

Intro to Machine Learning

May 1, 2022

Today, Machine Learning (ML) is a word that is in everyones mouth. Companies are looking for skills in ML despite not knowing what that actually means. Well, the good ones do but that does not apply to the other 80%. There is so much wuwu and blabla about ML but the techniques and algorithms used there are quite simple. The hard part is the finetunig and backtesting of the models but running and understanding any particular algorithm in that field is quite straight forward.

While there are many intros in ML and tons of books, exercises etc etc, I found myself still struggling to get started as the intros were way too mathematical so that understanding the theory takes a couple of days and in the end one is so far away from the practical side that one really don't want to continue or the guides were so practical that while being able to program it (it's usually super easy), I had no idea what the algorithm actually did. So, that is why I decided to really understand ML from a practical and slightly theoretical point of view. What this guide is about is a practical approach to ML but with a heuristic understanding of how the algorithm actually works. This include some maths but I decided to explain that in super simple term rather than proving theorems all the time. That said, I will go into the maths but won't use math terms to explain it.

This guide contains an introduction of the overall idea of ML and a list of algorithms that are usually assigned with ML. Each algorithm will be explained heuristically, but still with enough details to understand what to change if the output is not what one wants and a small exercise or application will be provided with some Python code. All code and explanation are written by me but can be surely found in similar forms in the internet.

1 The Idea

The ML algorithms can be divided into three different categories: supervised, unsupervised and reinforced learning. While there is a lot of literature on each of them let me give a simple explanation. Supervised learning means working with labeled data, that means we name the data. For example, we have pictures and we label them as dog, cat or whatever. This will then be easy for the computer to distinguish between pictures. A usual task would be to learn a certain thing about the data, for example what a cat looks like. Unsupervised learning means working with unlabeled data, that means no naming. The computer just gets pictures without knowing what a dog, cat or whatever is. A usual task would be to find a grouping of this data that makes sense. Reinforcement learning means working with rules and produce data yourself by simulation. For example, a computer knows the rules of a game and it plays it many times against itself. A usual task would be for the computer learn how to do a certain task like walking or playing a game.

1.1 Supervised Learning

As already mentioned supervised learning is helping the machine to learn things by naming data. This is one of the most popular directions as the algorithms cover classification and regression. The aim is to find an estimator for a function $f()$ in

$$Y = f(X) + \varepsilon,$$

where Y, X are our data (random variables) and ε is some random error. The X variable can be multidimensional (so it can contain more than one number) and is called the feature vector, while the Y variable is... . Y also determines also if it is a regression or a classification. If Y is continuous one speaks about regression, if Y is deterministic it's classification (the labels cat, dog etc. will then be transformed to 0,1,...).

A supervised ML algorithm is structured in the following way:

- Clean your data
- Divide your data into training and test sets
- Train the model with the training set

- Test the model with the test set
- Adjust some parameters

The whole process of testing your model on test data is called validation. If one mixes and matches test and training sets it is called cross validation. Cross validation is a key concept in ML, so it would be a good idea to look into how it works. Here are some solid sources: ...

1.2 Unsupervised Learning

1.3 Reinforcement Learning