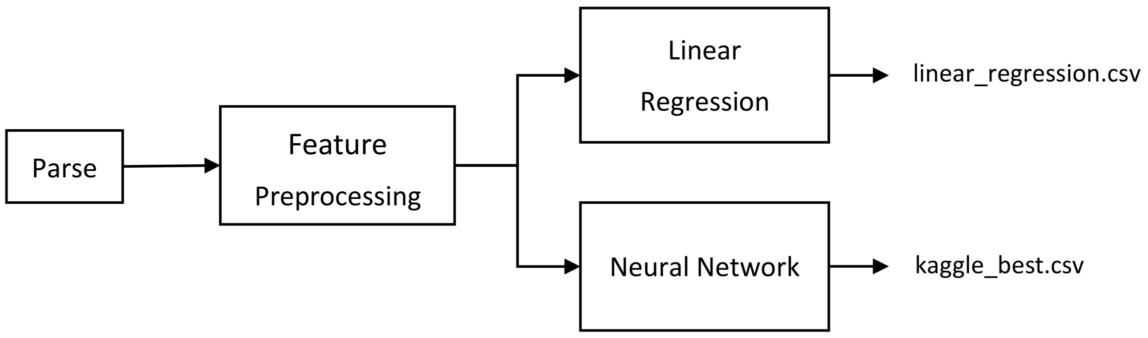
Machine Learning Homework 1 Report

Student ID: r05922063　Name: 陳啟中

Overview



Parse

In parsing phase, the purpose is to read train.csv and test_X.csv and interpret them into three matrices: training feature matrix, training label matrix and testing feature matrix.

Feature matrices, including training and testing, consist of rows of feature sets. Each row represents an instance. In a row, there are totally 9 * 18 columns, consisting 18 kinds of air observational data in 9 sequential hours. The design of feature sets is obtained by test_X.csv, in which these 9 * 18 features are given.

Training label matrix consists of rows corresponding to rows in training feature matrix. Each row records the PM2.5 index at 10th hour.

Feature Preprocessing

After some training on linear regression model with L1 regularization, which is embedded with feature selection function, I found some features are useless and even misleading to training models by observing some weights on features are very close to 0 and should be pruned. Eventually, I selected NO2, NOx, O3, PM10, PM2.5, RAINFALL, SO2 with recently 7 hours, totally 49 features.

Linear Regression

My linear regression model takes 49 features and 49 weights and a bias term to perform linear combination and output the prediction result. The following are some details:

Training Data Random Shuffling

Upon several iterations of training, the program shuffles the order of training data randomly. This prevents the training model from learning the bias obtained by the certain order of data.

Early stopping

To prevent overfitting, early stopping is an easy and efficient way. The key point is to find the ending point just before overfitting occurs. How? I firstly split the training data into 66% subset training data and 33% validation data. Upon each epoch (1000000 iterations), the program calculates the testing MSE on validation data. Once the testing MSE grows, we then find the stopping point. Finally, I apply the stopping criteria on the whole training data and train model without overfitting.

Discussion on L2 Regularization

Regularization prevents overfitting, as well as early stopping does, while it's even easier to implement because we don't need to run the program twice. Instead, we just run the program, the longer the better. However, it's harmful to well-fitting with excessive lambda.

The following chart is the testing MSE corresponding to different lambda chosen. Their stopping criteria is obtained by early stopping. We can observe that the effect of regularization is not significant after applying early stopping. And after applying feature selection, regularization becomes useless. Therefore, I disable regularization afterward due to the result.

	No regularization	**λ=0.01**	**λ=0.1**	**λ=1**	**λ=10**
W/o feature selection	31.5807	31.5797	<u>31.5729</u>	31.619	34.726
W/ feature selection	<u>31.001</u>	31.0025	31.018	*(not test)*	*(not test)*

Discussion on learning rate

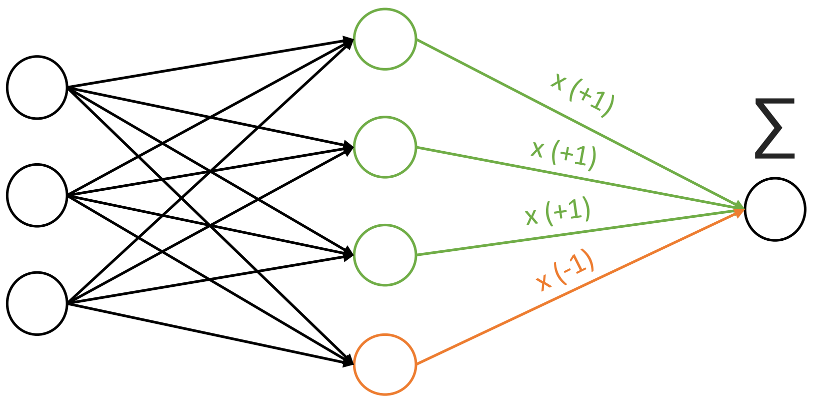
The following chart shows the stopping iteration count evaluated by validation (described in Early Stopping section) and testing MSE regarding different learning rates (**η**):

	η=1e-8	**η=1e-9**	**η=1e-10**
Stopping iteration count (x1e6)	15	242	2089
Testing MSE	32.4745	31.1976	31.1617

It’s clear to see as learning rate gets lower, the stopping iteration count gets higher and the MSE gets lower. In a word, it’s a trade-off between time and precision.

Neural Network

Topology



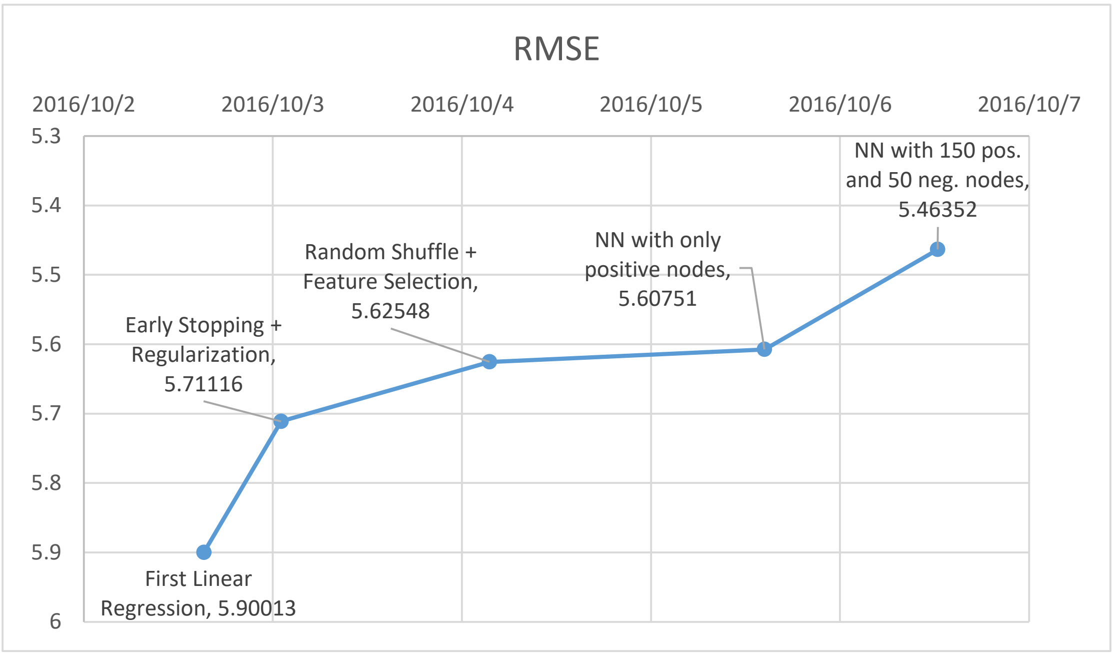
Input layer	**Hidden Layer**	**Output Node**
49 features	ReLU 150 positive nodes 50 negative nodes	Summation

Description

Between input layer and hidden layer, there are configurable synapses (weight and bias). Each hidden node perform linear combination and then rescale the result by ReLU function. Then, multiply the output by +1 or -1 depending on whether the hidden node is a positive node or a negative node. Finally, the output node sums up them and makes a prediction.

Why there should be positive nodes and negative nodes? Because ReLU function is always positive, making all negative factors diminishing if we don’t put negative nodes in the topology. The learning algorithm adopts stochastic gradient descent backpropagation.

Milestone



RMSE
2016/10/2
2016/10/3
2016/10/4
2016/10/5
2016/10/6
2016/10/7
5.3
5.4
5.5
5.6
5.7
5.8
5.9
6
First Linear Regression, 5.90013
Early Stopping + Regularization, 5.71116
Random Shuffle + Feature Selection, 5.62548
NN with only positive nodes, 5.60751
NN with 150 pos. and 50 neg. nodes, 5.46352