1 Gradient Descent

$$\begin{align}
dC(\omega) = \lim_{\varepsilon \to 0} \frac{C(\omega + \varepsilon) - C(\omega)}{\varepsilon} \tag{1}\\
\tag{2}
\end{align}$$

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:

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

The final form of our derivitive:

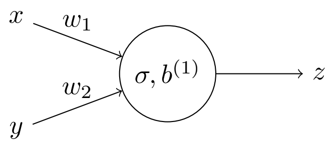
$$C'(w) = \frac{1}{n} \sum_{i=1}^{n} \left(2 \cdot (x_i \cdot w - y_i) \cdot x_i \right) \tag{12}$$

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

$$\begin{align} \sigma(x) &= \frac{1}{1+e^{-x}} \tag{14} \\ \sigma(x)' &= \sigma(x) \cdot (1 - \sigma(x)) \tag{15} \end{align}$$

Let's define the cost function C for our model

$$\begin{align} a_i &= \sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) \tag{16} \\ & \boxed{C(x) = \frac{1}{n}\sum_{i=1}^{n}(a_i - z_i)^2} \tag{17} \end{align}$$

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

$$a_i = \sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) = \tag{18}$$

$$\partial_{w_1} a_i = \partial_{w_1}(\sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) = \tag{19}$$

$$= a_i(1 - a_i)\partial_{w_1}(x_i \cdot w_1 + y_i \cdot w_2 + b) = \tag{20}$$

$$\partial_{w_1} a_i = a_i(1 - a_i) \cdot x_i \tag{21}$$

$$\tag{22}$$

$$\partial_{w_1} C = \partial_{w_1} \left(\frac{1}{n} \sum_{i=1}^{n} (a_i - z_i)^2 \right) = \tag{23}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \partial_{w_1} \left((a_i - z_i)^2 \right) = \tag{24}$$

$$= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i)\partial_{w_1} (a_i - z_i) = \tag{25}$$

$$= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i)\partial_{w_1} a_i = \tag{26}$$

$$\partial_{w_1} C = \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \cdot a_i(1 - a_i) \cdot x_i \tag{27}$$

$$\tag{28}$$

2. Partial Derivitive in respect to w_2

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

3. Partial Derivitive in respect to b

$$\begin{align}
a_i &= \sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) \tag{40}\\
\partial_b a_i &= \partial_b(\sigma(x_i \cdot w_1 + y_i \cdot w_2 + b) = \tag{41}\\
&= a_i(1 - a_i)\partial_b(x_i \cdot w_1 + y_i \cdot w_2 + b) = \tag{42}\\
&= a_i(1 - a_i) \cdot 1 = \tag{43}\\
\partial_b a_i &= a_i(1 - a_i) \tag{44}\\
& \tag{45}
\end{align}$$

$$\begin{align}
\partial_b C &= \partial_b \left(\frac{1}{n} \sum_{i=1}^{n} (a_i - z_i)^2 \right) = \tag{46}\\
&= \frac{1}{n} \sum_{i=1}^{n} \partial_b \left((a_i - z_i)^2 \right) = \tag{47}\\
&= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \partial_b \left(a_i - z_i \right) = \tag{48}\\
&= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \partial_b a_i = \tag{49}\\
\partial_b C &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i - z_i) \cdot a_i(1 - a_i) \tag{50}\\
& \tag{51}
\end{align}$$

To summarize the partial derivitives are:

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

1.2.2 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)

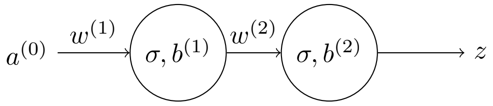
RESULTS:

Finite Difference : $\approx 1.556\ seconds$

Gradient Descent : $\approx 0.473\ seconds$

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

1.3 Two Neuron Model And 1 Input



Let's define the mathematical model

$$\begin{align}
a^{(1)} &= \sigma(x \cdot w^{(1)} + b^{(1)}) \tag{57}\\
a^{(2)} &= \sigma(a^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{58}\\
& \tag{59}\\
a^{(i)} &\text{ ... activation of the i-th layer} \tag{60}\\
& \tag{61}\\
z &= a^{(2)} \tag{62}\\
z &= \sigma(a^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{63}\\
& \tag{64}
\end{align}$$

1.4 Cost

1.4.1 Cost Of The 2nd Layer

Let's define the cost function $C^{(2)}$ for the second layer of our model:

$$\begin{align}
a_i^{(1)} &= \sigma(x_i \cdot w^{(1)} + b^{(1)}) \tag{65}\\
a_i^{(2)} &= \sigma(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{66}\\
& \tag{67}\\
a_i^{(l)} &\text{ ... activation of the i-th sample of the l-th layer} \tag{68}\\
& \tag{69}
\end{align}$$

$$\boxed{C^{(2)}(x) = \frac{1}{n}\sum_{i=1}^{n}(a_i^{(2)} - z_i)^2} \tag{70}$$

Before we start computing derivitives let's think about them

- Firstly we have to compute partial derivitives for $w^{(2)}$ nad $b^{(2)}$ which should not be hard becuase we already converd similar calculations in the past
- When we try to compute partial derivitives inner $w^{(1)}$ and $b^{(1)}$ we notice that the parameters $w^{(1)}$ and $b^{(1)}$ are deeply nested inside $a^{(1)}$ which can present a challange when trying to compute partial derivates
- Introduce a separate cost functions for each indivudual layer
  - Alexey Kutepov

- For each of these *specialized* layers you compute the cost only for the variables that are nearly accessable
- Let's treat the *previous activation* as a variable of the cost function and let's differentiate it
- The result of the differentiation of the cost function is actually an *differance(error)* that we can use for the computation of the *Difference(error)* of the inner layer
- We continue to compute these *differances(errors* for all layers all the way back to the *input layer* => This is where the idea of **back-propagation** comes to play

Let's compute the derivitive $C^{(2)\prime}$ of our function

Partial derivitive in regards to $w^{(2)}$

$$
\begin{align}
a_i^{(2)} &= \sigma(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{71}\\
\partial_{w^{(2)}} a_i^{(2)} &= \partial_{w^{(2)}} \sigma(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{72}\\
&= a_i^{(1)}(1 - a_i^{(1)}) \cdot \partial_{w^{(2)}}(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{73}\\
\partial_{w^{(2)}} a_i^{(2)} &= a_i^{(1)}(1 - a_i^{(1)}) \cdot a_i^{(1)} \tag{74}\\
& \tag{75}\\
& \tag{76}
\end{align}
$$

$$
\begin{align}
\partial_{w^{(2)}} C^{(2)} &= \partial_{w^{(2)}} \left(\frac{1}{n} \sum_{i=1}^{n} (a_i^{(2)} - z_i)^2 \right) \tag{77}\\
&= \frac{1}{n} \sum_{i=1}^{n} \partial_{w^{(2)}} (a_i^{(2)} - z_i)^2) \tag{78}\\
&= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot \partial_{w^{(2)}} (a_i^{(2)} - z_i) \tag{79}\\
&= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot \partial_{w^{(2)}} a_i^{(2)} \tag{80}\\
\partial_{w^{(2)}} C^{(2)} &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot a_i^{(1)}(1 - a_i^{(1)}) \cdot a_i^{(1)} \tag{81}\\
& \tag{82}
\end{align}
$$

Partial derivitive in regards to $b^{(2)}$

$$\begin{aligned}
a_i^{(2)} &= \sigma(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) && (83)\\
\partial_{b^{(2)}} a_i^{(2)} &= \partial_{b^{(2)}} \sigma(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) && (84)\\
&= a_i^{(1)}(1 - a_i^{(1)}) \cdot \partial_{b^{(2)}}(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) && (85)\\
\partial_{b^{(2)}} a_i^{(2)} &= a_i^{(1)}(1 - a_i^{(1)}) && (86)\\
& && (87)\\
& && (88)\\
\partial_{b^{(2)}} C^{(2)} &= \partial_{b^{(2)}} \left(\frac{1}{n} \sum_{i=1}^{n} (a_i^{(2)} - z_i)^2 \right) && (89)\\
&= \frac{1}{n} \sum_{i=1}^{n} \partial_{b^{(2)}} (a_i^{(2)} - z_i)^2) && (90)\\
&= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot \partial_{b^{(2)}} (a_i^{(2)} - z_i) && (91)\\
&= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot \partial_{b^{(2)}} a_i^{(2)} && (92)\\
\partial_{b^{(2)}} C^{(2)} &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot a_i^{(1)}(1 - a_i^{(1)}) && (93)\\
& && (94)
\end{aligned}$$

Partial derivitive in regards to $a_i^{(1)}$

$$a_i^{(2)} = \sigma(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{95}$$

$$\partial_{a_i^{(1)}} a_i^{(2)} = \partial_{a_i^{(1)}} \sigma(a_i^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{96}$$

$$= a_i^{(1)}(1 - a_i^{(1)}) \cdot \partial_{a_i^{(1)}} (a_i^{(1)} \cdot w^{(2)} + b^{(2)}) \tag{97}$$

$$\partial_{a_i^{(1)}} a_i^{(2)} = a_i^{(1)}(1 - a_i^{(1)}) \cdot w^{(2)} \tag{98}$$

$$\tag{99}$$

$$\tag{100}$$

$$\partial_{a_i^{(1)}} C^{(2)} = \partial_{a_i^{(1)}} \left(\frac{1}{n} \sum_{i=1}^{n} (a_i^{(2)} - z_i)^2 \right) \tag{101}$$

$$= \frac{1}{n} \sum_{i=1}^{n} \partial_{a_i^{(1)}} (a_i^{(2)} - z_i)^2) \tag{102}$$

$$= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot \partial_{a_i^{(1)}} (a_i^{(2)} - z_i) \tag{103}$$

$$= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot \partial_{a_i^{(1)}} a_i^{(2)} \tag{104}$$

$$\partial_{a_i^{(1)}} C^{(2)} = \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(2)} - z_i) \cdot a_i^{(1)}(1 - a_i^{(1)}) \cdot w^{(2)} \tag{105}$$

$$\tag{106}$$

1.4.2 Cost Of The 1st layer

Here is en example of **back-propagation** in the works:

- We know that the expected value for the 2nd layer is z_i
- Let's use e_i to donote the expacted value for i-th sample of the 1st layer
- e_i is the differance between the activation of the first layer and the derivitive of the cost function of the second layer

$$e_i = a_i^{(1)} - \partial_{a_i^{(1)}} C^{(2)}$$

- Now we can define the cost function of the 1st layer as such

$$C^{(1)} = \frac{1}{n} \sum_{i=1}^{n} (a_i^{(1)} - e_i)^2$$

Now that we can easily accessi whe weights and bias of the first layer let's compute the corresponding partial derivitives

$$\begin{aligned} a_i^{(1)} &= \sigma(x_i \cdot w^{(1)} + b^{(1)}) && (107) \\ & && (108) \\ \partial_{w^{(1)}} a_i^{(1)} &= a_i^{(1)}(1 - a_i^{(1)}) \cdot a_i^{(0)} && (109) \\ \partial_{b^{(1)}} a_i^{(1)} &= a_i^{(1)}(1 - a_i^{(1)}) && (110) \end{aligned}$$

We know that the activation of 0th layer - $a_i^{(0)}$ represnets the i-th input value - x_i

$$\boxed{\partial_{w^{(1)}} a_i^{(1)} = a_i^{(1)}(1 - a_i^{(1)}) \cdot x_i} \qquad \boxed{\partial_{b^{(1)}} a_i^{(1)} = a_i^{(1)}(1 - a_i^{(1)})} \qquad (111)$$

$$\begin{aligned} \partial_{w^{(1)}} C^{(1)} &= \partial_{w^{(1)}} \left(\frac{1}{n} \sum_{i=1}^{n} (a_i^{(1)} - e_i)^2 \right) = && (112) \\ &= \frac{1}{n} \sum_{i=1}^{n} \partial_{w^{(1)}} (a_i^{(1)} - e_i)^2 = && (113) \\ &= \frac{1}{n} \sum_{i=1}^{n} 3 \cdot (a_i^{(1)} - e_i) \cdot \partial_{w^{(1)}} (a_i^{(1)} - e_i) = && (114) \\ &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(1)} - e_i) \cdot \partial_{w^{(1)}} a_i^{(1)} = && (115) \\ &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(1)} - (a_i^{(1)} - \partial_{a_i^{(1)}} C^{(2)})) \cdot \partial_{w^{(1)}} a_i^{(1)} = && (116) \\ &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (a_i^{(1)} - a_i^{(1)} + \partial_{a_i^{(1)}} C^{(2)}) \cdot \partial_{w^{(1)}} a_i^{(1)} = && (117) \\ &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot \partial_{a_i^{(1)}} C^{(2)} \cdot \partial_{w^{(1)}} a_i^{(1)} = && (118) \\ &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot \partial_{a_i^{(1)}} C^{(2)} \cdot a_i^{(1)}(1 - a_i^{(1)}) \cdot x_i = && (119) \\ &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot \partial_{a_i^{(1)}} C^{(2)} \cdot a_i^{(1)}(1 - a_i^{(1)}) \cdot x_i && (120) \\ & && (121) \\ & && (122) \\ \partial_{b^{(1)}} C^{(1)} &= \frac{1}{n} \sum_{i=1}^{n} 2 \cdot \partial_{a_i^{(1)}} C^{(2)} \cdot a_i^{(1)}(1 - a_i^{(1)}) && (123) \end{aligned}$$

1.5 Arbitrary Neural Model With 1 Input

Let's say that our neuarl model has m layers

1.5.1 Feed-Forward

TODO: Describe

To calculate the activation of the l-th layer for the i-th sample you must multiply the activation of the $l-1$ layer with the weight of the l-th layer then add the bias of the l-th layer and apply an activation function to the result.

As an example the sigmoid-σ function is used.

Let's also consider the activation of the 0th layer $a_i^{(0)}$ to represent the i-th sample of the input data - x_i

$$a_i^{(l)} = \sigma(a_i^{(l-1)} \cdot w^{(l)} + b^{(l)}) \tag{124}$$

Partial derivitives

$$\partial_{w^{(l)}} a_i^{(l)} = a_i^{(l)} \cdot (1 - a_i^{(l)}) \cdot a_i^{(l-1)} \tag{125}$$

$$\partial_{b^{(l)}} a_i^{(l)} = a_i^{(l)} \cdot (1 - a_i^{(l)}) \tag{126}$$

$$\partial_{a_i^{(l-1)}} a_i^{(l)} = a_i^{(l)} \cdot (1 - a_i^{(l)}) \cdot w^{(l)} \tag{127}$$

1.5.2 Back-Propaggation

TODO: Describe
Let's denote this differance $a_i^{(m)} - z_i$ as a parial derivitive $\partial_{a_i^{(m+1)}}$

$$\boxed{C^{(l)} = \frac{1}{n} \sum_{i=1}^{n} (\partial_{a_l^{(l)}} C^{(l+0)})^2} \tag{128}$$

$$\tag{129}$$

$$\partial_{w^{(l)}} C^{(l)} = \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (\partial_{a_l^{(l)}} C^{(l+1)}) \cdot a_i^{(l)} \cdot (1 - a_i^{(l)}) \cdot a_i^{(l-1)} \tag{130}$$

$$\partial_{b^{(l)}} C^{(l)} = \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (\partial_{a_l^{(l)}} C^{(l+1)}) \cdot a_i^{(l)} \cdot (1 - a_i^{(l)}) \tag{131}$$

$$\partial_{a_i^{(l-1)}} C^{(l)} = \frac{1}{n} \sum_{i=1}^{n} 2 \cdot (\partial_{a_l^{(l)}} C^{(l+1)}) \cdot a_i^{(l)} \cdot (1 - a_i^{(l)}) \cdot w^{(l)} \tag{132}$$

1.6 Combining Feed-Forward and Back-Propaggation

Let's describe the way we will use conecpts descrabed in the previous two sections togeter.

Let's say we have k samples of trainig data

1. For each smaple we will **forward** the sample through the neural network until we reach the output. - **Feed-Forward**
Node: Neural Network output is the $a_i^{(m+1)}$ *activation*

2. Straing from the $m+1$ activation let's move backwards through the neual network propagating the differances of layer activations - **Back-Propaggation**