

ML Notes without the BS

vivian sedov

December 19, 2022

Contents

1	Types of Learning	2
1.1	Supervised Learning	2
2	Introduction To NN	4
2.1	How does it work ?	4
2.1.1	Step by step guide on how NN works	4

1 Types of Learning

- Supervised Learning
 - Algo is trained on labeled data
 - meaning that the data has been labeled with the correct output
 - *Goal:* is to make predictions about new unseen data based on patterns learned from the training data.
 - Examples
 - * Linear Regression
 - * Logistic Regression
 - * Support vector Machines
 - * Classification problems : Binary or Multi Class
- Unsupervised Learning
 - The algo is not given any labeled data and must find the patterns and relationships on its own.
 - *Goal:* Is to discover structure in the data and to identify relationships based on that structure.
 - Examples:
 - * Clustering
 - * Dimensionality reduction [Really cool concept]
- Reinforcement Learning
 - The learning algo is not given any labeled data and must find patterns and relationships in the data on its own.
 - *Goal :* is to discover structure, but based on its own ability to generalise and understand the current scope of what it can learn from.
 - Examples:
 - * Self driving cars
 - * Those small little hoovers that move around on their own. Question your self how do they map everything.

1.1 Supervised Learning

Due to this module purely focusing on this, i will pick certain info that i deem viable

Features

- Components of samples are features
- Each feature can be discrete or continuous $[0, \infty)$

Protocols used in supervised Learning There are two core protocols that are used within supervised learning: Batch and Online — Just google this, the explanation on the slides are bad.

Batch Learning In batch learning, there are two core steps, and stages that you have to work with

- Training: Exploration stage - You analyse the data, and the training set - and find some viable explanation for why it works
- Exploitation stage: Pretty much saying does this hypothesis work with respect to what the training data had presented.

Definition 1.1 – Induction Induction is a form of learning that involves making generalizations based on specific examples or observations. In induction, the goal is to learn a general rule or model that can be applied to new, unseen data. This is the approach used in many supervised learning algorithms, where the algorithm is trained on a labeled dataset and then makes predictions about new, unseen data based on the patterns learned from the training data.

has a model

Definition 1.2 – Transduction Transduction is a form of learning that involves making predictions about a specific instance based on the available information. In transduction, the goal is to make a prediction about a specific case rather than learning a general rule that can be applied to new data. This is the approach used in many unsupervised learning algorithms, where the algorithm is given a dataset but is not told what the correct output should be. Instead, the algorithm must find patterns and relationships in the data on its own and use that information to make a prediction about a specific instance.

No model

The main difference between induction and transduction is that induction involves learning a general rule or model that can be applied to new data, while transduction involves making a prediction about a specific instance based on the available information.

2 Introduction To NN

Definition 2.1 – Nearest Neighbor Algorithm The nearest neighbor algorithm is a method for classification that assigns a new data point to the class of the nearest training data point. This algorithm is based on the idea that similar data points are likely to belong to the same class.

Within binary classification problems, the label space is often taken to be 0 or 1.

2.1 How does it work ?

Here is a mathematical example of how the nearest neighbor algorithm works: Suppose we have a training set of data points with known classes, represented as a matrix $X \in \mathbb{R}^{n \times m}$, where each row corresponds to a data point and each column corresponds to a feature. The class labels for the data points are stored in a vector $y \in \mathbb{R}^n$.

Now, suppose we have a new data point, represented as a row vector $x \in \mathbb{R}^m$, that we want to classify. To classify this point using the nearest neighbor algorithm, we need to find the data point in the training set that is closest to x . We can measure the distance between x and each data point in the training set using a distance metric, such as the Euclidean distance:

$$d(x, X_i) = \sqrt{\sum_{j=1}^m (x_j - X_{i,j})^2}$$

where X_i is the i -th row of the matrix X , and x_j and $X_{i,j}$ are the j -th elements of the vectors x and X_i , respectively.

Once we have computed the distances between x and all the data points in the training set, we can find the nearest neighbor by selecting the data point with the smallest distance:

$$\hat{y} = y_i, \text{ where } i = \operatorname{argmin}_{j=1}^n d(x, X_j)$$

In this equation, \hat{y} is the predicted class for the new data point x , and y_i is the class of the nearest neighbor.

2.1.1 Step by step guide on how NN works

Suppose we have the following training set of data points, where each row corresponds to a data point and each column corresponds to a feature:

$$X = \begin{bmatrix} 1 & 1 & 2 & 2 & 3 & 3 & 4 & 4 \end{bmatrix}$$

The class labels for the data points are stored in the following vector:

$$y = \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix}$$

Now, suppose we have a new data point that we want to classify, represented as the following row vector:

$$x = \begin{bmatrix} 1.5 & 1.5 \end{bmatrix}$$

To classify this point using the nearest neighbor algorithm, we need to find the data point in the training set that is closest to x . We can measure the distance between x and each data point in the training set using the Euclidean distance, as shown in the previous equation.

For example, the distance between x and the first data point in the training set is:

$$d(x, X_1) = \sqrt{(1.5 - 1)^2 + (1.5 - 1)^2} = \sqrt{0.5^2 + 0.5^2} = 0.7071067811865475$$

We can compute the distances between x and all the other data points in the training set in a similar way. Once we have computed the distances, we can find the nearest neighbor by selecting the data point with the smallest distance:

$$\hat{y} = y_i, \text{ where } i = \operatorname{argmin}_{j=1}^n d(x, X_j)$$

In this case, the nearest neighbor is the first data point, since it has the smallest distance to x . Therefore, the predicted class for the new data point x is $\hat{y} = y_1 = 0$.

This is transduction : No model was formed