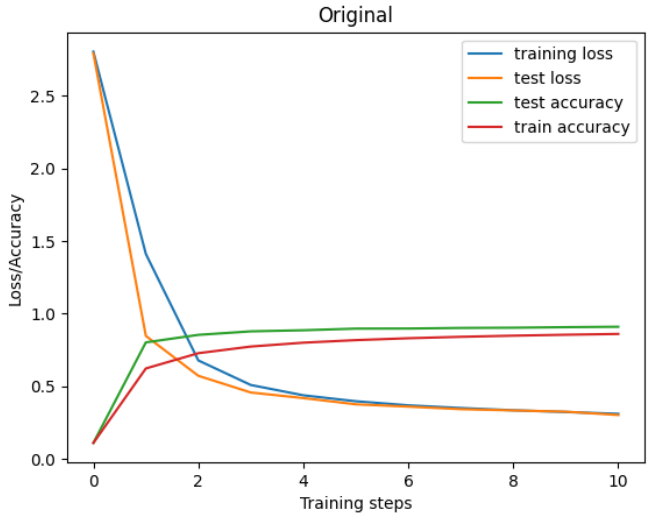
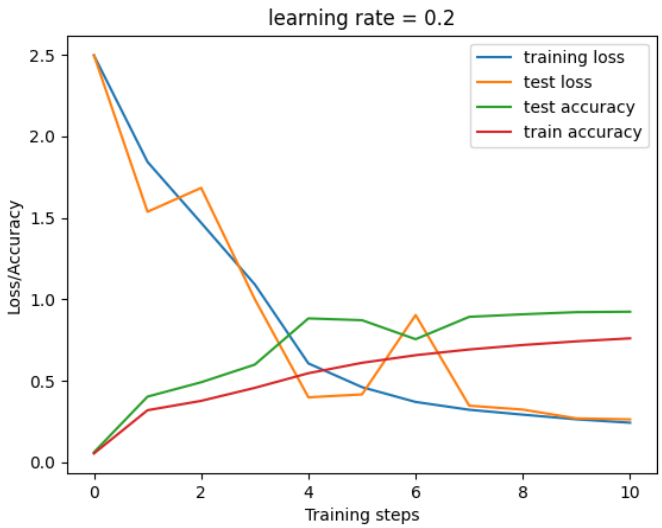
IANNwTF HW03 – Group 16

First, we want to show how each of the different hyperparameters learning rate, batch size, number of hidden layers , number of units per layer and optimizer influence the train losses, test losses, test accuracies and train accuracies.



learning rate = 0.1
batch size = 32
2 hidden layers
256 units per layer
10 epochs
SGD as optimizer

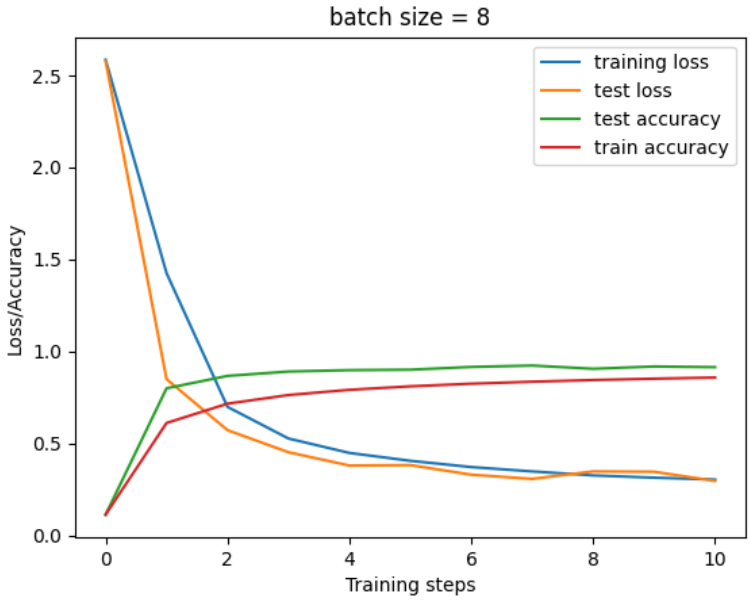
Learning rate:

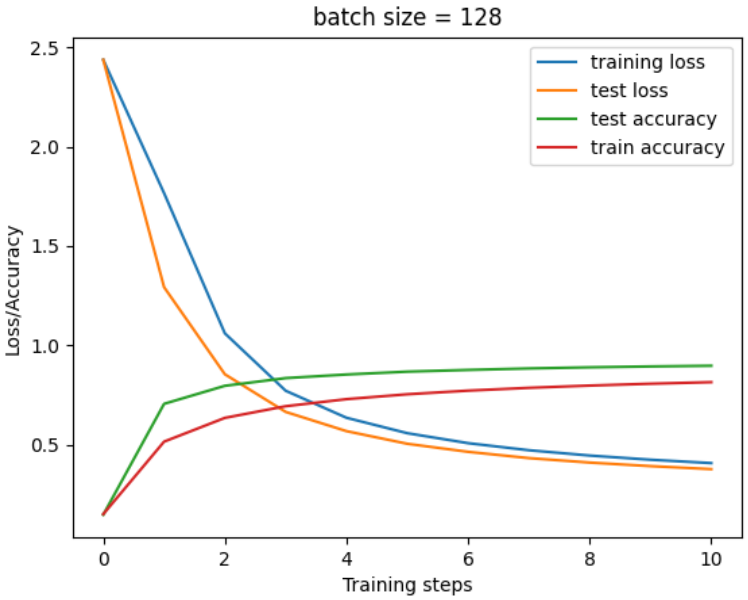


The learning rate defines how strongly a network updates its parameters.
As you can see, increasing the learning rate leads to making bigger steps in the direction of the gradient. Therefore a larger learning rate speeds up the learning but may not converge. We might miss the goal because the steps are too big.
Using a very small learning rate makes the process slower because we are making very small steps

into the direction of the gradient. Therefore we need to make more steps/epochs to get a high accuracy. However, we make sure that we don't miss the goal.

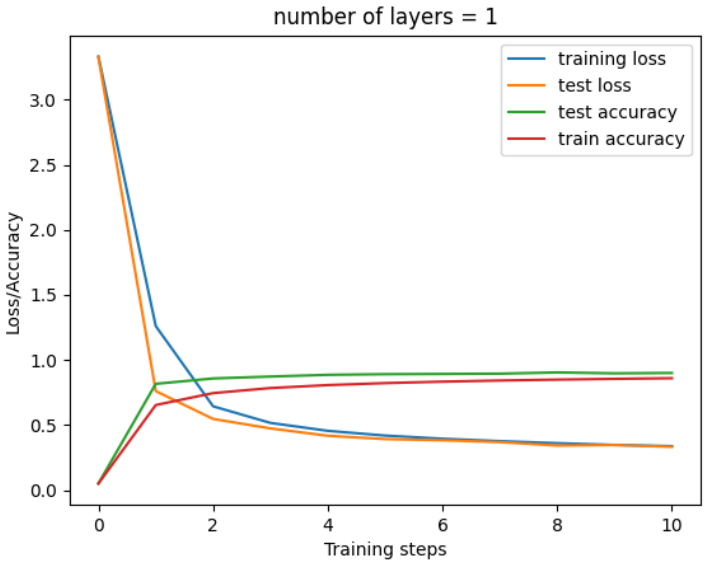
Batch size:

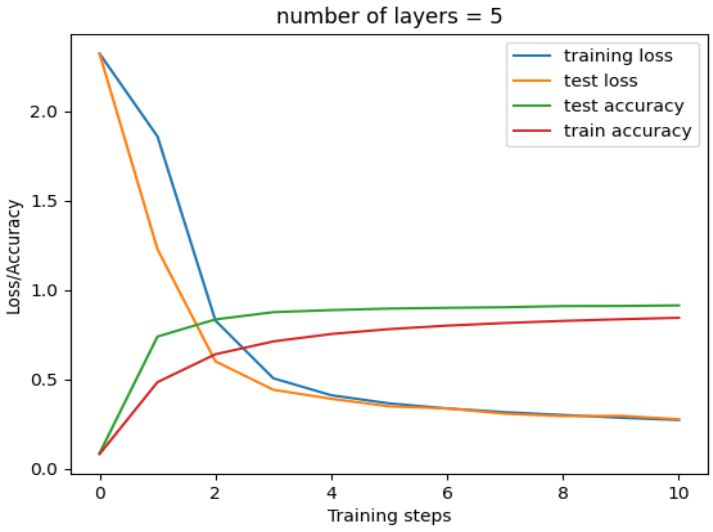




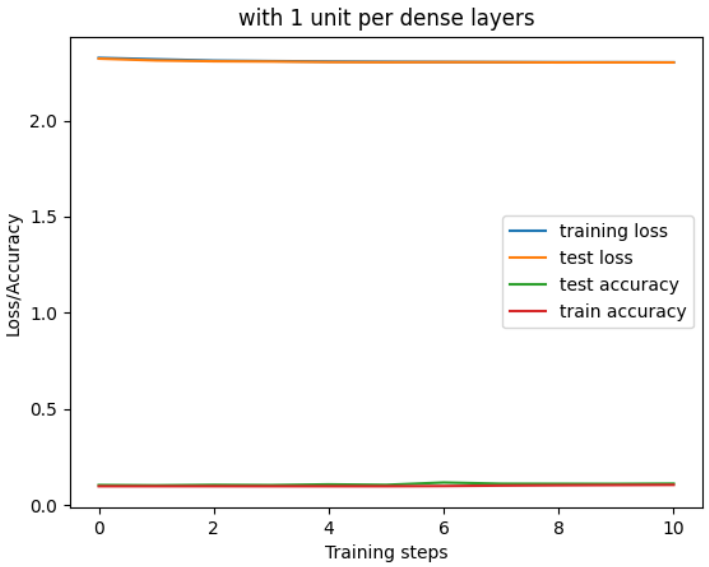
In the graphs we see that with a higher batch size we need more epochs to reach low losses and high accuracies, but we also see that the curves are more smoothly.
This is because a higher batch size leads to more accurate steps into the direction of the overall gradient. Also it takes less time because we test and train more data at once.
If we decrease the batch size we need to make more steps because we might not follow the direction of the overall gradient. Due to that we have more missteps and the graph is uneven. However, these missteps are also useful because we can escape local optima.

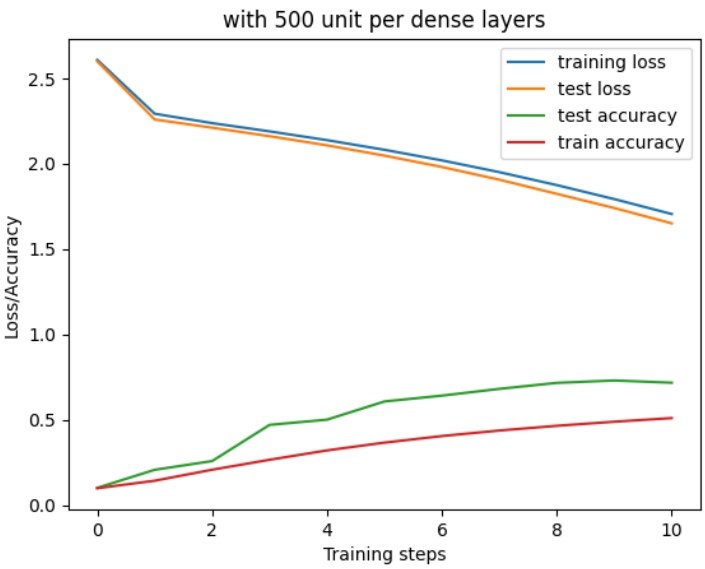
Number of hidden layers and units per layer:





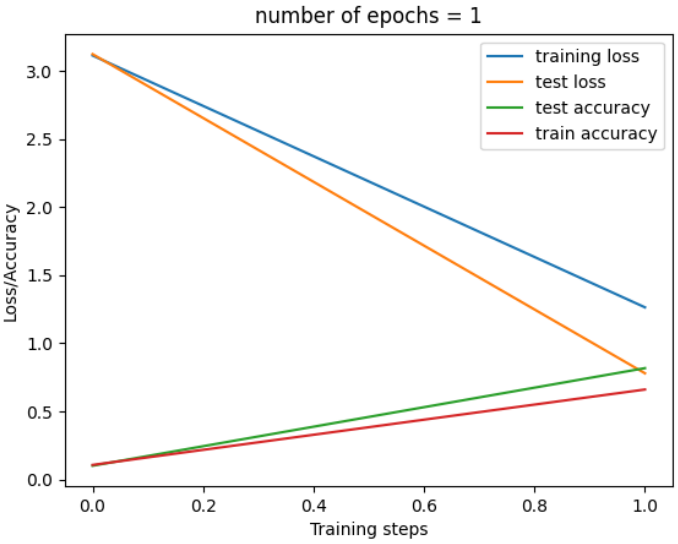
With less hidden layers we see that we have higher losses and lower accuracies. Too little layers can cause underfitting and too much can cause overfitting, both should be prevented by choosing a number of layers that is inside the optimal range between under- and overfitting, this range also depends on the context of the network and its inputs/targets. Besides that a higher number of layers need more time to calculate, so it can also be useful to look at the number of layers/accuracy ratio for efficiency reasons.

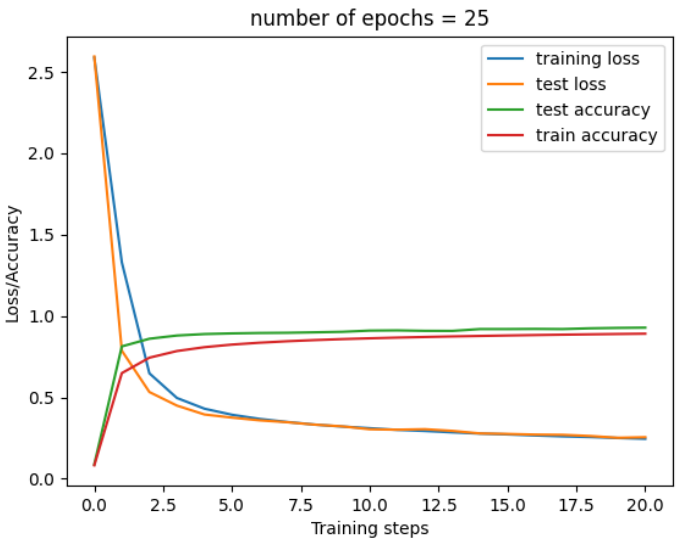




Many hidden units within a layer with regularization techniques can increase accuracy. Too many neurons in the hidden layers may result in overfitting. In this case, the case of overfitting is not clearly visible with a number of 5000 units per layer, so it still might be in the range of a good amount, but if we can achieve the same results with less units than it is probably more efficient/better to use less units. But smaller number of units may result in underfitting. The underfitting example is clearly visible in the upper plot with one unit per layer, there the accuracies are really low. Overall, it really depends on the content how much is too much or too little as before in number of layers.

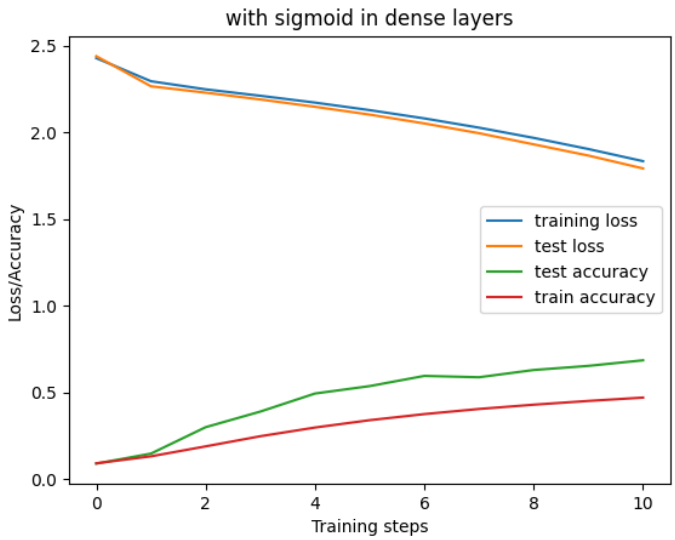
Number of epochs:





The number of epochs is the number of times the whole training data is shown to the network while training. We see that with a very low number of epochs our losses and accuracies are way worse than with more epochs. However, having too many epochs can lead to overfitting.

Sigmoid as activation:



We see that if we use sigmoid as the activation function we do not reach acceptable losses or accuracies. Therefore, using this activation function in our ANN does not make sense. This is because the sigmoid function “translates” a big input space into a number between 0 and 1 (a small input space). Due to that, large changes in the input turn to only small changes in the output and the derivative becomes small. This is especially problematic when the network has more than just a few layers: in backpropagation we multiply the derivatives and if they are all very small with each layer the gradient decreases further. If it becomes too small in the initial layers (which are often crucial for the performance of the network) the weights and biases will not be updated effectively and our network is inaccurate.