Classification -> Neural Network -> Deep Learning

Classification

- A broad/popular type of problem in computer science
- Neural Network/Deep learning is just a method or a strategy to solve classification problem.
- Definition: Given some **background knowledge**, you/computer will assign a label for a particular input data to a certain **group/category.**

e.g. the camera capture many images, you can sort these images by labeling them as "people", "building", "jungle", "animal", "food", …

Background Knowledge: a model that is trained before you do the task.

Different from **Clustering** problem, which separate the input data into multiple groups/categories without any background knowledge.

Classification – training on other (previous) data, supervised learning
Clustering – no training or training only on current data, unsupervised learning, "relative information in current data"

(1) Given a lot of images, let computer find out those images that have faces.
classification
(2) …… , let computer separate the images into brighter ones and darker ones.
Clustering

(3) Computer find a particular pattern from a single image.
classification

(4) Given a single image, group all the pixels into 4 parts according to their color distribution. Initially, get a rough estimation about the group assignment of this image and perform training on these groups.
Still Clustering/unsupervised

Linear Regression

Input: x
Output: y

y = f(x) if for the function f(), the exponential part of x is at most 1, then we can call this function as linear function.

y = 5x
y = 3x + 7
y = 2 (y = 2 + 0*x)

We can extend dimension of X into multi-dimensional data, e.g. x is 3D: (x1, x2, x3)

$X = (x_1, x_2, x_3)$, we use the lower case x with a number index as the value for the corresponding dimension.
y as the output data, we usually prefer it to be 1D for maths convenience.

Question: write an example of linear function between y and X

$y = 5x_1 + 2x_2 – 7x_3 + 10$
$y = 9\ x_2$

y – label (e.g. “face” = 1, “food” = 2, “building” = 3)
$X = (x_1, x_2, x_3)$ - is the input image

The goal of training is to learn the values for the coefficients ($\theta_0, \theta_1, \theta_2, \theta_3$):

$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

For the whole classification problem domain, most of all the methods, including neural network, deep learning, …, is just to find an optimal values for the coefficients ($\theta_0, \theta_1, \theta_2, \theta_3, \ldots$) through training, an optimal estimation of these coefficients would produce an accurate prediction on y; otherwise, the result maybe not satisfactory.

Definition of Linear Regression

Input Data: X $(x_1, x_2, \ldots)$ - it can be multidimensional

Output Data: y - by default, the output is 1D for mathematic computation reason. Sometimes the output data is also called "label"

Equation: between y and X is linear combination.

Goal: to learn an optimal model, more specifically, to learn those optimal parameters, $\theta_0, \theta_1, \theta_2, \ldots$ How many such parameters $\{\theta_j\}$ equals to the number of X dimension + 1. "j" indicates the dimension index.

Optimal:

Given a set of data pairs $\{<X^{(i)}, y^{(i)}>\} = <X^{(1)}, y^{(1)}>, <X^{(2)}, y^{(2)}>, \ldots, <X^{(m)}, y^{(m)}>$, coming in pairs. We will learn the parameters $\{\theta_0, \theta_1, \ldots, \theta_n\}$ that minimize the following equation:

One sample <X, y>: $\theta_0 X_0 + \theta_1 X_1 + \ldots + \theta_n X_n \xrightarrow{\text{predict}} y'$, we hope that y' is as to y as possible.

$$\underset{\{\theta_0, \theta_1, \ldots, \theta_n\}}{\text{minimize}}\left(\frac{1}{m} \sum_{i=1,\ldots,m} ((\theta_0 X_0^{(i)} + \theta_1 X_1^{(i)} + \ldots + \theta_n X_n^{(i)}) - y^{(i)})^2\right)$$

The reason of use square instead of "absolute operator" is to allows conveniently compute the derivative, i.e. df()/dx $(5x^2 + 7x + 5)' = 10x + 7$

$(12x + 9)' = 12$

m – the number of training data

n – the number of input data X's dimension

i – indicates the input training data (sample) index, placing as superscript

j – indicates the dimension index, placing as subscript

For example, a student has taken many courses, we can consider X (CPS 501, CPS 580, …), the output should be the grade for each course, which is multidimensional. But for optimization purpose of linear regression, we have to convert it into 1D. For instance, instead of considering all the student's scores, we can just use GPA as the output.