CHAPTER 1: OVERFITTING

1.1 Introduction

There is an un-detouring issue in supervised machine learning when the model does not generalize correctly from observed data to unseen data. That is called overfitting, a scenario when a model performs perfectly on a training set but fits poorly on a testing set. This is due to the fact that an over-fitted model has difficulty coping with pieces of the information in the testing set, which may be different from those in the training set. On the other hand, over-fitted models tend to memorize all the data, including unavoidable noise on the training set, instead of learning the discipline hidden behind it.

Overfitting occurs when a machine learning model tries to cover all the data points or more than the required data points present in the given dataset.

1.2 Cause of overfitting

- Lack of data: The data requirement for training a machine learning model is usually large. Because when a model tries to learn from a small dataset, it will tend to have greater control over the dataset and will make sure to satisfy all the data points exactly.

- Noisy data: Noise is the measure of irrelevant information or randomness in a dataset. If a dataset contains too much noise, the model tends to memorize the noise instead of learning the underlying patterns. Noisy data also causes an overly complex model.

- Model is too complex: A model is too complex and becomes overfitting when it is not learned from the sufficient data required.

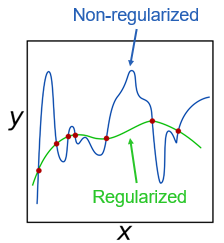
- Unbalance data: An unbalanced data is also a cause to overfitting, some features may dominate the others

- Overlearning: Overlearning is when a model has learned for many epochs, it may start to memorize the training data, rather than learning the underlying patterns.

1.3 Solutions

1.3.1 Regularization

The output of a model is generally affected by multiple features. The larger the number of features, the more complicated the model is. An overfitting model tends to take all features into consideration, even though some of them have very limited effect on the final output. Or even worse, some of them are noises which are meaningless to the output.



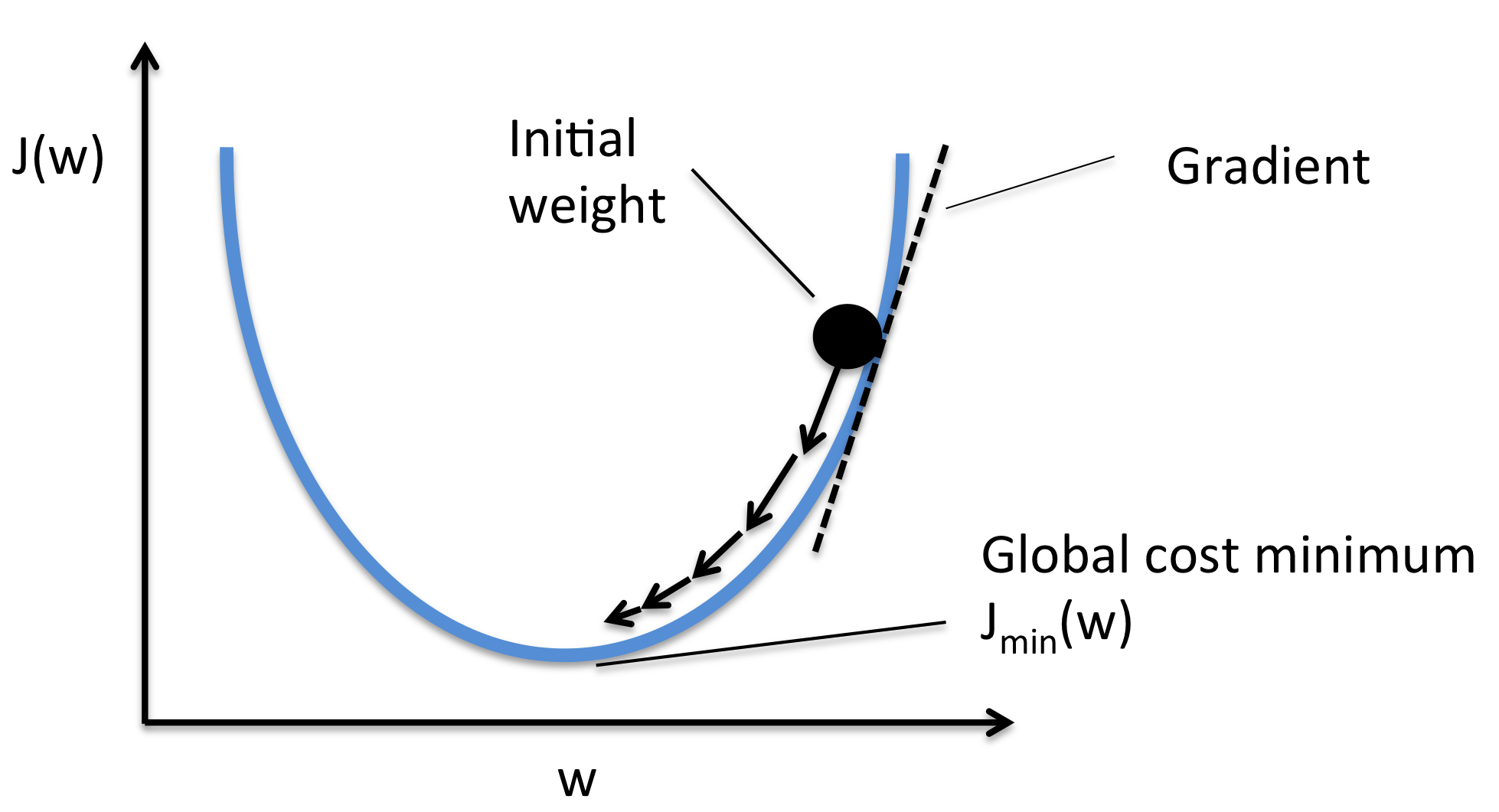
Picture 1.1 Regularization and Non-regularization

There are two kinds of solutions to limit these cases

* Select only the useful features and remove the useless features from the model.
* Minimize the weights of the features which have little influence on the final classification.

General idea of these solutions above is to limit the effect of useless features. However, it is hard to consider which features are useless, so trying to minimize the cost function of the model is usually applied in order to limit them. To do this, a regularization term or a regularizer is added to the cost function.

* ω is weight. Gradient Descent can be used to determine the set of weights.
* X is training set.
* y is the labeled value (true value).
* is regularization coefficient.
* is the penalty term.



#### 1.3.1.1 L1 Regularization

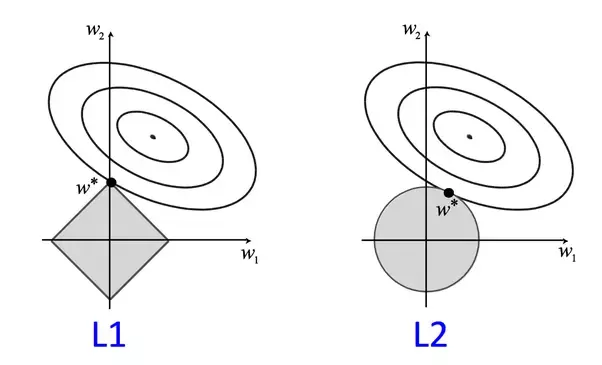
L1 regularization or L1 norm or Lasso Regression, prevent overfitting by shrinking the parameters towards 0. This makes some parameter is ignored during training process.

To minimize the cost function, the weight of some features may be set to be zero in order to remove it out of the model. This approach made the model simpler. However, some useful features with lower influence on the output may also be removed, which led to underfit.

#### 1.3.1.2 L2 Regularization

L2 regularization or L2 norm or Ridge Regression, prevent overfitting by forcing weights to be small, but not making them exactly 0.

This approach makes the networks prefer to learn features with small weights. Instead of rejecting those less valuable features like L1 norm, L2 norm gives them lower weights. So that the model can get as much information as possible. For those features that have considerable influence on the initial cost function, large weights would be given to them.



Picture 1.2 L1 regularization vs L2 regularization

#### 1.3.1.3 Dropout

Dropout is a technique where randomly selected neurons are ignored during training. It is an effective way to prevent overfitting and provides a way to approximately combining exponentially many different neural network architectures.

The idea is this method is temporarily removing a unit out of the network, along with all its incoming and outgoing connections. The choice of which units to drop is random.

A close-up of a diagram

Description automatically generated

Picture 1.3

1.3.2 Cross-validation

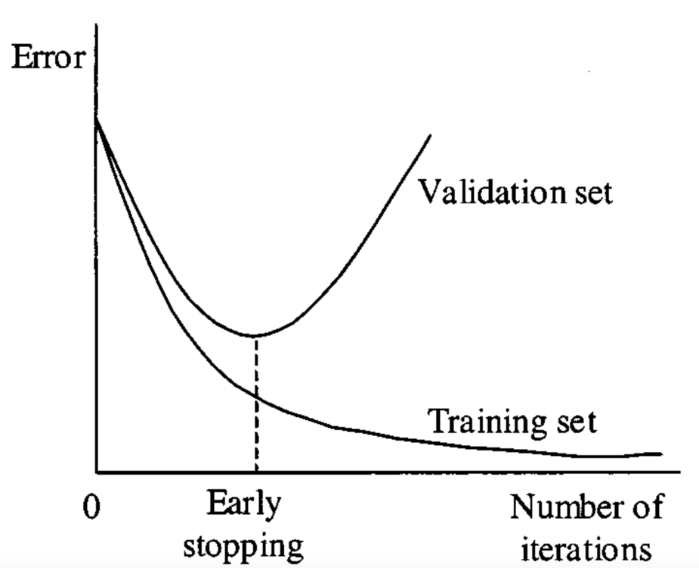
This is a technique used to check the generalization performance of a model on a new dataset. In this method, the dataset is divided into smaller folds, where the model is trained on the majority of the dataset and evaluated on the remaining data. This process is repeated for different combinations of the folds, allowing the model to be tested over all the data.



Picture 1.4 K-Fold cross validation

1.3.3 Early Stopping

This is a method used to prevent overfitting by stopping the training process before the model reaches its optimal accuracy on the training data. This method works by evaluating the model on a validation set, and when the accuracy on the validation set starts to drop or stops improving, the training is stopped.



Picture 1.5 Early stopping

1.3.4  Noise Reduction

Noisy data is one of the reasons led to overfitting. So logically, noise reduction becomes one researching direction for overfitting inhibition.

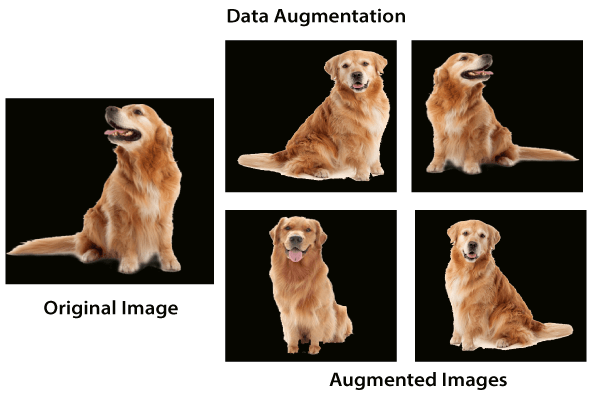
Pruning is a significant theory used to reduce classification complexity by eliminating less meaningful, or irrelevant data, and finally to prevent overfitting and to improve the classification accuracy.

* Pre-pruning algorithms function during the learning process. The tree growth is stopped based on a predefined condition.
* Post-pruning is simpler and faster than pre-pruning, but it may not be as effective as pre-pruning in reducing the model complexity. It also requires a larger validation set and costs more computational power.

1.3.5 Augment the training data

In some cases, overfitting is caused by a lack of training data. Datasets play a crucial role in affecting the performance of a model.

 Model training is a process of tuning hyper-parameters. Well-tuned parameters make a good balance between accuracy and regularity, and then inhibit the effect of overfitting, as well as that of underfitting.



Picture 1.6 Data augmentation