

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет Программной Инженерии и Компьютерной Техники

Системы искусственного интеллекта

Лабораторная работа № 4

Выполнил студент

Неизвестная Екатерина Павловна

Группа № Р33701

Преподаватель: Полещук Елизавета Александровна

г. Санкт-Петербург

2021

Вариант: 1

Задание:

Изучить гиперпараметры нейронной сети. Понять влияние различных параметров на точность.

Отчет:

Часть 1.

Изначальные данные:

Слоес: 2

Loss function:

Binary crossentropy

Batch size: 100

Learn rate:

-0.1	0	0.001	0.01	0.05	0.1	0.5	1	5
------	---	-------	------	------	-----	-----	---	---

Regularization L1:

-0.1	0	0.0001	0.0005	0.001	0.005	0.01	0.05	0.1
------	---	--------	--------	-------	-------	------	------	-----

Regularization L2:

-0.1	0	0.0001	0.0005	0.001	0.005	0.01	0.05	0.1
------	---	--------	--------	-------	-------	------	------	-----

Output layer activation type:

sigmoid

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:

relu

Layer 2 activation type:

relu

Layer 3 activation type:

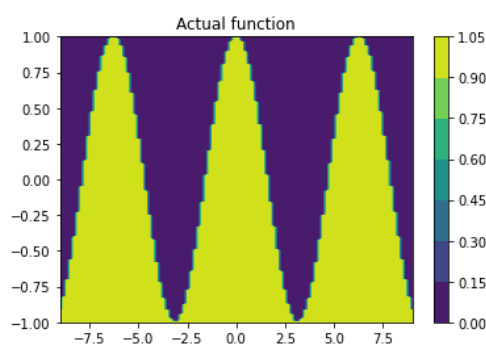
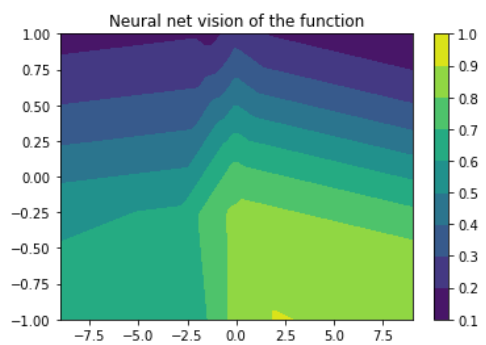
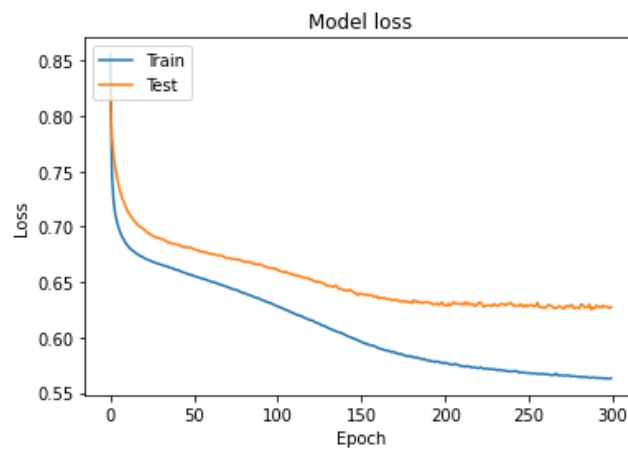
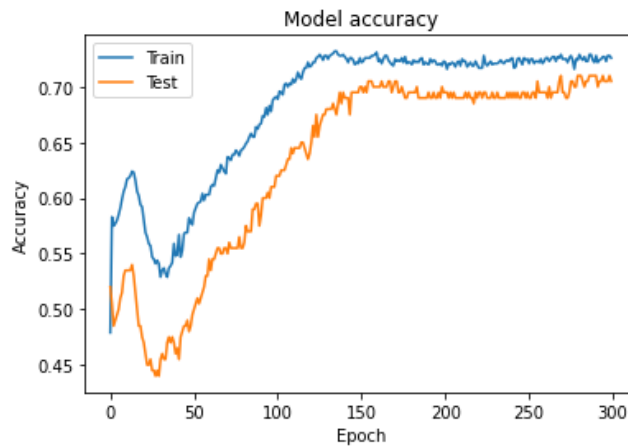
relu

Layer 4 activation type:

relu

Run Interact

Accuracy: 0.7049999833106995



- 1) Когда я меняла Neurons count in layer 1 (3 строчка): на более высокое значение, точность уменьшилась

2) При таких параметрах:

Epoch count: 300

Neurons count in layer 1: 7

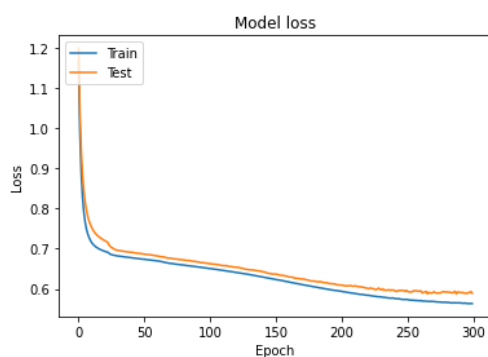
Neurons count in layer 2: 5

Neurons count in layer 1: 0

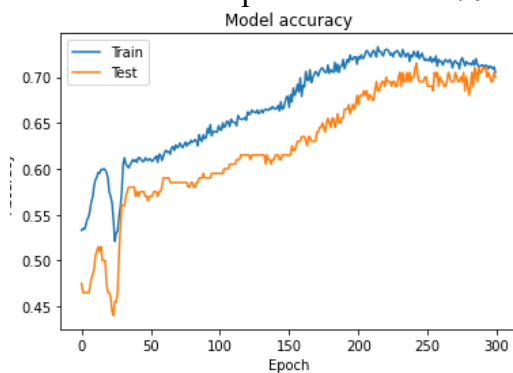
Neurons count in layer 1: 0

Layer 1 activation type:

Я почти до идеала довела график:



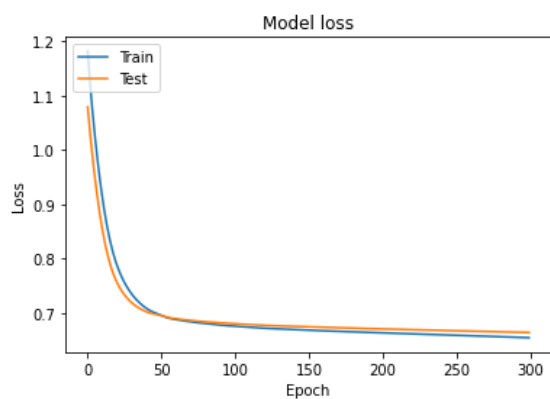
Но точность всё-равно осталась 70:



3) Когда я изменила learn rate:

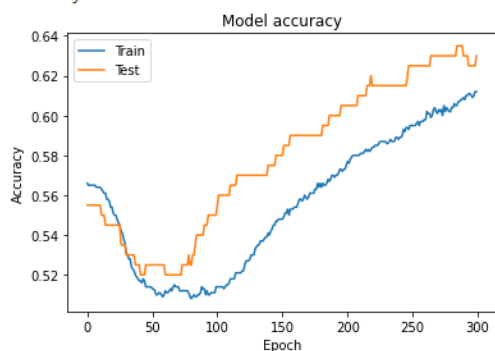
Learn rate:	-0.1	0	0.001	0.01	0.05	0.1	0.5
Regularization L1:	-0.1	0	0.0001	0.0005	0.001	0.005	

Второй график стал еще идеальнее!



Но первый.....:

Accuracy: 0.6299999952316284



И точность уменьшилась(

4) При таких параметрах:

Onnx: ☐ 2

Loss function:

Batch size:

Learn rate:

Regularization L1:

Regularization L2:

Output layer activation type:

Epoch count:

Neurons count in layer 1:

Neurons count in layer 2:

Neurons count in layer 3:

Neurons count in layer 4:

Layer 1 activation type:

Layer 2 activation type:

Layer 3 activation type:

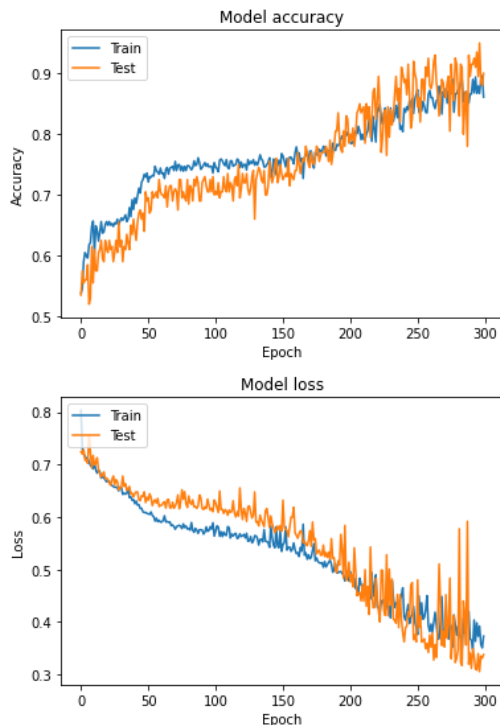
Layer 4 activation type:

Run interact

Accuracy: 0.900000007051581475

Я получила точность 90% !!!!!!! И точность ошибки стремится к 0!!

Accuracy: 0.8999999761581421



Это был лучший результат, который мне удалось получить

Часть 2.

Вариант: CIFAR100 small images classification dataset

Здесь представлены такие гиперпараметры, как

- Скорость обучения
- Регуляризация L1

Задание:

- 1) Изменяя эти гиперпараметры, попытайтесь достичь максимального значения точности (не менее 0,95) для модели Part2 с фиксированным количеством эпох 20
- 2) Измените значение 1-го гиперпараметра с минимального на максимальное с минимальным шагом, в зависимости от вашего варианта
- 3) Показать влияние на результат с помощью графиков
- 4) Опишите влияние каждого гиперпараметра на точность.
- 5) Установите значение гиперпараметра обратно на значение, обеспечивающее максимальную точность

- 6) Повторите 2-5 шагов для второго гиперпараметра

Сделайте отчет, включающий:

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

Отчет:

Поменяем параметры на такие:

Layers: 2

Loss function: Standard deviation

Batch size: 1000

Learn rate: 0.001

Regularization L1: 0.0001

Regularization L2: 0.0001

Output layer activation type: sigmoid

Epoch count: 20

Neurons per layer: 6

Layer 1 activation type: relu

Layer 2 activation type: relu

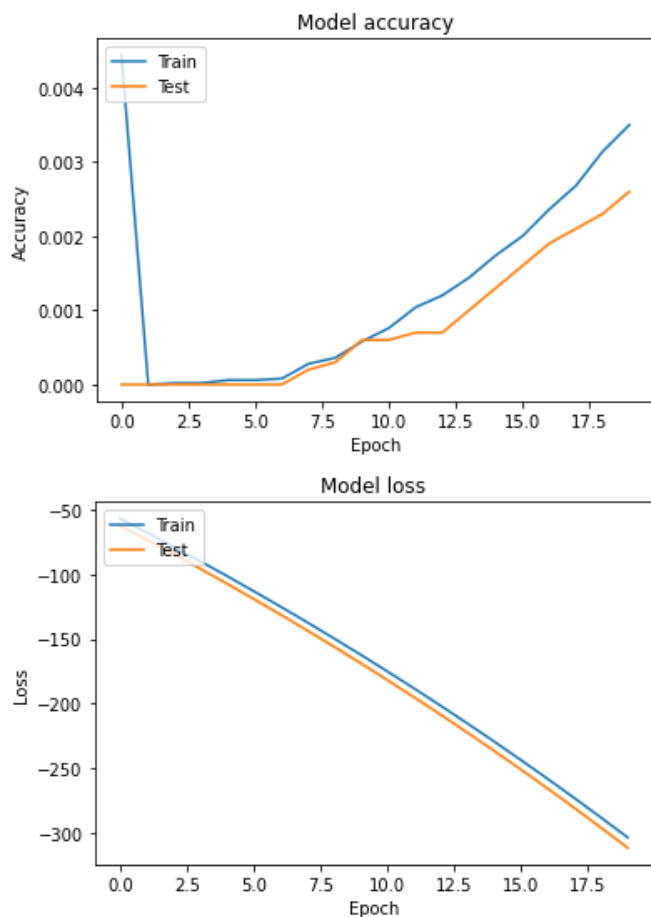
Layer 3 activation type: relu

Layer 4 activation type: relu

Run Interact

```
/usr/local/lib/python3.7/dist-packages/keras/optimizer_v2/optimizer_v2.py:356: UserWarning: The 'lr' argument is deprecated, use 'learning_rate' instead.
"The 'lr' argument is deprecated, use 'learning_rate' instead."
Epoch 1/20
1/50 [.....] - ETA: 24s - loss: -51.5515 - accuracy: 0.2220
3/50 [>.....] - ETA: 1s - loss: -51.7830 - accuracy: 0.0740
6/50 [==>....] - ETA: 1s - loss: -52.1196 - accuracy: 0.0370
9/50 [====>..] - ETA: 0s - loss: -52.4499 - accuracy: 0.0247
```

В результате получили такие графики:



Точность стремится к 1, но очень медленно. Попробуем увеличивать скорость обучения:

Loss function: Standard deviation

Batch size: 1000

Learn rate: -0.1 0 0.001 0.01 0.05 0.1 0.5 1 5

Regularization L1: -0.1 0 0.0001 0.0005 0.001 0.005 0.01 0.05 0.1

Regularization L2: -0.1 0 0.0001 0.0005 0.001 0.005 0.01 0.05 0.1

Output layer activation type: sigmoid

Epoch count: 20

Neurons per layer: 6

Layer 1 activation type: relu

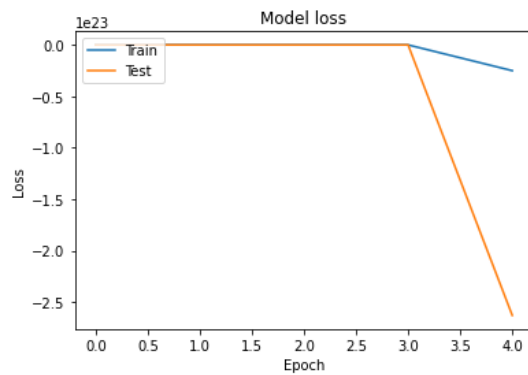
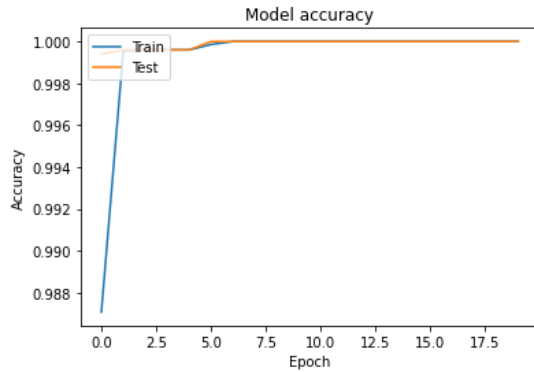
Layer 2 activation type: relu

Layer 3 activation type: relu

Layer 4 activation type: relu

Run Interact

50/50 [=====] - 1s 22ms/step - loss: nan
Accuracy: 1.0



Стало всё не очень хорошо, раньше было лучше....

И на максимальной скорости вообще всё сломалось..

Layers: 2

Loss function: Standard deviation

Batch size: 1000

Learn rate: -0.1 0 0.001 0.01 0.05 0.1 0.5 1 5

Regularization L1: -0.1 0 0.0001 0.0005 0.001 0.005 0.01 0.05 0.1

Regularization L2: -0.1 0 0.0001 0.0005 0.001 0.005 0.01 0.05 0.1

Output layer activation type: sigmoid

Epoch count: 20

Neurons per layer: 6

Layer 1 activation type: relu

Layer 2 activation type: relu

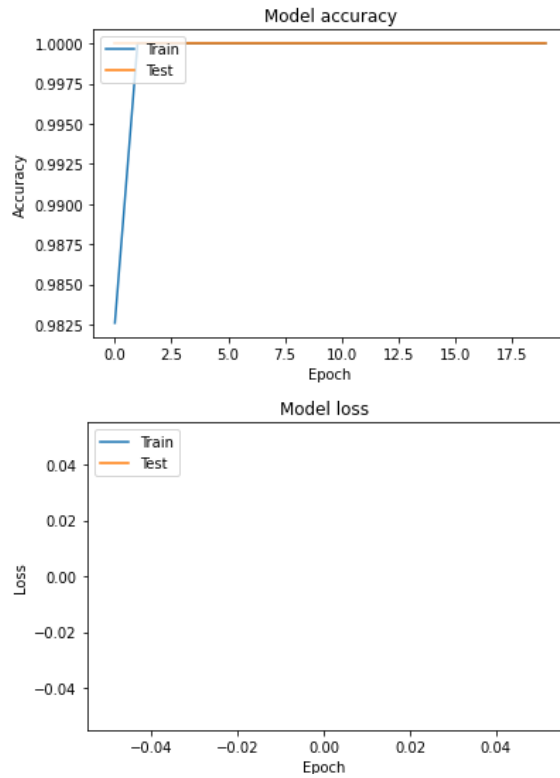
Layer 3 activation type: relu

Layer 4 activation type: relu

Run Interact

Epoch 1/20

50/50 [=====] - 1s 21ms/step - loss: 0.0000
Accuracy: 1.0



Попробуем поменять Regularization 1:

Layers:

Loss function:

Batch size:

Learn rate:

Regularization L1:

Regularization L2:

Output layer activation type:

Epoch count:

Neurons per layer:

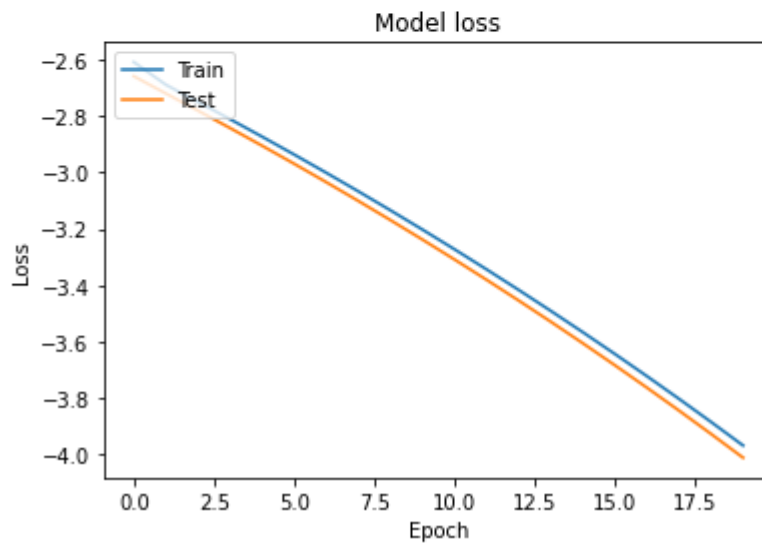
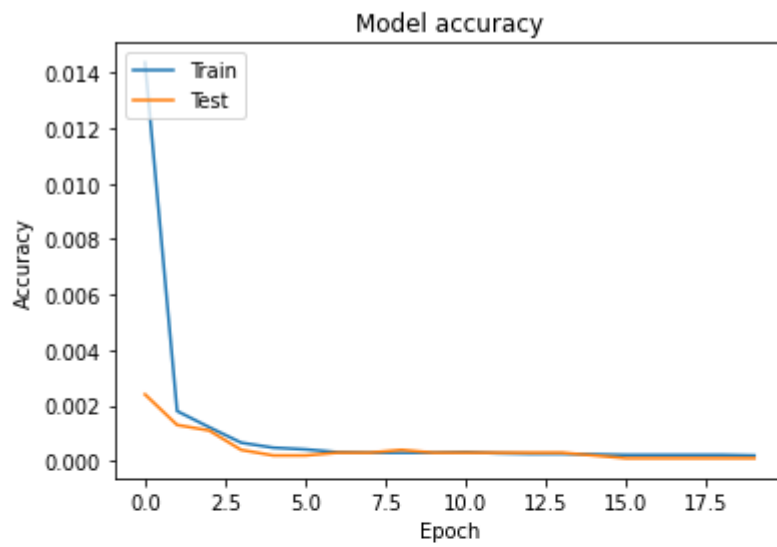
Layer 1 activation type:

Layer 2 activation type:

Layer 3 activation type:

Layer 4 activation type:

При этом графики получились:



И еще разок:

Layers:

Loss function:

Batch size:

Learn rate:

Regularization L1:

Regularization L2:

Output layer activation type:

Epoch count:

Neurons per layer:

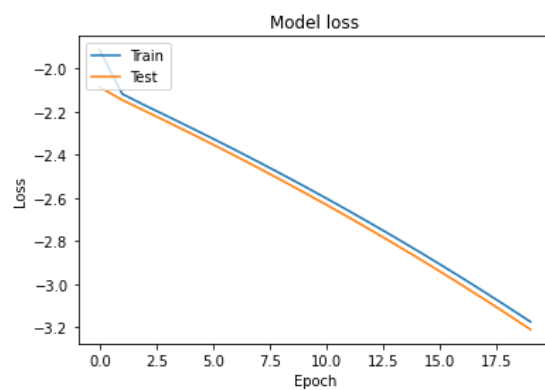
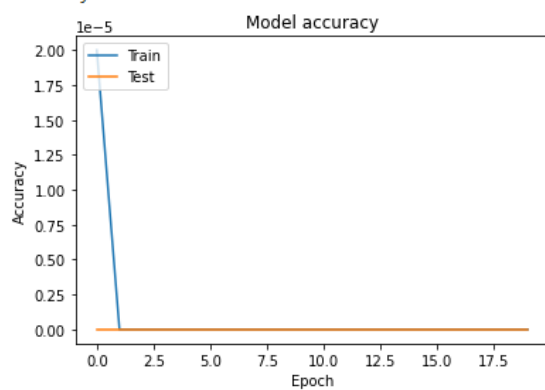
Layer 1 activation type:

Layer 2 activation type:

Layer 3 activation type:

Layer 4 activation type:

Accuracy: 0.0



Попробуем еще раз, поменять на максимум, вдруг что-то изменится...

Layers:

Loss function:

Batch size:

Learn rate:

Regularization L1:

Regularization L2:

Output layer activation type:

Epoch count:

Neurons per layer:

Layer 1 activation type:

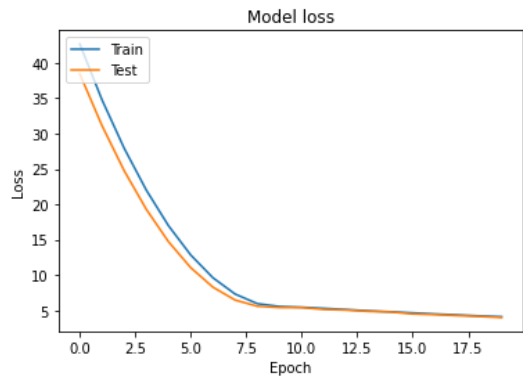
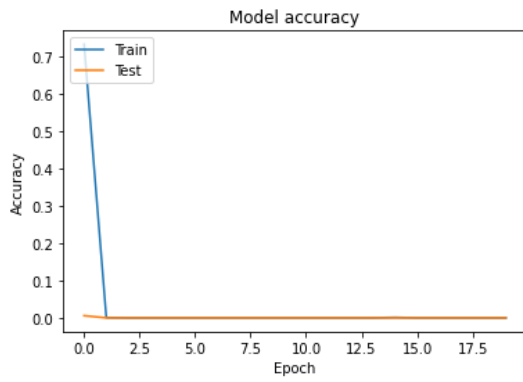
Layer 2 activation type:

Layer 3 activation type:

Layer 4 activation type:

Epoch 1/20

Accuracy: 0.0



Ничего не получилось, наша точность стала стремиться к нулю

Но есть еще Regularization 2!

Layers: 2

Loss function: Standard deviation

Batch size: 1000

Learn rate: 0.001

Regularization L1: 0.0001

Regularization L2: 0.0005

Output layer activation type: sigmoid

Epoch count: 20

Neurons per layer: 6

Layer 1 activation type: relu

Layer 2 activation type: relu

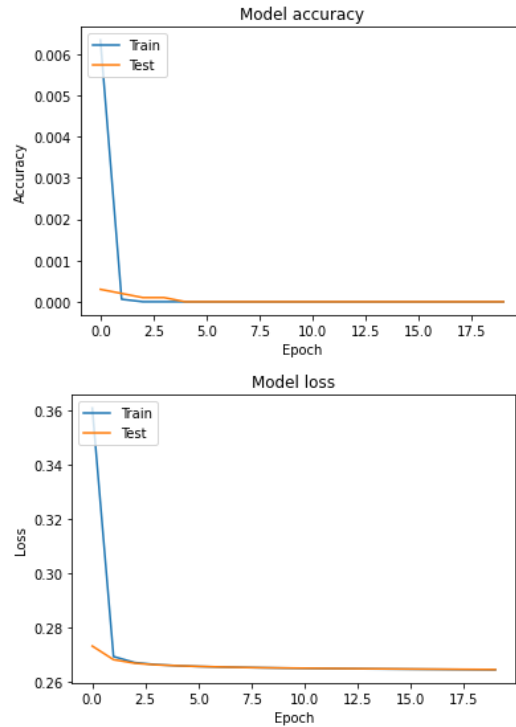
Layer 3 activation type: relu

Layer 4 activation type: relu

Run Interact

Epoch: 1/20
1 - ETA: 17s - loss: 0.0007 - accuracy: 0.0000

50/50 [=====] - 1s 22ms/step - loss:
Accuracy: 0.0



Снова всё плохо, значит лучше всего оставлять Regularization на минимум, а скорость обучения на середине.

Вывод:

Я делала лабораторную в Google Collab. Я узнала, что такое нейронная сеть, как происходит процесс ее обучения и какие параметры есть и как они влияют на точность и ошибку.