

Федеральное государственное автономное образовательное  
учреждение высшего образования

Университет ИТМО

Дисциплина: Системы искусственного интеллекта

## **Лабораторная работа 4**

Вариант 2

**Выполнил студент:**

Кривоносов Егор Дмитриевич

**Группа:** Р33111

**Преподаватель:**

Полещук Елизавета Александровна

Санкт-Петербург

2021 г.

# Задание

В файлах с именами Lab1-Часть 1 и Lab1-Часть 2 содержится 2 части работы соответственно. Обе части представляют собой работу с коллекциями обучающих и тестовых данных.

Часть 1 представляет собой распознавание основных математических функций с иллюстрациями нейросетевого видения функций для обучения.

Часть 2 представляет собой распознавание простых изображений для выполнения следующей работы.

2	Cos(x) X: -9..9 Y: -1..1	CIFAR100	Learn rate, regularization L1
---	--------------------------	----------	-------------------------------

Там представлены такие гиперпараметры, как:

- Layer count
  - Neurons count per layer (actually it's not hyperparameter but structure parameter)
  - Learn rate
  - Regularization L1 and L2
  - Output layer activation type
  - Layer activation type
  - Loss function type
  - Epoch count
- 1) они используются для отбора признаков, попытаться достичь максимальной точности значением (не менее 0,95) для Часть2 модели с фиксированным эпохи графа 20
  - 2) изменение 1-го hyperparameter значение от Min до Max с минимальный шагом зависит от вашего варианта
  - 3) показать влияние на результаты, используя графики
  - 4) описать влияние каждого hyperparameter на точность.
  - 5) установите значение гиперпараметра обратно на значение, обеспечивающее максимальную точность
  - 6) Повторите 2-5 шагов для второго гиперпараметра

Составьте отчет, включающий:

- Описание каждого гиперпараметра и его влияние на точность.
- Значения гиперпараметров, которые использовались для достижения значения точности 0,95

- Графики значений этих гиперпараметров

## Выполнение

Оптимизация гиперпараметров – задача машинного обучения по выбору набора оптимальных параметров для обучающего алгоритма. Одни и те же виды моделей машинного обучения могут требовать различные предположения, веса и скорости обучения для различных видов данных. Эти параметры называются гиперпараметрами и их следует настраивать так, чтобы модель могла оптимально решить задачу обучения.

### Часть 1

После перебора всех значений гиперпараметров Learn Rate (определяет порядок того, как мы будем корректировать наши веса с учётом функции потерь в градиентном спуске) и regularization L1 (регуляризация через манхэттенское расстояние – метод добавления некоторых дополнительных ограничений к условию с целью решить некорректно поставленную задачу или предотвратить переобучение), наилучшего результата (**точность = 0.87**) удалось достичь на значениях **LR = 0.1** и **L1 = 0.001**

Слоес:  2

Loss function:

Batch size:  100

Learn rate: 

-0.1	0	0.001	0.01	0.05	<b>0.1</b>	0.5
1	5					

Regularization L1: 

-0.1	0	0.0001	0.0005	<b>0.001</b>	0.005
0.01	0.05	0.1			

Regularization L2: 

-0.1	0	<b>0.0001</b>	0.0005	0.001	0.005
0.01	0.05	0.1			

Output layer activation type:

Epoch count:  300

Neurons count in layer 1:  6

Neurons count in layer 2:  3

Neurons count in layer 1:  0

Neurons count in layer 1:  0

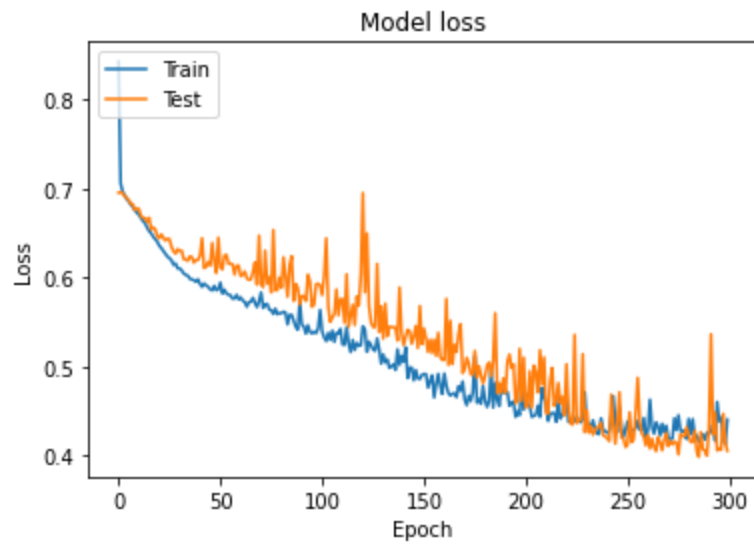
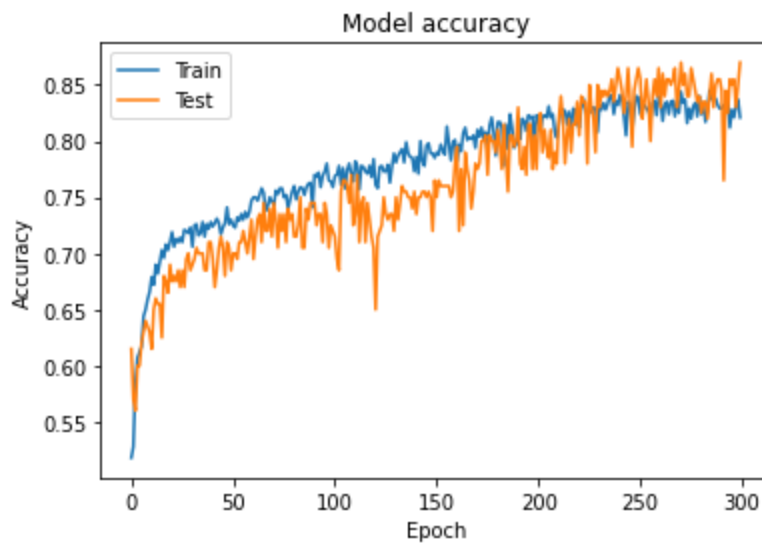
Layer 1 activation type:

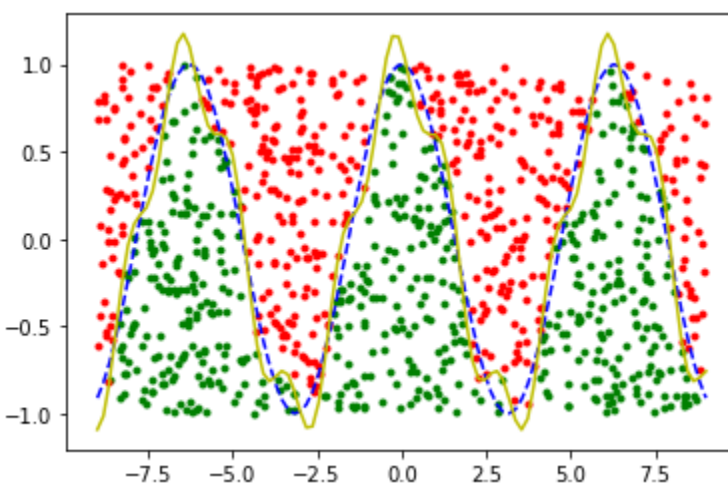
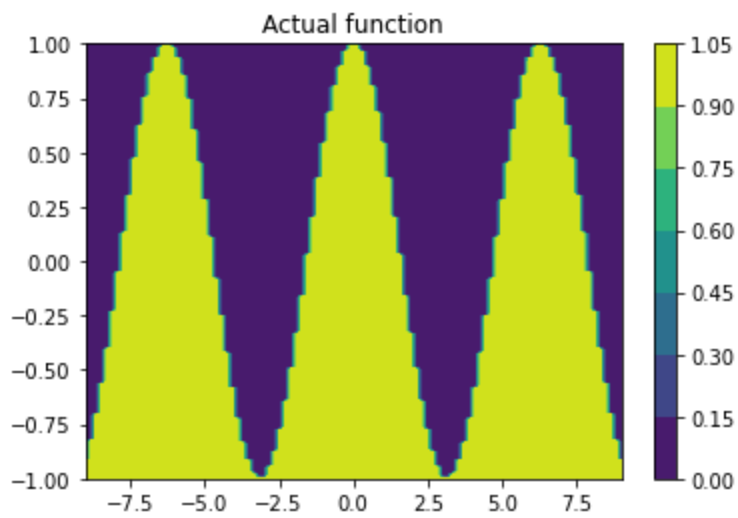
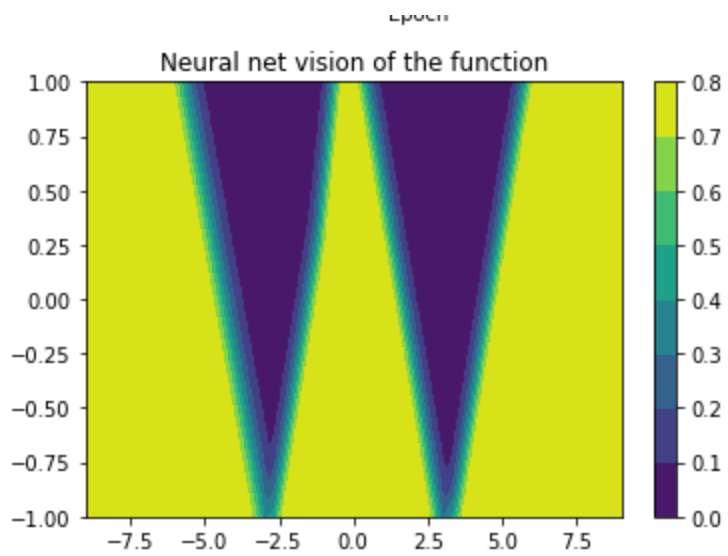
Layer 2 activation type:

Layer 3 activation type:

Layer 4 activation type:

Accuracy: 0.8700000047683716





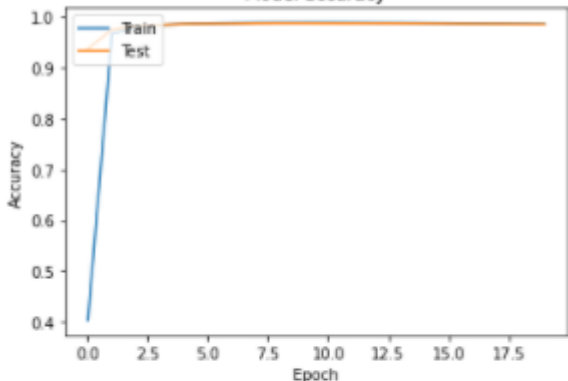
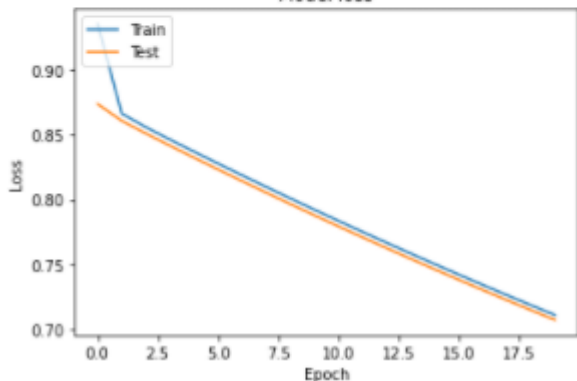
## Часть 2

Ссылка на датасет: <https://keras.io/api/datasets/cifar100/>

Команда: `tf.keras.datasets.cifar100.load_data()`

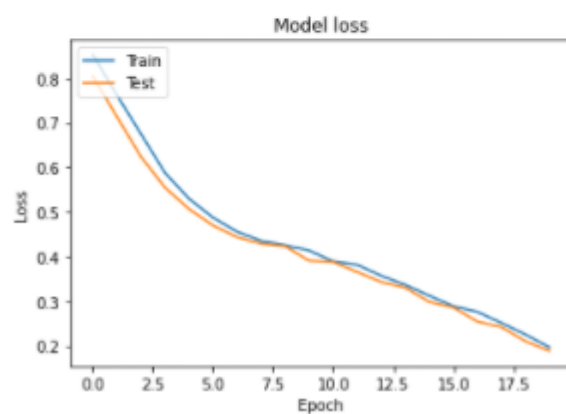
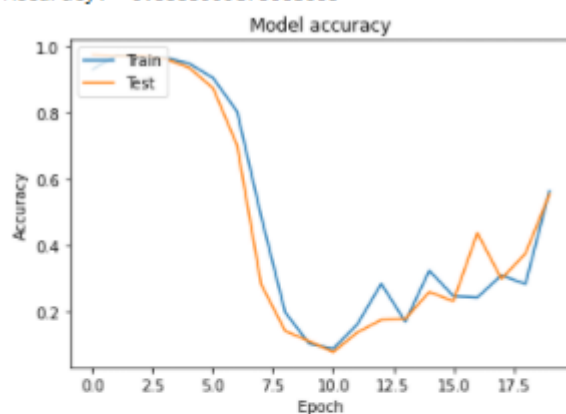
Выписал лучшие для Learn rate иначе отчет состоял бы из 64 \* 4 графиков

LR не может быть отрицательным и 0 ставить нет смысла т.к. корректировка весов вообще производиться не будет

Параметры	Графики
Lare rate = 0.001 Regularizatio n L1 = 0.001	<p>Accuracy: 0.9853000044822693</p> <p>Model accuracy</p>  <p>Model loss</p> 

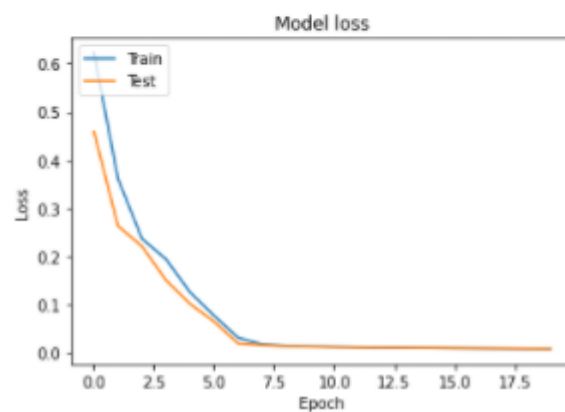
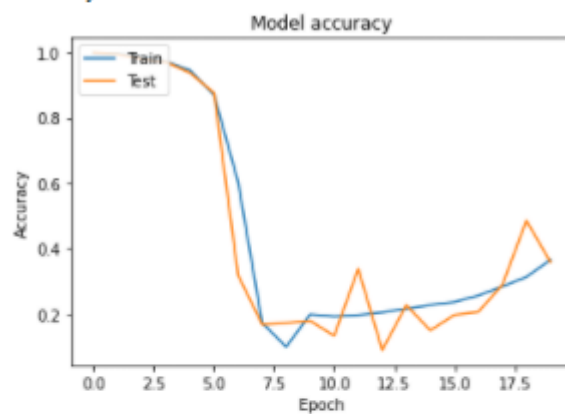
Lare rate =  
0.01  
Regularizatio  
n L1 = 0.001

Accuracy: 0.5555999875068665



Lare rate =  
0.05  
Regularizatio  
n L1 = 0.001

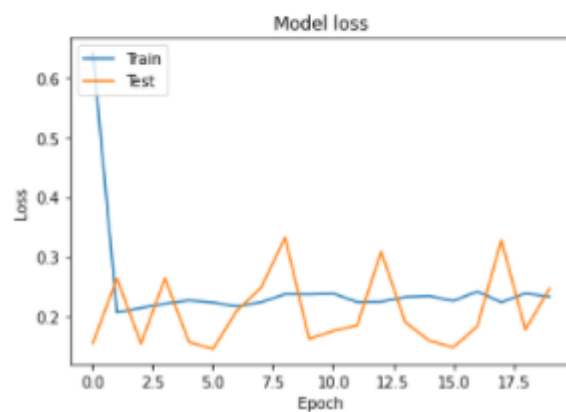
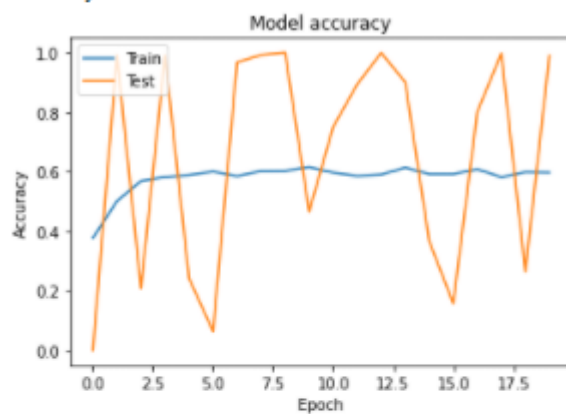
Accuracy: 0.3587000072002411





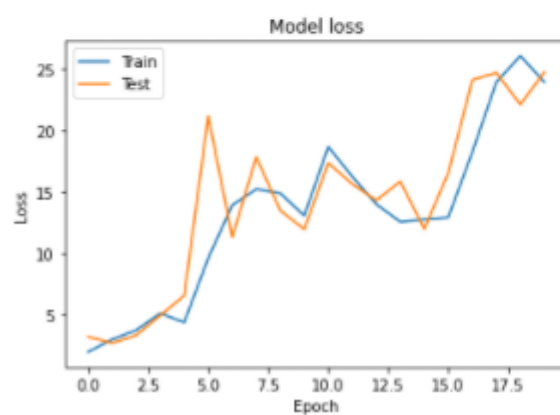
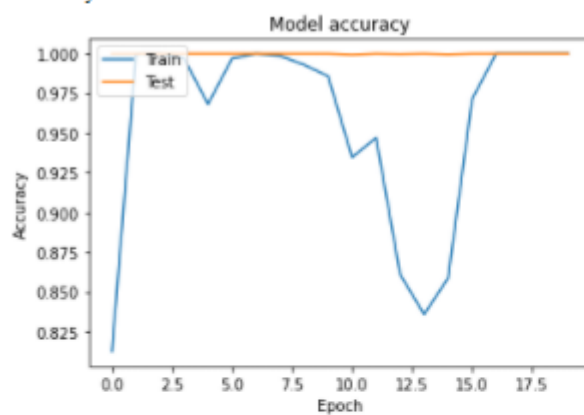
Lare rate =  
0.1  
Regularizatio  
n L1 = 0.01

Accuracy: 0.9889000058174133



Lare rate =  
0.5  
Regularizatio  
n L1 = 0.005

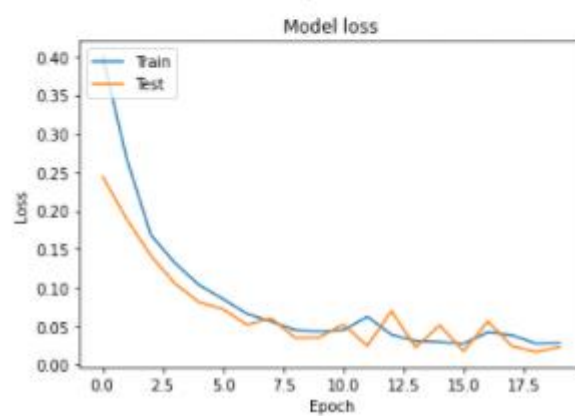
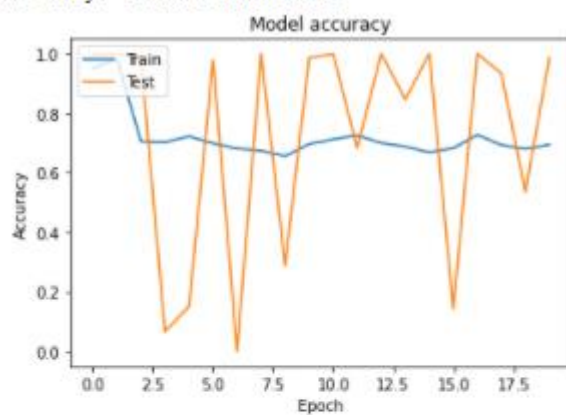
500/500 [=====] - 2s 5ms/step - loss: 23.9649 - accuracy: 1.00  
Accuracy: 1.0



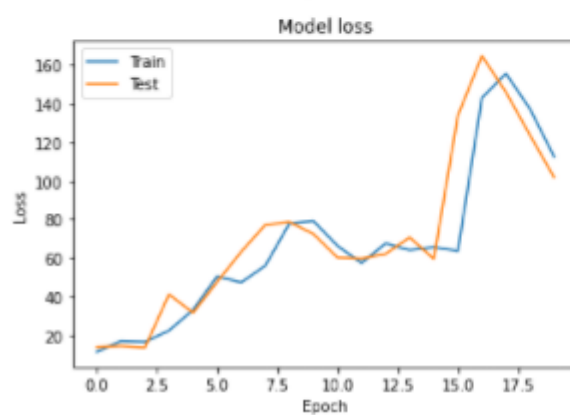
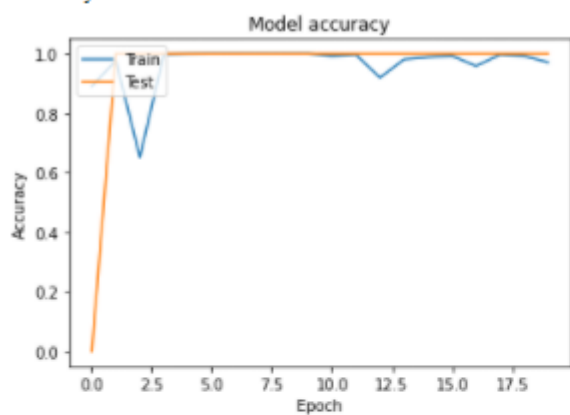
Lare rate = 1  
Regularizatio  
n L1 = 0.005

LR = 1  
L1 = 0.001

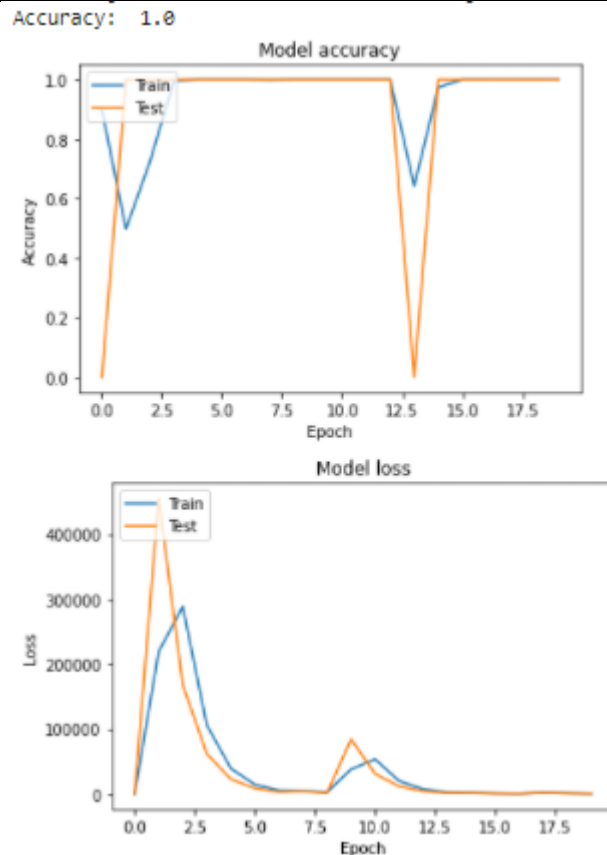
Accuracy: 0.9878000020980835



500/500 [=====] - 2s 5ms/step - loss: 112.4564 - accuracy: 0.976  
Accuracy: 1.0



Lare rate = 5  
Regularization  
n L1 = 0.001



Layers:

Loss function:

Batch size:

Learn rate:

Regularization L1:

Regularization L2:

Output layer activation type:

Epoch count:

Neurons per layer:

Layer 1 activation type:

Layer 2 activation type:

Layer 3 activation type:

Layer 4 activation type:

## Вывод

В процессе выполнения лабораторной, я “поигрался” с настоящей нейросетью, с целью оптимизировать гиперпараметры для улучшения результатов работы сети, на наборе данных из библиотеки Keras. Данная лабораторная работа является интересной точкой старта в data science. Также увидел, как влияют некоторые параметры на результаты (например, LR не может быть отрицательным). Для выполнения использовал Google Colab.