MNIST Challenge Report

Adnan Akhundov – adnan.akhundov@mytum.de

1 Pre-processing

Before applying each of the models described in the sections below, the process of "deskewing" was applied to the input images with the aim of improving predictive accuracies throughout the model applications. Also, the input image vectors were normalized before applying all models except k-NN.

2 K-Nearest Neighbors

Six-fold cross-validation was performed for different combinations of distance metrics (Manhattan, squared Euclidian, and cosine) and values of k (1, 3, 5, 7, 9, 17, and 33). The lowest average cross-validation error was obtained for cosine distance and $k = 3$. Corresponding test set error of 1.66% was observed.

3 Logistic Regression

Multi-class logistic regression model with softmax output was implemented. Bias was included by concatenating each input vector with a unit component. Training was performed by means of mini-batch stochastic gradient descent with L2-regularization. The model was trained for different combinations of batch sizes (50, 100, and 200), learning rates (0.02, 0.05, 0.1, 0.5, 1.0), and regularization coefficients (0.0, 0.0001, 0.0005). Each training lasted 100 epochs, each covering the entire input dataset. The lowest validation set error was obtained for batch size = 100, learning rate = 1.0, and regularization coefficient = 0.0001. Corresponding test set error of 4.57% was observed.

4 Neural Networks

Neural network with two hidden layers and softmax output was implemented. Bias was included by concatenating each input vector with a unit component. Training was performed by means of mini-batch stochastic gradient descent with dropout. The weights were initialized from the standard normal and normalized by the square root of the number of inputs. Momentum learning (with the coefficient = 0.9) was applied to speed up the learning process. The model was trained for different combinations of activation functions of hidden layers (*sigmoid*, *tanh*, and *relu*), numbers of units in hidden layers (250/150, 500/300, 1000/600), batch sizes (50, 100, and 200), learning rates (0.01, 0.05, 0.1), and dropout rates (0.0, 0.5). Each training lasted 100 epochs, each covering the entire input dataset. The lowest validation set error was obtained for *relu* activation function, 1000/600 units in hidden layers, batch size = 100, learning rate = 0.1, and dropout rate = 0.5. Corresponding test set error of 1.19% was observed.

5 Gaussian Processes

GP framework was applied for modelling ten separate one-vs-rest regression functions for each of the ten data classes (with assumed function values of 1 corresponding to an image belonging to the respective class, and -1 corresponding to an image belonging to some other class). As per standard GP procedure, the values of the regression functions at unobserved points (test set) were inferred given their values (1 or -1) at the observed ones (training set). The assignment of actual class labels to the images in the test set was done in accordance with the highest predicted mean among ten regression functions. Squared exponential kernel was used as a kernel function. The optimal value of $l^2 = 33$ was determined empirically. As predicted variance was not relevant for this particular modelling, σ_f^2 was taken equal to 1 for the sake of efficiency. As the target labels of the training set were assumed to be assigned correctly, σ_y^2 was set to 0 (no output noise). The limited computing resources available allowed the modelling restricted to the first $40,000$ training inputs only. The resulting test set error of 1.04% was observed.

6 Final Result

The resulting average error score over all four models applied was 2.115%.