

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Искусственные нейронные сети»**  
**Тема: Распознавание объектов на фотографиях**

Студентка гр. 8383

\_\_\_\_\_

Гречко В.Д.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2021

## Цель работы

Распознавание объектов на фотографиях (Object Recognition in Photographs) CIFAR-10 (классификация небольших изображений по десяти классам: самолет, автомобиль, птица, кошка, олень, собака, лягушка, лошадь, корабль и грузовик).

## Задачи

- Ознакомиться со сверточными нейронными сетями
- Изучить построение модели в Keras в функциональном виде
- Изучить работу слоя разреживания (Dropout)

## Требования

1. Построить и обучить сверточную нейронную сеть
2. Исследовать работу сети без слоя Dropout
3. Исследовать работу сети при разных размерах ядра свертки

## Выполнение работы

### 1. Построение и обучение сверточной нейронной сети

На первом шаге была построена сеть, представленная в методических указаниях. Однако при запуске программы оказалось, что длительность обучения (из-за слабых вычислительных мощностей моей машины) превышает несколько часов. В связи с этим было сокращено количество эпох до 20, а размер пакетов увеличен до 70.

Получили следующие результаты:

```
Epoch 20/20
643/643 [=====] - 120s 186ms/step - loss:
0.3449 - accuracy: 0.8761 - val_loss: 0.6271 - val_accuracy: 0.8082
313/313 [=====] - 6s 19ms/step - loss:
287.0916 - accuracy: 0.5244
```

Модель обладает высокой точностью на тренировочных данных (от 75% до 80%), но на тестовых данных точность понизилась до 52.44%

На рисунках 1 и 2 изображены графики ошибок и точности на тренировочном и тестовом множествах.

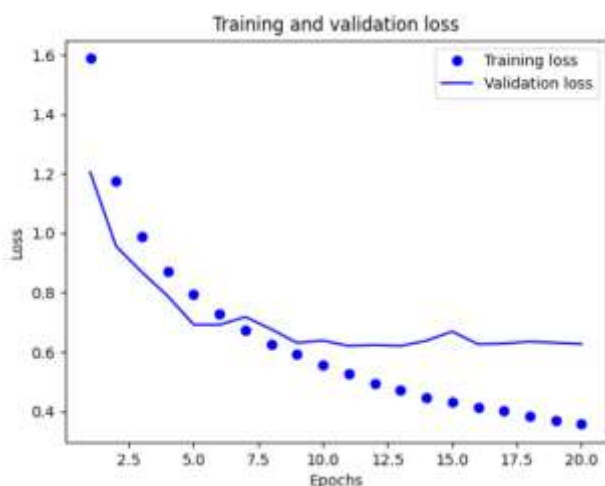


Рисунок 1 – график потерь

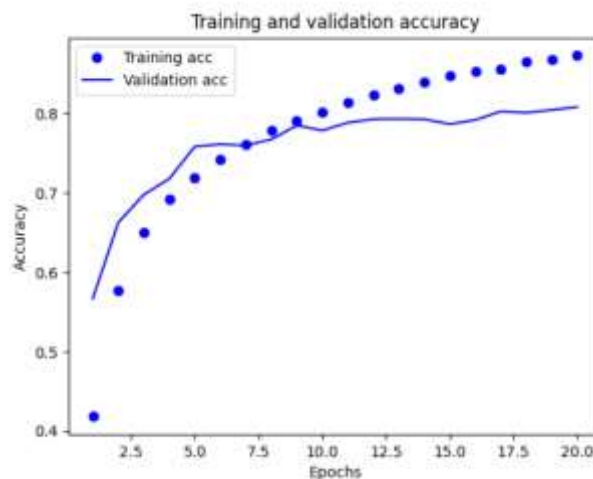


Рисунок 2 – график точности

## 2. Исследование работы сети без слоя Dropout

Посмотрим на результаты, полученные после удаления слоёв Dropout из сети:

643/643 [=====] - 117s 181ms/step - loss: 0.0493 - accuracy: 0.9838 - val\_loss: 1.6983 - val\_accuracy: 0.7378

313/313 [=====] - 6s 21ms/step - loss: 489.0543 - accuracy: 0.6358

Модель обладает 80% - 85% точностью на тренировочных данных, а на тестовых данных точность достигла 63.58%

На рисунках 3 и 4 изображены графики ошибок и точности на тренировочном и тестовом множествах.

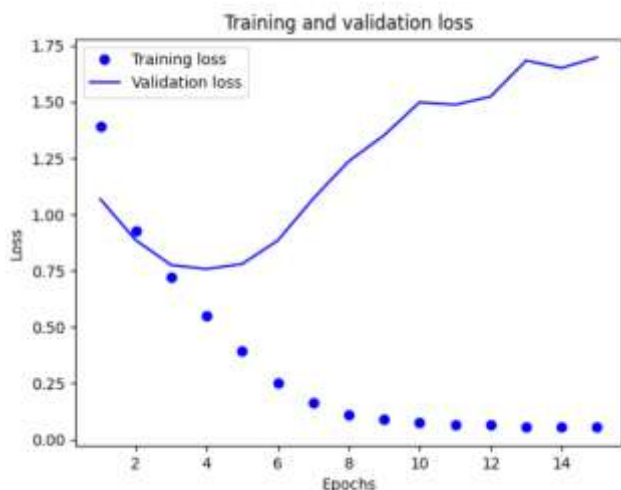


Рисунок 3 – график потерь

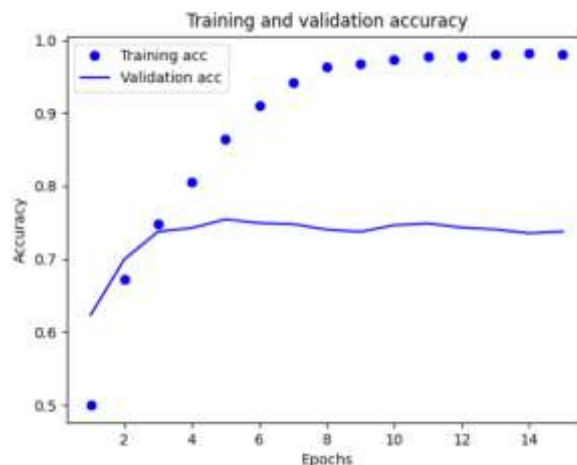


Рисунок 4 – график точности

Как видно из графиков произошло переобучение сети – точность на валидационных данных остается практически без изменений, а на тестовых данных она стремится к единице. В тоже время потери на валидационном множестве достигают максимальных значений, а не убывают. Следовательно, удаление слоя Dropout приводит к ухудшению обучения сети.

### 3. Исследование работы сети при разных размерах ядра свертки

В первую очередь добавим обратно слои Dropout и исследуем результаты работы сети при изменении параметра `kernel_size`.

`kernel_size = 5`

Результат:

643/643 [=====] - 274s 427ms/step - loss: 0.4739 - accuracy: 0.8318 - val\_loss: 0.6698 - val\_accuracy: 0.7804

313/313 [=====] - 14s 44ms/step - loss: 167.2173 - accuracy: 0.6252

На рисунках 5 и 6 изображены графики ошибок и точности на тренировочном и тестовом множествах.

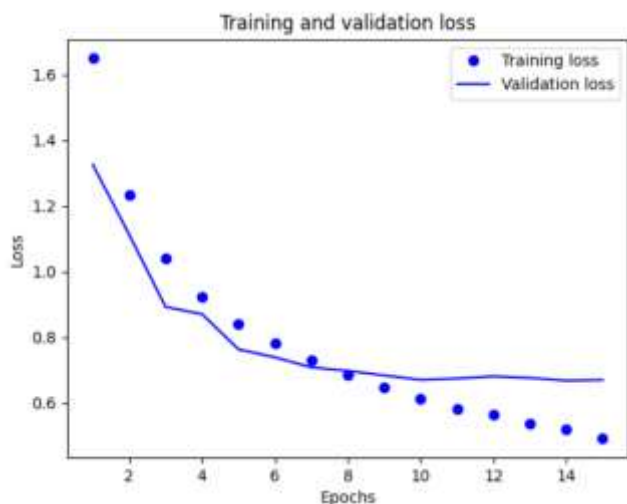


Рисунок 5 – график потерь

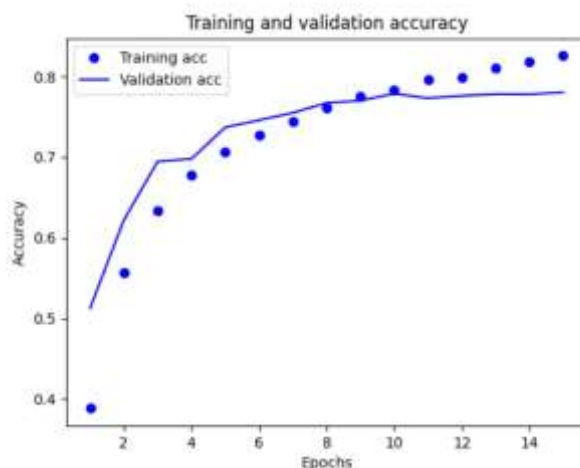


Рисунок 6 – график точности

`kernel_size = 7`

Результат:

```
643/643 [=====] - 425s 660ms/step - loss:
0.6311 - accuracy: 0.7768 - val_loss: 0.8133 - val_accuracy: 0.7378
313/313 [=====] - 21s 67ms/step - loss:
211.4076 - accuracy: 0.5598
```

На рисунках 7 и 8 изображены графики ошибок и точности на тренировочном и тестовом множествах.

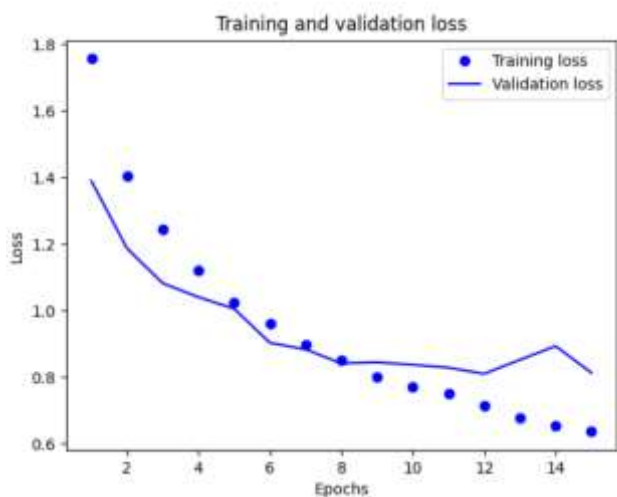


Рисунок 7 – график потерь

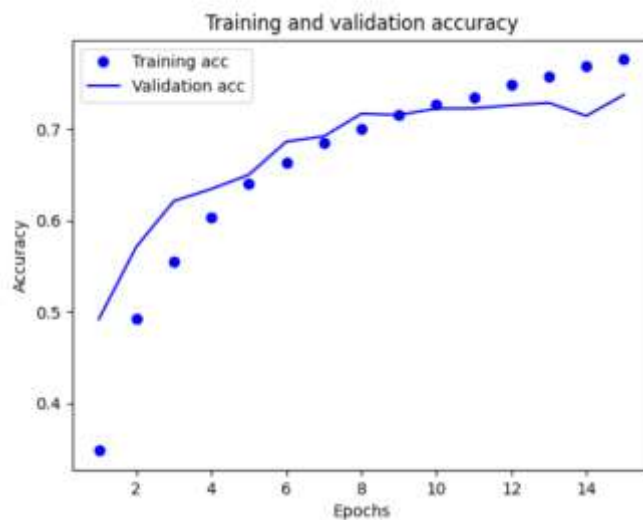


Рисунок 8 – график точности

`kernel_size = 2`

## Результат:

643/643 [=====] - 84s 131ms/step - loss: 0.4519 - accuracy: 0.8376 - val\_loss: 0.6503 - val\_accuracy: 0.7846

313/313 [=====] - 4s 13ms/step - loss: 227.2225 - accuracy: 0.4880

На рисунках 9 и 10 изображены графики ошибок и точности на тренировочном и тестовом множествах.

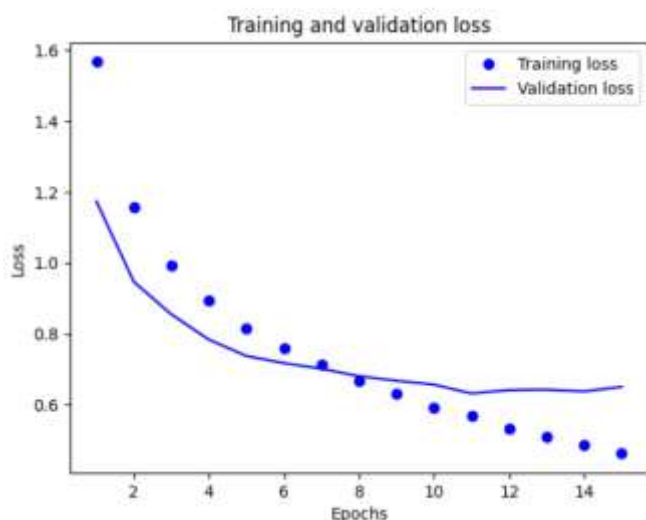


Рисунок 9 – график потерь

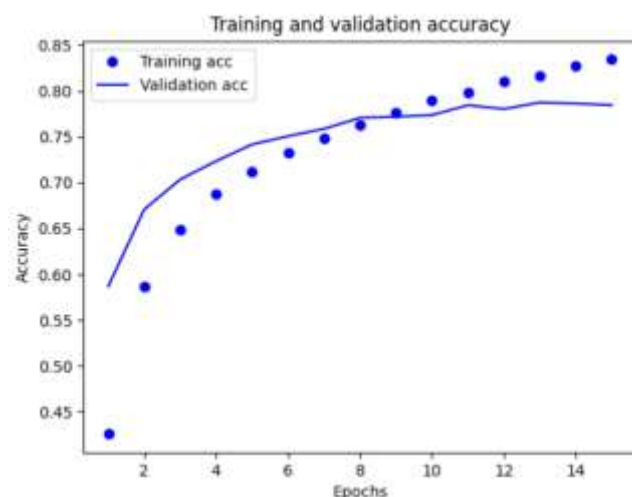


Рисунок 10 – график точности

На графиках видно, что увеличение размера ядра свёртки ухудшает обучение нейронной сети – все показатели падают. Рассмотрев два размера – 2 и 3 – можно сделать вывод, что самый оптимальный размер ядра – 3 (так как точность и на тестовом и на тренировочном наборах оказалась выше, а потери несколько меньше).

## Выводы

В ходе выполнения лабораторной работы была создана нейронная сеть со сверточной архитектурой, осуществляющая классификацию изображений. Также исследовано влияние слоя Dropout и различные размеры ядра свертки на процесс обучения.