

PS4 Raport Piotr Szewczul

1.

Batch size = 1

```
Epoch 1/3
60000/60000 [=====] - 103s 2ms/step - loss: 0.2209 - accuracy: 0.9346
Epoch 2/3
60000/60000 [=====] - 102s 2ms/step - loss: 0.1389 - accuracy: 0.9637
Epoch 3/3
60000/60000 [=====] - 101s 2ms/step - loss: 0.1248 - accuracy: 0.9691
313/313 [=====] - 1s 1ms/step - loss: 0.1733 - accuracy: 0.9683
0.1732817441225052
0.9682999849319458
CPU times: total: 52min 5s
Wall time: 5min 6s
```

Batch size = 10

```
Epoch 1/3
6000/6000 [=====] - 14s 2ms/step - loss: 0.2213 - accuracy: 0.9323
Epoch 2/3
6000/6000 [=====] - 13s 2ms/step - loss: 0.0992 - accuracy: 0.9689
Epoch 3/3
6000/6000 [=====] - 16s 3ms/step - loss: 0.0683 - accuracy: 0.9782
313/313 [=====] - 1s 3ms/step - loss: 0.0872 - accuracy: 0.9735
0.0872327908873558
0.9735000133514404
CPU times: total: 5min 21s
Wall time: 43.8 s
```

Batch size = 50

```
Epoch 1/3
1200/1200 [=====] - 6s 4ms/step - loss: 0.0286 - accuracy: 0.9915
Epoch 2/3
1200/1200 [=====] - 5s 4ms/step - loss: 0.0199 - accuracy: 0.9943
Epoch 3/3
1200/1200 [=====] - 6s 5ms/step - loss: 0.0167 - accuracy: 0.9948
313/313 [=====] - 1s 2ms/step - loss: 0.0796 - accuracy: 0.9779
0.07962647080421448
0.9779000282287598
CPU times: total: 2min 19s
Wall time: 17.1 s
```

Zwiększanie rozmiaru batch size zdecydowanie zmniejsza czas wymagany na naukę modelu, jednocześnie zwiększając nieznacznie dokładność. Prawdopodobnie przekroczenie określonego progu wpływa negatywnie na dokładność modelu, nie udało się tego jednak wykazać na zastosowanym zbiorze danych.

2.

2.1 Różne metody aktywacji

a) Relu

```
Epoch 1/3
6/6 [=====] - 1s 84ms/step - loss: 0.0668 - accuracy: 0.9814
Epoch 2/3
6/6 [=====] - 0s 74ms/step - loss: 0.0653 - accuracy: 0.9814
Epoch 3/3
6/6 [=====] - 0s 75ms/step - loss: 0.0639 - accuracy: 0.9820
313/313 [=====] - 1s 2ms/step - loss: 0.1039 - accuracy: 0.9678
0.10388515889644623
0.9678000211715698
CPU times: total: 19 s
Wall time: 2.53 s
```

b) Sigmoid

```
Epoch 1/3
6/6 [=====] - 1s 88ms/step - loss: 2.3404 - accuracy: 0.0973
Epoch 2/3
6/6 [=====] - 1s 87ms/step - loss: 2.2650 - accuracy: 0.1651
Epoch 3/3
6/6 [=====] - 0s 75ms/step - loss: 2.2251 - accuracy: 0.2270
313/313 [=====] - 1s 2ms/step - loss: 2.1915 - accuracy: 0.4808
2.1914899349212646
0.48080000281333923
CPU times: total: 19.2 s
Wall time: 3.25 s
```

c) Elu

```
Epoch 1/3
6/6 [=====] - 1s 85ms/step - loss: 2.0610 - accuracy: 0.4274
Epoch 2/3
6/6 [=====] - 0s 81ms/step - loss: 1.4730 - accuracy: 0.7635
Epoch 3/3
6/6 [=====] - 0s 78ms/step - loss: 1.0128 - accuracy: 0.8073
313/313 [=====] - 1s 2ms/step - loss: 0.8075 - accuracy: 0.8333
0.8075413107872009
0.833299994468689
CPU times: total: 21.1 s
Wall time: 3.43 s
```

2.2 Różna ilość warstw/neuronów

a) 2 warstwy ukryte / 128 neuronów

```
Epoch 1/3
60/60 [=====] - 1s 14ms/step - loss: 0.4724 - accuracy: 0.8755
Epoch 2/3
60/60 [=====] - 1s 15ms/step - loss: 0.2745 - accuracy: 0.9199
Epoch 3/3
60/60 [=====] - 1s 13ms/step - loss: 0.2192 - accuracy: 0.9368
313/313 [=====] - 1s 2ms/step - loss: 0.1993 - accuracy: 0.9420
0.19925925135612488
0.9419999718666077
CPU times: total: 27.8 s
Wall time: 3.66 s
```

b) 3 warstwy ukryte / 128 neuronów

```

Epoch 1/3
60/60 [=====] - 2s 15ms/step - loss: 0.8837 - accuracy: 0.7820
Epoch 2/3
60/60 [=====] - 1s 13ms/step - loss: 0.2748 - accuracy: 0.9190
Epoch 3/3
60/60 [=====] - 1s 11ms/step - loss: 0.2111 - accuracy: 0.9383
313/313 [=====] - 1s 2ms/step - loss: 0.1944 - accuracy: 0.9416
0.19442929327487946
0.9416000247001648
CPU times: total: 28.2 s
Wall time: 4.12 s

```

c) 7 warstw ukrytych / 128 neuronów

```

Epoch 1/3
60/60 [=====] - 3s 33ms/step - loss: 0.9522 - accuracy: 0.6789
Epoch 2/3
60/60 [=====] - 2s 27ms/step - loss: 0.2595 - accuracy: 0.9210
Epoch 3/3
60/60 [=====] - 2s 34ms/step - loss: 0.1758 - accuracy: 0.9469
313/313 [=====] - 1s 2ms/step - loss: 0.1632 - accuracy: 0.9501
0.16319994628429413
0.9501000046730042
CPU times: total: 58.8 s
Wall time: 7.83 s

```

d) 256 neuronów / 2 warstwy ukryte

```

Epoch 1/3
60/60 [=====] - 2s 16ms/step - loss: 0.7325 - accuracy: 0.8228
Epoch 2/3
60/60 [=====] - 1s 14ms/step - loss: 0.2504 - accuracy: 0.9276
Epoch 3/3
60/60 [=====] - 1s 14ms/step - loss: 0.1866 - accuracy: 0.9457
313/313 [=====] - 1s 3ms/step - loss: 0.1716 - accuracy: 0.9498
0.1715609729290085
0.9498000144958496
CPU times: total: 35.5 s
Wall time: 4.71 s

```

e) 512 neuronów / 2 warstwy ukryte

```

[73]: %%time
model.fit(x_train, y_train, batch_size=1000, epochs=3) # train the model
val_loss, val_acc = model.evaluate(x_test, y_test) # evaluate the out of sample data with model
print(val_loss) # model's loss (error)
print(val_acc) # model's accuracy

```

```

Epoch 1/3
60/60 [=====] - 2s 31ms/step - loss: 0.5762 - accuracy: 0.8542
Epoch 2/3
60/60 [=====] - 2s 33ms/step - loss: 0.2043 - accuracy: 0.9416
Epoch 3/3
60/60 [=====] - 2s 31ms/step - loss: 0.1421 - accuracy: 0.9582
313/313 [=====] - 1s 3ms/step - loss: 0.1292 - accuracy: 0.9628
0.12924960255622864
0.9628000259399414
CPU times: total: 1min 2s
Wall time: 7.74 s

```

f) 1024 neurony / 2 warstwy ukryte

```
Epoch 1/3
60/60 [=====] - 6s 82ms/step - loss: 0.4607 - accuracy: 0.8746
Epoch 2/3
60/60 [=====] - 5s 83ms/step - loss: 0.1579 - accuracy: 0.9528
Epoch 3/3
60/60 [=====] - 5s 83ms/step - loss: 0.0981 - accuracy: 0.9711
313/313 [=====] - 2s 4ms/step - loss: 0.1041 - accuracy: 0.9695
0.10410168766975403
0.9695000052452087
CPU times: total: 2min 24s
Wall time: 17.6 s
```

Zwiększanie ilości warstw podobnie jak neuronów przekłada się na nieznaczny wzrost dokładności modelu w badanych przykładzie. Jednocześnie wzrasta również czas wymagany na przeliczenie. Rozbudowana sieć może przetożyć się na poprawę wyników zmniejszając jednocześnie wydajność, powinniśmy więc zdecydować nad czym nam bardziej zależy i dobrać odpowiednie parametry.