Topics covered on ML and DL By Andrew Ng

Linear Regression with One Variable

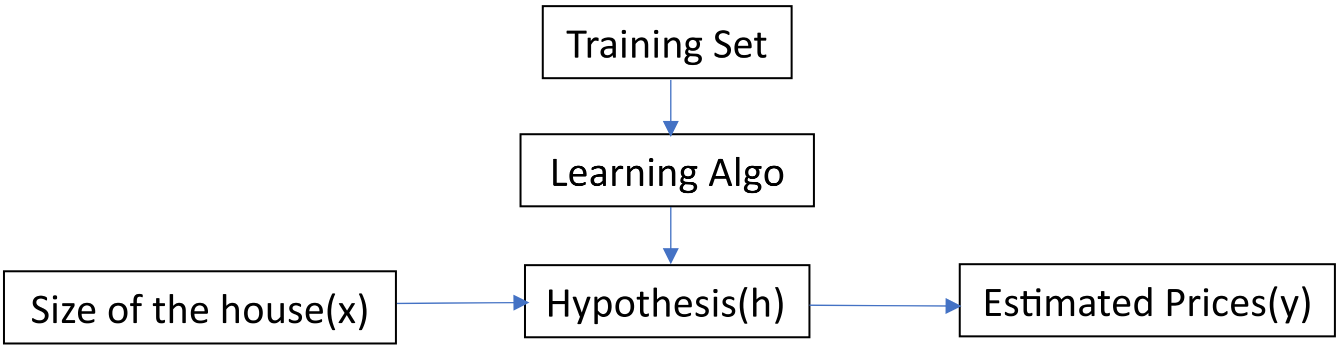
- Supervised Learning

Give the right answer for each example in the data

- Regression Problem

Predict real_valued output

Example of House price Prediction



- Gradient Descent

When you have some function $J(\Theta 1, \Theta 2)$ and you want the minium value of $J(\Theta 1, \Theta 2)$, we use gradient descent.

- Batch Gradient Descent

“Batch”: each step of the gradient descent uses all the training examples .

- Feature Scaling

Make sure features are on a similar scale.

- Mean Normalization

Replace x_i to $x_i - \mu_i$ to make features have approximately 0 mean .

- Normal Equation

Method to solve for Θ analytically.

- Difference Between Gradient Descent and Normal Equation

Gradient Descent	Normal Equation
Need to choose α.	No need to choose α.
Need many iteration.	Don’t need to iterate.
Works well even when n is large.	Works slow when n is large.

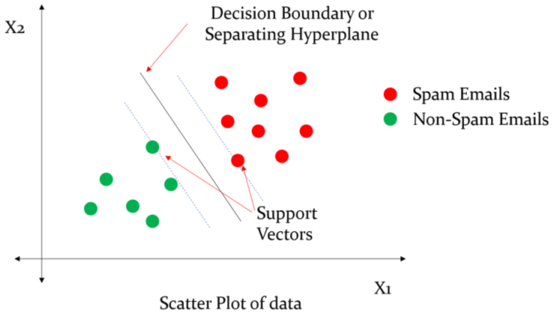
n = 10^6	n = 100
	n = 1000
	n = 10000

- Decision Boundary

The line which separates one class from another class.

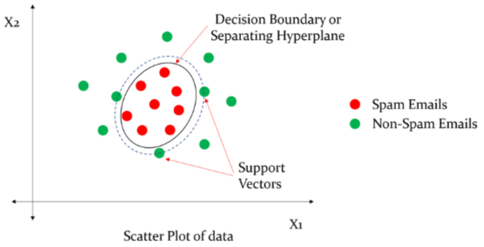
Can be classified into two types

1) Linear Decision boundary
   a) Used in linear models such as linear regression and support vector machine .
   b) These create straight line boundaries.



Linear Decision Boundary

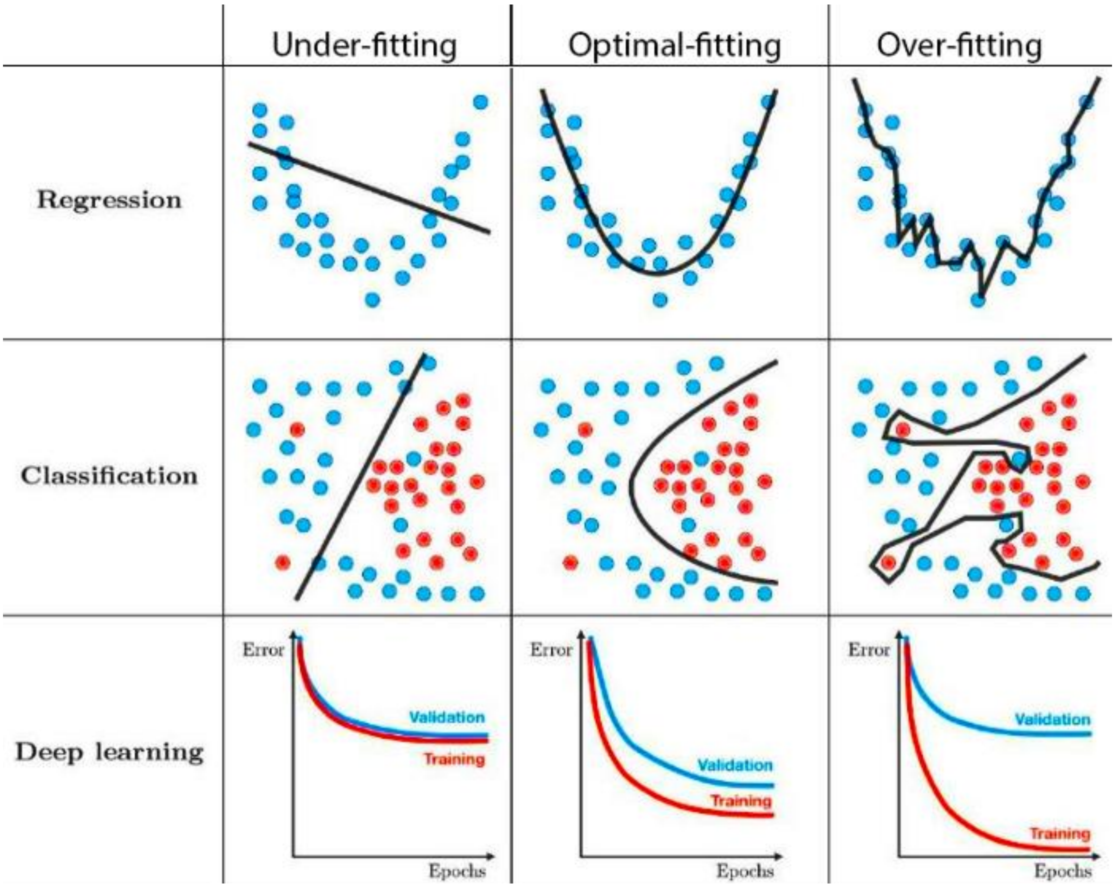
2) Non Linear decision boundary
   a) Used in non-linear models such as random forest , decision trees , neural networks etc.
   b) These create more complex and curved boundaries .



Non- Linear Decision Boundary

- Regularization

This process is basically used to solve the problem of over-fitting while mapping linear regression , classification and deep learning classes.



- What is Over-fitting?

When we have too many features, the learned hypothesis may fit the training set very well , but fail to generalize to new examples (predict precisely on new examples).

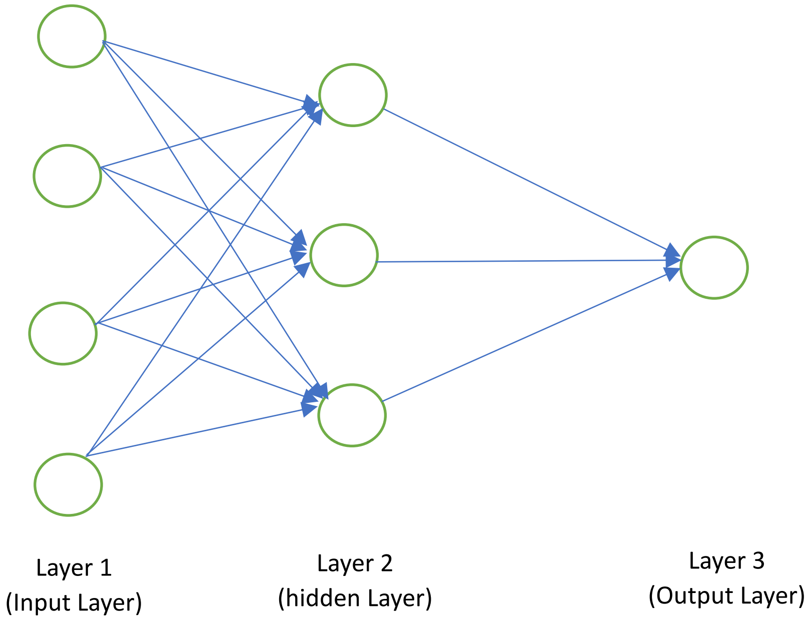
This situation basically occurs when we try to fit a large amount of features to training and testing models.

Options to overcome overfit:-

1) Reduce number of features
    - Manually select which features to keep.
    - Use Model selection algorithm.
2) Regularization
    - Keep all the features, but reduce magnitude / values of parameters.
    - Works well when we have a lot of features, each of which contributes a bit to predicting y.

- Neural Network Representation

Origin: Algorithms that try to mimic the brain.



- Weights and Biases

Weights determine the importance of each input.

Biases are additional parameters that help adjust the output along with weighted sum of inputs.

- Cost Function(J(Ө))

It is a measure of how well a model's prediction match the actual data. It quantifies (to express or measure) the error between the predicted values and the actual values, helping to evaluate the performance of the model.

- The goal is to minimize the cost function to improve the model's accuracy.

* In linear regression the cost function(J(Ө)) helps in finding the best fit line by minimizing the sum of the squared differences between the predicted and actual value.

Problem:

While testing your hypothesis on a new set of houses , you find that it can make unacceptably large errors in its predictions.

Solution:

a) Get more training examples.
b) Try smaller set of features.
c) Try getting additional features
d) Try adding polynomial features (x_1^2, x_2^2, x_1, x_2 etc.)

e) Try decreasing λ.

f) Try increasing λ.

- Precision and Recall

True Positive	False Positive
False Negative	True Negative

Let’s take an example of cancer prediction model !!

Precision

Of all the patients where we predicted $y = 1$, what fraction actually has cancer ?

True positives / predicted positives = True Positives / True pos + False pos

Recall

What fraction did we correctly detect as having cancer ?

True positives / actual positives = True positives / True pos + False neg

F1 score

Used to compare different algos based on the precision and recall

F1 score = 2PR/P+R

- Support Vector Mechanism (SVM)

Support Vector Machines (SVM) are a set of supervised learning methods used for classification, regression, and outliers detection.