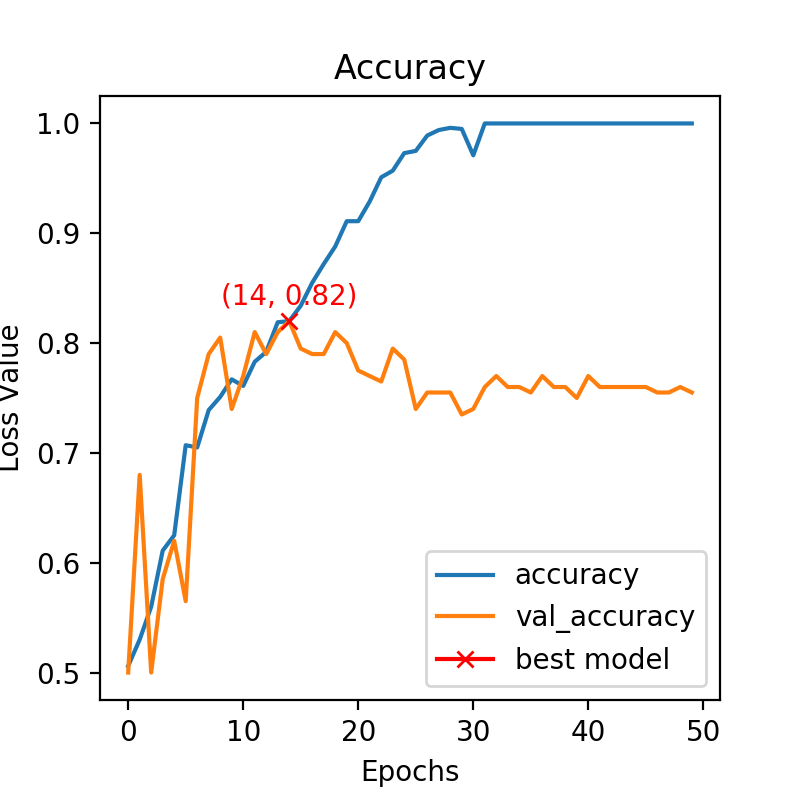
Chen Siyuan

Darçot Benjamin

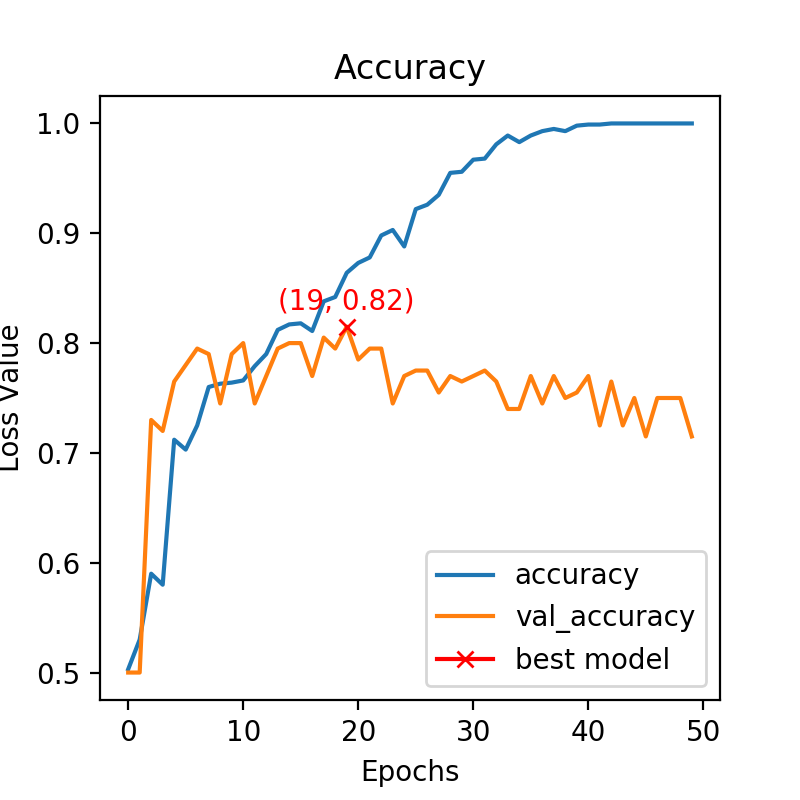
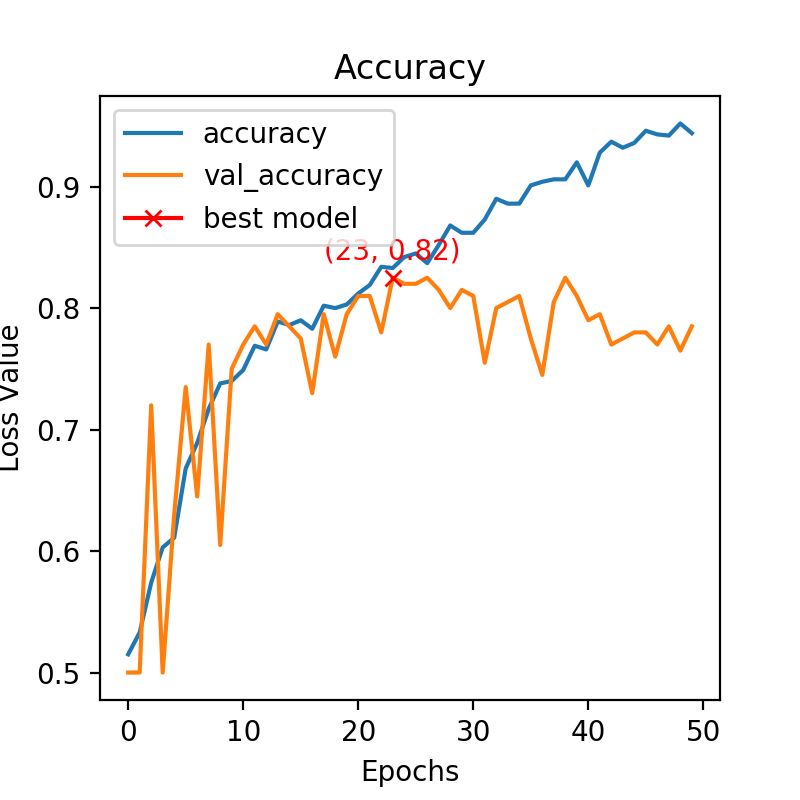
**Lab 2 : Report**

**Task 6A**

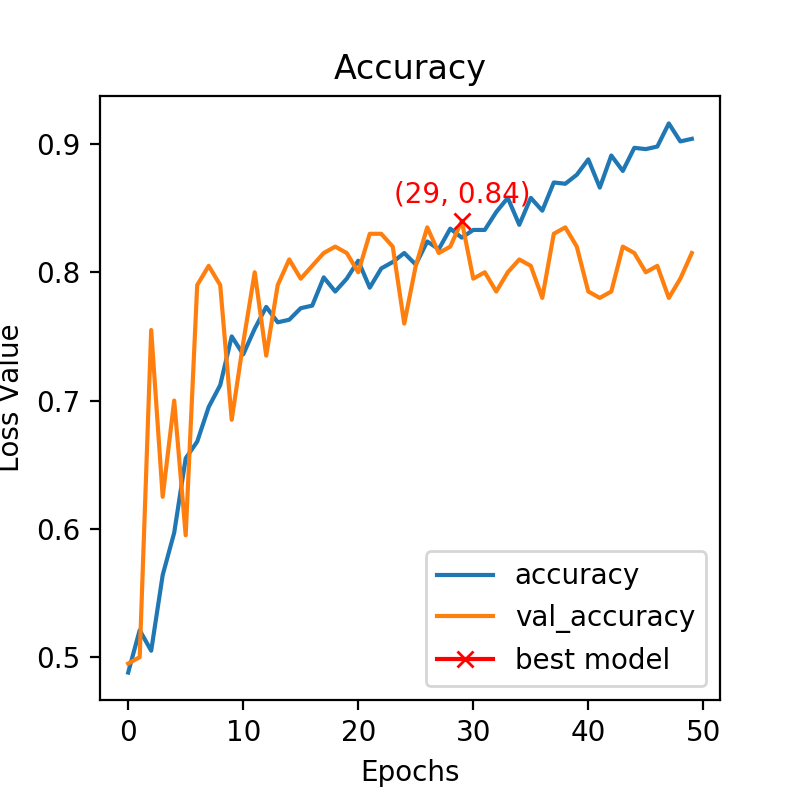
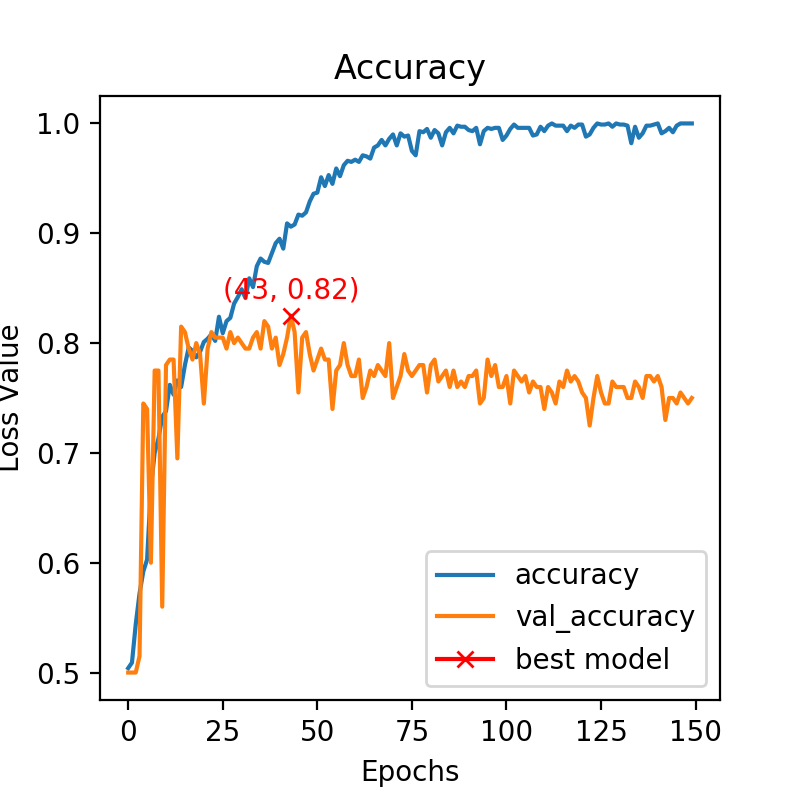
A classification of images was performed using a deep neural network with an AlexNet architecture. In this first step, a batch size of 8, a base number of 32, a learning rate of 0.001, 50 epochs and the Adam optimizer were used to train the model. The accuracy of the model through the iteration is presented in this graph. 

The training model is efficient because in the end the accuracy of the fitting for the training model is around 100%. But the validation dataset is not well fitted and the difference between the training model and the validation is increasing with time, which shows an overfit.

**Task 6B**

At this step, the influence of the base number is studied. The same test as before was performed but with a base number of 8 and 16. The results are shown on the graphs below (left for base number of 8 and right for base number of16).

So we can see that changing the base number only doesn’t change the results much, the best model is still at an accuracy of 0.82 and the overfitting is still there. The main change is on the training dataset because reducing the base number reduces the accuracy of the training fitting.

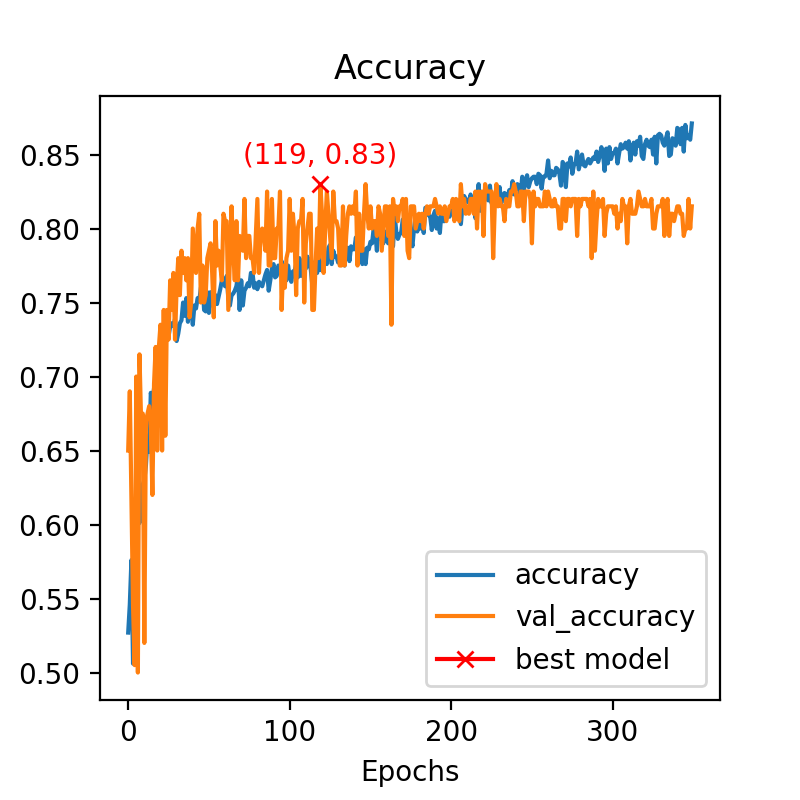
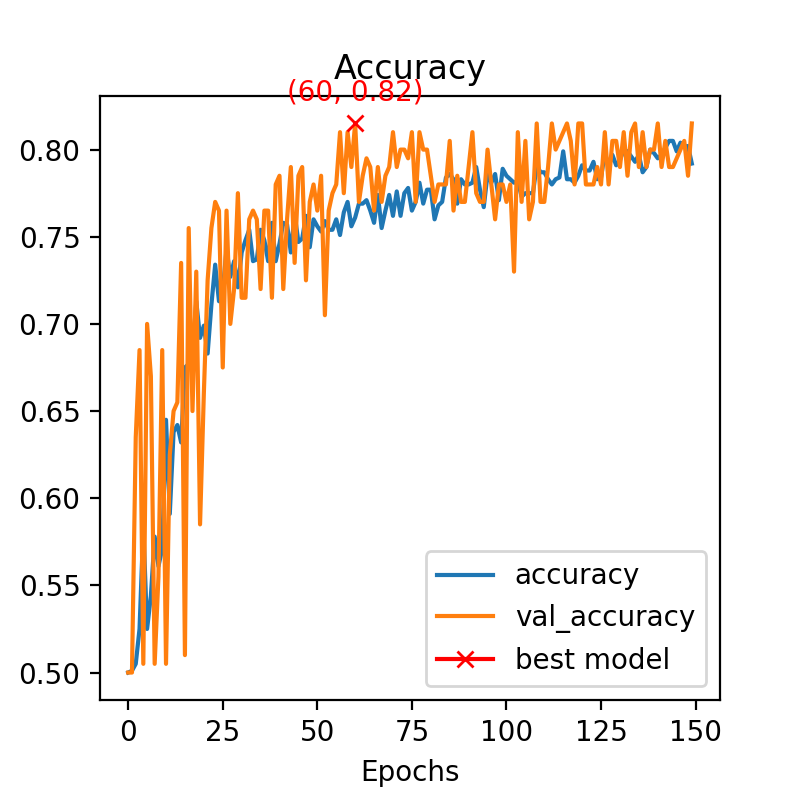


A dropout layer was then added after each dense layer in the network and then the same test was performed with a base number of 8 and both 50 (left) and 150 (right) epochs.

Therefore we can see that with 50 epochs the overfitting is slightly reduced at the end, which is normal because the dropout layer randomly deletes some neurons in the dense laker, thus reducing the learning power of the network which can no longer fit as well as before the training dataset. However, increasing the number of epochs increases the overfitting, which can be expected because with enough epochs the network can fit almost perfectly the training dataset.

**Task 6C**

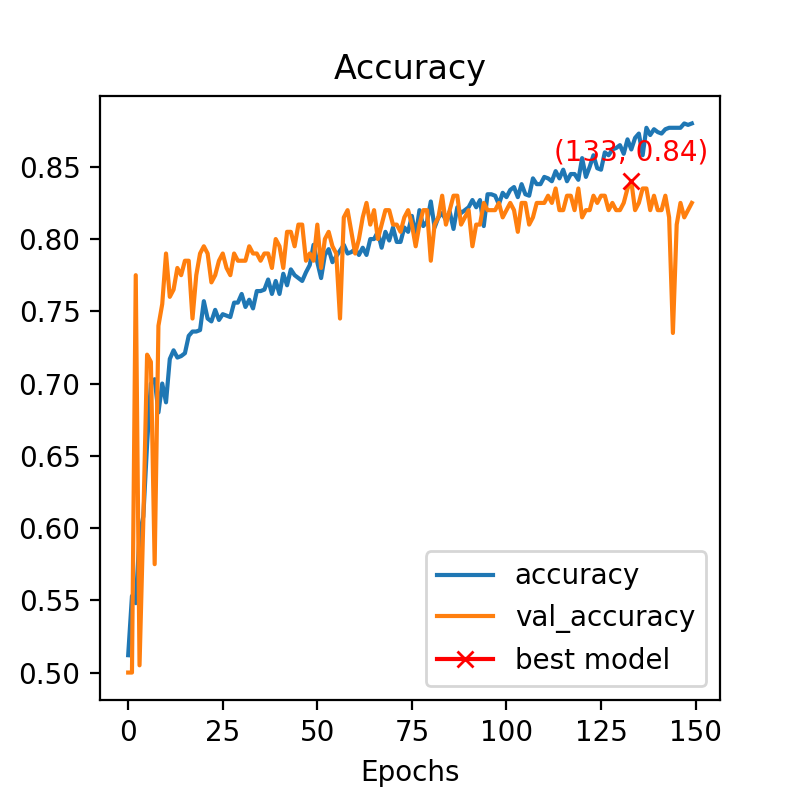
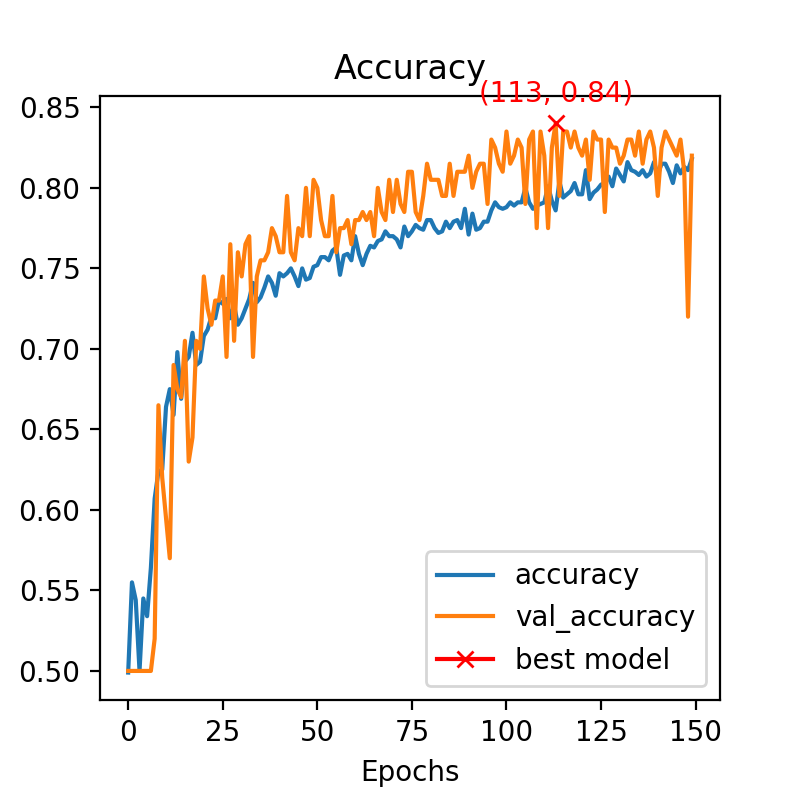
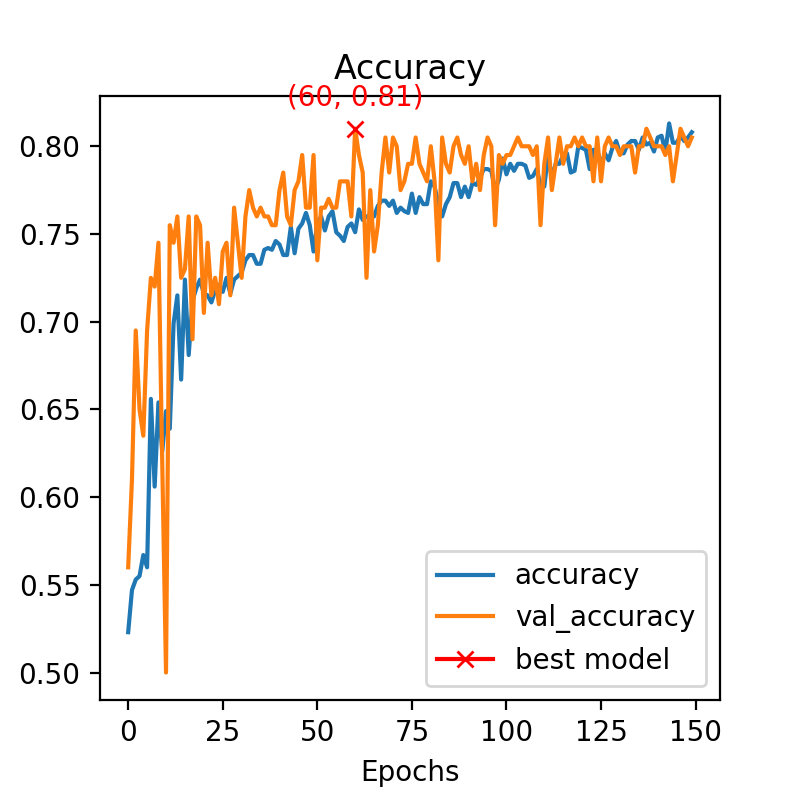
The influence of the learning rate is now studied. While before the learning rate was 1e-4, now a learning rate of 1e-5 is used. The dropout layers are now removed and the parameters are the same as at the beginning of task 6B. So we can compare these new results with the first figure of task 6B to evaluate the influence of the learning rate. To do that both 150 (left) and 350 (right) epochs are used.



First, we can see that the best accuracy has not changed, it is still around 0.82, so changing the learning rate does not increase the performance. However, reducing the learning rate reduces the time for the network to reach its best model. This seems normal as the learning rate is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Therefore reducing it, reduces the change of the model for each update in the weights.

**Task 6D**

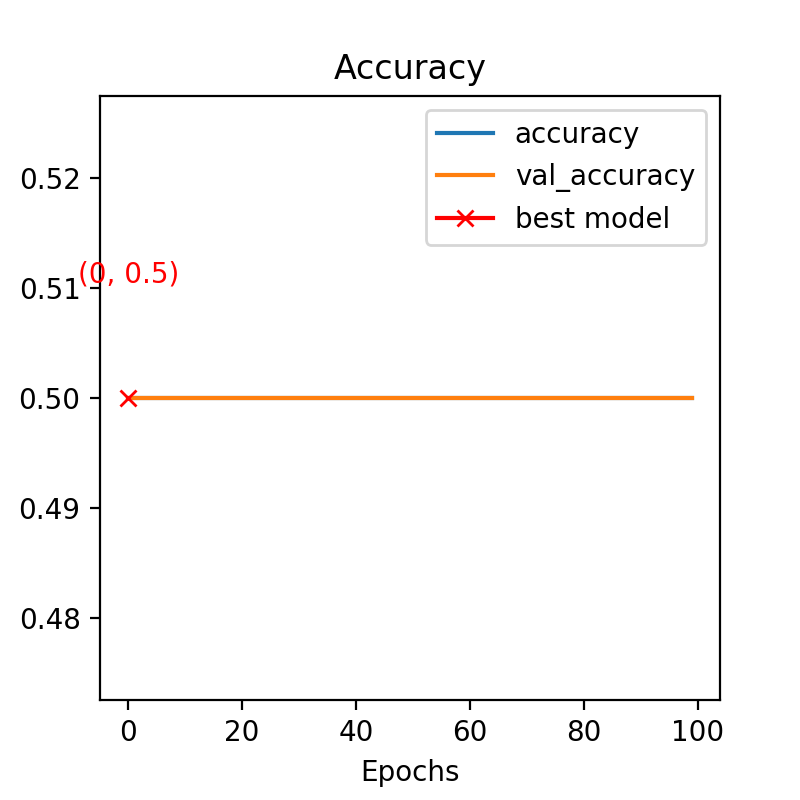
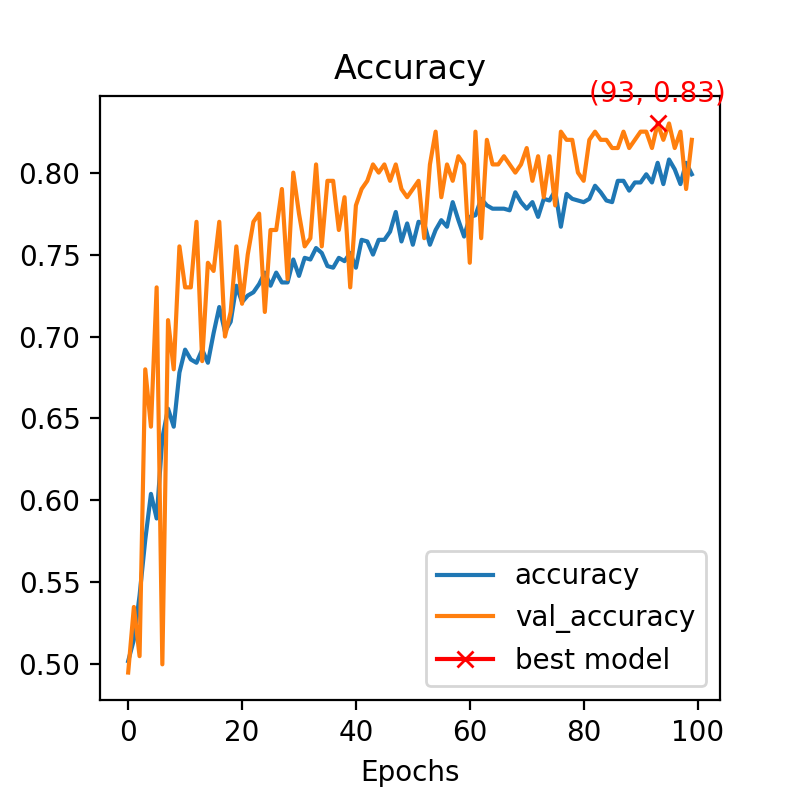
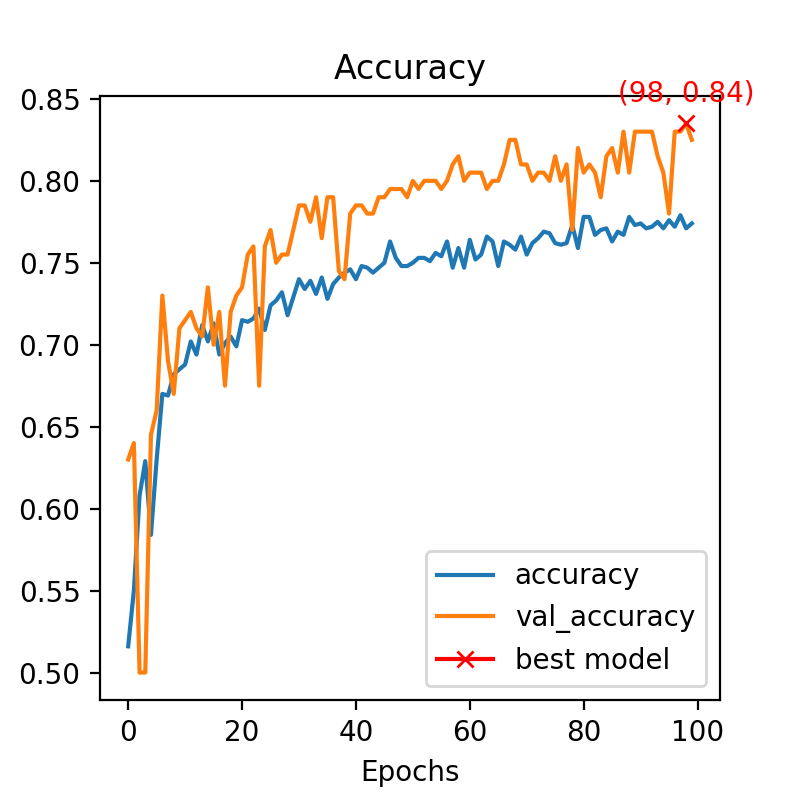
At this step, the influence of the batch size is studied. So now, the model is performed for a batch size of 2 (left), 4 (middle) and 8 (right).



Here, in terms of performance, the difference is not clear. The best model has almost the same accuracy in each case. The main difference is the speed to reach this best model (or almost the best model). Indeed, as the batch size decreases the number of iterations to reach a good model decreases too.

**Task 6E**

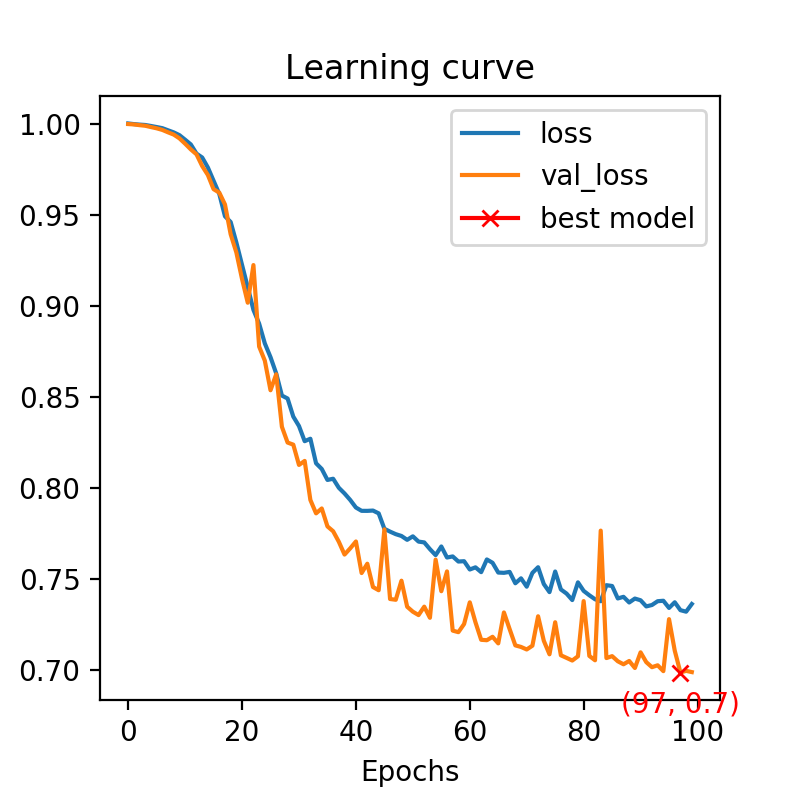
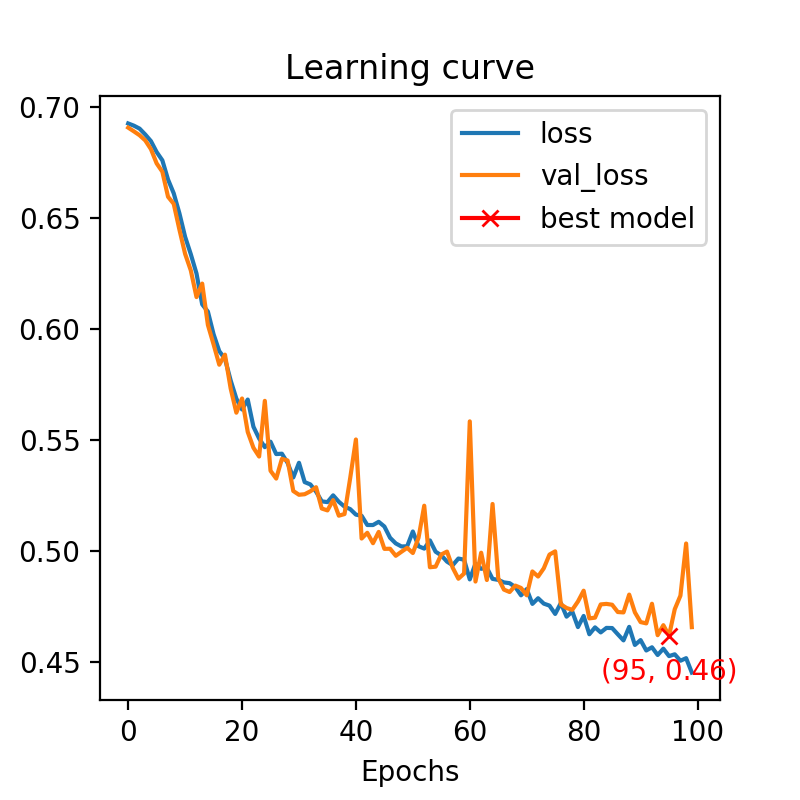
The influence of the optimizer is now studied. The same test was performed with Adam optimizer (left), RMSprop (middle) and SGD (right).



We can see that SGD is not at all efficient, the accuracy is always equal to 0.5 for both the training and the validation data. In this case, both Adam and RMSprop are good, but RMSprop is the most efficient. We can also note that with Adam the accuracy fluctuates more than with rmsprop.

**Task 6F**

In this part, the influence of the loss function is studied. In order to do that, the same test is performed using both ‘binary cross entropy’ (left) and ‘hinge’ (right) as a loss function.

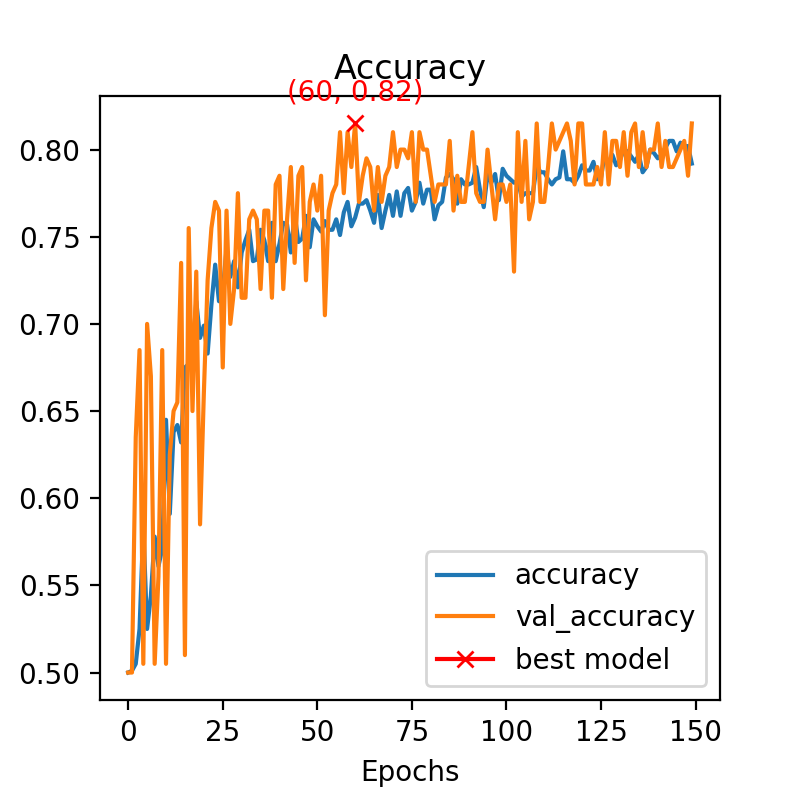
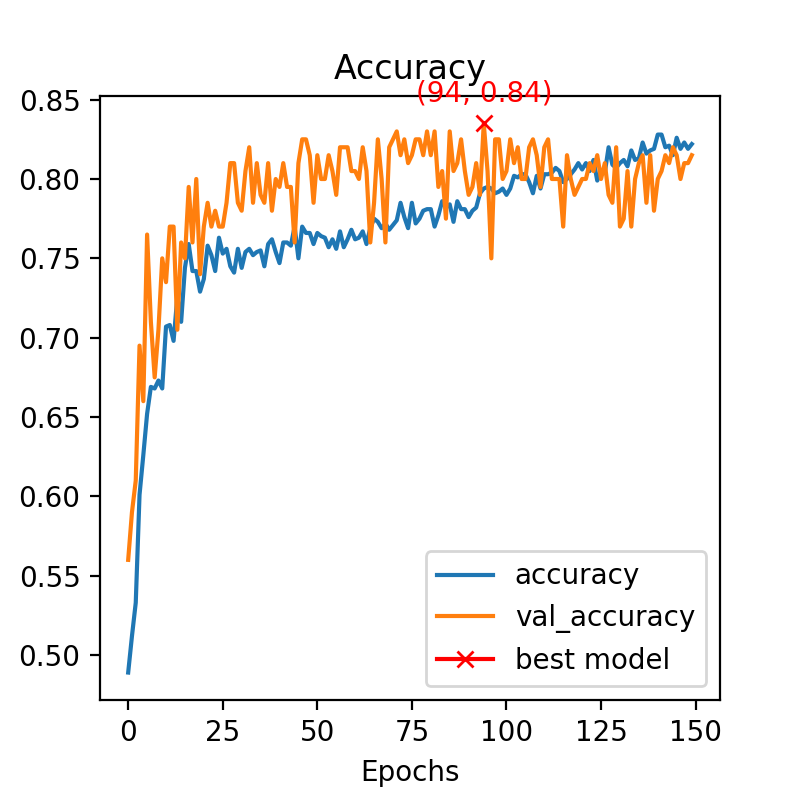


The loss values are higher with hinge as a loss function. This induces better performance and accuracy with the BCE.

**Task 7C**

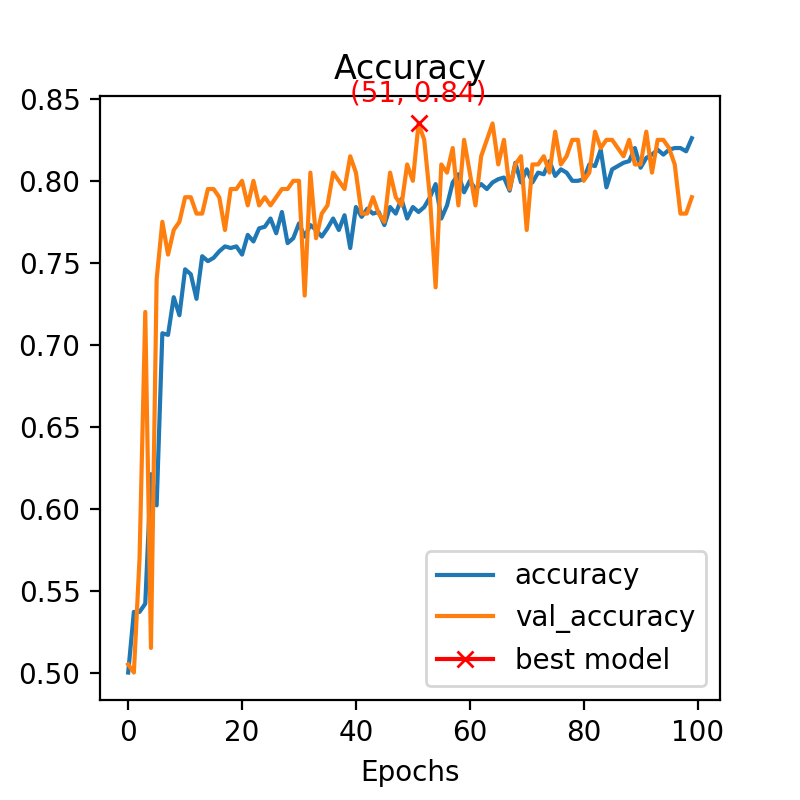
A new network architecture, called VGG16, is now used. After different tests, by looking at the accuracy of those, we can find that a proper learning rate for this network can be 1e-5 as well.

To compare both AlexNet et VGG16 architectures, a test is performed in the exact same conditions (same as task 6C). The results are shown below with on the left the one from the AlexNet architecture and on the right the one with VGG16.



We can see that the accuracy is slightly better with the VGG16. Moreover, the VGG16 architecture reaches a good model in fewer iterations than the AlexNet architecture. However the best accuracy is mainly the same, and it is due to the similar architecture of these two networks.

**Task 7D**

Increasing the number of features (from 8 to 16) does not really affect the model performance. Indeed, we can see below that the performance is mainly the same as before in the previous task.

However, adding a dropout layer after each dense layer slightly increases the accuracy of the test model.

