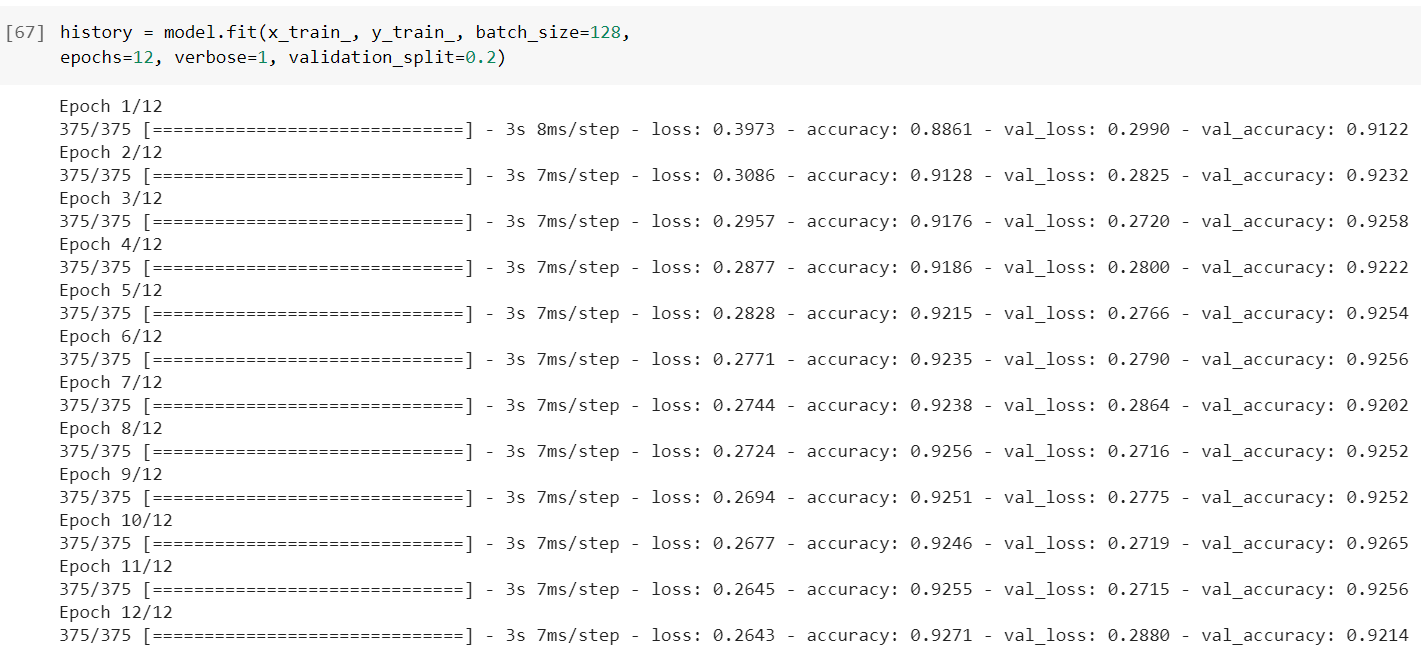
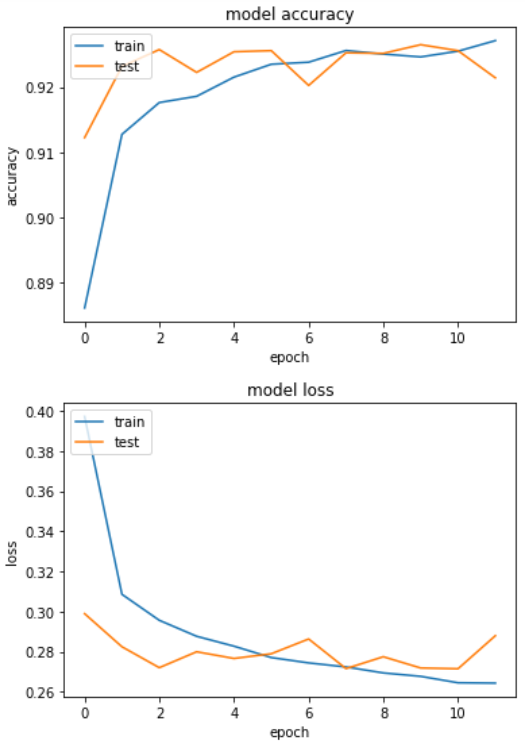
**Question 1**

* Bank fraud
* Doctors' recipes
* LaTeX translations for mathematical formulas. (To save paper. students could write their solutions on an iPad which will be translated automatically to Latex)
* Historical student performance in schools. (score written on paper --> digitalizing)
* Writing notes on paper for better memorization while learning and having the notes on the computer in a readable font.

**Question 2**



**Question 3**



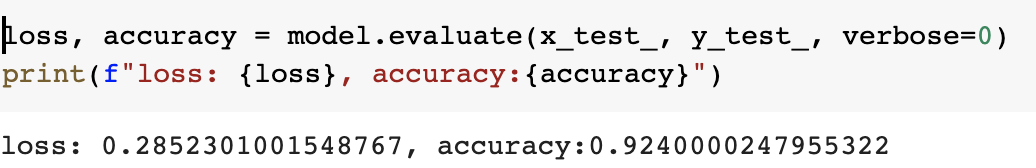
**Question 4**

Discuss with your group, then describe to your teacher, how the accuracy on the training and validation sets progress differently across epochs, and what this tells us about the generalisation of the model. (Question 4, 5 points)

After the first epoch, we saw that the accuracy of the model was the lowest, but already had quite good accuracy. After every next epoch, the model learned more from the training data, as the initial randomly assigned weights change with backpropagation. After 10 epochs, the model is still learning, but already in a lower learning phase. We’ve increased the number of epochs to 20 to see what happened. We found that the accuracy of the training model did not increase much (only 0.005 in 10 epochs). However, the difference in accuracy between the test data and training data decreased after every epoch, which makes the model more generalizable after every epoch. The lines are coming slightly closer together after every epoch. Concluding: the model works fine on both training data and data that it has never seen before and is more generalizable after 10 epochs than after e.g. 3 epochs.

This phenomenon is called vanishing gradient problem.

**Question 5**



**Question 6**

Discuss with your group, then describe to your teacher, whether this accuracy is sufficient for some uses of automatic hand-written digit classification. (Question 6, 5 points)

An accuracy of .92 is not high, especially when long integers need to be classified. Let’s say you have a 10 digit number, than the changes of classifying it correctly are 0.92^10 = 0.43. When comparing this accuracy with human accuracy, the system in which this accuracy is applied will not be acceptable. However, maybe for some quick and dirty analysis this might be high enough. Think of the LaTeX example, where we digitalize formulas. An accuracy of 0.92 is high enough, as this saves the user a lot of time in entering the formula in LaTeX.

**Question 7**

Discuss with your group, then describe to your teacher, how linear activation of units limits the possible computations this model can perform. (Question 7, 5 points)

When the activation function is linear, this means that it doesn’t make sense to add hidden layers to the neural network. Linear functions + linear functions will always stay a linear function. The extra hidden layers have no effect on the outcome of the model. So, with a linear function the model is limited and cannot learn about complex relationships between datapoints, as not all datapoints are linearly separable.

Besides, the model isn’t really learning as you cannot make useful backpropagation. It doesn’t improve upon the error it has made in the result. A linear function does not change the weighted sum of the input for the output. However, when training a model, you do want that certain nodes in your network become stronger and certain nodes become weaker, so that certain criteria for digit recognition are more important than others.

**Question 8**

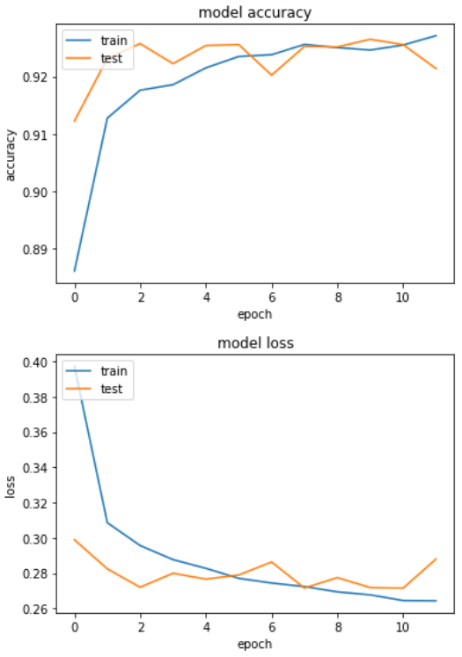
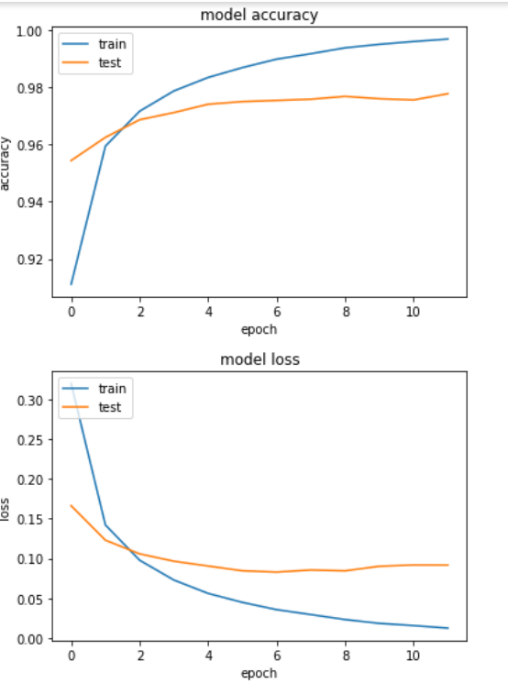
****

Figure 1 The old plot as comparison

**Question 9**

For the training data, after the first epoch, the accuracy is already a bit higher for the relu activation method (0.96 compared to 0.91). After every epoch, the model learns through backpropagation and eventually almost reaches an accuracy of 1.00, where the previous model only reached an accuracy of a bit less than 0.93 after 12 epochs. The new model is thus learning significantly better and more with the new activation method and every epoch really contributes to the improvement of the model.

The validation data (test data) also does a better job in the new model. The previous model had a capricious line compared to the new model, with accuracies fluctuating relatively a lot. After the first epoch, the accuracy of the test data is already better than throughout the entire previous model. After every epoch, the accuracy of the test data improves a bit, almost touching 0.98, making the model already more useful for cases like bank frauds. This makes the model more specific. In the end, less input will be classified wrong (e.g. a 6 as a 9).

The output is not very generalizable. As the graph shows, the difference between the test data and the training data is very large. This means that the model is trained very well on the training data, but it does not know what to do with the test data. Therefore, the generalizability is low. The model can not be used to test new data, because the model overfits.

It’s striking to see that the loss of the test data almost stays the same after 4 epochs in the new model. The model is becoming more accurate on classifying the data, but the misclassifications are always of the same impact after epoch 4. Our model gets close to being optimally trained, as the loss doesn’t change much anymore, and the loss is used in backpropagation to adjust the weights of the nodes.

**Question 10**

Chart, line chart

Description automatically generated

**Question 11**

As you can see for the training history, that doesn’t differ much from the previous model and the current model. Both almost have an accuracy of 1 and the road they took to have this accuracy is also almost the same. However, they second model learns at a different pace which is evident from the learning curve and the spike after the first epoch.

Another difference can actually be seen in the validation sets. Already after the first epoch, the loss is as high as after the last epoch from the previous model. Comparing the last epochs of both models, the loss went from 0.10 to 0.05, which is a notable difference and gain for the model. This means that the wrongly labeled cases are less of a surprise for the model. Also the accuracy increased a bit, from 0.98 to 0.99. For some cases, the change betw­­een models is preferable, for some not for the following reasons. The current model took significantly longer to train, so for use cases where extreme accuracy isn’t necessary this would not be a preferable upgrade. However, again, for cases like bank fraud this can actually be a major improvement. The model performs better on new data and is therefore more generalizable than previous model.

**Question 12**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Question 13**

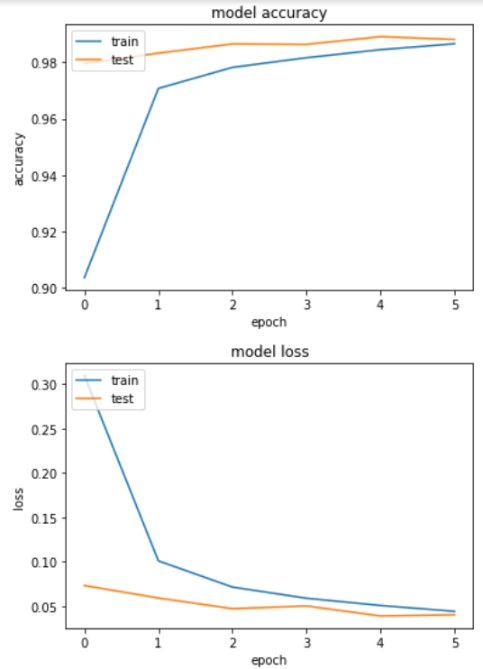
Discuss with your group, then describe to your teacher, whether this accuracy is sufficient for some uses of automatic hand-written digit classification. (Question 13, 5 Points)

The accuracy is almost perfect and sufficient for hand-written classification for critical and uncritical use. However, we should always keep the comparison to humans in mind. Is the model performance better than humans in different use cases? If yes, then the model is preferred. If no, there are two cases. 1) It doesn’t mind that the model is somewhat less accurate as this is allowed in the domain and 2) It does mind that the model is somewhat less accurate as this is not allowed in the domain. For 1, an example is digitalizing grades from test of a high school. For 2, an example might be the doctor recipes example. In that case, it is critical that the system should better perform than the doctor, as a difference between 900mg and 300 mg of a medicine can be a difference between life and death.

In the example above, we didn’t consider things like morality. This brings a whole new side to the discussion of whether the accuracy of digit classification is high enough or not. Considering morality: would we even accept a wrongly made doctors recipe by a machine learning system? Who is responsible in that case etc.?

**Question 14**

Chart, line chart

Description automatically generatedAdd dropout layers (model.add(keras.layers.Dropout(rate=x))) after the max pooling stage (x = 0.25) and after the fully-connected (dense) layer (x = 0.5). Compile and train the resulting model as before. Discuss with your group, then describe to your teacher, how the training history differs from the previous (convolutional) model, for both the training and validation sets, and for the time taken to run each model epoch (Question 14, 3 points)

*Current model previous model*

If we compare the training history of the current model with the previous model, we see that the training data lines almost have the same curve. However, after the second epoch (1) of the previous model, the accuracy of the training is similar (or even a bit higher) than the accuracy of the training data of the current model in epoch 6 (5). The previous model thus did a better job on the training data.

The curve of the validation data differ between the previous and current model. We see a really constant accuracy level of the validation data from the start in the current model, while we saw some jumps in the previous model. The current model has a stabler accuracy of .98, while the previous model had more difficulties with the validation data during the first epochs. Eventually, the accuracy became higher than in the current model, with an accuracy of 0. Also, the loss models follow the same curve as the accuracy models.

It’s noticeable that for the first time, our validation data outperforms the training data. This means that in real-life cases, it can be implemented better than in previous cases.

When comparing the runtime, we see some significant changes. Every epoch of the previous model took around 104 seconds, while the current model takes 120 seconds for every epoch. That’s an increase of 15% in every epoch.

**Question 15**

Discuss with your group, then describe to your teacher, what this tells us about the generalisation of the two models. (Question 15, 3 points)

The accuracy of the model for training and test data converges at the same value. This means it performs the same on training data as it does on new data. Therefore, one can say it is more generalizable.

**Question 16**

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

**Question 17**

Plot the training history and show it to your teacher (Question 17, 2 points)

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

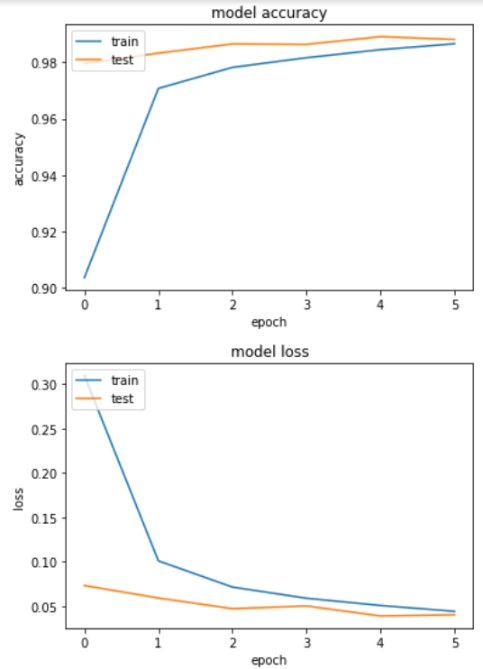
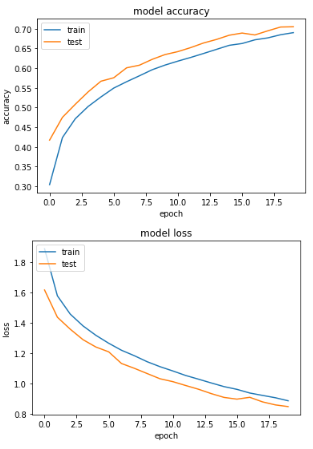
 

Figure 2 Old plot

Figure 3New plot

**Question 18**

Discuss with your group, then describe to your teacher, how the training history differs from the convolutional model for digit recognition and why. (Question 18, 5 points)

The slope of the loss function is less steep in the new model compared to the old model. The loss function of the training data is still higher than the loss function of the test data, which means that the loss of the model is lower on the new data, compared to the data that the model is trained upon. This means that the model is improving.

The new data set is more difficult than the digit recognition set. The new data set has high variations in object position, size and viewing angle. This can be seen when comparing the loss. The loss of the new model is really high compared to the loss of the relatively simple digit recognition data set.

Compared to the old model, the slope of the training data and the test data of the new model is the same. This means that there is not a lot of difference in both datasets. The new model follows almost a kind of root function, definitely when comparing it to the loss function of previous model.

We can see that the accuracy of the new model also follows this root function curve. It learns in a more graduate way compared to the previous model. Because the data has such a high variation in e.g. position, size and angle, it takes more than one epoch to come to a relatively high accuracy. With the easy digit recognition data, the model already has a lot of information about the data after the first epoch. It adjusted the weights already near optimal, as in further epochs not much improvement can be seen. On the contrary, after the first epoch of the new model, the weights of the nodes are not anywhere near their optimal settings. The data is more difficult and thus needs a longer time than one epoch to understand the data. Besides, the learning rate of the new model is way lower than the learning rate of the previous model, which facilitates this observation even more.

**Question 19**

Discuss with your group, then describe to your teacher, how the time taken for each training epoch differs from the convolutional model for digit recognition. Give several factors that may contribute to this difference (Question 19, 4 points)

The time per training epoch in the new model (around 200s) is higher than the time per training epoch of the digit recognition model (around 126s). One factor that could contribute to this, is the total size of the training data. Because the training data of our new model has three dimensions instead of two, this will give way more input points for the model for every epoch. Besides, as our new model has more layers than the digit recognition model, it needs to make more calculations. This also increases the time per epoch. The difference in training time per epoch would even be larger if we wouldn’t have changed the 64 filters to 32 in the second convolutional layer. What could also play a role, is the increased number of fully connected units (from 128 to 512). This increases computational load as there are more links.