KATAYOUN-KOBRAEI

99222084

Exercise 1

L1 AND L2 REGULARIZATION

L1 regularization:

In this type of regularization, the weight parameters are reduced close to zero. When the data is normalized or near zero, applying this regularization reduces the number of features in the learning algorithm. This is achieved by assigning zero weights to less important features and larger weights to more influential features.

$$L(x,y) \equiv \sum_{i=1}^{n}(y_i - h_\theta(x_i))^2 + \boxed{\lambda \sum_{i=1}^{n}|\theta_i|}$$

L2 regularization:

This type of regularization keeps the model's weights small, but unlike the previous regularization method, it does not reduce the values to zero. As a result, it does not lead to a sparse model. However, this regularization does not perform well when working with outliers. This is because the model's prediction error increases significantly at outlier points, and the penalty term causes the model's weights to become smaller.

This approach works better when all input features have an impact on the predicted target value, and when the model's weights are initially set to approximately equal values.

$$L(x,y) \equiv \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \boxed{\lambda \sum_{i=1}^{n} \theta_i^2}$$

THE DIFFERENCE BETWEEN L1 AND L2

Apart from the difference in formulas, there are other distinctions between these two regularizations. We know that **L1 regularization** creates a sparser model. Additionally, it generates different models by affecting different subsets of features. However, with **L2 regularization**, we only have one learning model because it does not operate based on different subsets of features. In **L1 regularization**, we have **feature selection**, whereas in **L2 regularization**, we do not. When the target value is a function of all the features, **L2 regularization** leads to better learning.

In general, **L1 regularization** can create a simpler and more interpretable model, but it may not perform well with complex algorithms. In contrast, **L2 regularization** can learn complex algorithms from the input data.

WHICH ONE SHOULD BE PREFERED?

Depending on the issue we are facing, our preference may vary. If we are looking for a stronger solution with more features, **L1** would be better. This software will have higher computational costs, whereas **L2** will have lower computational costs.

In **L1** regularization, by subtracting a small amount from the weights in each iteration, it forces the weights of non-informative features to zero, ultimately resulting in zero weights. For **L2** regularization, if we consider model complexity as a function of the weights, the complexity of a feature is proportional to the absolute value of its weight. **L2** regularization drives the weights towards zero but does not make them exactly zero. It works in such a way that only a small percentage of weights are reduced in each iteration, so the weights never become exactly zero.

Exercise 2

In the first part, we change the number of hidden layers. This might increase accuracy, or it might not, but ultimately it will help to some extent and may eventually lead to overfitting.

In the second part, by applying dropout to some neurons, we can increase accuracy. This will also help improve learning speed and reduce the risk of overfitting.