CAMA Hector
LOUSSOUARN Antoine
OUADAHI Fares

LAB 2 - Assignment

Deeper networks

TASK 6A

With the given parameters, AlexNet provides increasing values of val_loss and decreasing values of loss. The model is therefore overfitting.

TASK 6B

With a lower n_base parameter, the val_loss curve is slightly lower than with a higher n_base. Lowering base_dense decreases the number of weights that have to be trained : therefore, the model is better at generalizing (and the val_loss is lower). However, even if n_base is lower, the model still overfits. Finally, it can be inferred from the plots that the lower n_base is, the higher the number of epochs must be in order to find the model with the lowest val_loss.

Then, as expected, if we add a dropout layer, the model no longer overfits. However, the model is still not perfect because with only 50 epochs, the loss curve has not yet converged toward a stable value. Therefore, we cannot expect this model to give out the best outputs.

Increasing the number of epochs allows the loss curve to reach a minimum. Unfortunately, the model overfits if we go over 50 epochs. This can be due to the fact that the model is training on the same data over and over again, and is therefore bad at generalizing (hence higher and higher val_loss values).

TASK 6C

In this task we set the same values than in task 6A for each parameter, except for the epochs (150 instead of 50). If we lower the learning rate from 10-4 to 10-5, we no longer observe overfitting. But the loss curve has not yet reached its minimum : we can expect to get an even better model.

If we lower the learning rate again (10-6) with 150 epochs, we can see that the loss curve is very far from reaching its minimum. Therefore, a reduced learning rate must come with a significantly increased number of epochs. The model also takes more time to train with these changes.

TASK 6D

Lowering the batch_size parameter makes the loss curve decrease quicker, which means the model is trained more rapidly. However, the val_loss values don't seem to change drastically.

TASK 6E

The chosen parameters for the best not-overfitting model are the following : batch_size = 4, epochs = 100, learning rate = 10-5, base_dense = 8, no dropout layers.

If we chose the SGD optimizer, we see that the loss curve is far from reaching its minimum and the model therefore needs more epochs to be trained. If we chose the RMS-Prop optimizer, the loss curve is decreasing more slowly and the val_loss values are not stable.

TASK 6F

If we plot the loss curves, we can see that the BCE function makes the model reach a lower loss minimum than the Hinge loss function.

TASK 7B

For 100 epochs, a learning rate of 10-5 allows the best model performance.

TASK 7C

AlexNet and VGG16 accuracy values are both equal to 0.8 in average. The major difference lies in the loss curve. Indeed, VGG16’s loss values reach their minimum quicker than AlexNet. This can be explained by the fact that VGG16 is a deeper network than AlexNet (more layers). The other big difference is that large kernel-sized layers in AlexNet are replaced with several 3×3 kernel-sized layers put successively.

TASK 7D

Increasing the number of feature maps, it makes the model overfit sooner and sooner. There are too many feature maps for the model to generalize : it ends up overfitting. Adding a dropout removes the phenomena of overfitting.

TASK 7E

AlexNet is a version of LeNet on which there are more convolutional layers. AlexNet is therefore deeper than LeNet. VGG16 has even more small kernel-sized layers and is even deeper.

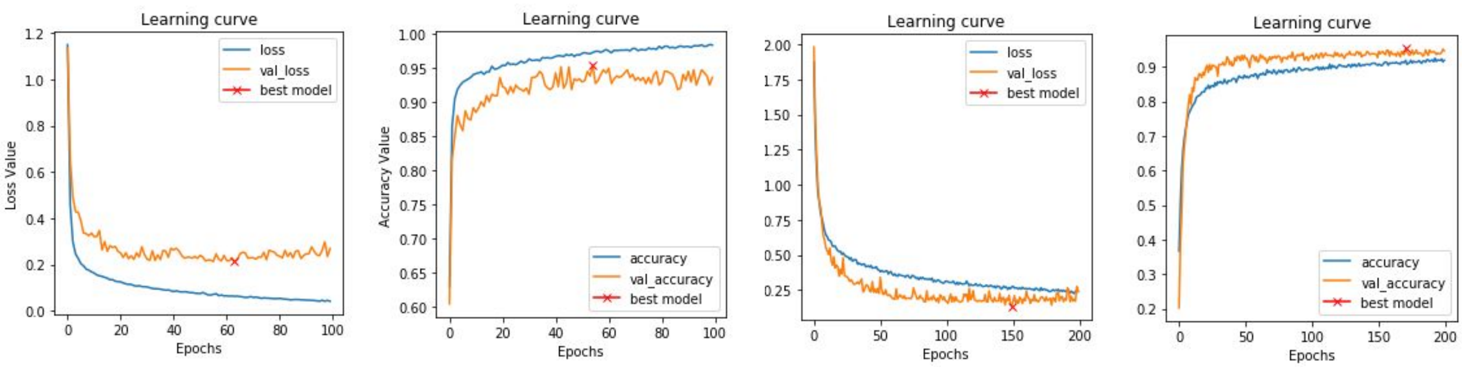
We noticed that the higher the number of layers is, the fewer iterations the model will need to give accurate predictions. However, with a higher number of weights to train, VGG16 is more likely to need a dropout layer to prevent overfitting (which is not necessary in LeNet given the lower number of weights to train).

TASK 9

The bone data set shows better results for classification (91% and val_loss = 0.2). We can guess that this happens because the bone has a rectilinear shape whilst the skin tumors can have very different sizes and shapes. The images are also in black and white here, when it was not the case for the skin images, which makes it less complicated for the model to recognize certain features. However, what could make the bone images hard to be recognized is the fact that the fractures can be very different from each other (position on the bone, displaced or not…).

In order to check whether the results are reliable or not, the accuracy value must be acceptable (around 0.9). For both tests here, the accuracy is around 0.9 so we can infer the results are reliable.

TASK 10



(a) **(b)** **(c)** **(d)**

Figure 1: Learning and accuracy curves for the LeNet (a),(b) and AlexNet (c),(d) models

By extending the previous models into multi-class classification models, we tuned them by finding the optimum values of hyperparameters:

- <u>for LeNet:</u> base_dense = 32, LR = 10-5, epochs = 100, batch_size = 8, no dropout, opt=Adam;
- <u>for AlexNet:</u> base_dense = 8, LR = 10-5, epochs = 100, batch_size = 8, DR = 0.2, opt = Adam;

Both models allow for a good classification (more than 90% accuracy and a low loss value). However, when looking at the curves (figure 1), the AlexNet model seems to have a better accuracy on the validation set and can achieve a lower loss value than the LeNet model.