Neural Networks in Tensorflow (Assignment 1)

JIANG Fan, fan.jiang@polytechnique.edu

Introduction

For this assignment, we are going to use neural network implemented with Tensorflow to tackle a classification problem. The data set contains 2000 points, splitted to train/test data set with a ratio 6/4. The dimension of X is 294, and the dimension of Y is 6. Since the classes of Y are independent, we have used sigmoid classifier at the output layer. The number of neurons in each layer of my network is 294-64-32-16-6, and the activation functions are all sigmoid functions. The final Hamming loss achieved by the network is around 0.85-0.95 (depending on the random state).

Hyper-parameter Tuning

The best learning rate is 0.005, and the best batch size is 100. The following left table shows the process of finding the best structure, and the following right table shows the process of finding the best learning rate:

Hidden layer structure	Epoches	Learning rate	Time	Hamming loss
16	2000	0.005	24s	0.10604
32-16	2000	0.005	39s	0.10208
64-32-16	2000	0.005	47s	0.08958
128-64-32-16	2000	0.005	76s	0.09583
256-128-64-32-16	2000	0.005	138s	0.09271

Learning rate	Epoches	Hidden layer structure	Time	Hamming loss
0.0005	4000	64-32-16	95s	0.09771
0.001	4000	64-32-16	98s	0.10125
0.005	4000	64-32-16	96s	0.08708
0.01	4000	64-32-16	96s	0.10104
0.05	4000	64-32-16	96s	0.11292

This table shows the process of finding the best batch size with training time of 2 minutes.

Batch size	time limits	Hidden layer structure	Learning rate	Hamming losses
50	2000	64-32-16	0.005	0.09021
100	2000	64-32-16	0.005	0.08729
200	2000	64-32-16	0.005	0.09333
400	2000	64-32-16	0.005	0.09833
800	2000	64-32-16	0.005	0.09916

Benchmark Comparison

I have also compared the training time and Hamming loss of my neural network with several benchmarks, including K Nearest Neighbour classifier, Decision Tree, One vs Rest classifier, classifier chain and also the neural network implemented by Scikit-learn.