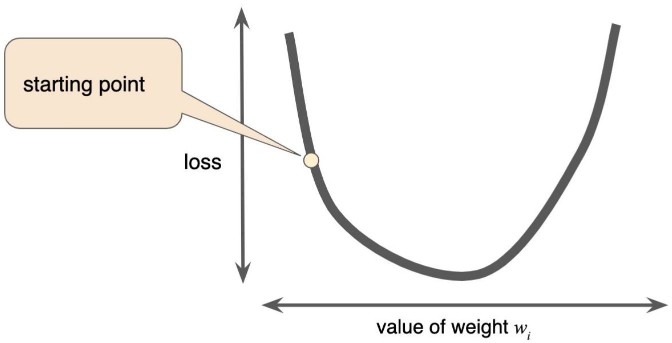
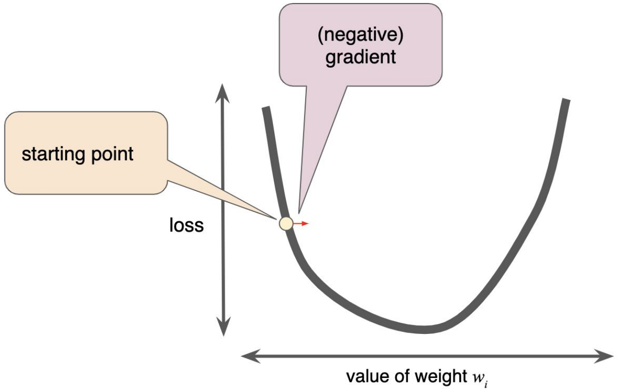
Gradient Descent

Anand Yati

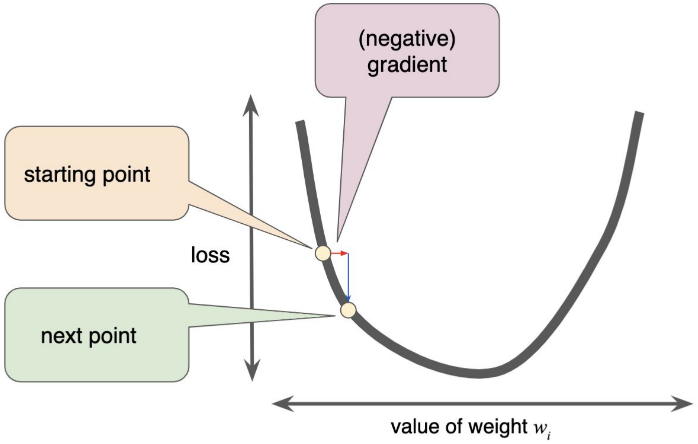
Introduction to gradient descent algorithm



Start: For a random value of weights, find the starting point on loss curve



Next: Calculate the gradient of the loss curve at the starting point. It is equal to the derivative (slope) of the curve. Thus figure out direction in which loss decreases.



Next: To determine the next point along the loss function curve, the gradient descent algorithm adds some fraction of the gradient's magnitude to the starting point as shown in the following figure.

Repeat the process of calculating gradient and updating weights to get the next point till we reach minima.

Learning rate, the deciding factor for step size

Mathematics behind gradient descent

Loss Function

The loss is the error in our predicted value of **m** and **c**. Our goal is to minimize this error to obtain the most accurate value of **m** and **c**.
We will use the Mean Squared Error function to calculate the loss. There three steps in this function:

1. Find the difference between the actual y and predicted y value(y = m + c), for a given x.
2. Square this difference.
3. Find the mean of the squares for every value in X.

$$E = \frac{1}{n} \sum_{i=0}^{n} (y_i - \bar{y}_i)^2$$

Mean Squared Error Equation

Here y_i is the actual value and $\bar{y}_i$ is the predicted value. Lets substitute th value of $\bar{y}_i$:

$$E = \frac{1}{n} \sum_{i=0}^{n} (y_i - (mx_i + c))^2$$

Substituting the value of $\bar{y}_i$

Calculate the partial derivative of the loss function with respect to m, and plug in the current values of x, y, m and c in it to obtain the derivative value **D**.

$$D_m = \frac{1}{n} \sum_{i=0}^{n} 2(y_i - (mx_i + c))(-x_i)$$
$$D_m = \frac{-2}{n} \sum_{i=0}^{n} x_i(y_i - \bar{y}_i)$$

Derivative with respect to **m**

D_m is the value of the partial derivative with respect to **m**. Similarly lets fin the partial derivative with respect to **c**, Dc :

$$D_c = \frac{-2}{n} \sum_{i=0}^{n} (y_i - \bar{y}_i)$$

Derivative with respect to **c**

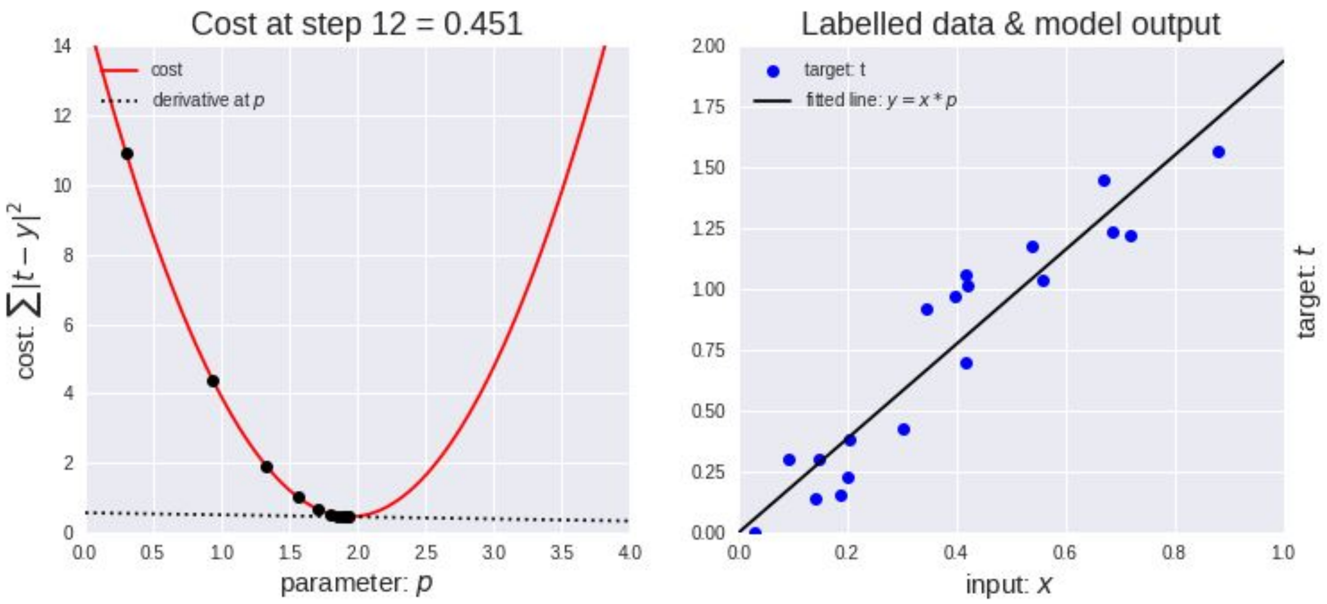
Now we update the current value of **m** and **c** using the following equation:

$$m = m - L \times D_m$$

$$c = c - L \times D_c$$

We repeat this process until our loss function is a very small value or ideally 0 (which means 0 error or 100% accuracy). The value of **m** and **c** that we are left with now will be the optimum values.

Gradient descent in action



The loss function for this model is a convex/bowl/U shape with 1 minima