftMax.

Задание 3 и 4.

Для обучения CNN в качестве функции потерь используется CrossEntropyLoss. Используется оптимизатор сигмоиды, коэффициент скорости обучения равен 0.001, импульс равен 0.9. Был опробован коэффициент скорости обучения равный 0.01 и 0.0001, но в обоих случаях это приводило к падению точности сети на 5-15%.

Для регуляризации после свёрток и между линейными слоями были добавлены дропауты, исключающие часть связей между нейронами из сети. В ходе экспериментов, изначально отбрасывались 20% связей, но для наиболее склонных к переобучению сети 3 процент между некоторыми слоями был повышен до 30.

Размер батча при обучении равен 16. Также в ходе экспериментов было замечено, что большие (16 или 32 против 4 или 8) размеры батчей положительно влияют на генерализацию сети. Согласно наблюдениям, ситуации, когда сеть значительно (более чем на 40%) лучше предсказывает один из классов, чем другой, встречаются реже при большем размере батча. Однако обучение при размере батча больше или равном 32 иногда оканчивалось внештатно из-за внештатного завершения процессов, подгружающих данные.

Далее приведёны результаты обучения моделей. В них включены изменение значений функции потерь в процессе тренировки, точность на тестовом датасете, а также точность для каждого из классов. Чтобы было возможно оценить размер модели, также подсчитано количество весов в модели.

---- net 0 ----

<class '\_\_main\_\_.Net\_1'>

[1, 30] loss: 2.075

[1, 60] loss: 2.062

[2, 30] loss: 2.026

[2, 60] loss: 1.985

[3, 30] loss: 1.881

[3, 60] loss: 1.938

[4, 30] loss: 1.843

[4, 60] loss: 1.776

Finished Training

./cats\_net\_0.pth

Accuracy of the network on the 464 test images: 69 %

Accuracy for class: lastik is 77.1 %

Accuracy for class: other is 61.4 %

Model size: 9542486 parametres

---- net 1 ----

<class '\_\_main\_\_.Net\_2'>

[1, 30] loss: 2.040

[1, 60] loss: 1.898

[2, 30] loss: 1.737

[2, 60] loss: 1.650

[3, 30] loss: 1.551

[3, 60] loss: 1.658

[4, 30] loss: 1.615

[4, 60] loss: 1.452

Finished Training

./cats\_net\_1.pth

Accuracy of the network on the 464 test images: 78 %

Accuracy for class: lastik is 75.3 %

Accuracy for class: other is 79.8 %

Model size: 46521037 parametres

---- net 2 ----

<class '\_\_main\_\_.Net\_3'>

[1, 30] loss: 2.079

[1, 60] loss: 2.077

[2, 30] loss: 2.069

[2, 60] loss: 2.055

[3, 30] loss: 2.018

[3, 60] loss: 1.930

[4, 30] loss: 1.852

[4, 60] loss: 1.834

Finished Training

./cats\_net\_2.pth

Accuracy of the network on the 464 test images: 67 %

Accuracy for class: lastik is 65.4 %

Accuracy for class: other is 71.2 %

Model size: 34631324 parametres

---- net 3 ----

<class '\_\_main\_\_.Net\_4'>

[1, 30] loss: 2.080

[1, 60] loss: 2.049

[2, 30] loss: 2.012

[2, 60] loss: 1.983

[3, 30] loss: 1.896

[3, 60] loss: 1.878

[4, 30] loss: 1.778

[4, 60] loss: 1.771

Finished Training

./cats\_net\_3.pth

Accuracy of the network on the 464 test images: 71 %

Accuracy for class: lastik is 74.5 %

Accuracy for class: other is 67.4 %

Model size: 79501762 parametres

Таблица 1. Точность моделей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модель | Net\_1 | Net\_2 | Net\_3 | Net\_4 |
| Точность модели | 69% | 78% | 67% | 71% |
| Точность по классу 1 | 77.1% | 75.3% | 65.4% | 74.5% |
| Точность по классу 2 | 61.4% | 79.8% | 71.2% | 67.4% |

На рисунке 6 представлен график изменения функции потерь моделей при обучении.

Рисунок 6. Изменение функции потерь при обучении

В итоговой версии общая точность каждой из моделей составила более 65%. В процессе экспериментов с разными сидами для разбиения данных и обучения были встречены модели, показывающие большую точность на тестовых данных, однако при дальнейшей проверке на отдельных примерах те сети показывали худшие результаты, скорее всего, из-за переобучения.

Модели были сохранены на диск и вновь загружены с него.

Далее произведено тестирование моделей на конкретном примере. Была подобрана новая фотография Ластика, несодержащаяся в исходном датасете, к ней были применены аналогичные трансформации, после чего она была подана на вход моделей.



Рисунок 7. Загрузка единичного изображения из класса 1

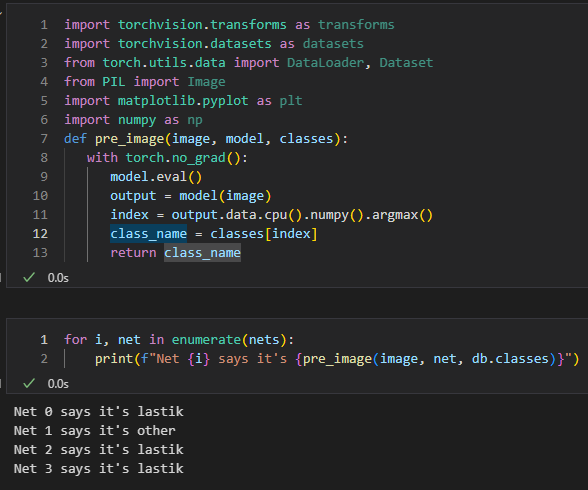


Рисунок 8. Результаты работы моделей на загруженном примере

Модели 1, 3 и 4 корректно определили класс изображения, модель 2 определяет класс ошибочно.

# Вывод

В ходе выполнения лабораторной работы был самостоятельно собран набор данных. Были созданы четыре архитектуры свёрточных нейронных сетей, после чего модели были обучены на собранных данных. Полученные модели достигают необходимый уровень точности.

Больший объём датасета мог бы помочь в увеличении точности распознавания. Однако ручной сбор большего числа фотографий в разных условиях значительно затруднен. Необходимо отметить, что нарезка видео со многими схожими кадрами помогает нарастить датасет, но они запечатлевают кота с одного ракурса на схожем фоне, что негативно влияет на качество собранного набора данных и приводит к переобучению.

По результатам обучения наилучшие метрики показала сеть Net\_2, однако более худшие результаты при проверке на новых изображениях позволяют отдать предпочтение менее точным, но более генерализированным моделям Net\_1 и Net\_4.