Deep Learning Lab 2
Neural Networks

Garben Tanghe
Olivier Van den Nest
Group 03

May 20, 2020

1 Analysis of the provided model

First, the provided model is analyzed.

After training the model on the 50000 samples in the training set and for 20 epochs, the training set accuracy is 97 % and the loss is 0.0952. When evaluating that model on the validation set, the validation set accuracy is 95 % and the validation set loss is 0.1712. After retraining that same model one last time on all 60000 samples in the training set, the training set accuracy is 97 % and the training set loss is 0.0858. When evaluating that model on the test set, the test set accuracy is 96 % and the set test loss is 0.1638.

The validation curves are shown in Figure 1. There can be seen that the model is not complex enough and is not able to capture the ground truth just yet. A more complex model is thus required in the next step. The train set accuracy and loss for training on all 60000 samples of the provided model can be seen in Figure 2.

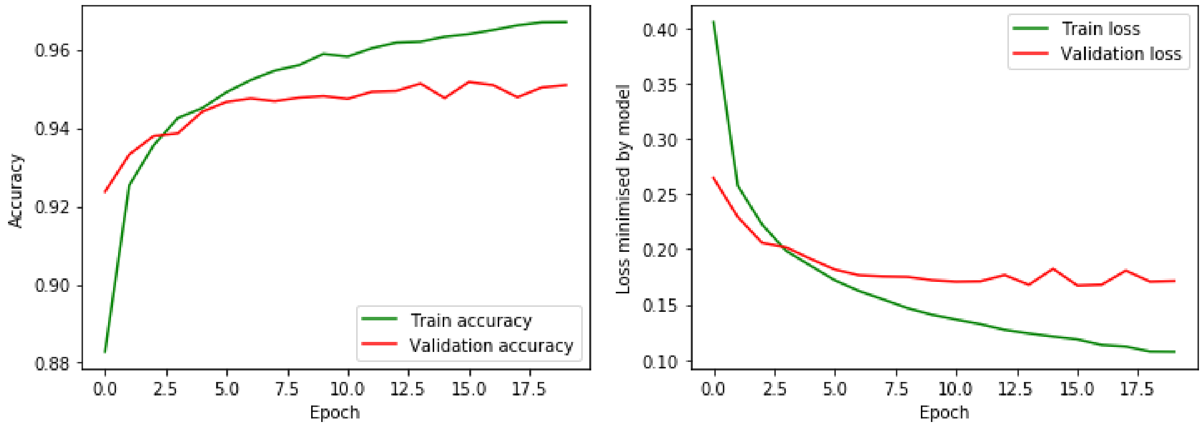


Figure 1: Validation curves for the provided model

Looking at the data could give more insights, so first, there is looked at some of the wrongly classified images. The first 9 miss-classifications are shown in Figure 3. Some of these digits are quite vague or do have some characteristics as samples from other

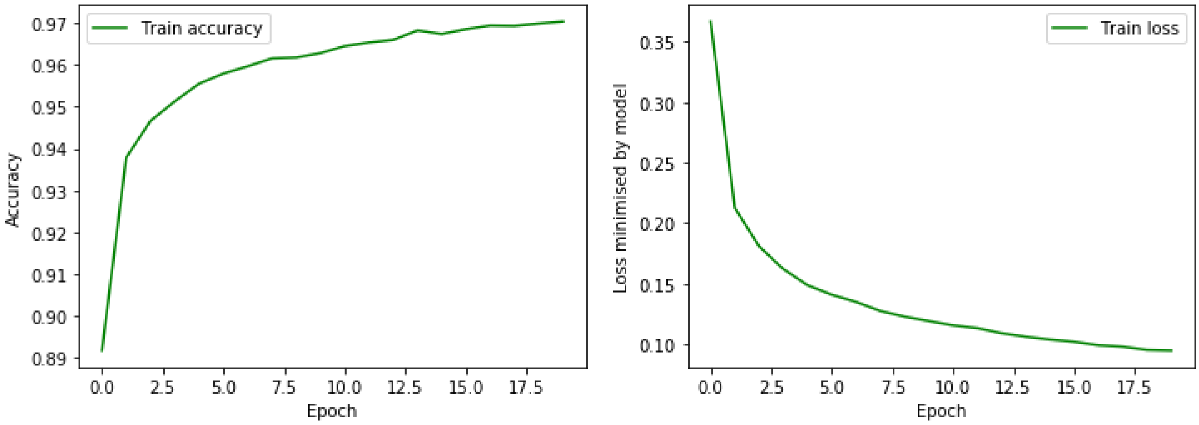


Figure 2: Validation curve for the provided model when training on the complete train set

classes. Even for us humans, it might be hard to be sure what the author of the character wanted to write down.

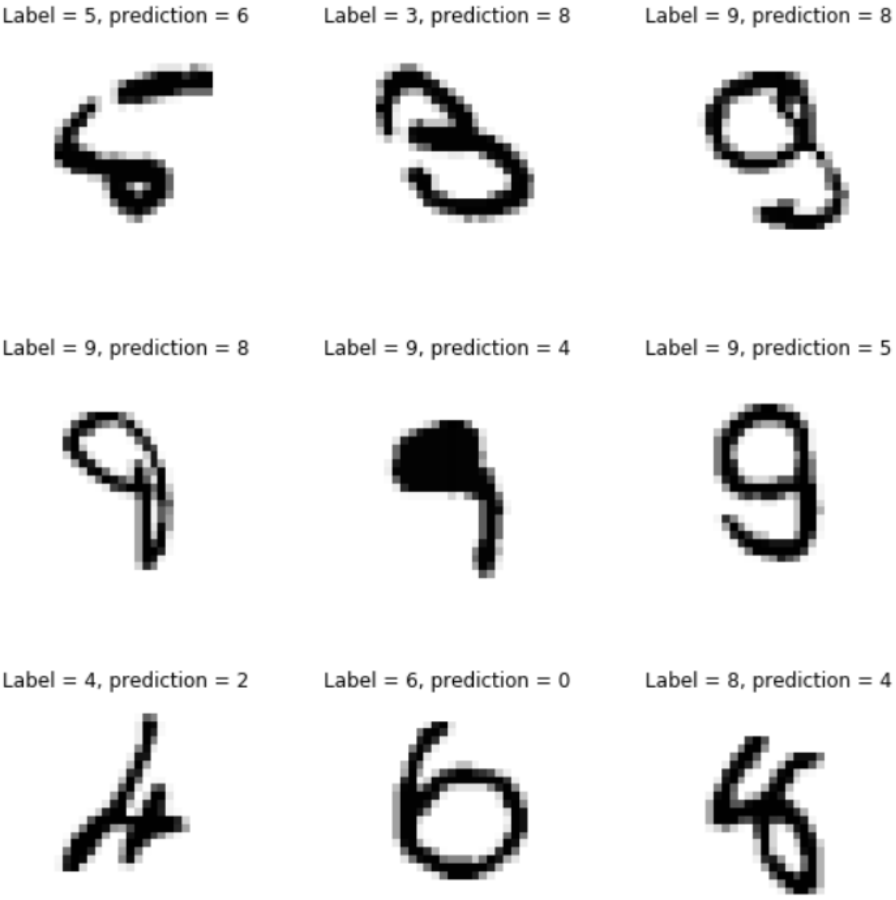

Label = 5, prediction = 6
Label = 3, prediction = 8
Label = 9, prediction = 8
Label = 9, prediction = 8
Label = 9, prediction = 4
Label = 9, prediction = 5
Label = 4, prediction = 2
Label = 6, prediction = 0
Label = 8, prediction = 4

Figure 3: 9 wrongly classified images by the provided model

It is also interesting to plot some metrics. The precision, recall, and F1 score for each class are given in Figure 4. The normalized confusion matrix is shown in Figure 5. Since the classes are balanced, the confusion matrix with absolute values will not give us any more information than the normalized one. From all these figures, there can be concluded that there are not some classes that are harder to distinguish from the others. All classes are classified approximately equally correct.

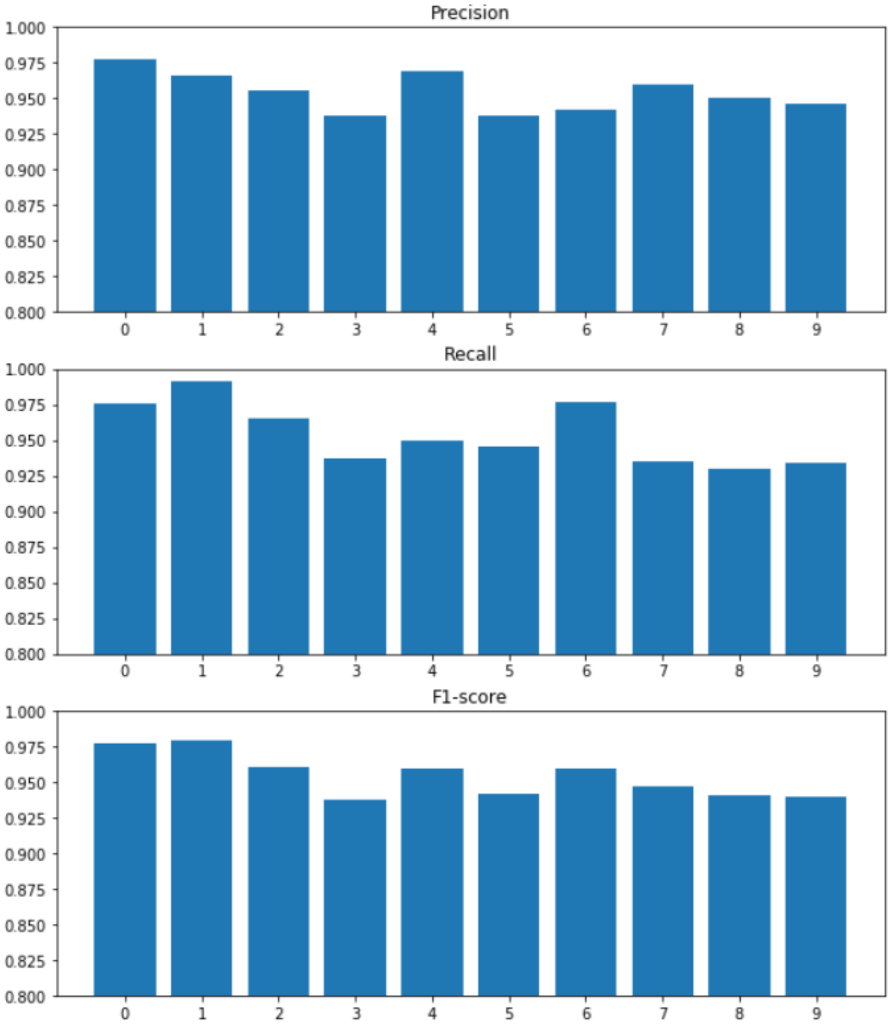


Figure 4: Precision, recall, and F1 score for each class for the provided model

2 Making the model powerful enough

Now, the model must be made more complex, so that it is capable to capture the ground truth. This might not be possible, but for such a simple task, we are quite certain it will be. Following Occam's Razor, the simplest model that can explain the ground truth will be better than an overly complex one. Such model is devised by adding more layers, adding more nodes to its layers or changing the non-linear activation function of those nodes. We found the following multi-layer perceptron: 6 Dense layers with 128, 128, 64, 64, 32, and 10 neurons respectively.

We also make the batch size larger, as a bigger batch size it better (as long as a batch still fits in GPU memory). It assures that the steps that are taken are well calculated and consequently in the right direction. This in contrary to many, small steps, of which many can be in the wrong direction (not towards the minimum). The batch size is now the complete train set (so batch gradient descent instead of mini-batch gradient descent).

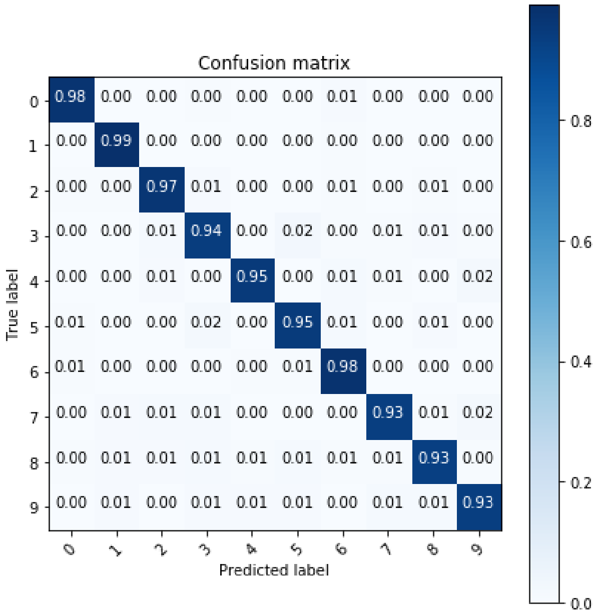


Figure 5: Normalized confusion matrix for the provided model

The default learning rate of 0.001 for the Adam optimizer was a bit to high, so it is now lowered to 0.0001. If the learning rate is too low, the slope of the validation curves would not be as steep which means that learning is very slow. If it is too high, the validation curves would not look smooth at all, the accuracy and loss would jump up and down. The number of epochs is also increased from 20 to 50 to allow convergence.

The validation curves are shown in Figure 6. There can be seen that the model is now complex enough and is able to capture the ground truth quite well. Now that we have low bias, regularization is needed to reduce the variance. The train set accuracy and loss for training on all 60000 samples of the provided model can be seen in Figure 7.

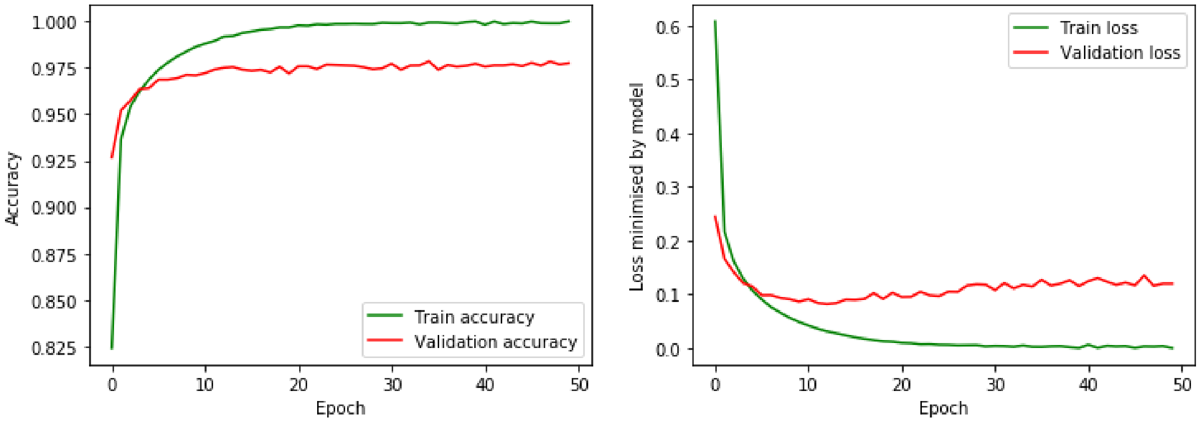


Figure 6: Validation curves for the more powerful model

After training the model on the 50000 samples in the training set and for 20 epochs, the training set accuracy is $100\,\%$ and the loss is 0.0043. When evaluating that model on the validation set, the validation set accuracy is $98\,\%$ and the validation set loss is 0.1438. After retraining that same model one last time on all 60000 samples in the

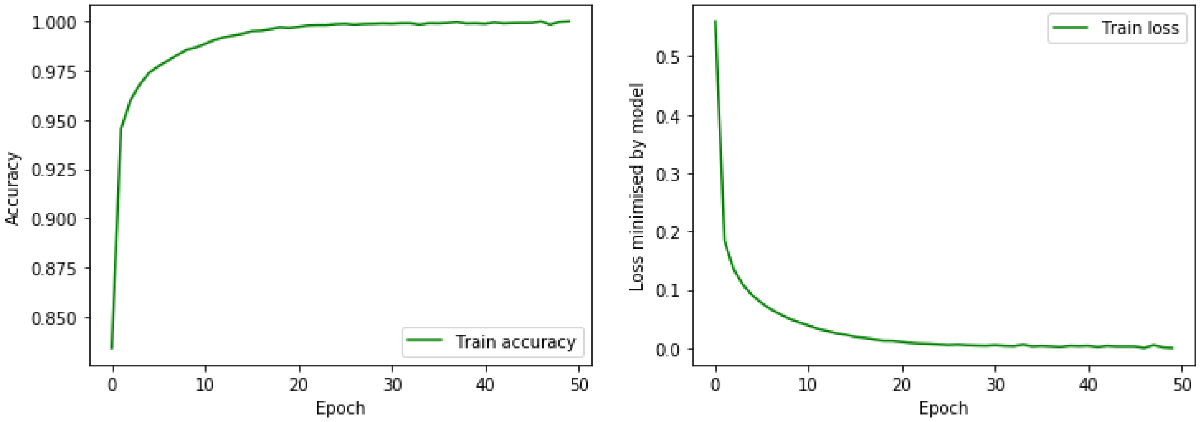


Figure 7: Validation curve for the more powerful model when training on the complete train set

training set, the training set accuracy is $100\,\%$ and the training set loss is 0.0024. When evaluating that model on the test set, the test set accuracy is $98\,\%$ and the set test loss is 0.1426.

From the precision, recall, and F1 scores (Figure 8) and the confusion matrix (Figure 9), we can conclude that the overall quality of predictions is already better.

3 Regularizing

We explore 3 ways of regularizing a neural network: L1/2 regularization, dropout and early stopping. The number of epochs is each time set to where the validation loss has converged. Note: in TensorFlow's Keras, the extra loss for regularizers in included in the metrics.

Since L2 regularization is the most widely used form of regularizing, we start applying this technique only. We set the L2 regularization parameter to 0.00001. After training the model on the 50000 samples in the training set and for 50 epochs, the training set accuracy is $100\,\%$ and the loss is 0.0090. When evaluating that model on the validation set, the validation set accuracy is $98\,\%$ and the validation set loss is 0.1244. The validation curves are visualized in Figure 10. After retraining that same model one last time on all 60000 samples in the training set, the training set accuracy is $100\,\%$ and the training set loss is 0.0088. When evaluating that model on the test set, the test set accuracy is $98\,\%$ and the set test loss is 0.1246. If we train the model with a higher value for the regularization factor, 0.01 in the extreme case, training and validation loss get closer to each other, but the model performs worse. After training the model on the 50000 samples in the training set and for 50 epochs, the training set accuracy is $95\,\%$ and the loss is 0.7935. When evaluating that model on the validation set, the validation set accuracy is $95\,\%$ and the validation set loss is 0.7959. It illustrates the typical bias-variance trade-off. The model does not overfit to the training data anymore. Because train and validation loss are equal, the variance is very small. In exchange, the loss and bias are higher than in the case of the non-regularized model. We conclude to keep the regularization factor at 0.00001.

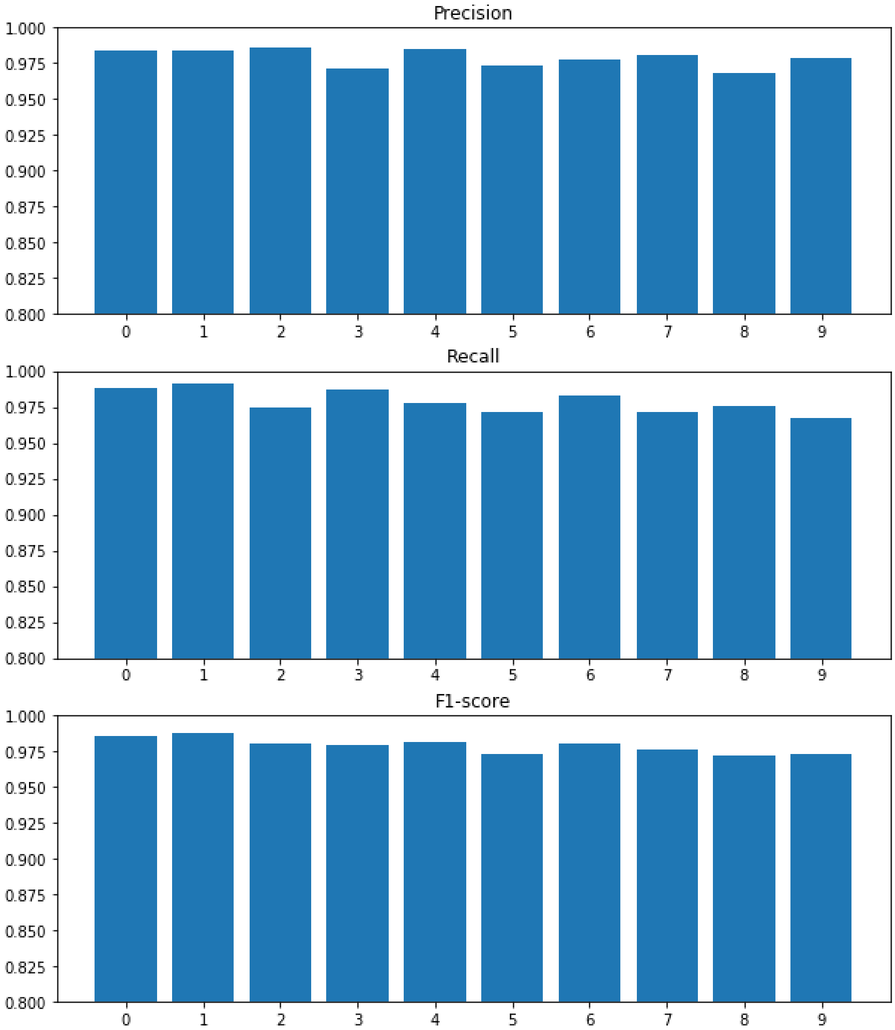


Figure 8: Precision, recall, and F1 score for each class for the more powerful model

Training the model with L1-regularization resulted in a classifier with even more bias, so this will not be discussed into more detail.

Next up, dropout is explored. Before each fully connected layer, a dropout layer with dropout rate 0.1 is inserted to reduce overfitting in a simple, elegant way. After training the model on the 50000 samples in the training set and for 50 epochs, the training set accuracy is $100\,\%$ and the loss is 0.0080. When evaluating that model on the validation set, the validation set accuracy is $98\,\%$ and the validation set loss is 0.0783. The validation curves are shown in Figure 11. After retraining that same model one last time on all 60000 samples in the training set, the training set accuracy is $100\,\%$ and the training set loss is 0.0071. When evaluating that model on the test set, the test set accuracy is $98\,\%$ and the set test loss is 0.0732. So, applying dropout resulted in a slightly better model with lower bias and variance.

The last regularization technique to explore is early stopping. When looking at the learning curves of the powerful model in figure 6, the early stopping technique will probably not have a variance-reducing effect. We expect it to only have a reduced amount of training time. After training the model on the 50000 samples in the training set and for 25 epochs, the training set accuracy is $100\,\%$ and the loss is 0.0062. When evaluating

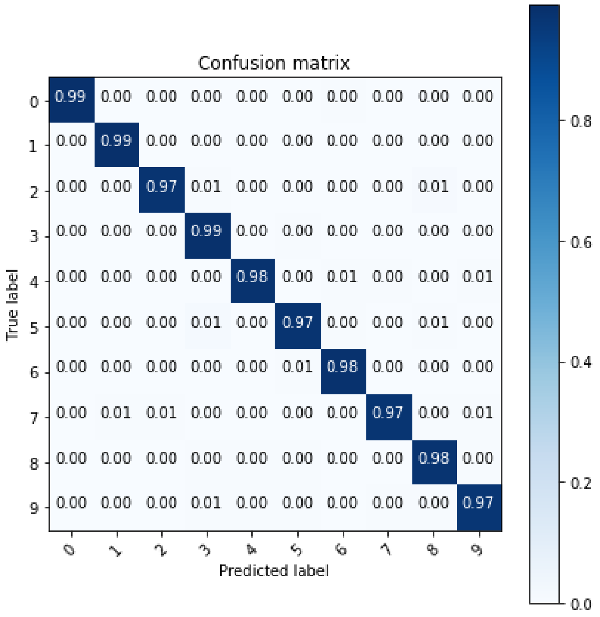


Figure 9: Normalized confusion matrix for the more powerful model

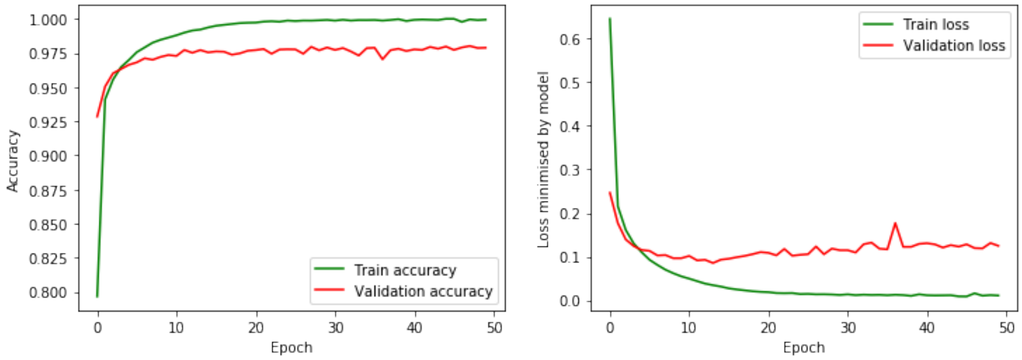


Figure 10: Validation curves for the L2-regularized model

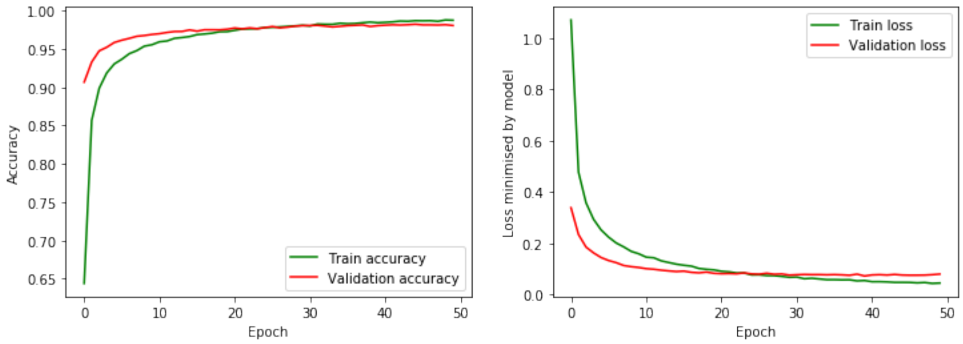


Figure 11: Validation curves for the regularized model using dropout

that model on the validation set, the validation set accuracy is $98\,\%$ and the validation set loss is 0.1032. After retraining that same model one last time on all 60000 samples in the training set, the training set accuracy is $100\,\%$ and the training set loss is 0.0041. When evaluating that model on the test set, the test set accuracy is $97\,\%$ and the set test loss is 0.1610. The validation curves for the training with early stopping are found in Figure 12. As expected, early stopping did not help reducing variance of our model.

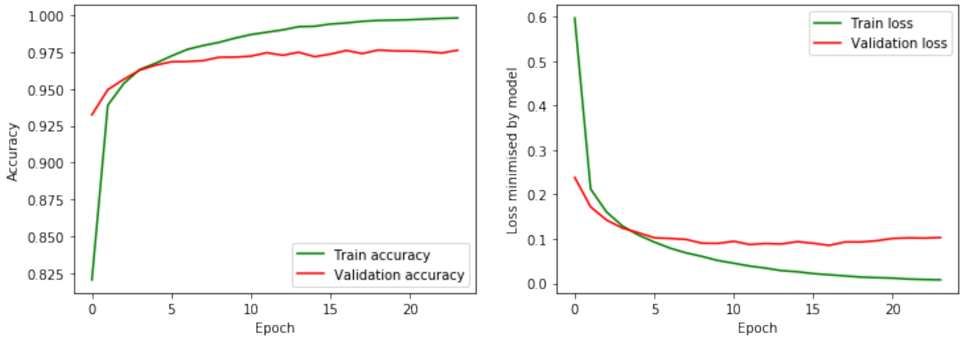
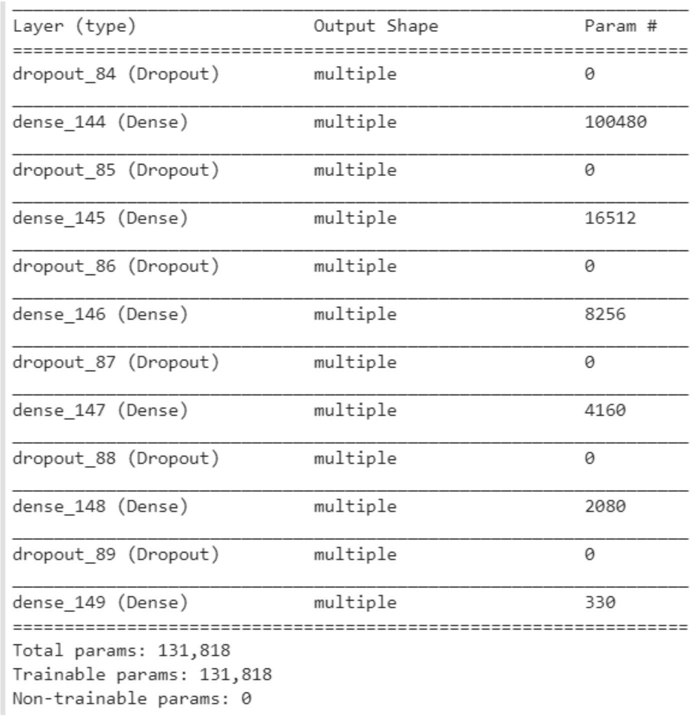


Figure 12: Validation curves for regularized model with early stopping

4 Final Model

Figure 13 shows a summary of our final model. The six dense layers were necessary to make the model complex enough for the classification task. The number of nodes per layer were 128, 128, 64, 64, 32 and 32 from the first to last dense layer. The six dropout layers were necessary for regularization, such that the model overfits less to the training data and generalizes better. The widely accepted Adam optimizer with learning rate 0.0001 is used, to shrink the categorical cross entropy loss while training. A batch size equal to the whole training set and 50 epochs turned out to be the equilibrium between performance and training time.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dropout_84 (Dropout)         multiple                  0         
_________________________________________________________________
dense_144 (Dense)            multiple                  100480    
_________________________________________________________________
dropout_85 (Dropout)         multiple                  0         
_________________________________________________________________
dense_145 (Dense)            multiple                  16512     
_________________________________________________________________
dropout_86 (Dropout)         multiple                  0         
_________________________________________________________________
dense_146 (Dense)            multiple                  8256      
_________________________________________________________________
dropout_87 (Dropout)         multiple                  0         
_________________________________________________________________
dense_147 (Dense)            multiple                  4160      
_________________________________________________________________
dropout_88 (Dropout)         multiple                  0         
_________________________________________________________________
dense_148 (Dense)            multiple                  2080      
_________________________________________________________________
dropout_89 (Dropout)         multiple                  0         
_________________________________________________________________
dense_149 (Dense)            multiple                  330       
=================================================================
Total params: 131,818
Trainable params: 131,818
Non-trainable params: 0
```

Figure 13: Summary of the final model