ML By Examples

Mister Nobody

January 4, 2021

Contents

1 Linear Regression

Take the following linear equation:

$$y = w \times x + b$$

In ML the naming convention is y: label, x: feature, w: weight, b: bias.

We can use Linear Regression **Model** and to find w and b that minimize the error (find the best equation).

We need pairs (x_i, y_i) to **train** a mathematical equation (find best w and b).

The most common type of Machine Learning is called Supervised Learning, where we need both Features (x) and Labels (y) so we can train the Model. This examples are on Supervised Learning.

1.1 Best Parameters

To find the best weight and bias we minimize the Mean Square Error:

$$MSE(w, b) = \frac{1}{m} \sum_{i=0}^{m} (y_d - y_p)^2$$

m is the number of samples, d stands for data point, p for predicted.

In ML the error function is called the Cost Function. And Loss function is just the element-wise difference $(y_d - y_p)$

Let's write this on a small piece of code:

```
def cost_function(X, Y, w, b):
    dPoints = len(X)
    err = 0.0
    for i in range(dPoints):
        err += (Y[i] - (w*X[i] + b))**2
    return err / dPoints
```

We can also write that down using vectorization:

```
import numpy as np

def cost_function(X, Y, w, b):
    dPoints = len(X)
    err = 0.0
    for i in range(dPoints):
        err = np.sum(np.square(Y - np.dot(w,X)+ b))
    return err / dPoints
```

We want the MSE(w,b) to be as small as possible. We can compute the gradient by deriving the MSE.

$$dMSE = \frac{dMSE}{db} + \frac{dMSE}{dw}$$

So the total variation is adding up variations on b when w is constant, and the opposite.

We think of MSE as $MSE = A(w,b)^2$ and use the Chain Rule (very frequent on ML) $dMSE = 2A \cdot (dA/db + dA/dw)$. The gradient is:

$$\frac{dMSE}{dw} = \sum 2(y_i - (mx_i + b)) \cdot -x_i \tag{1}$$

$$\frac{dMSE}{db} = \sum 2(y_i - (mx_i + b)) \cdot -1 \tag{2}$$

We can use it to minimize the Cost by updating w and b.

$$w = w - \frac{dMSE}{dw} \cdot \alpha b \qquad = b - \frac{dMSE}{db} \cdot \alpha \tag{3}$$

There is no graphical justification why we update w and b like that, but think of it like this: we are moving w against the gradient multiplied by a constant (alpha) which is normally called learning rate.

The minus sign is because the gradient always points away from the minimum and we want towards it (in one dimension there are only 2 directions).

Because the $y_d - y_p$ is squared, MSE is a parabola for b and w then it makes sense: the farther away we are from the minimum the larger the gradient, and the more we want to update w and b

This is not that simple for other methods and many variables. But gives the general idea.

1.2 Summary

- Model: equation to fit the data,
- Cost: Evaluates the error,
- Training. Compute dw, db to update parameters (w, b)

The first 2 steps are called forward propagation, the third is backward propagation.

2 Binary Regression

Linear regression fits linear data, Binary fits binary data. Binary data is 0 or 1 on a plot. Then we can't use linear regression.

The **Model** used here is not a straight line but a sigmoid function:

$$\sigma(y) = \frac{1}{1 + e^{-y}} y = w \times x + b$$

The Cost function (computes error) is different too.