Convolutional Neural Networks for Image Classification

Introduction. Regular neural networks do not scale well to full images. Consider MNIST, which is composed of 28×28 images. A fully connected layer of a regular neural network would thus use 784 weights per neuron in the next layer. This is manageable, but consider a more realistic image of, say, 1000×1000 pixels and 3 color channels (red, green, and blue). In this case, a fully connected neural network would need $3,000,000$ weights per neuron. Clearly, this full connectivity becomes intractable and wasteful and the huge number of parameters would quickly lead to overfitting. As a solution, convolutional neural networks allow to encode certain properties of images into the architecture of the network. This results in a more efficient implementation and a vast reduction on the number of parameters.

There are two main ideas behind convolutional networks: small connectivity regions, and parameter sharing. Small connectivity means that neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons. Parameter sharing means that the weights corresponding to different output neurons are set to the same value, which implies that we need to learn a smaller number of weights.

Project description. In this project, the student will learn to implement a deep convolutional network on TensorFlow. The network will be applied on a multiclass image classification problem. The student will also learn about the operations of convolution, pooling, and dropout. The student will implement a standard network architecture and will report the predictive performance on each dataset.

Datasets. We will use these two datasets:

- The MNIST dataset, a collection of $60,000$ images of handwritten digits, of 28×28 pixels each. This has been used in previous lab sessions.
- The CIFAR-10 dataset, which contains $50,000$ color images of 32×32 pixels that are classified into one of the following categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

We will not use larger datasets because it takes long to train.