ML3 – Logistic Regression

Introduction

If instead of a “hard” binary classifier where the output is only one of two states ([+1, -1] or [1, 0]) we now want a classifier that outputs the probability that the input belongs to a class, then we can use Logistic Regression (LR). An example where LR is useful is in medical diagnostics where you want to know the probability that the patient has a certain disease. Another example is in weather predictions wherein you want to know the percent chance of rain.

In Logistic Regression we introduce the ***logit*** function which is the logarithm of the odds ratio of probability p,

$$logit(p)=\log\frac{p}{1-p} \quad . \tag{1}$$

For example, p can be $p(y=1|\boldsymbol{x})$ or the probability that the input belongs to class 1 given its feature vector **x.** The inverse of the logit function is the logistic function

$$\phi(a)=\frac{1}{1+\exp(-\beta a)}=p \quad . \tag{2}$$

Because the shape of this curve is nearly an “S” it is also called the sigmoid function (Figure 1). The coefficient β controls the steepness of the sigmoid function. The sigmoid function resembles a step function except that the transition from one state to the other is smooth and continuous. It is also known as a “squashing” function because it squashes any input between -∞ to +∞ to a number between 0 and 1.

We can use our artificial neuron model to find p using logistic regression. To train a Logistic Regression neuron we once again compute the network input

$$a=w_0+x_1 w_1+x_2 w_2+x_3 w_{3.}\ldots=\sum x_i w_i=\boldsymbol{x}^T \boldsymbol{w} \quad .$$

Note that $x_0 = 1$. But this time, we pass it to the logistic (sigmoid) function activation function,

$$z=\phi(a) \quad .$$

The weight change rule for the jth weight will once again be

$$\begin{aligned} \Delta w_j &= \eta(d^i - z^i)x_j^i \\ w_j &= w_j + \Delta w_j \end{aligned} \tag{3}$$

where d^i is the desired output of the ith data, z^i is the network output, and x_j^i is the jth feature of the ith data. This time, d, instead of being -1 will be 0 and d =1 will remain the same. The network output is then the probability that the input belongs to a certain class or not.

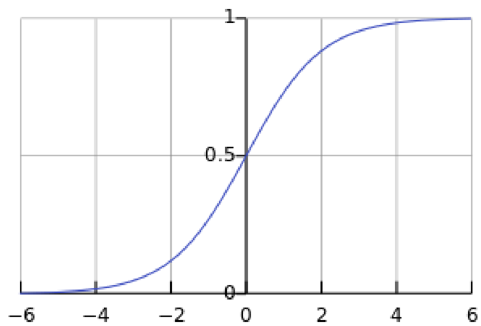

Figure 1: Logistic or Sigmoid function. Image from wikipedia.

Procedure

1. Select a fruit that changes color as it ripens. For example, banana or mango. Gather several images of ripe and unripe fruit, including those in between. For each image get the average red, green and blue color of the fruit (instead of a 0 to 255 digital range, normalize it to 0 to 1.0 for each channel) and label it 1 for ripe, 0 for unripe.
2. Apply logistic regression to train an artificial neuron to give the degree of ripeness of that the fruit. Plot the Test with images not yet seen by the neuron and comment if the output agrees with the visual appearance.

Reference

1. Raschka, Sebastian. Python machine learning. Packt Publishing Ltd, 2015.

