University of Louisiana at Lafayette

School of Computing and Informatics

INFX-598 Machine Learning Applications

Homework # 4

Submitted By

Navid Yousuf

ULID: C00419219

Fall 2021

1. Which Linear Regression training algorithm can you use if you have a training set with millions of features?

   If I have a training set with millions of features, I will use Stochastic Gradient Descent or Mini-batch Gradient Descent, and perhaps Batch Gradient Descent if the training set fits in memory. But, the Normal Equation or the SVD approach can't be used because the computational complexity grows quickly with the number of features.

2. Suppose the features in your training set have very different scales. Which algorithms might suffer from this, and how? What can you do about it?

   If the features in my training set have very different scales, the cost function will have the shape of an elongated bowl, so the Gradient Descent algorithms will take a long time to converge. To solve this, the data need to be scaled before training the model. Moreover, regularized models may converge to a suboptimal solution if the features are not scaled: since regularization penalizes large weights, features with smaller values will tend to be ignored compared to features with larger values.

3. Can Gradient Descent get stuck in a local minimum when training a Logistic Regression model?

   Gradient Descent cannot get stuck in a local minimum when training a Logistic Regression model because the cost function is convex.

4. Do all Gradient Descent algorithms lead to the same model, provided you let them run long enough?

   No. If the learning rate is too high, then the model can diverge. It can also only reach the local minimum based on where the initialization is.

5. Suppose you use Batch Gradient Descent and you plot the validation error at every epoch. If you notice that the validation error consistently goes up, what is likely going on? How can you fix this?

   If the validation error consistently goes up after every epoch, then one possibility is that the learning rate is too high and the algorithm is diverging. If the training error also goes up, then this is clearly the problem and the learning rate should be reduced. However, if the training error is not going up, then the model is overfitting the training set and training needs to be stopped.

6. Is it a good idea to stop Mini-batch Gradient Descent immediately when the validation error goes up?

   No, because it will be erratic in approaching the minimum. Therefore, it is better to revert to the best case if the error does not improve for a while.

7. Which Gradient Descent algorithm (among those we discussed) will reach the vicinity of the optimal solution the fastest? Which will actually converge? How can you make the others converge as well?

   The Stochastic Gradient Descent will reach the fastest since we are using one random training data at each iteration. However, the Batch Gradient Descent is the only one to actually converge. Others cannot be converged; they will only approach close to the global minimum.

8. Suppose you are using Polynomial Regression. You plot the learning curves and you notice that there is a large gap between the training error and the validation error. What is happening? What are three ways to solve this?

   If the gap exists between the training and the validation error, that means the model is overfitting the data. To avoid overfitting the data, the three things we can do: 1. train more data, 2. regularize the model, and 3. reduce the complexity of the model.

9. Suppose you are using Ridge Regression and you notice that the training error and the validation error are almost equal and fairly high. Would you say that the model suffers from high bias or high variance? Should you increase the regularization hyperparameter α or reduce it?

   The model suffers from high bias, because the errors are both high, indicating wrong assumptions and therefore underfitting. In order to reduce high bias, we have to decrease alpha.

10. Why would you want to use: a. Ridge Regression instead of plain Linear Regression (i.e., without any regularization)? b. Lasso instead of Ridge Regression? c. Elastic Net instead of Lasso?

    a. Ridge Regression instead of plain Linear Regression
       We would use ridge regression to regularize the Linear Regression so that overfitting can be avoided.

    b. Lasso instead of Ridge Regression
       We would use Lasso, which uses L1 norm regularization, automatically eliminates the weights of the least important features and therefore performs feature selection.

\

c. Elastic Net instead of Lasso
Elastic Net is preferred over Lasso if there are lots of features or lots of strongly correlated features.

11. Suppose you want to classify pictures as outdoor/indoor and daytime/nighttime. Should you implement two Logistic Regression classifiers or one Softmax Regression classifier?

    If we want to classify pictures as outdoor/indoor and daytime/nighttime, since these are not exclusive classes we should train two Logistic Regression classifiers.

12. Implement Batch Gradient Descent with early stopping for Softmax Regression (without using Scikit-Learn).

    https://github.com/navid015/Assignment-4-Ques-12/blob/main/Assignment%2004.ipynb