Deep Learning for Computer Vision
Assignment Sheet 2 - Due 27.11.2017 23:59

Second exercise group: 28.11.2017, 13:30
This time will try to train a classifier(s) to recognize a Japanese characters. You will work with a dataset with 19909 training and 3514 testing images, see figure 1. The dataset contains 10 characters therefore we have to train 10 classes.

`http://codh.rois.ac.jp/char-shape/software/pmjt_sample_20161116.tar.gz`

し　に　の　て　り　を　か　く　き　も

Figure 1: A example pictures from the dataset

Exercise 2.1: multi-class logistic regressor (10 + 30 points)

- Normalize the dataset in the way that each image has zero mean and unit length. Transform the desired outputs from training and testing data into one-hot format compatible with Tensorflow [1]
- Train a simple logistic regression. The probability that a datum $\mathbf{x}$ belongs to a class i is

$$p\left(y=i|\mathbf{x}\right)=\exp\left(\mathbf{w}_i^T\mathbf{x}\right)/\sum_{c\in C}\exp\left(\mathbf{w}_c^T\mathbf{x}\right) \tag{1}$$

  The value which has the highest likelihood correspond to a class currently assigned to the datum $\mathbf{x}$. There is a different input-output relationship, therefore, the $L2$ loss does not work for us and you need to use softmax cross entropy with logits. A simple step-by-step tutorial how to build a classifier can be found here:

`https://www.tensorflow.org/get_started/mnist/beginners`

Exercise 2.2: training conv-nets (20+10+5+10+10 points)

- Build a simple conv-net architecture with at least two convolution layers to train features and one fully connected layer to classify the features. The last fully connected layer must have 10 units to fit into number of classes.
- Train the prosed architecture on the normalized dataset from Exercise 2.1. Use smaller batches instead of entire training dataset.
- Evaluate accuracy on the test and training data on the network.
- Remove one convolution layer and describe by your own words how does it influence training and testing data.
- Try different step sizes for the training. Explain what happens when the step-size is too large and too small. Draw graphs which how does the accuracy changes over iterations with different step sizes.
- (*Bonus*) Perform a hyper-parameter tunning. Play with the architecture and evaluate accuracy with k-fold cross-validation. Write with your own words what and how do the changes influence the overall accuracy of training data.

`https://www.tensorflow.org/get_started/mnist/pros`

[1] See https://www.tensorflow.org/api_docs/python/tf/one_hot. One-hot encoding encodes a desired output $c \in C$ into a vector of length card (C) where only the bit that encodes the class c is one.