Project Imaging Group 10 – Assignment 2

Helgers, V.M.J. 1329332
Jong, de R.L.P.D. 1328328
Korsten, T. 1340522
Moharir, S. 1296256

Exercise 1

The example neural network classification model in this assignment is relatively simple — it contains a single hidden layer of 64 neurons. Perform a set of experiments with more complex models, e.g., with more layers (deeper models), more neurons per layer or a combination. Describe the set of experiments that you have performed. What is the accuracy of the best model? How did you determine which model is the best?

To evaluate the performance of the model the number of neurons is varied, and the number of layers is varied. In this exercise there has been chosen to create 4 additional models on top of the given model, with 1 layer, containing 64 neurons. These models were all made in Jupyter notebook.

It has been chosen to first create a model with 2 layers, still containing 64 neurons. Then a model with just 1 layer, but with 128 neurons instead of 64. The third model was a combination of the previous 2 model additions. The model had 2 layers, both consisting of 128 neurons. And finally, a model was created with 4 layers, containing 128 neurons per layer. The layers inside these models all contain ReLU activations.

The loss values and the accuracy values for all 5 of the models can be found in table 1 and 2 respectively. There are no values for the model with 4 layers, containing 64 neurons per layer, since this model has not been made. The other models showed that 128 neurons is more accurate so therefore it was decided not to create this model.

Table 1: Loss values for the created models

Nr. Layers / Nr. Neurons	**64**	**128**
1	0.184	0.173
2	0.144	0.124
4	-	0.099

Table 2: Accuracy values for the created models

Nr. Layers / Nr. Neurons	**64**	**128**
1	0.947	0.949
2	0.958	0.962
4	-	0.969

Figures 1 and 2 respectively show, for both the training and validation set, the accuracy and loss curves. The curves are shown for the original model, containing 1 layer with 64 neurons per layer and for the model containing 2 layers with 128 neurons per layer.

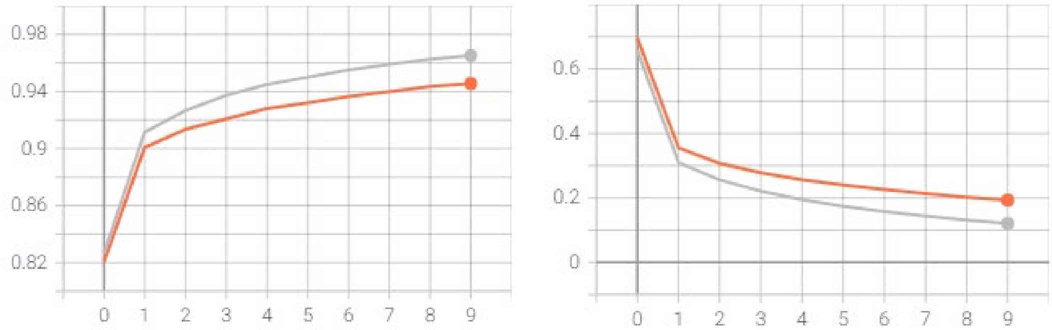

*Figure 1: Left: Accuracy of training set vs. epochs. Right: loss of training set vs. epochs. The model with 1 layer, 64 neurons per layer in **orange**, the model with 2 layers, 128 neurons per layer in **grey**.*

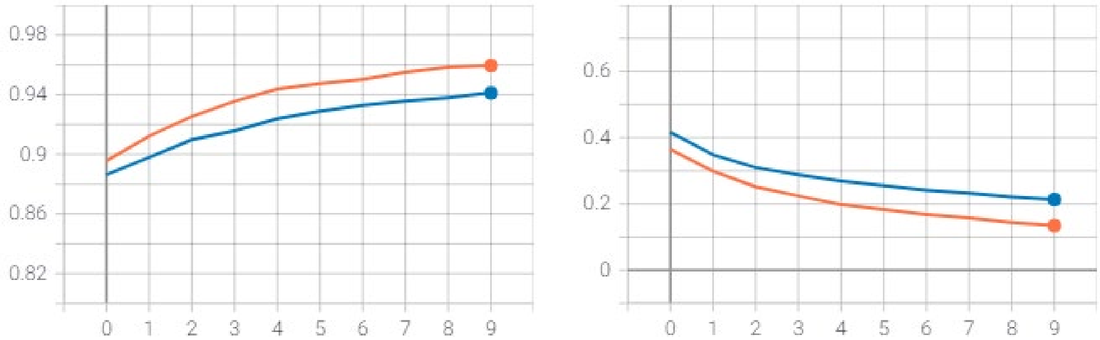

*Figure 2: Left: Accuracy of validation set vs. epochs. Right: loss of validation set vs. epochs. The model with 1 layer, 64 neurons per layer in **blue**, the model with 2 layers, 128 neurons per layer in **orange**.*

The figures show that a model with more layers and containing more neurons in those layers results in a higher accuracy and a lower loss for both the validation and the training set. The accuracy and loss curves for the remaining models can be found in the Appendix, which show the same result.

As can be seen from the results, presented in tables 1 and 2, the accuracy becomes higher, and the loss becomes lower the more layers and neurons are added. It can be seen that adding another layer has greater effect on the increase in accuracy and decrease of the loss than adding more neurons.

Concluding, the best model in this set of experiments is the model with 4 layers, containing 128 neurons each. The criteria for the best model were the loss and accuracy values, obtained during the evaluations of the models. For this best model, the loss value was 0.099, and the accuracy value was 0.969. Adding on more layers increases the accuracy and decreases the loss. These values, however, will start to decrease and increase less with each addition of layers. It is not a linear continuity, with each additional layer the accuracy and the loss converges.

Exercise 2

Compare the performance of the following three models:

1. **Neural network without any hidden layers (the input layer connects directly to the output layer).**
2. **Neural network with 3 hidden layers with ReLU activations.**
3. **Neural network with 3 hidden layers with linear activations (i.e., without nonlinearities between the layers).**

Analyze the performance of the three models. What is the reason behind the difference in performance between the second and third models?

The 3 neural network models as posed in question 2 were made in Jupyter Notebook. The validation loss and the accuracy for the 3 cases is shown in table 3.

Table 3: Validation Loss values and accuracy values for the 3 different questions in exercise 2.

	Validation loss	Accuracy
Question 1	0.310	0.914
Question 2	0.119	0.964
Question 3	0.281	0.922

It can be observed from table 3 that the model of question 1 has the highest validation loss and the lowest accuracy. This can be explained by the fact that this model has no hidden layers, whereas the other two models have 3 hidden layers. The hidden layers in the models of question 2 and 3 make the models more complex which results in better loss and accuracy values.

It can also be seen from table 3 that the model of question 2 has a lower validation loss and a higher accuracy than the model of question 3. This in turn can be explained by the fact that the model in question 2 makes use of ReLU activations, whereas the model of question 3 makes use of linear activations. In the model of question 3 only linear layers are added which means that the final model will still be linear. In other words, it is not beneficial to add multiple linear layers. Therefore, the loss and accuracy of the model with 3 linear layers is almost the same as that of the model without any hidden layers. Thus, the conclusion is that the model with 3 hidden layers with ReLU activations shows the best performance.

In figure 3 the accuracy and loss of the training set is shown over the epochs for the 3 different models of the assignment. In figure 4 the same is shown for the validation set.

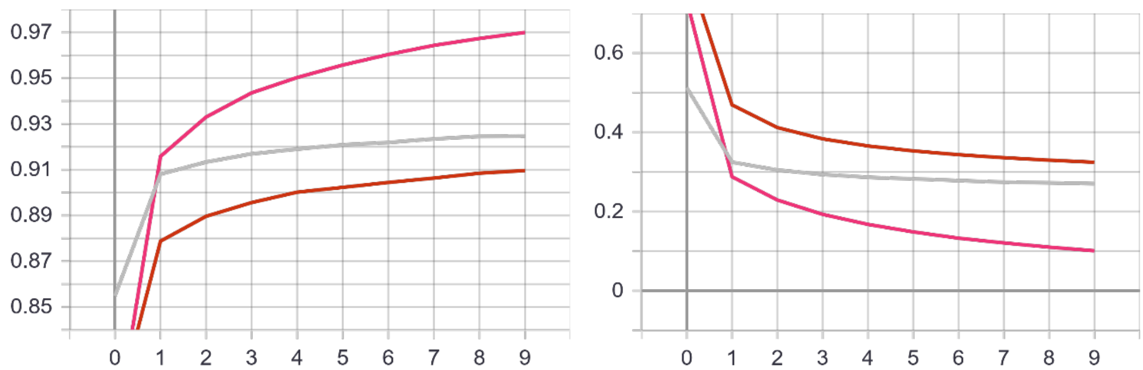

*Figure 3: Left: Accuracy of training set vs. epochs. Right: loss of training set vs. epochs. Model of question 1 in **red**, model of question 2 in **magenta**, model of question 3 in **grey**.*

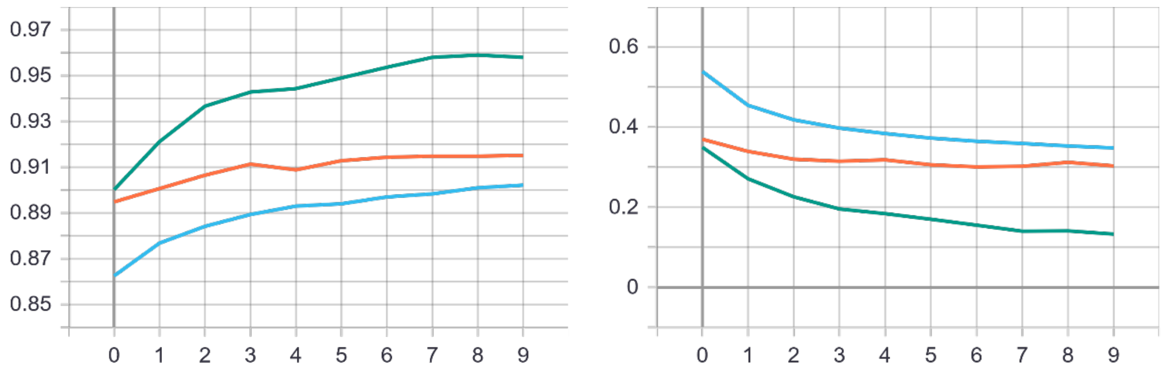

*Figure 4: Left: Accuracy of validation set vs. epochs. Right: loss of validation set vs. epochs. Model of question 1 in **light blue**, model of question 2 in **orange**, model of question 3 in **turquoise**.*

From figures 3 and 4 the same observations can be made as before; the model of question 2 performs best followed by the model of question 3 and 1 respectively. The curves for the training dataset have a similar course compared to that of the validation set. From the graphs it can be concluded that the models are not overfitting the data.

Exercise 3

Train a neural network model (the specific architecture is up to you) for a four-class classification problem derived from MNIST in the following way:

- **"vertical digits": 1, 7**
- **"loopy digits": 0, 6, 8, 9**
- **"curly digits": 2, 5**
- **"other": 3, 4**

For exercise 3, a code has been written in the notebook that was also in the submission. A function has been made to change from 10 categories to the 4 specified categories. This function was then applied to change the labels of the training, validation, and test set. Finally, a model has been made consisting of 3 layers with ReLU activations, with 128 neurons per layer.

Appendix

In this appendix the accuracy and loss curves are displayed of both the training and validation set, for the 3 remaining models.

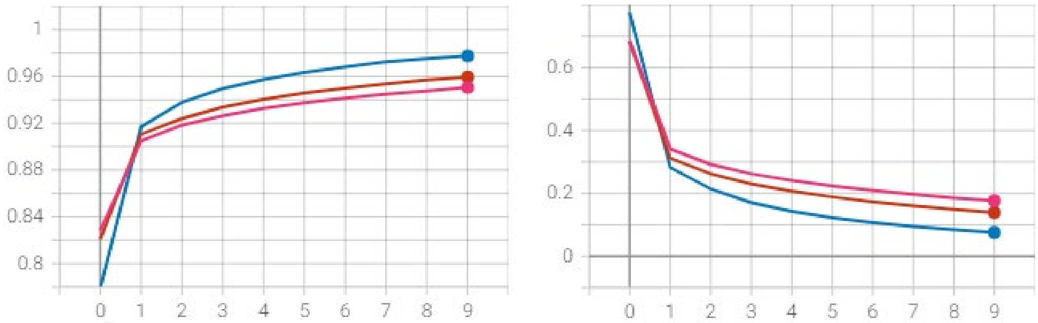

*Figure 5: Left: Accuracy of training set vs. epochs. Right: loss of training set vs. epochs. The model with 1 layer, 128 neurons per layer in **magenta**, the model with 2 layers, 64 neurons per layer in **red**, the model with 4 layers, 128 neurons per layer in **blue**.*

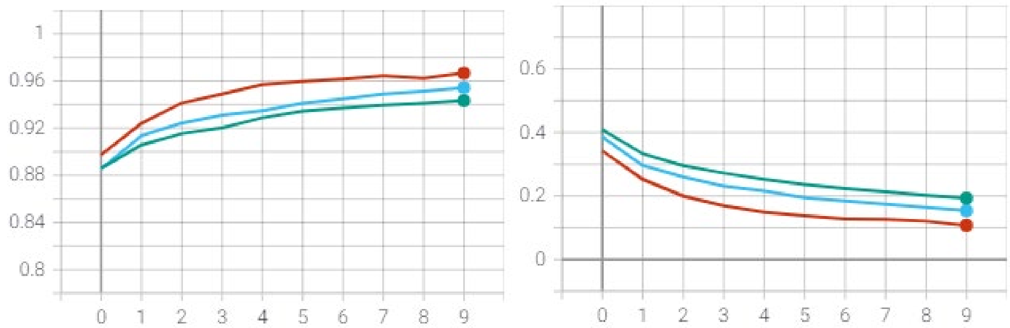

*Figure 6: Left: Accuracy of validation set vs. epochs. Right: loss of validation set vs. epochs. The model with 1 layer, 128 neurons per layer in **turquoise**, the model with 2 layers, 64 neurons per layer in **light blue**, the model with 4 layers, 128 neurons per layer in **red**.*