b03901057 詹晉誠

A. Linear regression function by Gradient Descent.

• Code :

```
while not converged :
    curIter = curIter + 1 # current iteration
    tmp = np.dot(trainData, beta) - gndData # X*beta - y
    new_beta = beta - 2*eta*np.dot(np.transpose(trainData), tmp) + lamb *
    beta # beta_hat = beta - 2*eta*X'*(X*beta-y) + lambda*beta
    diff = np.linalg.norm(new_beta - beta, 2) # change of beta
    beta = new_beta # update beta
```

Notation : trainData - X; gndData - Y

B. Describe your method

• Data Preparation

I use the concentration of PM2.5, PM10, NO2, and the ambient temperature (AMB_TEMP) over the past 9 hours as features. The training data are collected in a sliding window manner. More specifically, the values of PM2.5, PM10, NO2, and AMB_TEMP in hour 0-8 are concatenated into a row vector and used as a feature. The value of PM2.5 in hour 9 is used as the ground-truth of the training data. Then the window moves rightward — features in hour 1-9 are extracted, and the ground-truth is the value of PM2.5 in the tenth hour. Also, the window may cross two consecutive days. Finally, training data are divided into two parts, one for training and the other for validation.

• Training

I use the simple linear regression method in this homework. The mathematical formation is as follows. Let X be the training data, y be the ground truth data of the training data, β be the linear combination coefficients. We have to optimize the following equation :

$$min \quad \|y - X\beta\|_2^2 + \lambda\|\beta\|_2^2$$

The coefficient vector β can be updated by the following equation,

$$\hat{\beta} = \beta - 2 * \eta - X^T(X\beta - y) + \lambda\beta$$

In addition, since this is a least square problem, there is a closed form solution. In this homework, I also use the closed form solution to solve the least square problem without regularization. Experiment results show that the results of gradient descent (without regularization) converge to those of the closed form solution method. Therefore, we know that the gradient descent method has arrived at the global minimum, instead of being stuck at local minimum or flat regions.

C. Discussion on regularization

Regularization is used in this homework to prevent overfitting. However, regularization doesn't help much in my model. Experiment results (rmse of errors) are shown in the table below: (learning rate = 5e-9)

λ	0	1E-10	1E-05	1E-04	1E-03
Training error	5.9394	5.9394	5.9590	3E+17	nan
Validation error	5.5689	5.5689	5.5794	2.2E+17	nan

As the regularization parameter λ increases, the error of validation set increases as well. The reason is that my model is already oversimplified, so regularization makes the model even more simpler, and thus increases the error rate.

D. Discussion on learning rate

Learning rate in gradient descent determines the "step size" of each iteration. According to my experiment, the learning rate should be chosen around 1e-10. If the learning rate is too low, the program will converge very slowly, resulting in very long training time. On the other hand, if the training rate is too large, the program will diverge.

The table below displays the error on the validation set after running 50,000 iterations.

learning rate	1E-08	5E-09	1E-09	1E-11	1E-13
Validation Error	nan	5.568	5.568	6.004	10.199
Time (s)	-	75	75	82	75

When the learning rate to too low, the program does not arrive at the global minimum even after 50,000 iterations. Experiment results show that 5e-9 would be a fair choice of the learning rate.

E. Other discussions

• Selection of features

I have also tried other features. For example, I have used all the measurements in the training data as features, but the results show that this method does not make any improvement. Indeed, more data/features does not guarantee better results since linear regression assumes that input features (independent variables) are uncorrelated to each other. Using all the measurements as features may violate this assumption. Also, overfitting may also be a reason since the model now has become much more complex than before.

• Higher order polynomial

I also tried to use higher order polynomials to fit the data. However, experiment results show that the first order model outperforms the second order model.

• Kernel trick

I also used the kernel trick to map features to higher dimensional space because I suspect that linear model may not be powerful enough to explain the dependency of current PM2.5 concentration on previous hours' data. But unfortunately, this kernel trick doesn't help either.

In all, based on my experiment results, first order linear model seems to be the best one.