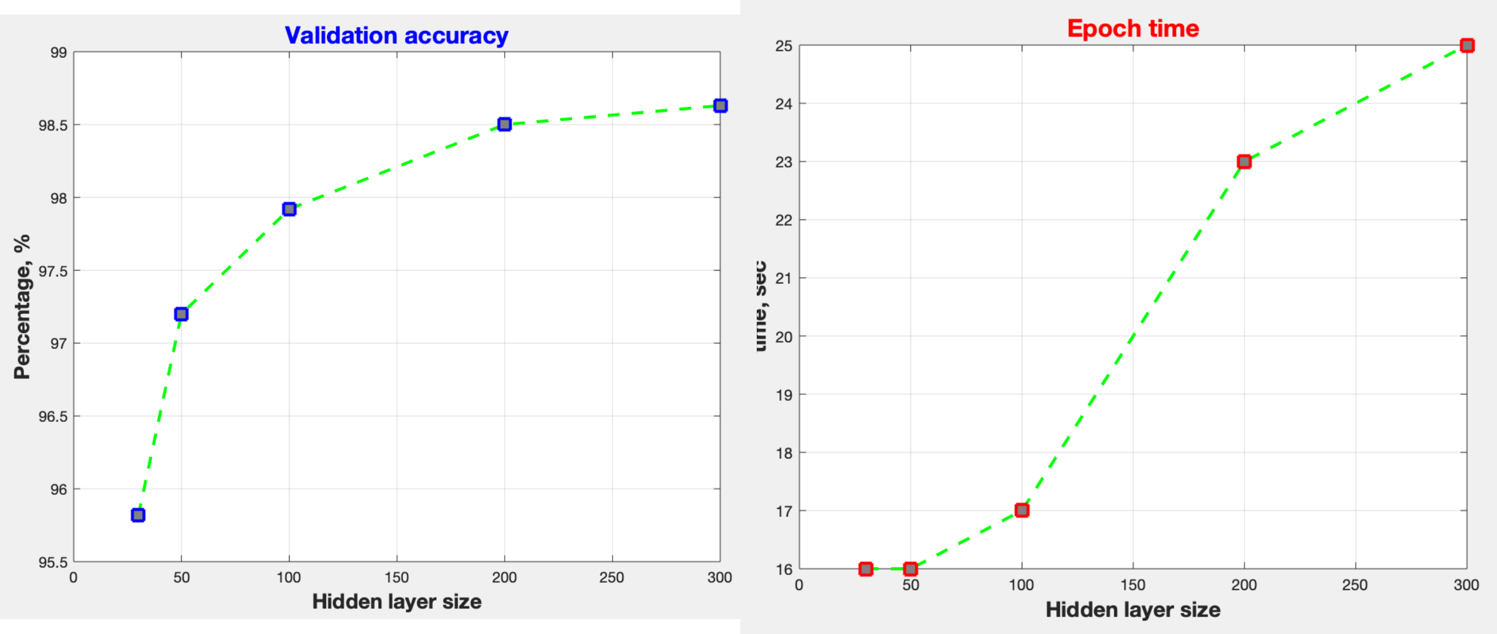
Mini Project II

Deep Learning with Tensorflow and Keras

(Baglan Aitu)

1. Hidden layers

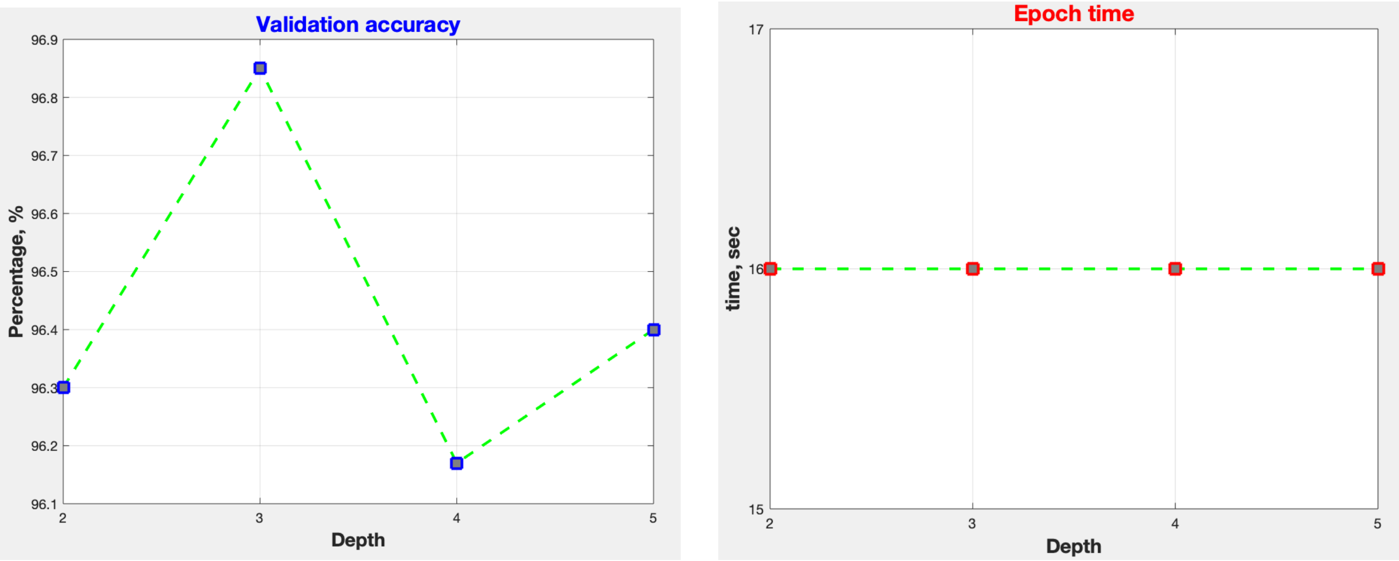


(5 epochs, depth is 2 hidden layers, buffer size = 10000, batch size = 100, relu-softmax)

As can be seen from the results above, the increase of hidden layer size is directly proportional to the increase of validation accuracy and epoch time.

The problem is the large size of the hidden layer can lead to overfitting, because in this work the solving methods like dropout weren't implemented. This is why the low size of the hidden layer is probably a good solution. Moreover, the difference is not too much.

2. Depth

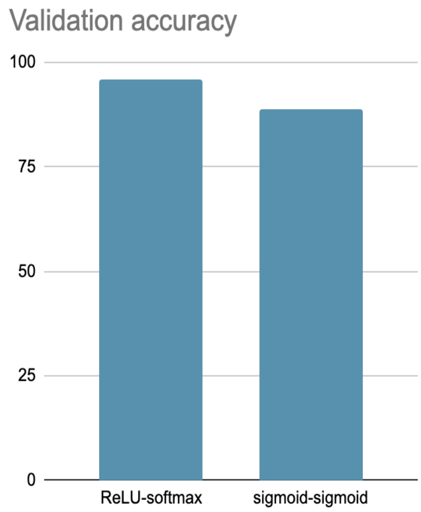


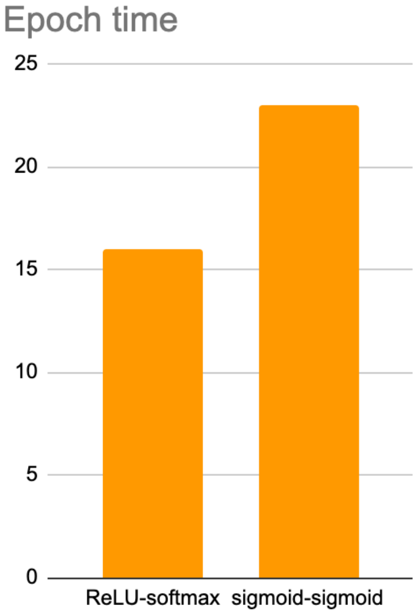
(5 epochs, hidden layer size = 30, buffer size = 10000, batch size = 100, relu-softmax)

Validation accuracy slightly changes between 96 and 97% for given parameters.
Epoch time is the same for 2-5 hidden layers (depth).

3. Activation functions

As it was in the 3rd part of exercise, I set 5 hidden layers. According to the article: https://heartbeat.fritz.ai/benchmarking-deep-learning-activation-functions-on-mnist-3d174e729735 . For MNIST task, the most effective activation functions are: ReLU-softmax, sigmoid-sigmoid:





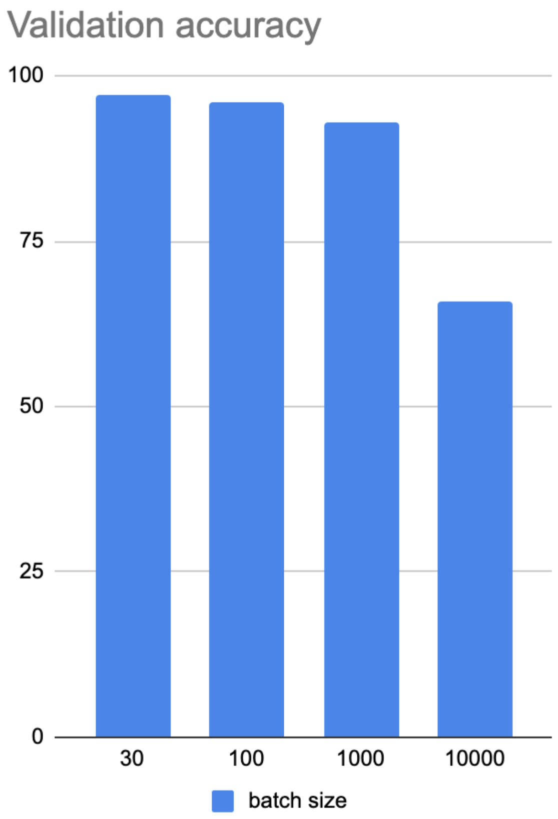
Sigmoid activation function didn't give good results.
I decided to stick to ReLU-softmax. It is also a possible combination of other functions like tanh to maintain more flexibility (nonlinearity).

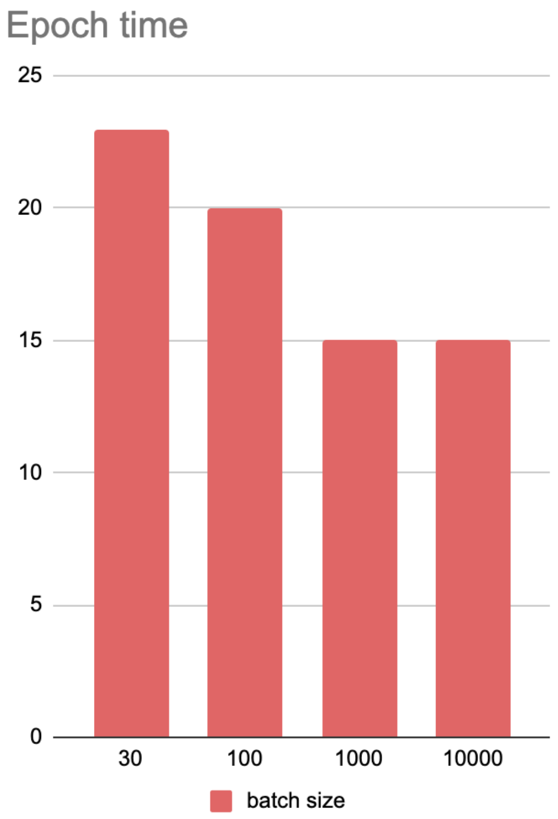
4. **Optimizer**

According to following article:
https://onlytojay.medium.com/mnist-cnn-optimizer-comparison-with-tensorflow-keras-163735862ecd

The best optimizers for given tasks are “adam” and “rmsprop”.
In my case, they gave almost the same result and the “Adam” optimizer was chosen.

5. **Batch size**





By increasing batch size, the accuracy will drop and epoch time decreases. In our case, it is better to keep a low value of batch size (30-50) as time difference is small.

6. **Batch size = 1**

It takes too much time to calculate. Results are worse. There was no point in continuing it.

7. **Learning rate**

Learning rate is necessary to stabilize the process of backpropagation. Its values should vary between 0.1 and 0.0001. The learning process will be better if the rate is low, but it affects the processing time. As I set a lower number of epochs (less time), the low learning rate is good for the given task.

8. Conclusion:

After analysis discussed above and playing with parameters, I achieved **100%** validation accuracy and epoch time **15 sec.** The validation loss is decreasing after each epoch which shows there is no case of overfitting.

The parameters:

- Learning rate = 0.001 (decreased).
- Buffer size = 10 000 (by default).
- batch size = 30 (decreased)
- Hidden layer size = 50 (by default).
- Optimizer: adam (by default)
- 3 epochs (decreased)
- Number of hidden layers = 5 (increased)
- Activation functions: relu-tanh-sigmoid-relu-relu-softmax (added new functions)