

# Introduction to Machine Learning Summary

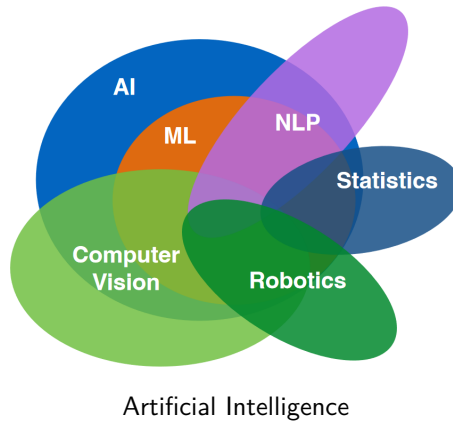
Leonardo Baldo

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Artificial Intelligence . . . . .	3
1.2	Machine Learning . . . . .	3
1.3	Learning Algorithm . . . . .	3
1.4	Bias . . . . .	4
1.4.1	Bias vs Variance . . . . .	4
<b>2</b>	<b>Linear Regression, Gradient Descent</b>	<b>5</b>
2.1	Gradient Descent . . . . .	5
2.1.1	Simple Implementation . . . . .	5
<b>3</b>	<b>Linear Classification, Logistic Regression</b>	<b>6</b>
<b>4</b>	<b>Regularization</b>	<b>7</b>
<b>5</b>	<b>Evaluation, Learning Curves</b>	<b>8</b>
<b>6</b>	<b>Artificial Neural Networks</b>	<b>9</b>

# 1 Introduction

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



## 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  $T$  and 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.

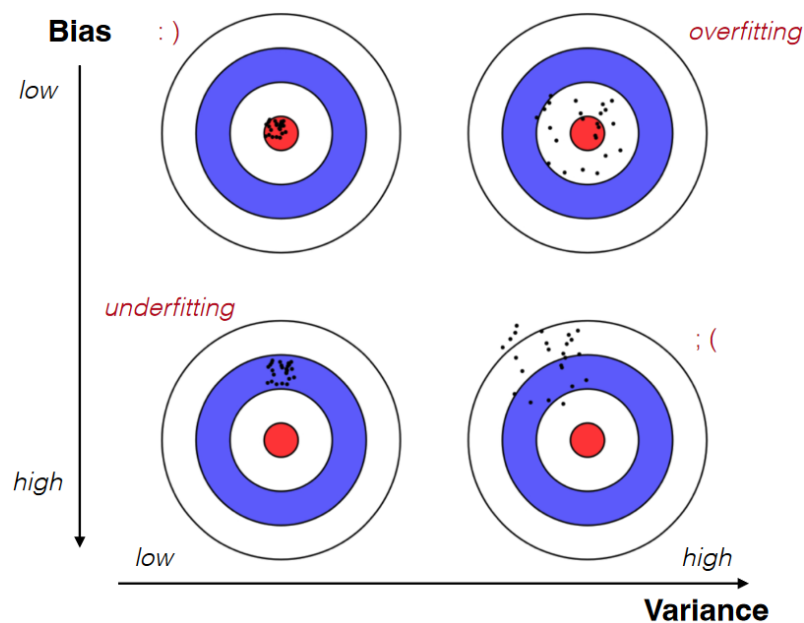
### 1.4.1 Bias vs Variance

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

The variance is an error produced by an over-sensitivity 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, Gradient Descent

**Linear Regression** is defined by the notation:

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

If two parameters are used, the function is:

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

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**:

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

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.

### 2.1 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.

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

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  $\nabla J$ , the cost is computed over the entire training set, and after each update the gradient is recomputed for the new vector  $\theta_{k+1}$ .

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

```
for i=1 to m do{
     $\theta_0 = \theta_0 - \eta(\theta_0 + \theta_1 x_i - y_i)$ 
     $\theta_1 = \theta_1 - \eta(\theta_0 + \theta_1 x_i - y_i)x_i$ 
}
```

#### 2.1.1 Simple Implementation

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

```
repeat until convergence{
     $\text{tmp}_0 = \theta_0 - \eta \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$ 
     $\text{tmp}_1 = \theta_1 - \eta \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$ 
     $\theta_0 = \text{tmp}_0$ 
     $\theta_1 = \text{tmp}_1$ 
}
```

### **3 Linear Classification, Logistic Regression**

## 4 Regularization

## 5 Evaluation, Learning Curves



## **6 Artificial Neural Networks**