Artificial Neural Networks

Advanced Machine Learning

April 5, 2018

In this lab you are going to experiment with ANN. To facilitate you with the task we provide the file `utils.py` with methods to read the input dataset.

1. Read the dataset file `mnist_small.csv`. It contains the unrolled 28x28 images of handwritten digits. Originally there are 10 classes that correspond to digits from 0 to 9.

2. Divide the data into two classes: numbers less than 5, and numbers greater or equal to 5.

3. Implement a simple feed-forward neural network architecture with 100 hidden units. Use sigmoid activation for the hidden layer. Tensorflow object for instantiating a simple ANN layer is `tf.layers.dense`. Last layer should have a sigmoid activation, and consists from one single neuron (binary classification). To to specific rules of tensorflow, you should use linear activation for the last layer, and the activation is computed within the loss function. Use `tf.nn.sigmoid_cross_entropy_with_logits` as your loss function. Remember that since you apply the loss to many data samples at once, you need to average the loss.

4. Compare the performance of a single hidden layer ANN with KNN classifier (you used it on this dataset before).

5. Check whether you have overfit the training data.

6. Try to implement an ANN that performs multi-class classification. You need to transform the target label into one-hot embedding format. You can use `sklearn.preprocessing.OneHotEncoder`. For multi class problem, you should use `softmax` activation for the last layer. The tensorflow method is `tf.nn.softmax_cross_entropy_with_logits.` The size of the last layer should be equal to the number of classes.

7. Compare the results with the multi-class KNN classifier.

8. Check whether you have overfit the training data.

9. Try to use more hidden layers.