**ML2016\_hw1\_report 姓名: 徐有慶 學號: r05922162**

1. **Linear regression function by Gradient Descent.**

double LinearRegression::lossFunction(vector<vector<double> > train\_set, vector<double\*> &past\_gradients)

{

double error\_value = 0;

int parameters\_num = parameters.size();

double \*gradients = new double [parameters\_num];

for(int i = 0; i < parameters\_num; i++)

{

gradients[i] = 0;

}

for(int m = 0; m < train\_set.size(); m++)

{

double y\_head = y\_heads[m];

double features[parameters\_num] = {0};

features[0] = 1;

for(int i = 1; i < parameters\_num; i++)

{

features[i] = train\_set[m][i-1];

}

double y = 0;

for(int i =0; i < parameters\_num; i++)

{

y = y + parameters[i]\*features[i];

}

for(int i = 0; i < parameters\_num; i++)

{

gradients[i] = gradients[i] + (-2)\*(y\_head - y) \* features[i]; //compute every feature's gradient

}

error\_value = error\_value + (y\_head - y)\*(y\_head - y);

}

past\_gradients.push\_back(gradients);

gradientDescent(gradients, past\_gradients);

return error\_value;

}

void LinearRegression::gradientDescent(double gradients[], vector<double\*> past\_gradients)

{

int gradients\_num = parameters.size();

double learning\_rate = 1;

double sigma\_past[gradients\_num] = {0};

for(int i = 0; i < past\_gradients.size(); i++)

{

for(int j = 0; j < gradients\_num; j++)

{

sigma\_past[j] = sigma\_past[j] + past\_gradients[i][j] \* past\_gradients[i][j]; // Agagrad, compute the past gradient's sum

}

}

for(int i = 0; i < parameters.size(); i++)

{

parameters[i] = parameters[i] - learning\_rate \* gradients[i] / sqrt(sigma\_past[i]); // Adjust the weight

}

}

1. **Describe your method**

取每項的前5個小時來預測第6個小時PM2.5的值，每天有18項，所以會取90 筆資料當作features來訓練，而要跑test set時就只取最後的5個小時來當作features。初始weight則在0~0.01當中隨機產生出來，會這樣選的原因在於這樣子算出來的y值會比較符合實際上PM2.5的值。在本次作業中使用Adagrad來動態調整learning rate，並加上了regularization。

1. **Discussion on regularization**

根據實驗結果，在其他條件不變的情況下在lambda設為100時訓練出來的model與lambda設為0時訓練出來的model差異並不太大，有時反而lambda設為0訓練出的model在test set當中的表現較佳。

**4. Discussion on learning rate**