Artificial Intelligence

Arland Barrera

November 7, 2025

Contents

1 Statistics 6
1.1 Basic Concepts 6
1.1.1 Mean 6
1.1.2 Weighted Average 7
1.1.3 Median 7
1.1.4 Mode 7
1.1.5 Range 7
1.1.6 Variance 8
1.1.7 Standard Deviation 8
1.1.8 Gaussian Distribution 9
1.2 Metrics 10
1.2.1 Accuracy 10
1.2.2 Recall 10
1.2.3 Precision 10
1.2.4 F1-Score 11

2 Mathematics 12
2.1 Geometry 12
2.1.1 Euclidean Distance 12
2.2 Activation Functions 12
2.2.1 Rectified Linear Unit ReLU 12
2.2.2 Sigmoid 14
2.2.3 Tanh 15
2.2.4 Softmax 15

3 Algorithms 16
3.1 Supervised Learning 16
3.1.1 Linear Regression 16
3.1.2 Logistic Regression 16
3.1.3 K-Nearest Neighbor 16
3.1.4 Decision Tree 16
3.1.5 Support Vector Machine 16
3.1.6 Naive Bayes 16
3.2 Unsupervised Learning 17
3.3 Optimization 17

4 Neural Networks 18
4.1 Perceptron 18
4.1.1 Simple Perceptron 18
4.1.2 Multi Layer Perceptron MLP 18
4.2 Dense Neural Network DNN 18

4.3 Convolutional Neural Network CNN 19
4.4 Recurrent Neural Network RNN 19

List of Plots

1.1 Gaussian distribution . . . 10
2.1 ReLU function . . . 13
2.2 Sigmoid function . . . 14
2.3 Tanh function . . . 15

List of Equations

1.1 Sample mean 6
1.2 Population Mean 6
1.3 Weighted mean 7
1.4 Odd median 7
1.5 Even median 7
1.6 Range 8
1.7 Sample variance 8
1.8 Population variance 8
1.9 Sample standard deviation 9
1.10 Population standard deviation 9
1.11 Probability density function 9
1.12 Accuracy 10
1.13 Recall 10
1.14 Precision 11
1.15 F1-Score 11
2.1 General Euclidean distance 12
2.2 ReLU function 13
2.3 Sigmoid function 14
2.4 Tanh function 15
2.5 Tanh function 15
3.1 Naive Bayes theorem 17

Statistics

Statistics is a field of mathematics that pertains to data analysis. Statistical methods and equations can be applied to a data set in order to analyze and interpret results, explain variations in the data, or predict future dara.

Two important concepts in Statistics are **population** and **sample**. Population is all the data available, the larger group. Sample is a subset of the population, a fraction or portion of the total data available.

1.1 Basic Concepts

1.1.1 Mean

The mean, also known as the **arithmetic mean** or **average**, is obtained by dividing the sum of observed values, data values, by the number of observations. This gives a good idea as to what the points as closest to. Occasionally a value that differs greatly can be seen, the mean would incorporate the occasional outlying data. This means the average is susceptible to outlying values. The idea is to find the center of the data set. It is the most common type of average.

The **sample mean** is represented by an x with an over line. The numbers of samples is represented by the letter n.

$$\boxed{\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i} \tag{1.1}$$

Equation 1.1: Sample mean

The **population mean** is represented by the greek letter mu, μ. The total number of values is represented by the letter N.

$$\boxed{\mu = \frac{1}{N}\sum_{i=1}^{N} x_i} \tag{1.2}$$

Equation 1.2: Population Mean

Both equations 1.1 and 1.2 are the same, the only change are the symbols employed. They can be used when the error associated with each measurement is the same or unknown.

1.1.2 Weighted Average

Also known as **weighted mean** or **weighted arithmetic mean**. The weighted average gives more importance to certain data points by multiplying each value a specific weight before summing them up. The result is then divided by the sum of the weights to find the average. This method is used when data have different levels of importance or significance.

$$\boxed{\bar{X}_w = \frac{\sum\limits_{i=1}^{n} w_i x_i}{\sum\limits_{i=1}^{n} w_i}} \tag{1.3}$$

Equation 1.3: Weighted mean

1.1.3 Median

The median is the middle value of a set of data containing an odd number of values, or the average of the two middle values of a set of data with an even nummber of values. The median is specially helpful when separating data into two equal sized distributions. It is useful if there is an interest in knowing the range of values the system could be be operating in. Half the values should be above and half the values should be below, so there is an idea of where the middle operating point is. Because of this the median less sentitive to outlier data.

For **odd numbers** the median is the data points n plus 1 divided by 2.

$$\boxed{\text{Median} = \frac{n+1}{2}} \tag{1.4}$$

Equation 1.4: Odd median

For **even numbers** the median is the average of the data points n divided by two, and n divided by two plus one.

$$\boxed{\text{Median} = \frac{1}{2}\left(\frac{n}{2} + \left(\frac{n}{2} + 1\right)\right)} \tag{1.5}$$

Equation 1.5: Even median

1.1.4 Mode

The mode of a set of data is the value which occurs most frequently.

1.1.5 Range

The range is the difference between larget and smallest values. This a way to find the dispersion of the data set. However, if there is outliers in the data the range is of little use.

$$\boxed{\text{Range} = x_{\max} - x_{\min}} \tag{1.6}$$

Equation 1.6: Range

1.1.6 Variance

Measure of dispersion that indicates how far each individual data point is from the mean. This is the spread of each individual data point.

The deviation of each data point from the mean is, the data value minus the mean:

$$x_i - \bar{X}$$

Adding up each deviation and then divide by the number of data points, averages the deviation of every point from the mean. This is the average variance or spread of the data from the mean.

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{X})$$

This has a problem. The values to the left of the mean result in negative values, and the ones to the right are positive. To solve this, the deviation of each data point is squared. This results in the **variance**,

The **sample variance** is represented by the letter s squared, s^2. This is divided by the sample data minus 1, $n - 1$.

$$\boxed{s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2} \tag{1.7}$$

Equation 1.7: Sample variance

The **population variance** is represented by the greek letter sigma to the power of two, σ^2. This is divided by the total population data, N.

$$\boxed{\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{1.8}$$

Equation 1.8: Population variance

1.1.7 Standard Deviation

The variance squares the spread of each data point, this raises a problem. The measure of units of the data set is lost because it has been squared. To solve this problem the square root of the variance can found, that way the measure of the original data is more close. This is known as standard deviation.

The standard deviation gives an idea of how close the entire set of data is to the average value. Data sets with a small standard deviation have tightly grouped, precise data. Data sets with large standard deviations have data spread out over a wide range of values.

The **sample standard deviation** is represented by the letter s.

$$\boxed{s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{X})^2}} \tag{1.9}$$

Equation 1.9: Sample standard deviation

The **population standard deviation** is represented by the greek letter σ.

$$\boxed{\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}} \tag{1.10}$$

Equation 1.10: Population standard deviation

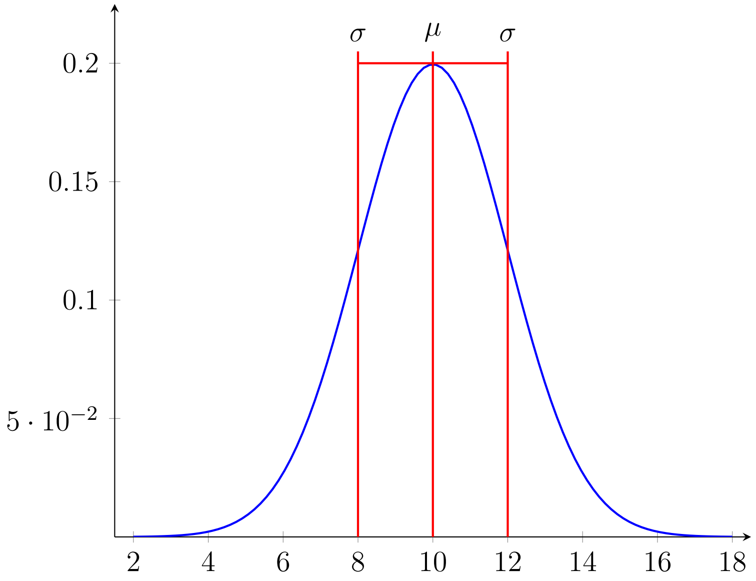
1.1.8 Gaussian Distribution

Gaussian distribution, also known as **normal distribution**, is represedted by the following **Probability Density Function**:

$$\boxed{\mathrm{PDF}_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\dfrac{(x-\mu)^2}{2\sigma^2}}} \tag{1.11}$$

Equation 1.11: Probability density function

where μ is the mean and σ is the standard deviation of a very large data set. The Gaussian distribution is a bell-shaped curve, symmetric about the mean value. Probability density functions represent the spread of a data set. An example is shown below.



Plot 1.1: Gaussian distribution

In this specific example, $\mu = 10$ and $\sigma = 2$.

Normal distribution is completely defined by the mean (μ) ans the standard deviation (σ). The line never touches the x-axis. The y-axis acts as the probability. Total area under curve is 1, because the sum of all probabilities is 1.

1.2 Metrics

1.2.1 Accuracy

Proportion of correct predictions and total predictions.

$$\boxed{\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}} \tag{1.12}$$

Equation 1.12: Accuracy

1.2.2 Recall

Proportion of true positives and total positives.

$$\boxed{\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}} \tag{1.13}$$

Equation 1.13: Recall

1.2.3 Precision

Proportion of true positives and positive predictions.

$$\boxed{\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}} \tag{1.14}$$

Equation 1.14: Precision

1.2.4 F1-Score

Mean of precision and recall.

$$\boxed{\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}} \tag{1.15}$$

Equation 1.15: F1-Score

Mathematics

2.1 Geometry

2.1.1 Euclidean Distance

Euclidean distance is the straight-line distance between two points in Euclidean space, calculated using the Pythagorean theorem. It's the most intuitive way to measure distance, found by taking the square root of the sum of the squared differences between the corresponding coordinates of the points.

For **two points**, $A(x_1, y_1)$ and $B(x_2, y_2)$, the formula is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

For **three points**, $A(x_1, y_1, z_1)$ and $B(x_2, y_2, z_2)$, the formula is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

The general formula extends by adding the squared difference for each additional coordinate.

$$\boxed{d = \sqrt{\sum_{i=1}^{n} (A_i - B_i)^2}} \tag{2.1}$$

Equation 2.1: General Euclidean distance

where n is the number of dimensions.

2.2 Activation Functions

2.2.1 Rectified Linear Unit ReLU

Rectified Linear Unit (ReLU) is a popular activation function used in neural networks, especially in deep learning models. It has become the default choice in many architectures due to it's simplicity and efficiency. The ReLU function is a piecewise linear function that outputs the input directly if it is positive, otherwise it outputs zero.

It allows positive values to pass through unchanged while setting all negative values to zero. This helps the neural network maintain the necessary complexity to learn patterns while avoiding some of the pitfalls associated with other activation function, like the **vanishing gradient problem**.

The ReLU function can be described mathematically as follows:

$$\boxed{f(x) = \max(0, x)} \tag{2.2}$$

Equation 2.2: ReLU function

Where:

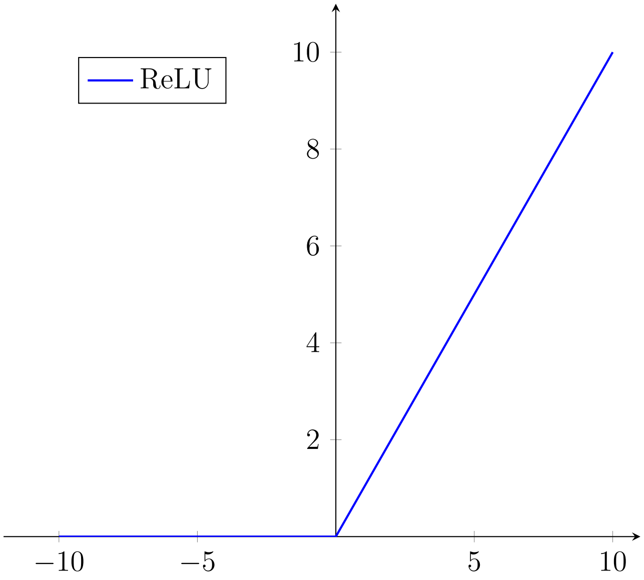
- x is the input to the neuron.
- The function returns x if x is greater than 0.
- if x is less than or equal to 0, the function returns 0.
- The output range is $[0, \infty)$.

The function can also be written as:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases}$$

This simplicity is what makes ReLU so effective in training deep neural networks, as it helps to maintain non-linearty without complicated transformations, allowing models to learn more efficiently.

The plot of the function is shown bellow.



Plot 2.1: ReLU function

This activation function is the most widely used for the following:

- **Simplicity**: is computationally efficient as it involves only a thresholding operation. This makes it easy to implement and compute, which is important when training deep neural networks with millions of parameters.
- **Non-linearty**: although it seems like a piecewise linear function, it is still a non-linear function. This allows the model to learn more complex data patterns and model intricate relashionships between features.
- **Sparse Activation**: it's ability to output zero for negative inputs introduces sparsity in the model, meaning that only a fraction of neurons activate at any given time. This can lead to more efficient and faster computation.
- **Gradient Computation**: it offers computational advantages in terms of backpropagation, as it's derivative is simple, either 0 or 1. This helps to avoid the vanishing gradient problem.

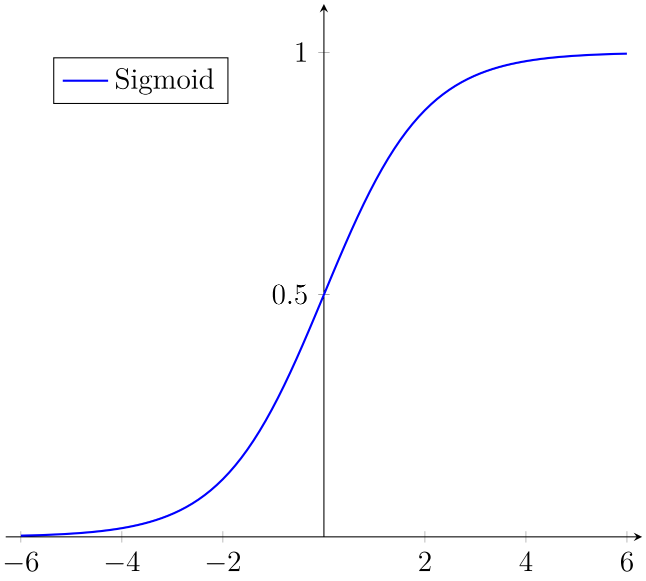
2.2.2 Sigmoid

This function takes any real-value input and maps it to an output between 0 and 1, creating a charasterisic S-shaped curve. It is also known as **logistic curve**.

$$\boxed{f(x) = \frac{1}{1+e^{-x}}} \tag{2.3}$$

Equation 2.3: Sigmoid function

The S-shaped curve of the function is shown below.



Plot 2.2: Sigmoid function

Charasterisics:

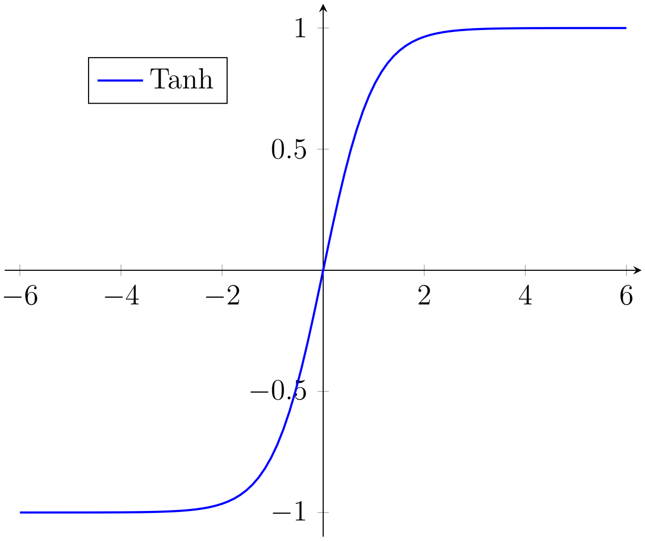
- Domain: $(-\infty, \infty)$
- Range: $(0, 1)$

- Intersection: $f(0) = 0.5$

2.2.3 Tanh

$$\boxed{f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}} \tag{2.4}$$

Equation 2.4: Tanh function



Plot 2.3: Tanh function

Charasterisics:

- Domain: $(-\infty, \infty)$
- Range: $(-1, 1)$

2.2.4 Softmax

$$\boxed{f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}} \tag{2.5}$$

Equation 2.5: Tanh function

Algorithms

Artificial intelligence algorithms can be classified by their learning approach, task type, domain, and others.

Learning approach

- Supervised learning.
- Unsupervised learning.
- Reinforcement learning.
- Semi-supervised learning.

Task type

- Classification: used to sort data into categories.
  - Binary classification: separates data into two classes.
  - Multi-class classification: assigns an item to three or more classes.
- Regression: used to predict a continuous value.

Domain

- Natural Language Procesing (NLP).
- Computer vision.

3.1 Supervised Learning

3.1.1 Linear Regression

3.1.2 Logistic Regression

3.1.3 K-Nearest Neighbor

3.1.4 Decision Tree

3.1.5 Support Vector Machine

3.1.6 Naive Bayes

The Naive Bayes algorithm in machine learning is a probabilistic classification algorithm based on Bayes' theorem that makes a "*naive*" assumption that all features are independent of each other given the class. It calculates the probability of a class given a set of features, using the formula shown below.

$$\boxed{P(\text{class}|\text{data}) = \frac{P(\text{data}|\text{class})P(\text{class})}{P(\text{data})}} \tag{3.1}$$

Equation 3.1: Naive Bayes theorem

The independence assumption simplifies the calculation.

3.2 Unsupervised Learning

3.3 Optimization

Neural Networks

4.1 Perceptron

4.1.1 Simple Perceptron

4.1.2 Multi Layer Perceptron MLP

4.2 Dense Neural Network DNN

A dense neural network (DNN), also known as a **fully connected neural network** (FCN), is one of the fundamental architectures in deep learning. This architecture is well-suited for tasks involving structured data, like tabular data, as well as unstructured data when combined with other layers in hybrid models.

In this type of artificial neural network each neuron in one layer is connected to every neuron in the following layer. This complete connectivity between neuron forms the core idea behind DNNs, allowing them to learn complex relationships in the data. Dense layers are commonly used in various deep learning models, and in practice, the excel when working with numerical or structured data.

Architecture of Dense Neural Networks

1. Input Layer

The input layer is the first layer, receiving raw data features.

2. Hidden Layers

Hidden layers in DNNs are fully connected layers between the input and output layers. The network's depth, or the number of hidden layers, is what makes it “deep”. Each layer transforms the data by applying weights and biases, adjusted during training.

Weights are matrices that connect neurons between layers, determining the strength and direction of the connections.

Biases are values added to the neuron's input to control the activation.

3. Output Layer

The output layer produces the final predictions. For classification tasks, this layer typically uses a softmax activation to produce probabilities, while for regression, a linear activation function may be more appropiate.

4. Activation Functions

Dense layers use activation functions to introduce non-linearities into the model, enabling it to capture complex patterns. Common functions include: ReLu, Sigmoid and Tanh.

4.3 Convolutional Neural Network CNN

4.4 Recurrent Neural Network RNN