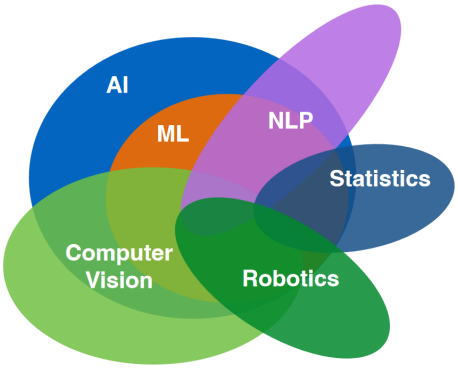
Introduction to Machine Learning Summary

Leonardo Baldo

Contents

1 Introduction

- **Artificial Intelligence:** the science to make things smart
- **Machine Learning:** building machines that can learn
- **Deep Learning:** class of ML algorithms



Artificial Intelligence

1.1 Artificial Intelligence

J.McCarthy 1956 definition: the science and engineering of making intelligent machines.

Modern definition: the ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings.

1.2 Machine Learning

Arthur Lee Samuel 1959 definition: ML is the fields of study that gives computers the ability to learn without being explicitly programmed.

Tom Mitchell 1998 definition: A computer program is said to learn from experience E with respect to some class of tasks T ans performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

1.3 Learning Algorithm

The learning algorithm is an algorithm that is able to learn from data. There are 3 main ingredients:

- **Task**
  - Is described in terms of how the machine learning algorithm should process an example
  - How is an example represented? as a collection of features
- **Performance Measure**
  - How good is the machine learning system? we need to measure its performance
  - The performance measures depends on the task
- **Experience**
  - The experience is provided by the available data

1.4 Bias

Inductive Bias: all the assumptions about the nature of the target function and its selection.

Example of Inductive Bias:

- **Linear regression:** assume that the output or dependent variable is related to independent variable linearly
- **Nearest neighbors:** assume that most of the cases in a small neighborhood in feature space belong to the same class

Algorithmic Bias: It describes systematic and repeatable errors in a system that create unfair outcomes, such as privileging one arbitrary group of users over others.

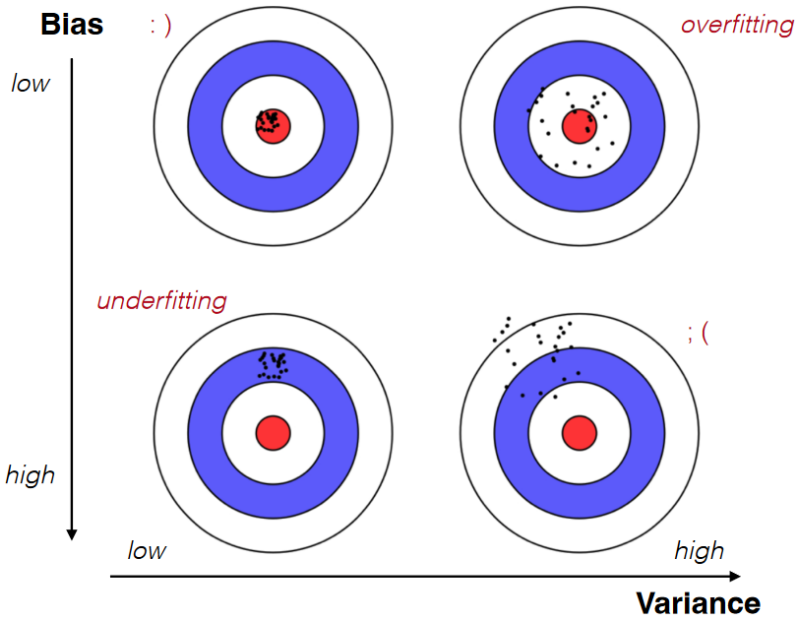
1.4.1 Bias vs Variance

The bias error is produced by weak assumptions in the learning algorithm.

The variance is an error produced by an oversensitivity to small fluctuations in the training set.

Underfitting: high bias can cause an algorithm to miss the relevant relations between features and target outputs.

Overfitting: high variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs.



Bias vs Variance

2 Linear Regression

Linear Regression is defined by the notation:

$$h_\theta x = \theta^T x \qquad \text{(Linear Regression)}$$

If two parameters are used, the function is:

$$h_\theta = \theta_0 + \theta_1^T x \qquad \text{(Linear Regression)}$$

The goal of this approach is to find parameters such that $\theta_{0,1}$, the result of the function $h_\theta(x)$, is close to the value of the target y. One way to quantify this distance is through the **Cost Function**:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)^2 \qquad \text{(Cost Function)}$$

Where:

- m: number of examples
- x: input variable
- y: output variable

The goal of the learning algorithm will then be to minimize the value of this function. To do this, the **Gradient Descent** method can be used.

3 Gradient Descent

Gradient Descent is a technique for obtaining the minimum points of a function.

The basic idea of this method is to always go in the opposite direction from the gradient, so as to arrive at the optimum value, which is at the lowest point, as quickly as possible.

$$\theta_{k+1} = \theta_k - \eta \nabla J(\theta_k) \qquad \text{(Gradient Descent)}$$

The parameter $\eta > 0$ is called the **Learning Rate** and represents the length of the "step" the algorithm takes in the direction of steepest descent. A learning rate that is too high risks that the algorithm will not converge while one that is too low increases the convergence time.

Batch Gradient Descent: To compute the gradient ∇J, the cost is computed over the entire training set, and after each update the gradient is recomputed for the new vector θ_{k+1}.

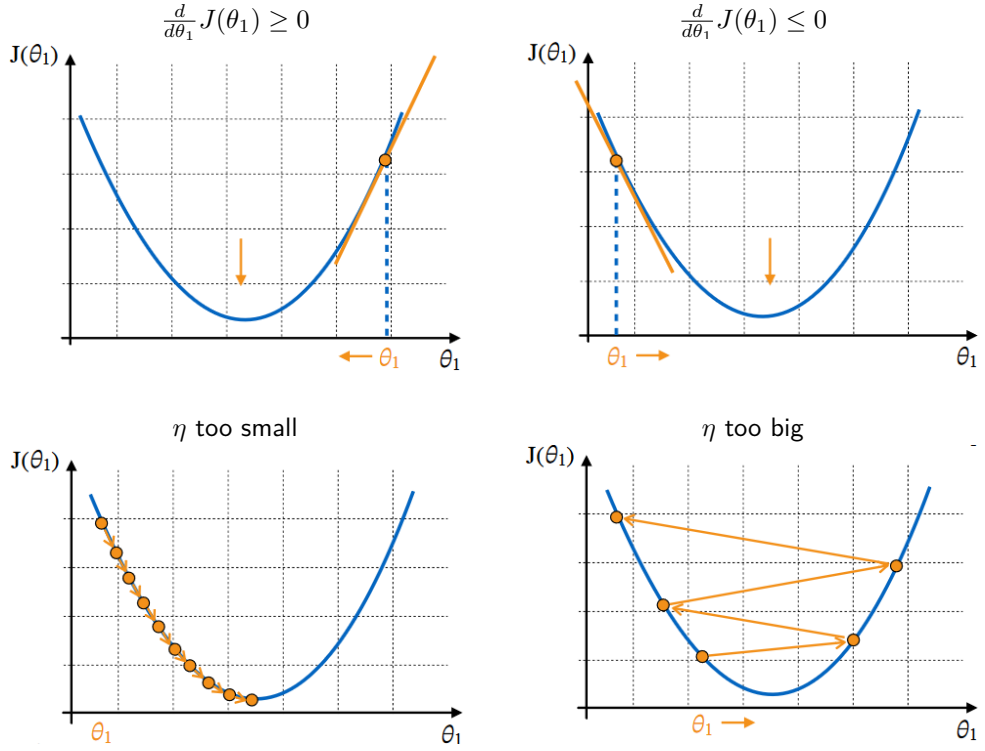
Stochastic Gradient Descent: method in which the parameters are updated for each cycle according to the gradient:

```
for i=1 to m do{
    θ0 = θ0 − η(θ0 + θ1 xi − yi)
    θ1 = θ1 − η(θ0 + θ1 xi − yi) xi
}
```

3.1 Simple Implementation

To proper implementation of gradient descent, it is necessary to update all parameters simultaneously:

```
repeat until convergence{
    tmp0 = θ0 − η ∂/∂θ0 J(θ0, θ1)
    tmp1 = θ1 − η ∂/∂θ1 J(θ0, θ1)
    θ0 = tmp0
    θ1 = tmp1
}
```



4 Classification

Linear Classification: used to determine to which discrete category a specific example belongs. (classification of emails into spam or ham)

Output category:

- **Binary classification:** 2 possible categories
- **Multi-class classification:** n possible categories

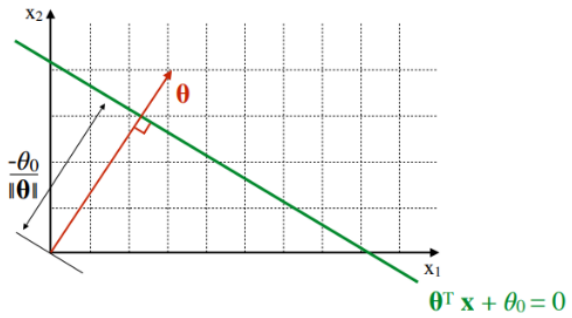
We can use binary classification by applying a threshold:

- $h_\theta(x) = \theta^T x$
- if $h_\theta(x) \geq 0$, then the output is $y = 1$
- if $h_\theta(x) < 0$, then the output is $y = -1$

$$y = sign(h_\theta(x)) \qquad \text{(Classification)}$$

Applying Linear Regression, for classification task, is not a good idea. Instead, it is preferable to use Logistic Regression, which, despite containing regression in its name, is a classification algorithm.

For larger dimensions, we assume a straight line, a shift of a certain term (bias term):



We then estimate a good decision boundary by deciding the direction θ and position θ_0 of the boundary, and we need a parameter selection criterion. For example, the Loss Functions.

4.1 Loss Functions

$$J_{01}(\theta) = \frac{1}{m}\sum_{i=1}^{m}\{0 \text{ if } h_\theta(x_i) = y_i, \text{else } 1\} \quad \text{(Zero/One)}$$

$$J_{abs}(\theta) = \frac{1}{m}\sum_{i=1}^{m}|h_\theta(x_i) - y_i| \quad \text{(Absolute)}$$

$$J_{sqr}(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2 \quad \text{(Squared)}$$

The loss function for the Logistic Regression is:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\text{cost}(h_\theta(x_i, y_i)) \quad \text{(Loss function)}$$

where:

$$\text{cost}(h_\theta(x_i, y_i)) = \begin{cases} -\log(h_\theta(x_i)) & \text{if } y_i = 1 \\ -\log(1 - h_\theta(x_i)) & \text{if } y_i = 0 \end{cases}$$

Because this function is convex, we can use gradient descent. In fact, the method used to calculate the Linear Regression loss would not be correct in a convex function if used with the Logistic Regression function.

5 Logistic Regression

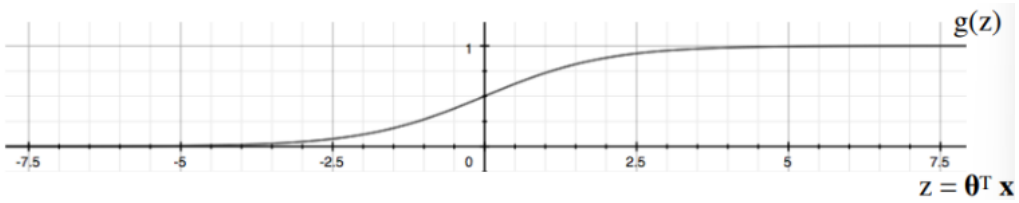
In logistic regression, the result of the prediction function $h(x)$ is between 0 and 1. This allows this value to also be considered a probability, making it ideal for classification tasks.

$$h_\theta(x) = g(\theta^T x) \qquad \text{(Prediction Function)}$$

Where:

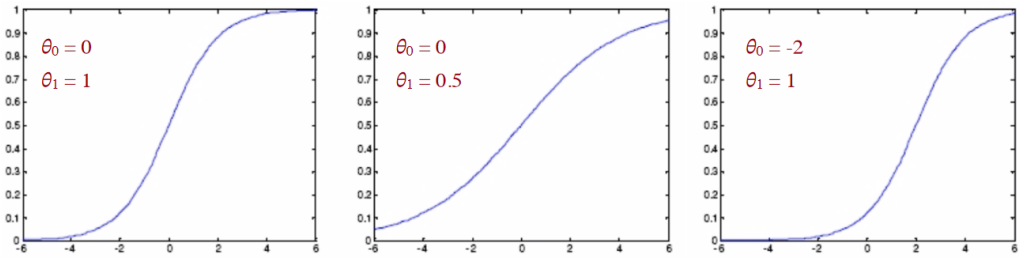
$$g(z) = \frac{1}{1 + e^{-z}}$$

The graphical representation is a sigmoid:



This function can "flatten" or "widen", depending on the values within θ, going to favor one class over another thereby succeeding in more accurately approximating a function that can divide the plane between classes.

$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$



5.1 Probabilistic Interpretation

Logistic regression benefits from the marginalization property, so:

$$h_\theta(x) = P(y = 1|x; \theta)$$

where $P(y = 1|x; \theta)$ denotes the probability that $y = 1$ given an input x, then:

$$P(y = 1|x; \theta) + P(y = 0|x; \theta) = 1$$

6 Regularization

6.1 Linear

Since the goal of the machine learning model is to minimize loss, a simple way to regularize the parameters is to multiply them by a number λ so as to force the algorithm to assign smaller and therefore less influential values, making the resulting parameters more accurately represent the totality of the examples.

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)^2 + \lambda\sum_{j=1}^{n}\theta_j^2 \qquad \text{(Loss Function)}$$

The parameter update step then becomes:

repeat until convergence{
$\theta_0 = \theta_0 - \eta\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)x_{0i}$
$\theta_j = \theta_j - \eta\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)x_{ji} + \frac{\lambda}{m}\theta_j$
}

6.2 Logistic

One can apply the same principle to logistic regression by adding a regularization term to the loss function, which becomes:

$$J(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m} y_i \log(h_\theta(x_i)) + (1 - y_i)\log(1 - h_\theta(x_i))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \qquad \text{(Loss Function)}$$

The parameter update step then becomes:

repeat until convergence{
$\theta_0 = \theta_0 - \eta\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)x_{0i}$
$\theta_j = \theta_j - \eta\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i) - y_i)x_{ji} + \frac{\lambda}{m}\theta_j$
}

7 Evaluation, Learning Curves

8 Artificial Neural Networks