OSAMA ABUZAID & ANTTI LUIRO

Team Pytlab - Data Analysis Challenge

Our approach to the challenge included a few different methods, but clearly the best results were achieved by utilizing neural networks. Hence this report discusses only neural network approaches.

Classification: Neural network

In classification we used neural network with 50 input neurons, 2 output neurons and 2 hidden layers of 30 neurons. In initialization we randomized the values for biases and weights from normal distribution with mean 0 and variance 1/sqrt(n), where n is the number of neurons in the previous layer. Each neuron was a sigmoid neuron with output sigmoid(wa+b), where w denotes the weight vector, a the activation vector of the previous neurons and b the bias of the vector. The output neurons represented the classes of ”good review” and ”bad review”.

We split the training set into two parts: 4500 for training and 500 for validation. In training we used logloss-function with regularization term λ|w|^2, λ=0.5. For each epoch we further split the data in random disjoint subsets of 500 data points and used backpropagation for each subset separately to train network. The step factor in gradient descent was 0.005, we ran epochs until there was 200 full epochs without improvement, and then chose the network with smallest validation error. About 300 full epochs were needed before minimum was reached.

Regression: Neural Network

In regression we used neural network with 50 input neurons, 1 output neuron and 2 hidden layers of 30 neurons. In initialization we randomized the values for biases and weights from normal distribution with mean 0 and variance 1/sqrt(n), where n is the number of neurons in the previous layer. Each neuron was a linear neuron with output max(0, (wa+b)), where w is the weight vector, a the activation vector of the previous neurons and b the bias of the vector. The output was the number of votes.

We split the training set into two parts: 4900 for training and 100 for validation. In training we used square loss with regularization term λ|w|^2, λ=0.5. For each epoch we further split the data in random disjoint subsets of 500 data points, and used backpropagation for each subset separately to train network. The step factor in gradient descent was 0.005. We ran 2000 epochs in one iteration, and if validation error seemed to continue decreasing, we made a new iteration. Total of 14 iterations were needed. We calculated the estimates for number of votes with our trained network and rounded them into closest whole number, and used it as an estimate. Final error to the test data was 0.0090.