AIFS ML Lecture 8: Machine Learning Basics

Suraj Narayanan Sasikumar

Hessian AI Labs

May 20, 2020

Overview

1. Recap

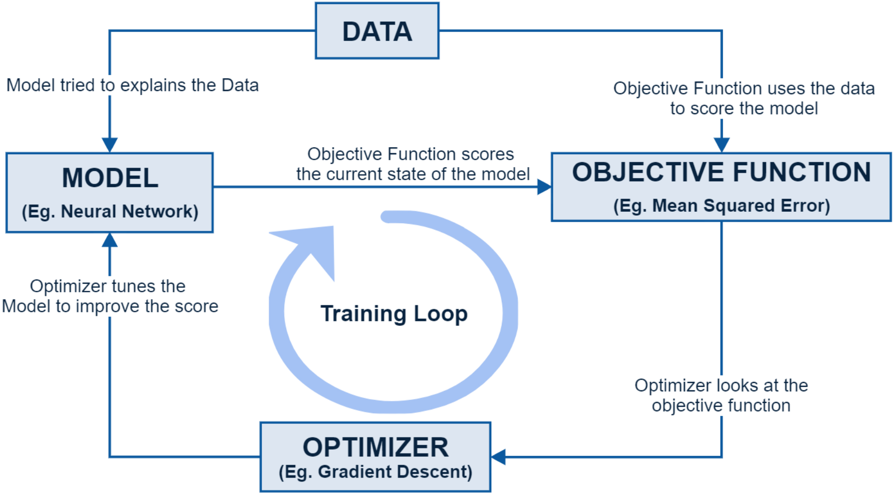
2. Ensuring Generalization

3. Generalization in Polynomial Regression

Table of Contents

1. Recap
2. Ensuring Generalization
3. Generalization in Polynomial Regression

Recap - Learning Algorithm



DATA
Model tried to explains the Data
Objective Function uses the data to score the model
MODEL
(Eg. Neural Network)
Objective Function scores the current state of the model
OBJECTIVE FUNCTION
(Eg. Mean Squared Error)
Training Loop
Optimizer tunes the Model to improve the score
Optimizer looks at the objective function
OPTIMIZER
(Eg. Gradient Descent)

Recap - Optimization vs Learning

- The goal or machine learning is to learn about the underlying data-generating process in order to perform tasks like classification, regression, clustering etc.
- Overfitting happens when we treat a machine learning algorithm purely as an optimization algorithm; by explaining the noise, it perfectly models all the data points and achieves zero mean-squared error.
- The key difference is that a properly trained machine learning algorithm is able to **generalize** to data points not seen before, ie it has predictive power.

Table of Contents

1 Recap

2 Ensuring Generalization

3 Generalization in Polynomial Regression

How to ensure generalization?

- Split the full dataset into two separate datasets, one meant for training, one meant for testing its generalization capability. The ratio of split is usually 80% for training and 20% for testing.
- After training the model on the training data, the trained model is evaluated on the test data to score it for its generalization capability.
- How generalization is achieved depends on the learning algorithm and the size of dataset
- For example, in Polynomial Regression generalization is achieved by choosing the right capacity model(polynomial degree: M), whereas in Neural network models generalization is achieved by checking the validation error after each epoch (when the network had sees all the data once) and stopping the training when the validation error starts to diverge from the test error.

Test vs Train Error

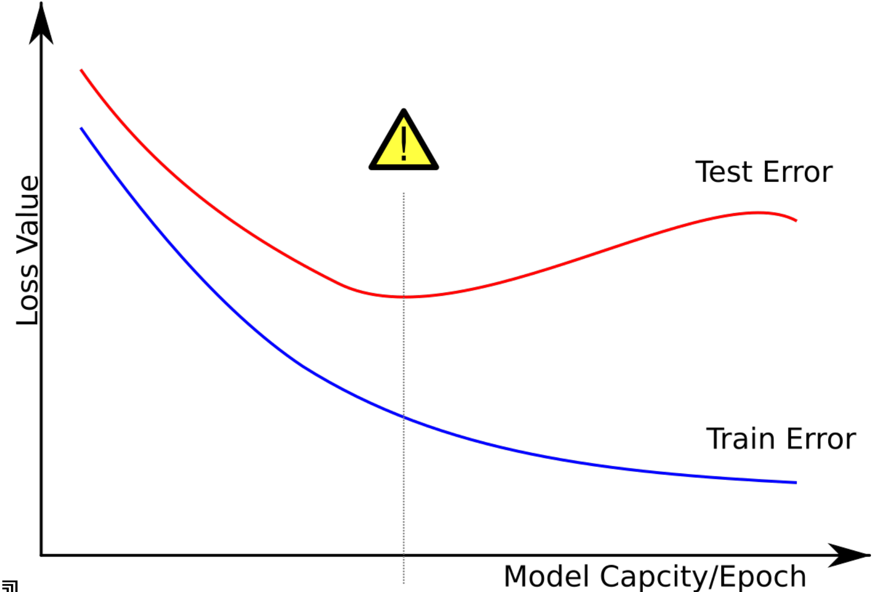


Table of Contents

1. Recap

2. Ensuring Generalization

3. Generalization in Polynomial Regression

Model Selection

- In polynomial regression the degree of the model, M controls the capacity of the model.

$$\hat{y} = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \ldots + w_M x^M$$

- If the loss function is regularized, then the regularization coefficient λ also controls the capacity of the model.

$$\mathrm{MSE}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x} - y_i)^2 + \lambda ||\mathbf{w}||_2^2$$

- In order to ensure that the learning algorithm generalizes the right capacity model has to be chosen. This means choosing the right values for the hyperparameters M and λ
- This process is called *model selection*.

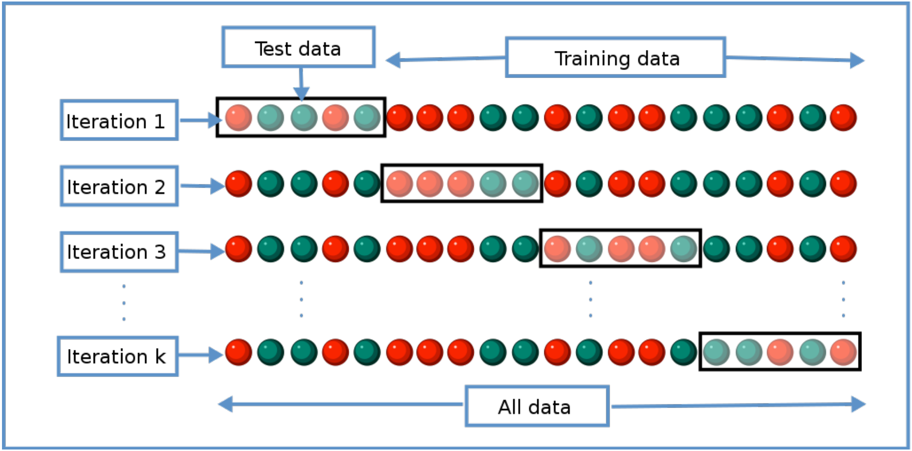
Model Selection

- Depending on the size of the dataset there are two approaches for model selection.
- **Data is abundant**:
  1. Split the dataset into three parts, *training set*, *validation set*, and *test set*.
  2. Train multiple models (polynomial model with different M and λ values) on the training set.
  3. Evaluate the loss (score) of each model on the validation set and choose the model with the lowest loss.
  4. Finally evaluate the loss of chosen model on the test set. This ensures that there is no over-fitting to the validation dataset.

Model Selection

- **Data is Scarce**: We use a method called K-fold cross-validation to choose the model.
    1. Partition the data into K groups
    2. One group is held-out as the test set and the remaining $(K-1)$ groups is used for training. This step is repeated for all K groups to be the held-out set.
    3. The loss from the K iterations from step 2 is averaged to get the final loss for the model.

Cross Validation



Drawbacks of Cross Validation

- Number of training iterations is increased by a factor of K
- When there are multiple hyperparameters to be tuned, to try different combinations the number of training iterations become exponential in the number of hyperparameters.