Machine Learning and Deep Networks

Assignment 1: CNN

Due: 1/11/2018

In this assignment you will implement a simple Convolutional Neural Network (CNN), train it to perform a simple classification task, and experiment with different architectures and hyper-parameters. Your implementation and experiments should be done using TensorFlow. As a basis you may start from the simple CNN-based MNIST classifier that I will provide you (or you may look for or implement yourselves a better one).

A. **Train and Test Error.** The provided simple CNN has two convolution layers (each followed by pooling) and two fully connected layers, and it is able to achieve an accuracy above 99% after less than 12000 iterations. How many free parameters does this network have? The training process is often visualized/monitored by plotting the training accuracy and the test accuracy as a function of the number of training iterations. Plot these accuracy measures (perhaps using a data point every 500 iterations).

B. **Linear vs. Non-Linear**. Identify what layers/components of the network are non-linear and remove them (except for the logits/softmax in the loss function, also assume the max-pooling is a uniform stride - or change it to one if you wish). Compare the performance of this linear network to the original non-linear version. What is the effect on the training and the final test accuracy of the network?

C. **Deep vs. Shallow**. Construct a shallow CNN consisting of a single convolutional layer, followed by one fully connected layer, such that it is able to achieve accuracy above 95% using the smallest number of free parameters. How many parameters does your network have (explain how you count them), and how quickly did it reach the target accuracy? Plot the training of this network. Next, reduce the number of parameters of the original deeper network as much as you can, while still being able to achieve 95% accuracy. How many parameters does this network have (detail the count), and how quickly did it reach the target accuracy? Plot the training of this network.

D. **Two Digits Sum.** Design, implement and train a CNN architecture that takes as input two MNIST images and infers their sum. Note that this should be solved as a pure classification problem, and not by running the single image MNIST classifier twice and adding the results. Describe the reasoning behind your choice of architecture. Plot the training of this network.

Submission: submit a ZIP or RAR archive that contains your code, and a document, which clearly describes your solutions and reports your results.