1 Gradient Descent

$$dC(\omega) = \lim_{\varepsilon \to 0} \frac{C(\omega + \varepsilon) - C(\omega)}{\varepsilon} \tag{1}$$

$$\tag{2}$$

1.1 Derivitive Of One Parameter Function

Within the *Twice* example we described a model with one parameter - w

The formula had a form like this:

$$f(x) = x \cdot w \tag{3}$$

Function C which takes one parameter w is defined as:

$$C(w) = \frac{1}{n} \sum_{i=1}^{n} (x_i \cdot w - y_i)^2 \tag{4}$$

Let's compute the derivitive C' of our function:

$$\begin{aligned}
C'(w) &= (C)' && (5)\\
&= \left(\frac{1}{n} \sum_{i=1}^{n} (x_i \cdot w - y_i)^2 \right)' = && (6)\\
&= \left(\frac{1}{n} \sum_{i=1}^{n} (x_i \cdot w - y_i)^2 \right)' = && (7)\\
&= \frac{1}{n} \left(\sum_{i=1}^{n} (x_i \cdot w - y_i)^2 \right)' = && (8)\\
&= \frac{1}{n} \sum_{i=1}^{n} \left((x_i \cdot w - y_i)^2 \right)' = && (9)\\
&= \frac{1}{n} \sum_{i=1}^{n} \left(2 \cdot (x_i \cdot w - y_i)(x_i \cdot w - y_i)' \right) = && (10)\\
&= \frac{1}{n} \sum_{i=1}^{n} \left(2 \cdot (x_i \cdot w - y_i) \cdot x_i \right) && (11)
\end{aligned}$$

The final form of our derivitive:

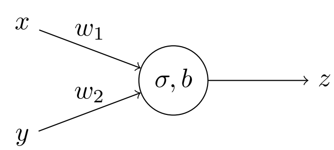
$$C'(w) = \frac{1}{n} \sum_{i=1}^{n} \left(2 \cdot (x_i \cdot w - y_i) \cdot x_i \right) \tag{12}$$

1.2 One Neuron Model With 2 Inputs

One neuron model is defined as:

$$z = \sigma(x \cdot w_1 + y \cdot w_2 + b) \tag{13}$$

x_1 ... input parameter
x_2 ... input parameter
w_1 ... weight paramter
w_2 ... weight paramter
b ... bias parameter
σ ... sigmoid activation function



1.2.1 Cost

Let's recall the Sigmoid activation function

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

$$\sigma(x)' = \sigma(x) \cdot (1 - \sigma(x)) \tag{15}$$

Let's define the cost function C for our model

$$a_i = \sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) \tag{16}$$

$$C(x) = \frac{1}{n}\sum_{i=1}^{n}(a_i - y_i)^2 \tag{17}$$

Let's compute the derivitive C' for our function

We have to modify TWO parameters w, b

For this we will use PARTIAL DERIVITIVES this means that first we compute a derivitive in respect to w_1, w_2 and then we compute another derivitive in respect to b

1. Partial Derivitive in respect to w_1

$$\begin{aligned}
a_i &= \sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) = && (18)\\
\partial_{w_1} a_i &= \partial_{w_1}(\sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) = && (19)\\
&= a_i(1-a_i)\partial_{w_1}(x_i \cdot w_1 + y_i \cdot w_2 + b) = && (20)\\
\partial_{w_1} a_i &= a_i(1-a_i) \cdot x_i && (21)\\
& && (22)\\
\partial_{w_1} C &= \partial_{w_1}\left(\frac{1}{n}\sum_{i=1}^{n}(a_i - z_i)^2\right) = && (23)\\
&= \frac{1}{n}\sum_{i=1}^{n}\partial_{w_1}\left((a_i - z_i)^2\right) = && (24)\\
&= \frac{1}{n}\sum_{i=1}^{n} 2 \cdot (a_i - z_i)\partial_{w_1}(a_i - z_i) = && (25)\\
&= \frac{1}{n}\sum_{i=1}^{n} 2 \cdot (a_i - z_i)\partial_{w_1} a_i = && (26)\\
\partial_{w_1} C &= \frac{1}{n}\sum_{i=1}^{n} 2 \cdot (a_i - z_i) \cdot a_i(1-a_i) \cdot x_i && (27)\\
& && (28)
\end{aligned}$$

2. Partial Derivitive in respect to w_2

$$\begin{aligned}
a_i &= \sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) = && (29)\\
\partial_{w_2} a_i &= \partial_{w_2}(\sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) = && (30)\\
&= a_i(1 - a_i)\partial_{w_2}(x_i \cdot w_1 + y_i \cdot w_2 + b) = && (31)\\
\partial_{w_2} a_i &= a_i(1 - a_i) \cdot y_i && (32)\\
& && (33)\\
\partial_{w_2} C &= \partial_{w_2}\left(\frac{1}{n}\sum_{i=1}^{n}(a_i - z_i)^2\right) = && (34)\\
&= \frac{1}{n}\sum_{i=1}^{n}\partial_{w_2}\left((a_i - z_i)^2\right) = && (35)\\
&= \frac{1}{n}\sum_{i=1}^{n} 2 \cdot (a_i - z_i)\partial_{w_2}(a_i - z_i) = && (36)\\
&= \frac{1}{n}\sum_{i=1}^{n} 2 \cdot (a_i - z_i)\partial_{w_2} a_i = && (37)\\
\partial_{w_2} C &= \frac{1}{n}\sum_{i=1}^{n} 2 \cdot (a_i - z_i) \cdot a_i(1 - a_i) \cdot y_i && (38)\\
& && (39)
\end{aligned}$$

3. Partial Derivitive in respect to b

$$
\begin{aligned}
a_i &= \sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) && (40)\\
\partial_b a_i &= \partial_b(\sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) = && (41)\\
&= a_i(1 - a_i)\partial_b(x_i \cdot w_1 + y_i \cdot w_2 + b) = && (42)\\
&= a_i(1 - a_i) \cdot 1 = && (43)\\
\partial_b a_i &= a_i(1 - a_i) && (44)\\
& && (45)\\
\partial_b C &= \partial_b \left(\frac{1}{n} \sum_{i=1}^{n} (a_i - z_i)^2 \right) = && (46)\\
&= \frac{1}{n} \sum_{i=1}^{n} \partial_b \left((a_i - z_i)^2 \right) = && (47)\\
&= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \partial_b \left(a_i - z_i \right) = && (48)\\
&= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \partial_b a_i = && (49)\\
\partial_b C &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \cdot a_i(1 - a_i) && (50)\\
& && (51)
\end{aligned}
$$

To summarize the partial derivitives are:

$$
\begin{aligned}
a_i &= \sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) && (52)\\
\partial_{w_1} C &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \cdot a_i(1 - a_i) \cdot x_i && (53)\\
\partial_{w_2} C &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \cdot a_i(1 - a_i) \cdot y_i && (54)\\
\partial_b C &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \cdot a_i(1 - a_i) && (55)\\
& && (56)
\end{aligned}
$$

1.3 Execution Time Comparison

Let's compare computation time differance between **Finite Difference** and **Gradient Descent**
My machine is Lenovo Legion Slim 5:

- All computations are run on the CPU
- CPU: AMD Rayzen 7 7840HS (16) 5.137Ghz

The test:

- Neural network will try to learn the proper configuration for simulating NAND gate
- Comparison of training the model using the *Finite Difference* method and *Gradient Descent*
- 8.000.000 iterations(epochs) of training will be run (overkill I know)

RESULTS:

$$\textbf{Finite Difference} : \approx 1.556 \ \textit{seconds}$$

$$\textbf{Gradient Descent} : \approx 0.473 \ \textit{seconds}$$

Let's not forget that NAND gate simulation is preatty much trivial and both methods of computation would have approximatly the same time when not doing as much iterations(epochs) of training