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«Разработка нейросетевых систем»

Тема: «Свёрточная нейросеть»

ИСПОЛНИТЕЛЬ: \_\_Журавлев Н.В.\_\_\_

ФИО

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

подпись

"27"\_\_02\_\_\_\_\_2024 г.

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

ФИО

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

подпись

"\_\_"\_\_\_\_\_\_\_\_\_202\_ г.

Москва - 2024

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

**Задание**

Необходимо познакомиться с фреймворком машинного обучения PyTorch и выполнить три задания:

1. По заданию выбрать свои классы и обучить сверточную нейронную сеть из примера, используя GPU, а затем повысить точность модели.
2. 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.
3. Загрузить в colab готовую уже обученную на cifar100 модель. Преобразовать в onnx и сохранить локально.
4. Скачать каталог с html-файлом и встроить в него два файла моделей - обученную на ЛР1 и на ЛР2.
5. Скачать картинки из интернета согласно варианту и открыть их в html по кнопке. Автоматически в скрипте масштабируется изображение.
6. 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.

Использовались следующие классы из CIFAR100: 19, 61, 26.

Задания для самостоятельной работы:

1. Проанализируйте результаты обучения вашей модели. Как изменилась точность на обучающей и тестовой выборке по сравнению с полносвязной моделью?
2. Измените вашу модель - сравните три варианта пуллинга по заданию. Сравните результаты обучения этих 3-ех вариантов
3. Измените вашу модель - поменяйте количество сверточных слоев, размер ядра свертки, шаг свертки и заполнение. Проанализируйте результаты обучения.
4. Возникает ли переобучение вашей модели?
5. Измените гиперпараметры обучения для повышения точности модели: количество эпох, размер батча, скорость обучения
6. Сравните полносвязную и сверточную нейронную сеть с помощью визуализации. Сравните результаты моделей и их реакцию на сдвиги изображения.
7. Укажите в отчете отличия моделей и поясните их. 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 с полносвязной моделью?**

После выполнения получим график на рисунке 1.

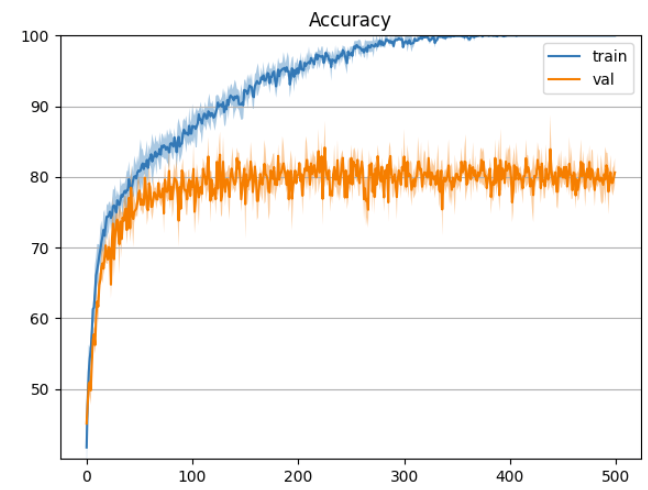


Рисунок 1 - Результат обучения

Даже при переобучении видно, что точность больше, чем в полносвязной моделью.

**Часть 2. Измените вашу модель - сравните три варианта пуллинга по заданию. Сравните результаты обучения этих 3-ех вариантов**

Сначала используем пуллинг с помощью шага свёртки stride, что показано на рисунке 2.

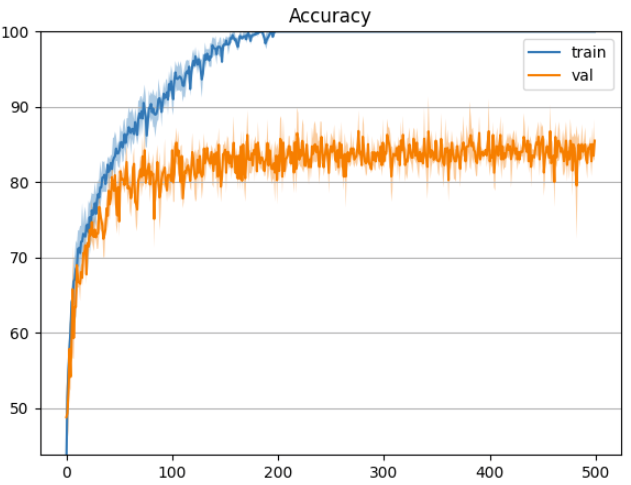


Рисунок 2 – Результат обучения с помощью stride

В результате, мы получили улучшение точности оценки, что связано с тем, что мы сделали более плавное выделение признаков.

Затем попробуем с макс пуллингом. Результат представлен на рисунке 3.

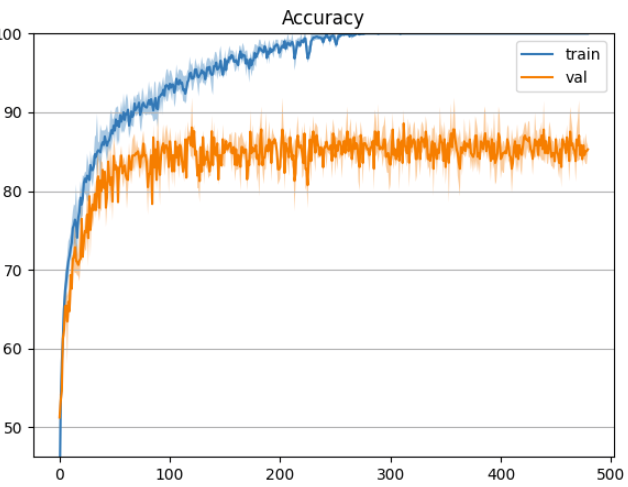


Рисунок 3– Результат обучения с помощью макс пуллинга

В результате, мы получили улучшение точности оценки, что связано с тем, что мы выделяем более выделяющиеся признаки.

В конце попробуем с усредняющим пуллингом. Результат представлен на рисунке 4.

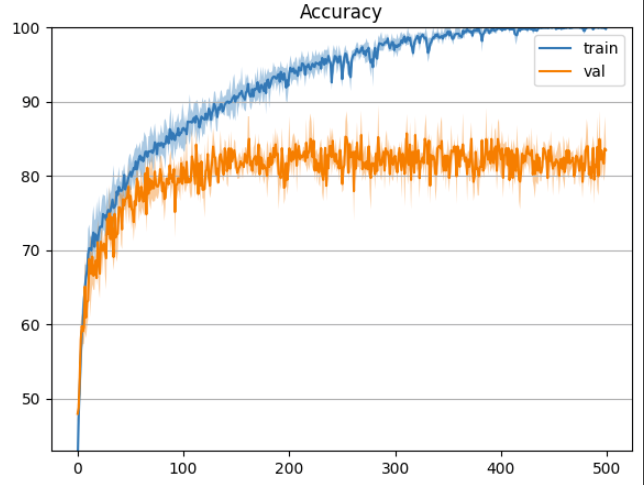


Рисунок 4 – Результат обучения с помощью усредняющего пуллинга

В результате, мы получили улучшение точности оценки, что связано с тем, что выделение максимального значимого важнее в данном случае.

**Часть 3. Измените вашу модель - поменяйте количество сверточных слоев, размер ядра свертки, шаг свертки и заполнение. Проанализируйте результаты обучения.**

Попробуем задать дополнительный сверточный слой

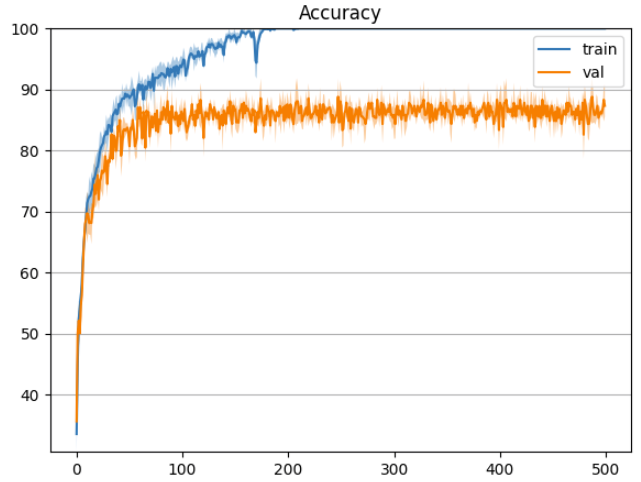


Рисунок 5– Результат обучения с помощью свёрточного слоя

В результате, мы получили улучшение точности оценки, что связано с тем, что мы сделали более точным обобщением признаков.

Теперь попробуем задать более большой размер ядра свёртки. Получим:

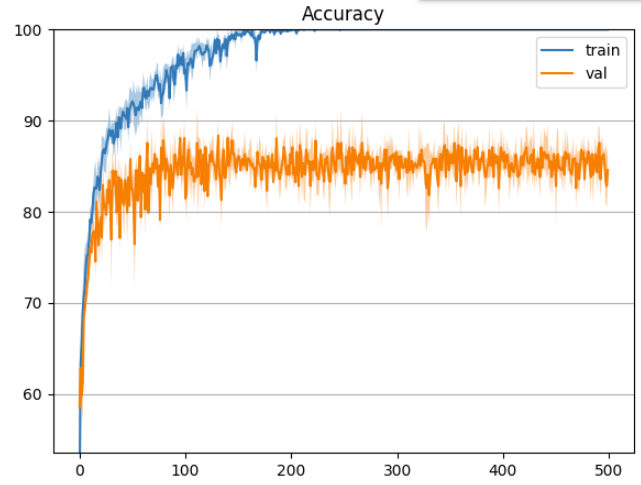


Рисунок 6 – Результат обучения с помощью большего размера ядра

Получили некоторое уменьшение точности за счёт того, что при увеличении размера ядра уменьшается возможность учитывать мелкое количество деталей.

Изменим stride на 2 и padding на 5. Получим:

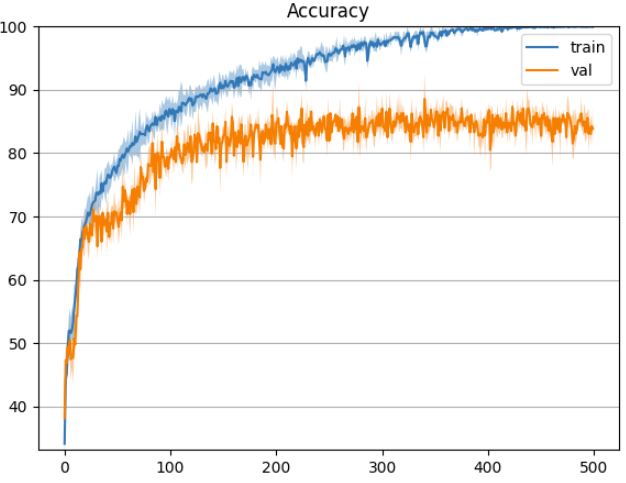


Рисунок 7– Результат обучения с помощью измения stride и padding

Результаты незначительно улучшилась, что связано с более точным обходом по карте за счёт уменьшения stride, но это нивелируется за счёт уменьшением padding.

**Часть 4. Возникает ли переобучение вашей модели?**

На рисунке 5 видно, что переобучении модели наступает переобучения примерно на 100 эпохах, т.к. сильно разнится точность на обучающей и тестовой выборках.

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

После изменения параметров были выявлены лучшие значения для повышения точности. В результате получается результат на рисунке 8.

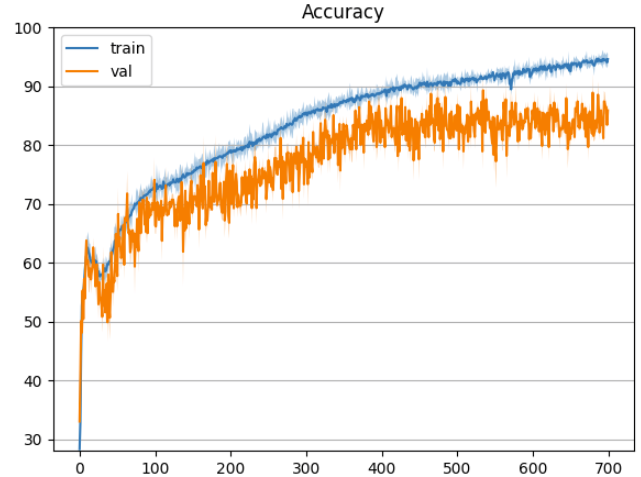


Рисунок 8 – Результат обучения с помощью оптимальных гиперпараметров

**Часть 6. Сравните полносвязную и сверточную нейронную сеть с помощью визуализации. Сравните результаты моделей и их реакцию на сдвиги изображения.**

При загрузке и последующего изучения нейросетей в каталоге, были выявлено, что полносвязная модель значительно хуже справляется с поставленной задачей при даже незначительных перемещений рисунка, однако сверточная модель показывает хороший результат, даже при больших смещениях, что связано со свойством инвариантности у свёрточной нейросети. Результат при смещении для полносвязной сети представлен на рисунке 9, а для свёрточной на рисунке 10.



Рисунок 9 – Результат работы свёрточной нейросети



Рисунок 10 – Результат полносвязной нейросети

**Часть 7. Укажите в отчете отличия моделей и поясните их. Укажите, какие действия помогли повысить точность вашей модели и объясните почему.**

Различия между свёрточной и полносвязной нейросетями заключается в следующем:

* Сверточная модель для каждый нейрон связан с ограниченным числом входных нейронов
* Свёрточная нейросеть имеет разные типы слоёв
* Свёрточные нейросети имеют инвариантность
* В свёрточной нейросети в карте признаков все нейроны имеют одинаковый набор параметров

Для повышения точности модели мне помогло:

* Уменьшение количество эпох для преодоления переобучения
* Выбор оптимальной конфигурации пулинга
* Добавление ещё одного свёрточного слоя
* Подбор наилучших гиперпараметров

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

Таблица 1 - Итоговая таблица с результатами для всех вариантов обучения

|  |  |  |  |
| --- | --- | --- | --- |
| Конфигурация нейросети | Гиперпараметры | Точность | Комментарий |
| CL(32, 5, stride=4, padding=2),  CL(64, 3, stride=1, padding=1),  AvgPool(4),  FC(256) | lr=0.005  batch\_size = 128  epoch = 500 | train = 100%  test= 79.67% | Базовый вариант |
| ~~CL(32, 5, stride=1, padding=2),~~  ~~CL(32, 5, stride=3, padding=1),~~  ~~CL(64, 3, stride=1, padding=1),~~  ~~AvgPool(4),~~  ~~FC(256)~~ | ~~lr=0.005~~  ~~batch\_size = 128~~  ~~epoch = 500~~ | ~~train = 100%~~  ~~test= 84,00%~~ | ~~Добавление свёрточного слоя~~ |
| CL(32, 5, stride=1, padding=2),  MaxPool(4),  CL(64, 3, stride=1, padding=1),  AvgPool(4),  FC(256) | lr=0.005  batch\_size = 128  epoch = 500 | train = 100%  test= 85,00% | Добавление макс пуллинга |
| CL(32, 5, stride=1, padding=2),  AvgPool(4),  CL(64, 3, stride=1, padding=1),  AvgPool(4),  FC(256) | lr=0.005  batch\_size = 128  epoch = 500 | train = 99,73%  test= 82,67% | Добавление среднего пуллинга |
| CL(32, 5, stride=1, padding=2),  MaxPool(4),  CL(64, 3, stride=1, padding=1),  AvgPool(4),  CL(64, 3, stride=1, padding=1),  FC(256) | lr=0.005  batch\_size = 128  epoch = 500 | train = 100%  test= 86,33% | Увеличение количества свёрточных слоёв |
| CL(32, 5, stride=1, padding=2),  MaxPool(4),  CL(64, 7, stride=1, padding=1),  AvgPool(4),  FC(64) | lr=0.005  batch\_size = 128  epoch = 500 | train = 100%  test= 85,33% | Увеличение размера ядра |

|  |  |  |  |
| --- | --- | --- | --- |
| CL(32, 5, stride=1, padding=2),  MaxPool(4),  CL(64, 3, stride=2, padding=5),  AvgPool(4),  FC(256) | lr=0.005  batch\_size = 128  epoch = 500 | train = 100%  test= 85,67% | Увеличение padding и уменьшение stride |
| CL(32, 5, stride=1, padding=2),  MaxPool(4),  CL(64, 3, stride=1, padding=1),  AvgPool(4),  CL(64, 3, stride=1, padding=1),  FC(256) | lr=0.001  batch\_size = 256  epoch = 700 | train = 94,93%  test= 84,00% | Оптимизация гиперпараметров |

**Вывод:** была создана и оптимизирована свёрточная нейросеть, которая с самого начала была точнее полносвязной и имела точность 79,67%. Затем было проверено влияние разных типов пуллинга и как это влияет на точность. Самым лучшим по точности было использование макс пуллинга с точность на тестовой выборке 85%, так как он среди признаков определяет более значимые признаки, в то время как при увеличении stride теряются какие-либо части изображения за счёт смешения, но быстрее происходит обучение. Точность при среднем пуллинге равна 82,67%, так как из-за особенности классов изображения, в которых важнее выделять более значимые признаки, а не среднее их значение.

Затем было произведено варьировании параметров модели, наибольшая точность 86,33% была достигнута при добавлении нового слоя, так как нейросеть извлекает более сложные и абстрактные признаки из входных данных, но может потребовать больше вычислительных ресурсов и времени для обучения. Точность при уменьшении stride и увеличения padding равна 85,67%, Маленький страйд позволяет учесть больше деталей изображения при формировании признаков, что может привести к более точному извлечению информации, но Использование маленького страйда может привести к увеличению использования памяти, так как больше признаковых карт будет генерироваться на каждом слое. Точность при увеличении размера ядра равна 85,33%, так как слишком большой размер ядра может привести к потере мелких деталей и ухудшению точности модели, что и произошло в данном случае.

После получения наиболее точной нейросети и оптимизации гиперпараметров, было произведено сравнение с полносвязной нейросетью и выявлены сильные различия в результате между ними при смещении из-за свойства свёрточной нейросети - инвариантности.