Artificial Neural Network - Convolutional Neural Networks

Summary

The exercises here aim for some understanding and some hands-on exeprience with CNNs.

Before these exercises, I also suggest that you have a look at the exampleFilters.ipynb to see some simple filters that detect vertical and horizotal edges.

Exercise 1

Consider a CNN that takes in 32 × 32 grayscale images and has a single convolution layer with three 5 × 5 convolution filters (without padding).

- What is the size of the feature map? (feature map is the output of one filter applied to the previous layer)
- With what size of padding we will end up to a feature map with the same size as the input (this is sometimes called "same padding")?
- How many parameters are in this model?
- Explain how this model can be thought of as an ordinary feedforward neural network with the individual pixels as inputs? are there any kind of constraints on the weights?

Exercise 2

Consider a CNN composed of three convolutional layers, each with 3 × 3 kernels, a stride of 2, and "same" padding. The lowest layer outputs 100 feature maps (i.e., channels), the middle one outputs 200, and the top one outputs 400. The input images are RGB images of 200 × 300 pixels.

- What is the total number of parameters in the CNN? If we are using 32-bit floats for every parameter, at least how much RAM will this network require when making a prediction for a single instance?
- What about when training on a mini-batch of 50 images?
- Why would you want to add a max pooling layer rather than a convolutional layer (with the same stride)?

Exercise 3

Solving a Fashion_MNIST with LeNet architecture

Here you will implement a LeNet architecture of Convolutional Neural Networks (as briefly introduced in the lecture and shown in LeNet.pdf here) with pyTorch (you may get inspired by the code in exampleCNN.ipynb).

First you will download the Fashion-MNIST dataset. Split into train/validation/test datasets and train the network. Finally, plot the learning curves (train/validation loss and accuracy) and show the confusion matrix.

1. Download Fashion-MNIST
2. Split the data into train / validation / test subsets. Make mini-batches, if necesssary.
3. Build a CNN model
4. Train the model on the dataset
5. Plot the training curves (Loss and accuracy)
6. Show the confusion matrix and accuracy on the test dataset.
7. Try adding Droput layers; play with the hyperparameters. Use cross-valudation to find the best hyperparameters
8. When you train the best model, visualize the filters of the first convolutional layer. You may look at an example on how to visualize filters in PyTorch: https://stackoverflow.com/questions/55594969/how-to-visualise-filters-in-a-cnn-with-pytorch