DATA.ML.200 Pattern Recognition and Machine Learning

Exercise Set 2: Multi-layer Percepteron (MLP)

1. `pen&paper` Count the number of parameters in a neural network (2 points)

   a) Consider the conventional full-connected neural network architecture of Figure 1. Suppose our inputs are 64×64 RGB images of two different traffic signs.

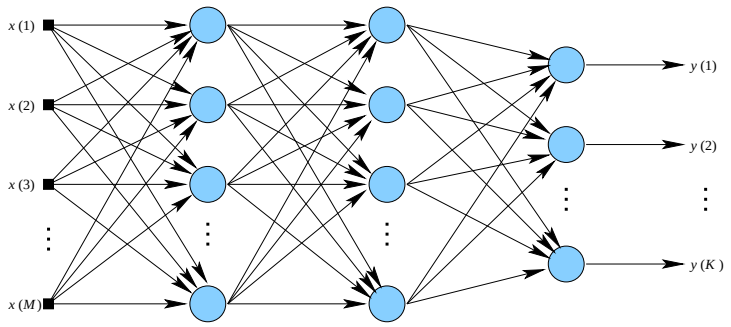

Figure 1: Vanilla neural network.

   Let the network structure be the following:

   - The input is "flattened" $3 \times 64 \times 64$-dimensional
   - On the 1st layer there are 100 nodes (marked in blue)
   - On the 2nd layer there are 100 nodes (marked in blue)
   - On the 3rd (output) layer there are 10 nodes (marked in blue; one for each class)

   Compute the number of parameters (weights) in the net.

   b) An old rule of thumb states that the number of training samples should be at least 5 times the number of coefficients. Compute the desired sample size based on this rule for (a).

2. `python` *Load Traffic sign data for deep neural network processing.*

   Download an extended version of the two class German Traffic Sign Recognition Benchmark (GTSRB) dataset (GTSRB_subset_2.zip) from the course Moodle page.

   This time, images are in color and there are about 400 from both classes. Split your data into two parts - 80% for training and 20% for testing. Note that there are ready-made functions for that.

3. `python` *Define the network in Keras.*

   Define the above network in your code. You may in the beginning reduce the number of neurons from 100 to 10 in the two layers.

4. `python` *Compile and train the net.* (8 pts)

   Use the following parameters:

   - **Loss:** binary/categorial crossentropy
   - **Optimizer:** stochastic gradient descent (SGD)
   - **Number of epochs:** 10

   Compute the test set accuracy for your network.

NOTE: if you have GPU in your laptop/desktop report training times per epoch for the both GPU training and CPU training. In the case of GPU you may need to adjust the *batch_size* option to make sure the data fits to your GPU memory.