Advanced Machine Learning

Likhit Nayak

GitHub for the course

Link:

https://github.com/likhitnayak/Advanced-Machine-Learning-SiliconTech

This repository will contain all the lecture slides and any relevant codebases.

Regularization

Definition:

Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.

Types of regularization:

1. L2 regularization
2. L1 regularization
3. Dropout

Regularization

<u>Definition:</u>

Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.

<u>Types of regularization:</u>

1. **L2 regularization**
2. **L1 regularization**
3. Dropout

L2 regularization

Let’s begin with an example of linear regression:

$$\mathbf{y = w_0 + w_1x_1 + w_2x_2 + \ldots\ldots + w_nx_n}$$

$$\mathrm{RSS} = \sum_{j=1}^{m}\left(Yi - Wo - \sum_{i=1}^{n} Wi\,Xji\right)^2$$

L2 regularization

To regularize the loss function (or the objective function), we add a parameter norm penalty that gives us the new loss function:

$$\text{Loss} = \sum_{j=1}^{m}\left(Yi - Wo - \sum_{i=1}^{n} Wi\,Xji\right)^2 + \lambda \sum_{i=1}^{n} Wi^2$$

L2 regularization

If we extend this to neural networks, our loss function becomes:

$$\text{Loss} = \frac{1}{m}\sum_{i=1}^{m} L\left(\hat{y}^{(i)}, y^{(i)}\right)$$

L2 regularization

If we extend this to neural networks, our loss function becomes:

$$\text{Loss} = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) + \frac{\lambda}{2m}\sum_{l=1}^{L} \|w^{[L]}\|_F^2$$

where *m* is the size of the training set, and *L* is the number of layers in the neural network

L1 regularization

Instead of L2 norm, we add L1 norm as the penalty:

$$\text{Loss} = \sum_{j=1}^{m}\left(Yi - Wo - \sum_{i=1}^{n} Wi\,Xji\right)^2 + \lambda \sum_{i=1}^{n} |Wi|$$

L1 regularization results in a **sparse solution,** and is often used for feature selection.

Why does parameter norm penalty reduce overfitting?