Assignment 1

1 Part 1

You are supposed to implement a convolutional neural network for MNIST in TensorFlow (see the Notes section). Your solution should contain the following features:

- it should contain both convolutional layers and fully connected layers,
- it should contain batch normalization for convolution, which you implement by basic tensor operations (without using tf.nn.moments nor tf.nn.batch_normalization) - it is fine if you use the train mode even for the test check,
- implementation should allow painless changes in the architecture, like adding new layers or changing number of filters by updating numbers in one place in your code,
- you need to obtain at least 99.1% test accuracy reliably in front of a person checking your solution (if your program runs for more than 3 minutes please have logs from previous runs).

2 Part 2

In the second part of the problem the goal is to visualize neurons for each convolutional layer. In particular for each filter of each conv layer you should find 10 patches (of size being equal to the filters receptive field in the input image) over all train images that excite the filter most.

This part should use a checkpoint with weights from Part 1. Visualize your obtained images nicely.

3 Additional solution features (nonobligatory)

If you want, you can additionally implement:

- dropout - check how it works with and without batch normalization,
- data augmentation,
- try different learning algorithms and different learning rates.

4 Deadline

You should submit your solution by email by 23:59 on 08.05.2017 to **both** `cygan@mimuw.edu.pl` and `maciej.jaskowski@gmail.com` with email title "Assignment 1 - Deep neural networks". Your code will be inspected either during lab session on 09.05 (or 16.05). Note that even if you are one minute late after the deadline, your solution will not be inspected. We have no mercy whatsover so you better not count on that.

5 Notes

You are allowed to use a framework different than Tensorflow, but in that case you should consult on the functions you use, so that solution in your favourite framework is not substantially simpler.