МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

**Лабораторная работа №\_\_2\_\_**

по дисциплине«Проектирование интеллектуальных систем»

Тема: «Сверточная нейросеть»

ИСПОЛНИТЕЛЬ: \_\_\_Ширшов А.С.\_\_\_

ФИО

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

подпись

"17"\_февраля\_2024 г.

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

ФИО

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

подпись

"\_\_"\_\_\_\_\_\_\_\_\_2024г.

Москва - 2024

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

# Задание

По заданию выбрать свои классы и обучить сверточную нейронную сеть из примера, используя GPU, а затем повысить точность модели. Провести три обучения для 3 разных тактик пуллинга: пуллинг с помощью шага свёртки stride, макс пуллинг, усредняющий пуллинг. Сравнить достигнутое качество, время обучения и степень переобучения. Выбрать лучшую конфигурацию. Сохранить модель. Перезапустить среду выполнения - теряются все текующие переменные.

Загрузить в colab готовую уже обученную на cifar100 модель. Преобразовать в onnx и сохранить локально.

Скачать [каталог](https://github.com/iu5git/Deep-learning/blob/main/ONNX_script) с html-файлом и встроить в него два файла моделей - обученную на ЛР1 и на ЛР2.

Скачать картинки из интернета согласно варианту и открыть их в html по кнопке. Автоматически в скрипте масштабируется изображение.

Выбрать в js нужные классы для готовой модели. Проверить на устойчивость обе модели, полносвязную и свёрточную, двигая картинку, убедиться в наличии свойства инвариантности сверточного слоя.

Отчет должен содержать: титульный лист, задание с вариантом, скриншоты и краткие пояснения по каждому этапу лабораторной работы, результаты работы модели ONNX в скрипте в браузере на загруженных изображениях, итоговую таблицу со результатами для всех вариантов обучения.

Варианты классов использовать из 1 лабораторной работы.

# Часть 1. Использование модели из примера

Возьмём исходную модель из примера, поместим в неё свои классы и посмотрим результаты. Результаты представлены в таблице 1.

Таблица 1. Результаты модели из примера

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 1.000 | 1.000 | 1.000 |
| Девочки | 1.000 | 1.000 | 1.000 |
| Цветы | 1.000 | 1.000 | 1.000 |
| Accuracy |  | | 1.000 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.8889 | 0.88 | 0.8844 |
| Девочки | 0.8333 | 0.85 | 0.8416 |
| Цветы | 0.9192 | 0.91 | 0.9146 |
| Accuracy |  | | 0.88 |
| **Время обучения** | 29,6 секунд | | |
| **Кол-во эпох** | 500 | | |
| **Размер батча** | 128 | | |
| **Скорость обучения** | 0.005 | | |

По результатам модели, а также графикам, можно заявить о переобучении модели. Возьмём кол-во эпох равным 160, так как на графиках именно около данного значения начинает ухудшаться точность на тестовой выборке. Значения базовой модели при уменьшенном кол-ве эпох представлены в таблице 2

Таблица 2. Результаты модели из примера с уменьшенным кол-вом эпох

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.9657 | 0.958 | 0.9618 |
| Девочки | 0.9506 | 0.962 | 0.9563 |
| Цветы | 0.9839 | 0.980 | 0.982 |
| Accuracy |  | | 0.9667 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.9192 | 0.91 | 0.9146 |
| Девочки | 0.8614 | 0.87 | 0.8657 |
| Цветы | 0.92 | 0.92 | 0.92 |
| Accuracy |  | | 0.9 |
| **Время обучения** | 9,9 секунд | | |
| **Кол-во Эпох** | 160 | | |
| **Размер батча** | 128 | | |
| **Скорость обучения** | 0.005 | | |

Можно отметить, что показатели на тестовой выборке возросли, по сравнению с предыдущей.

Увеличим размер батча в два раза. Также увеличим кол-во эпох в два раза. Результаты представлены в таблице 3.

Таблица 3. Результаты модели из примера с увеличенным размером батча и кол-вом эпох

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.9223 | 0.9740 | 0.9475 |
| Девочки | 0.9645 | 0.9240 | 0.9438 |
| Цветы | 0.9878 | 0.9740 | 0.9809 |
| Accuracy |  | | 0.9573 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.8835 | 0.91 | 0.8966 |
| Девочки | 0.8351 | 0.81 | 0.8223 |
| Цветы | 0.91 | 0.91 | 0.91 |
| Accuracy |  | | 0.8767 |
| **Время обучения** | 15,5 секунд | | |
| **Кол-во Эпох** | 320 | | |
| **Размер батча** | 250 | | |
| **Скорость обучения** | 0.005 | | |

Момент переобучения не успел возникнуть, поэтому попробуем сделать ещё одну модификацию модели. Уменьшим learning rate в 2 раза и увеличим кол-во эпох в два раза.

Таблица 4. Результаты модели из примера с увеличенным размером батча и кол-вом эпох

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.9732 | 0.9440 | 0.9584 |
| Девочки | 0.9364 | 0.9720 | 0.9539 |
| Цветы | 0.9819 | 0.9740 | 0.9779 |
| Accuracy |  | | 0.9633 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.914 | 0.85 | 0.8808 |
| Девочки | 0.7982 | 0.87 | 0.8325 |
| Цветы | 0.9184 | 0.9 | 0.9091 |
| Accuracy |  | | 0.8733 |
| **Время обучения** | 30,45 секунд | | |
| **Кол-во Эпох** | 640 | | |
| **Размер батча** | 250 | | |
| **Скорость обучения** | 0.0025 | | |

По результатам было решено оставить модель с параметрами, представленными таблице 2.

# Часть 2. Пуллинг с помощью шага свёртки stride

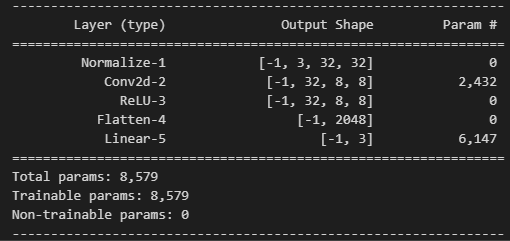


Рисунок 1 – Описание модели

Параметры модели - размер ядра = 5, stride = 4, padding = 2

Таблица 5. Результаты модели, слой с использованием stride

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 1.000 | 1.000 | 1.000 |
| Девочки | 1.000 | 1.000 | 1.000 |
| Цветы | 1.000 | 1.000 | 1.000 |
| Accuracy |  | | 1.000 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.8333 | 0.80 | 0.8163 |
| Девочки | 0.7407 | 0.80 | 0.7692 |
| Цветы | 0.9062 | 0.87 | 0.8878 |
| Accuracy |  | | 0.8233 |
| **Время обучения** | 10,9 секунд | | |
| **Кол-во Эпох** | 250 | | |
| **Размер батча** | 128 | | |
| **Скорость обучения** | 0.005 | | |

Произошло переобучение примерно около эпохи 45, обучим модель заново. Результаты в таблице 6

Таблица 6. Результаты модели, слой с использованием stride

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.9782 | 0.988 | 0.9831 |
| Девочки | 0.986 | 0.984 | 0.9850 |
| Цветы | 0.998 | 0.99 | 0.9940 |
| Accuracy |  | | 0.9873 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.84 | 0.84 | 0.84 |
| Девочки | 0.7714 | 0.81 | 0.7902 |
| Цветы | 0.9263 | 0.88 | 0.9026 |
| Accuracy |  | | 0.8433 |
| **Время обучения** | 2,24 секунд | | |
| **Кол-во Эпох** | 45 | | |
| **Размер батча** | 128 | | |
| **Скорость обучения** | 0.005 | | |

# Часть 3. Пуллинг с помощью maxPool

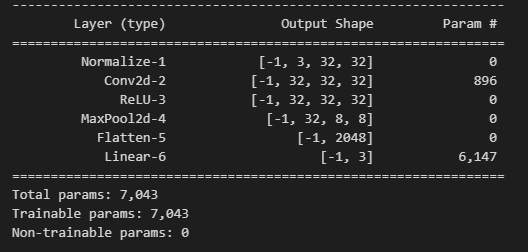


Рисунок 2 – Описание модели

Параметры модели – padding = 1, stride = 1, MaxPool2d (4)

Таблица 5. Результаты модели, слой с использованием maxPool

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 1.000 | 1.000 | 1.000 |
| Девочки | 1.000 | 1.000 | 1.000 |
| Цветы | 1.000 | 1.000 | 1.000 |
| Accuracy |  | | 1.000 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.9121 | 0.83 | 0.8691 |
| Девочки | 0.7982 | 0.87 | 0.8325 |
| Цветы | 0.92 | 0.92 | 0.92 |
| Accuracy |  | | 0.8733 |
| **Время обучения** | 12,31 секунд | | |
| **Кол-во Эпох** | 250 | | |
| **Размер батча** | 128 | | |
| **Скорость обучения** | 0.005 | | |

Произошло переобучение примерно около эпохи 90, обучим модель заново. Результаты в таблице 6

Таблица 6. Результаты модели, где используется добавочный слой с использованием stride

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.969 | 1.000 | 0.9843 |
| Девочки | 1.000 | 0.970 | 0.9848 |
| Цветы | 1.000 | 0.998 | 0.999 |
| Accuracy |  | | 0.9893 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.8835 | 0.91 | 0.8966 |
| Девочки | 0.866 | 0.84 | 0.8528 |
| Цветы | 0.920 | 0.92 | 0.92 |
| Accuracy |  | | 0.89 |
| **Время обучения** | 4,49 секунд | | |
| **Кол-во Эпох** | 90 | | |
| **Размер батча** | 128 | | |
| **Скорость обучения** | 0.005 | | |

# Часть 4. Пуллинг с помощью усредняющего пуллинга

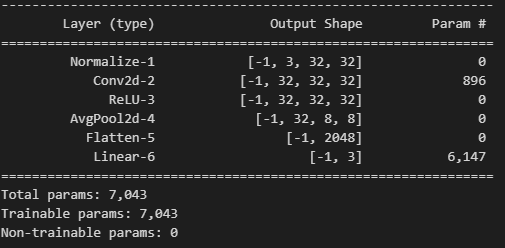


Рисунок 2 – Описание модели

Параметры модели – padding = 1, stride = 1, AvgPool2d (4)

Таблица 5. Результаты модели, слой с использованием AvgPool

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.998 | 1.000 | 0.999 |
| Девочки | 0.998 | 0.998 | 0.998 |
| Цветы | 1.000 | 0.998 | 0.999 |
| Accuracy |  | | 0.9987 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.8966 | 0.78 | 0.8342 |
| Девочки | 0.7434 | 0.84 | 0.7887 |
| Цветы | 0.89 | 0.89 | 0.989 |
| Accuracy |  | | 0.8367 |
| **Время обучения** | 13,8 секунд | | |
| **Кол-во Эпох** | 250 | | |
| **Размер батча** | 128 | | |
| **Скорость обучения** | 0.005 | | |

Произошло переобучение примерно около эпохи 100, обучим модель заново. Результаты в таблице 6

Таблица 6. Результаты модели, слой с использованием AvgPool

|  |  |  |  |
| --- | --- | --- | --- |
| **Train** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.9456 | 0.9740 | 0.9596 |
| Девочки | 0.9693 | 0.9480 | 0.9585 |
| Цветы | 0.9899 | 0.982 | 0.9859 |
| Accuracy |  | | 0.9680 |
| **Test** |  | | |
| Классы | **Precision** | **Recall** | **F1-score** |
| Рогатый скот | 0.8646 | 0.83 | 0.8469 |
| Девочки | 0.7714 | 0.81 | 0.7902 |
| Цветы | 0.899 | 0.89 | 0.8945 |
| Accuracy |  | | 0.8433 |
| **Время обучения** | 5,42 секунд | | |
| **Кол-во Эпох** | 100 | | |
| **Размер батча** | 128 | | |
| **Скорость обучения** | 0.005 | | |

# Часть 5. Использование модели ONNX

Сравним использование двух моделей. Начнём с полносвязной нейронной сети из лабораторной номер 1. На рисунках ниже представлены результаты.

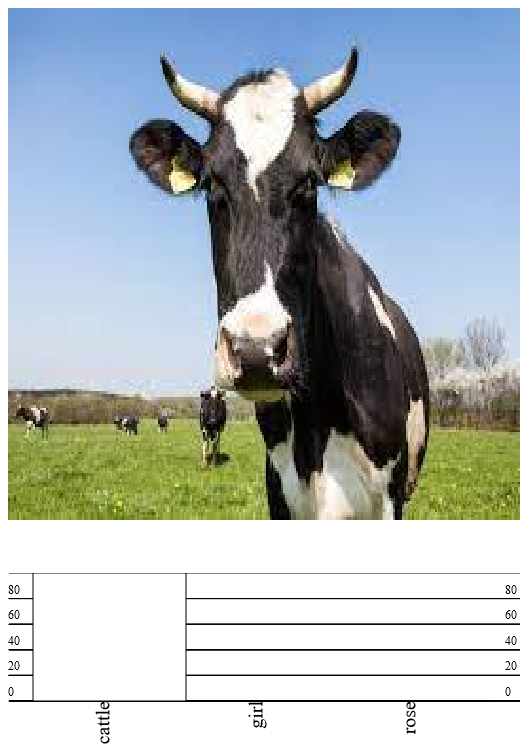


Рисунок 3 – MLP вход – корова



Рисунок 4 -МLP вход – девочка

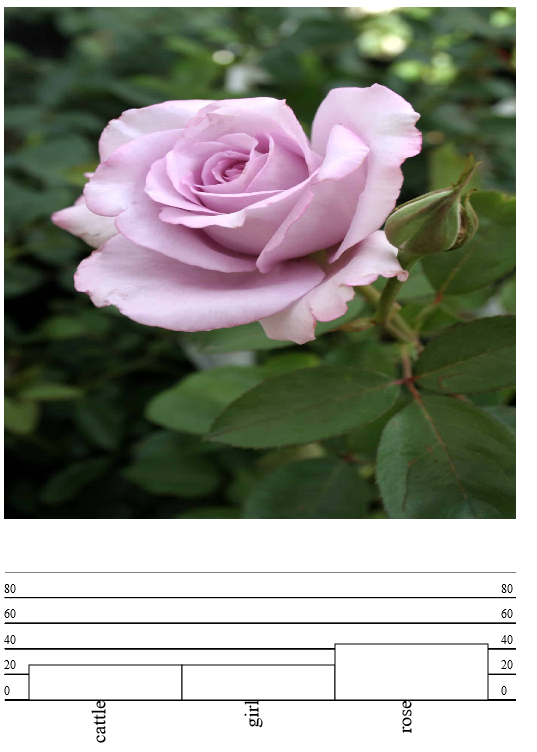


Рисунок 5 – MLP – вход роза

На примере картинкой с коровой, спрячем рога и посмотрим, что выдаст MLP. Смотря на рисунок ниже, можно увидеть, как модель начинает ошибаться в распознавании коровы на изображении.



Рисунок 6 – MLP. Вход – корова (сдвинутая)

Теперь посмотрим на результаты свёрточной нейронной сети.

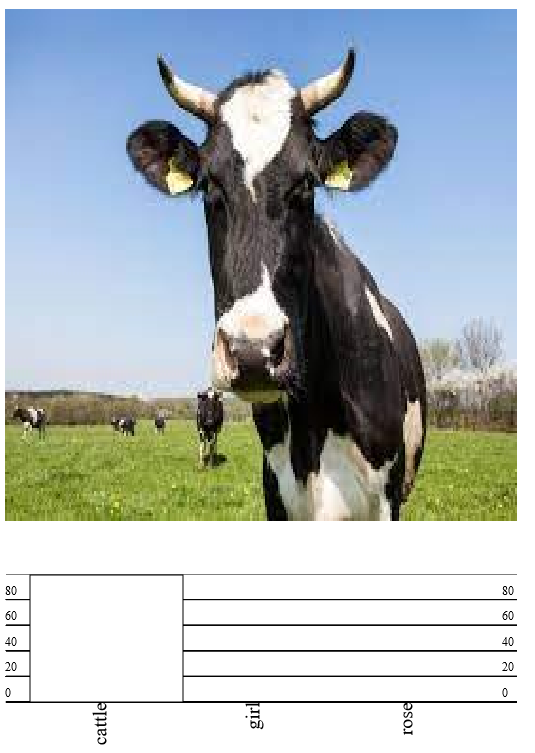


Рисунок 7 – CNN. Вход – Корова.



Рисунок 8 – CNN. Вход – девочка



Рисунок 9 – CNN. Вход – роза

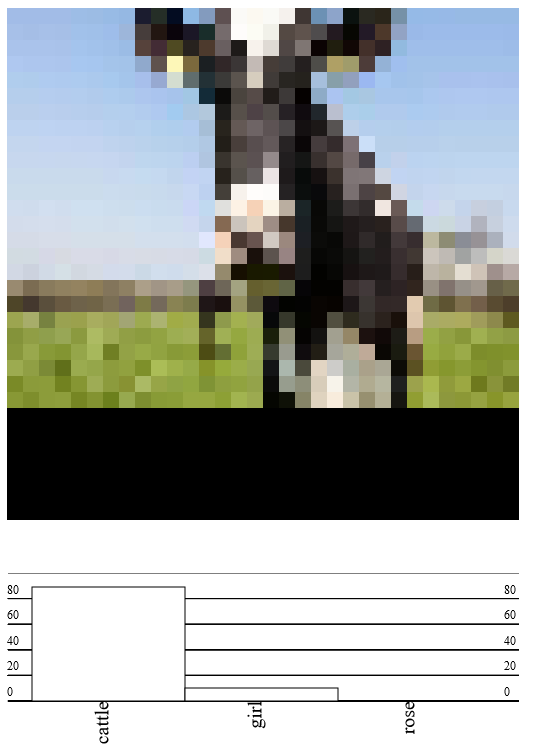


Рисунок 10 – CNN Вход – сдвинутая корова.

На примере видно, что результаты сверточной нейронной сети оказываются не только точнее, чем у обыкновенной полносвязной. Но и к различным изменениям положения изображения сверточная нейросеть оказвается более устойчивой.

|  |  |  |
| --- | --- | --- |
| Описание  модели | Результаты в % | Комментарии |
| Модель из примера | Train = 100  Test = 88 | Обучили исходную модель, обнаружили переобучение |
| Модель из примера, кол-во эпох = 160 | Train = 97  Test = 90 | Убрали момент переобучения, увеличилась точность на тестовой выборке |
| Модель из примера – Кол-во эпох = 320 Размер Батча = 250 | Train = 95,8  Test = 87,6 | Изменили размер батча вдвое, во столько же увеличили кол-во эпох |
| Модель из примера –  Кол-во эпох = 640  Размер Батча = 250  Скорость обучения = 0.0025 | Train = 96,3  Test = 87,3 | Уменьшили скорость обучения вдвое, кол-во эпох увеличили вдвое |
| Модель с одним слоем conv2d. Размер ядра – 5. Stride = 4. Padding = 2 | Train = 100  Test = 82,3 | Тактика – пуллинг через stride. Заметили переобучение. |
| Модель с одним слоем conv2d. Размер ядра – 5. Stride = 4. Padding = 2. Кол-во эпох - 45 | Train = 98,7  Test = 84,3 | Убрали переобучение. Улучшились показатели. |
| Модель с использованием MaxPool | Train = 100  Test = 87,3 | Поменяли тактику на MaxPool. Заметили переобучение. |
| Модель с использованием MaxPool. Кол-во эпох - 90 | Train = 98,9  Test = 89 | Убрали момент переобучения. Показатели улучшились. |
| Модель с использованием AvgPool | Train = 99,9  Test = 83,7 | Поменяли тактику. Заметили переобучение. |
| Модель с использованием AvgPool. Кол-во эпох - 100 | Train = 96,8  Test = 84,3 | Убрали переобучение. |

# Часть 6. Вывод.

На основе полученных результатов, можно сделать вывод о наличии большей эффективности у сверточной нейронной сети по сравнению с полносвязной. Даже при использовании одиночных слоёв пуллинга, либо stride – результаты получались лучше, чем у полносвязной нейронной сети. Наилучший результат вышел при использовании комбинации данных тактик. Среди трёх тактик наибольшей эффективностью выделился maxPool, скорее всего это связано с тем, что при попадании в часть изображения какого признака на фоне, он сохраняется, а фон отбрасывается. Также можно отметить, что сверточная нейронная сеть более устойчива к различным изменениям расположения изображения, чем полносвязная сеть.