Introduction to Basic Supervised and Unsupervised Learning I

IRS ML Section*

Hwa Chong Institution

1 Supervised Learning

1.1 Definition

Supervised learning is the machine learning task of learning a function that maps an input to an output based on sample input-output pairs. It infers a function from labelled training data consisting of a set of training examples.

$$\text{training set} = \{x_i, y_i \mid i = 1, 2, ..., n\}, \tag{1}$$

where $x_i \in R^k$ is the **input data**; $y_i \in R$ is the **ground truth**.

1.2 Regression

A regression model is used to predict the value of a **continuous** outcome variable (y) based on the value of one or multiple input variables (x). It is essentially the *best-fit line* (but it can have a very high dimension).

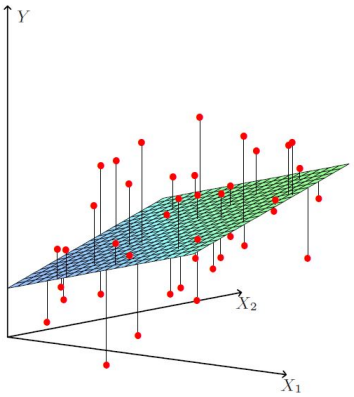

Fig. 1. Regression. In this specific case, there are 2 input variables, so regression produces a 'best-fit plane'.

* Written by Chenghao and Robert

Linear Regression

To understand how regression works, let's first try out the most basic form of regression — linear regression. As the name suggests, *linear* regression is a model that assumes a *linear* relationship between the n input variables $(x_1, x_2, \cdots, x_n)$ and the single output variable (y).

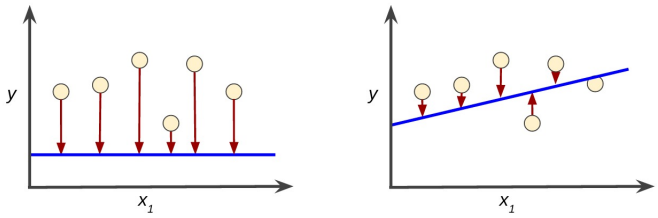

Fig. 2. Linear regression. In this particular case, the line was initialised to be a horizontal line (left), and as the machine learns, the line is gradually adjusted such that the loss is minimised (right).

How does one perform linear regression?

1. Randomly initialise a line.
2. Calculate the loss of that line.
3. Minimise the loss by tuning the gradient and the y-intercept.
4. Repeat steps 2 and 3 until the best-fit line is obtained.

Random Initialisation

The first step is random initialisation. We randomly generate the gradient and y-intercept for the line. People usually will just set both to be 0 or 1. Mathematically, this can be described by the linear function h

$$h = mx + c, \tag{2}$$

where m is the gradient and c is the y-intercept.

Loss Function

Next, a loss function l is used to assess how accurate or inaccurate the randomly generated line is. We have many types of loss functions to choose from: the cross-entropy loss, the mean-squared error, and the hinge loss — just to name a few. However, these are not the focus of today's lesson; today, we will use a rather basic type of loss function.

As a simple illustration, let us define the loss function to be the average deviation of the predicted values and the actual values. Mathematically, this can be expressed as

$$l = \frac{1}{n} \sum_{k=1}^{n} |h_k - y_k| \tag{3}$$

This loss function is the basis on which the gradient and y-intercept are tuned. The values of m and c are determined by minimising the loss function.

Note: From here on out, we will be talking about a general case. We will leave the rest of the linear regression part for your first homework.

Non-linearity

Before we delve into how parameter tuning works, we will first learn about non-linearity. Non-linearity can be introduced to the model using activation functions.

An activation function is just like any function $f(x)$, that gives an output from the input. Some common examples of activation function are *Rectified Linear Unit (ReLU), sigmoid, tanh.* As you can see, all of these functions are non-linear. The reason for the introduction of non-linearity is that, most of the time, the input and output do not have a linear relationship. (Imagine a image classification task).

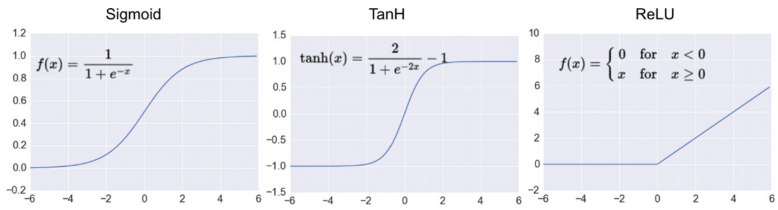

Fig. 3. Activation functions. Sigmoid (left), tanh (centre) and ReLU (right) are three commonly used activation functions.

When non-linearity is introduced, the loss function will have a rather irregular shape, as shown in Fig. 4. The example shown has multiple local minima, but note that the ultimate goal is to eventually reach the global minimum.

Tuning parameters and constant

Recall from the previous part that the end goal of tuning the parameters is to minimise the loss or — as is often the case in practice — reduce it to an acceptable value. Consider a parameter m and the loss function l w.r.t. m. To minimize the loss, we gradually tune m towards m_{min} by updating m as

$$m = m - \eta \frac{\partial l}{\partial m}, \tag{4}$$

where η is called the learning rate. The learning rate essentially determines how big you want the step to be as we gradually nudge the value of m along the loss function towards the minimum.

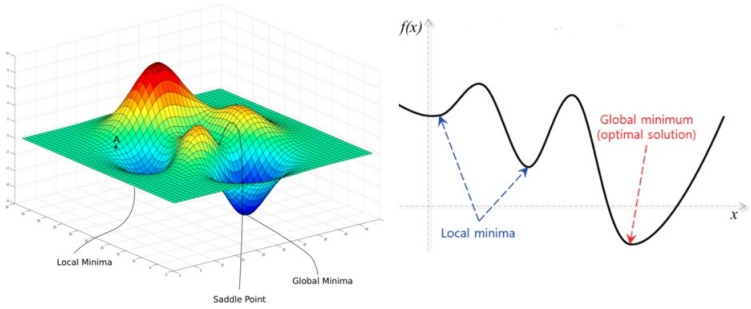

Fig. 4. Loss function. This is a sample loss function with non-linearity is introduced.

Thinking Question: What will happen if the learning rate is too big, or too small?

1.3 First Project: Linear Regression

Build a linear regression model on your own. Things to take note:

- Use MSE as the loss function
- Plot the final result to better visualize