Homework 4: Deep Learning

Out May 24; Due June 7, beginning of exercise session*

Kristian Kersting, Alejandro Molina, Jinseok Nam
{kersting, molina, nam}@cs.tu-darmstadt.de

upload link: `https://www.dropbox.com/request/4B0KMkPqUZMHv4uoOIKU`

1. Convolutional Neural Networks.

We have discussed during the last lecture that convolutional neural networks (CNNs) are inspired by the cat's visual cortex. CNNs have been very successful on a variety of learning problems, including, but not limited to, image classification.

The goal of this homework is to implement CNNs using deep learning frameworks such as TensorFlow, PyTorch and MXNet. There have been proposed a lot of CNNs in the literature, but we will focus on the most simple CNN architecture. LeCun et al. [1] have proposed LeNet that is composed of pairs of a convolution layer and subsampling layer, followed by fully connected layers, illustrated in fig. 1.

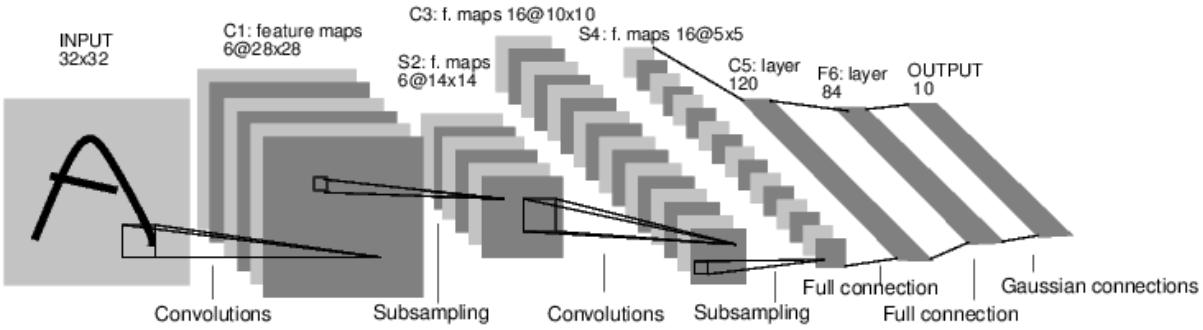

Figure 1: LeNet-5

As we use MNIST again for this homework, that is, inputs are gray-scale (single channel) images, the actual input size of your CNNs should be 32×32 if the kernel size is 5×5 and stride is 1. In other words, MNIST digit images (28×28) are padded with zeros around the border, so that inputs are of 32×32 matrix. If you want more examples about convolution operations and zero padding, please go check the following web page: `http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html`

*We will discuss the solutions in the exercise session. It is my suggestions that you try to address at least 50% of the exercise questions. Simply try hard to solve them. This way, you will get familiar with the technical terms and with the underlying ideas of the lecture.

In the subsampling layers, the kernel size is 2×2 and stride is 2. Please note that subsampling in LeNet is mean (average) pooling. Your first CNN architecture should have the same number of parameters with LeNet-5.

Although you are supposed to use deep learning frameworks, please remember that your task is to implement ***your own convolution layer and pooling layer***. **DO NOT** use convolution layer class and pooling class provided in such a deep learning framework.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.