

The following assignment is given here, as a pdf and not as a notebook. The purpose is that you will be able to read a task and transform it into code.

Ex4

In this exercise, you will touch the two previous exercises and try to change them by the following instructions:

(1) Convolutional Network

Ex2 - In the last task in ex2, you played with the hyper-parameters of the network and reached the highest accuracy you could. This was done using a fully-connected network only.

(a) You should change the network to have convolutional layers.

We do not guide you about the number of layers, number of filters and filter sizes.

The general guideline is to start with a rather small and simple network (no resnet blocks, skip-connections...). Once this works well, try to tune the hyper-parameters to squeeze the performance even more. Then, you can make the network more complex.

In order to define your simple network we encourage you to try some concepts from well-known architecture.

For example,

- Convolutional layers always come before fully-connected.
- You can stack two 3x3 convolutions instead of a single 5x5 one.
- Use a convolution with a larger number of filters before you downscale (pooling/stride). As the spatial resolution decreases, the number of channels often increases.
- Use a rather small kernel size (and not 11x11 as alexnet suggested).

Pay attention to the number of parameters in the network you define (use the 'summary' command). It can sometimes be a clue for overfitting.

a.1) Please give some explanations on the selection of the model. Have you tried other architecture? Different hyper-parameters? We would like to see the way you reached your final model.

a.2) Why did you get better or worse accuracy when you use convolutional layers?

(b) Add batch normalization layers to your network. Report your results.

* If you already used convolutional layers when you submitted ex2, we encourage you to experiment with them more. At every lecture you gain more knowledge...

(2) Transfer Learning

Ex1 - in ex1, you classified politicians using pre-defined networks. Specifically, we loaded the weights of a model that was trained on a different dataset. We talked in class that this procedure is called “transfer learning”. We also saw that we can fix the weights of the pre-trained networks. In this exercise, you should explore four different configurations when using vgg-16 network:

1. Training the network with the pre-trained weights and fixating all the weights up to the last layer (this is like we apply a classifier on a feature vector)
2. Training the network with the pre-trained weights and fixing the weights up the X layer (you decide what is X).
3. Training the network from scratch (just use the architecture without the weights)
4. Training the network with the pre-trained weights without fixating (this is what was done in ex1).

(a) Implement configurations 1-3.

(b) In what configuration did you get the best accuracy? Please also provide an explanation on why this is the best option and why the rest give worse performance.