096260 - Deep learning course: Warm-up exercise

Submission date: 07/12/2016

1 Exercise

In the following exercise you will create a classifier for the MNIST database. I strongly recommend to overview tutorial 4. You should write your own training code and satisfy the mentioned constraints. The performance parameters appears below. You can submit your work only after achieving these performance levels.

1.1 Submission instructions

Submission will be in pairs (course partners) and will contain a short (two pages) pdf report containing:

- Model architecture description, training procedure (data augmentation, regularization, optimization details etc).
- Two convergence graphs - for error and loss as a function of time (epochs). Each graph should depict both training and test performance.
- A short summary of your attempts and conclusions.

In addition, you should also supply:

- Code able to reproduce your results - we might test it on different variants on these datasets.
- The trained network (with trained weights) in Torch format. I recomend to use the following api for that:

  https://github.com/torch/torch7/blob/master/doc/serialization.md

  usage: http://stackoverflow.com/questions/33288006/torch-lua-how-to-save-a-trained-neural-network-model-to-file
- A function that loads your's trained network and returns the average error on the test set.

1.2 Classification Network

Write a complete training procedure for a classification network on MNIST dataset. Design and train your network so that it will satisfy the 2 following goals:

- Final accuracy on the test-set should be > **90%**
- Number of trainable parameters (weights) within the network should be < **6500 .**

1.3 Data

Dataset in Torch format can be used by loading the 'mnist' package:
local mnist = require 'mnist';

You will train the network **ONLY** on mnist.traindataset() and test the network on mnist.traindataset()

1.4 Grades policy

- Successful submission - 65 points.
- Report - 15 points.
- Competition - 20 points: the teams will be sorted according to test error results, then will be splitted to batches of size 5. The grade is given uniformly $[0, 20]$ w.r.t number of batches.