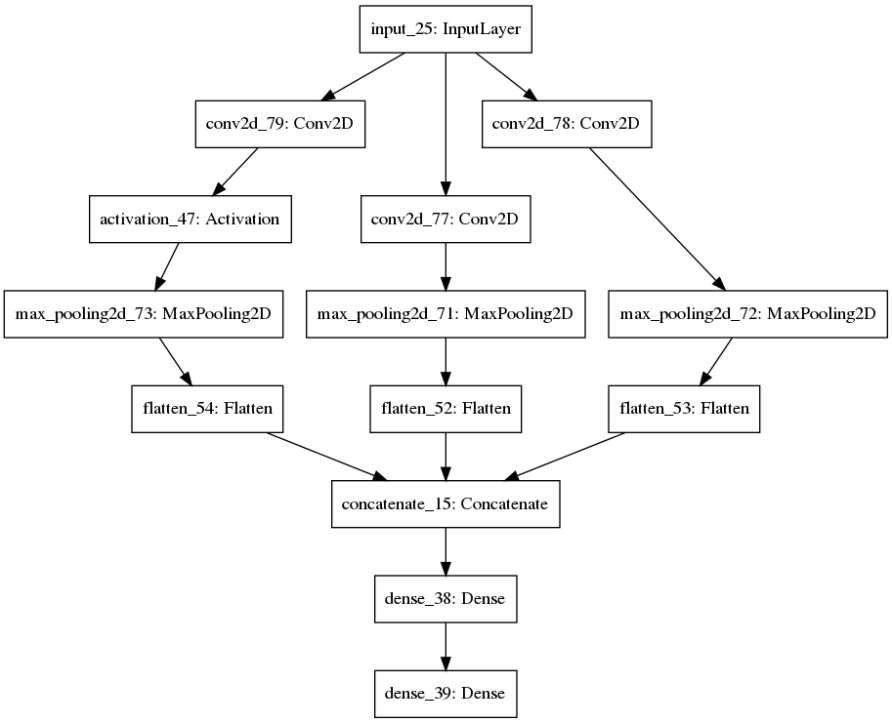
Practical work 10 – Conv Neural Networks with Keras - part 3

Exercise 1

a/b/d) CIFAR10-CNN_week10_ex1.ipynb

c) We used tree different padding size (2,2), (3,3) and (4,4) and then merged it to one.



The accuracy on the training data wasn't better than before, because we stopped after 10 epochs

Train accuracy: 0.77694

Test accuracy: 0.6662

Comparing with the previous results we see that the test data has approx the same accuracy.

CNN	Architecture description	Acc. train	Acc. test
No DA	CONV(32F,same)-RELU-CONV(32F,same)-RELU-MAXP(2)-CONV(32F,same)-RELU-MAXP(2)-DENSE	0.7040	0.7052
With DA v1	CONV(32F,same)-RELU-CONV(32F,same)-RELU-MAXP(2)-CONV(32F,same)-RELU-MAXP(2)-DENSE	0.7416	0.7486
Transferlerarning with VGG19	-	0.7925	0.7925
CNN With multiple paths (see model.png)	-	0.6662	0.7925

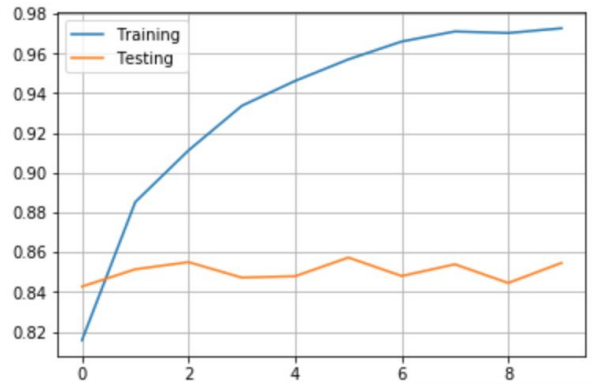
Exercise 2 Transfer learning (see transfert_mobilenet_v2.ipynb)

We used the mobilenet_v2 for transfer learning, with an image size of (96, 96). When using the image size (224, 224) we got an memory exception. Because the original images have the size (32,32) this should change a lot in the results.

We extracted the results of the mobilenet_v2 and added a simple MLP layer:

```
using inpit shape (output shape of model) (3, 3, 1280)
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 3, 3, 128)         163968
_________________________________________________________________
flatten_1 (Flatten)          (None, 1152)              0
_________________________________________________________________
dense_1 (Dense)              (None, 100)               115300
_________________________________________________________________
dense_2 (Dense)              (None, 10)                1010
=================================================================
Total params: 280,278
Trainable params: 280,278
Non-trainable params: 0
_________________________________________________________________
```

We get an accuracy of:
Train accuracy: 0.98546
Test accuracy: 0.8546



Then we tried to do data augmentation with using the mobilnet_v2
But the results didn't change significantly:
Train accuracy: 0.98486
Test accuracy: 0.854

Exercice 3 Auto-encoders (see notMNIST-auto-encoder-stud.ypnb)

a) With encoding, we have to choose between reducing to the minimum the number of features extracted and so have a really compressed result. That could be good because very light. Or use more hidden layers and have a better quality result because keeping more features from the original image. Often a good number of features (number of hidden layers) can be the number of high eigenvalues of the matrix-image.

b) The result of remove noisy from the characters: