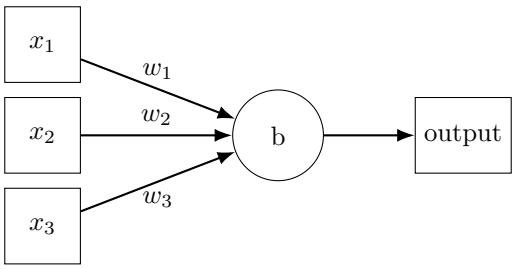
1 Code implementation



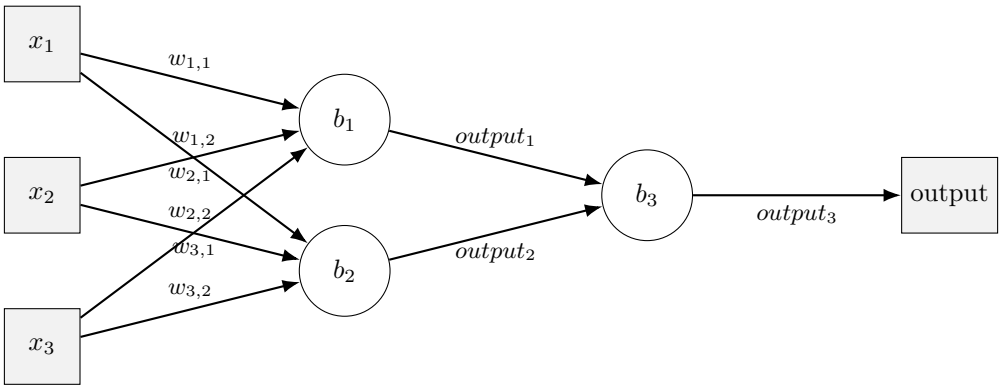
$$\text{output'} = \mathbf{x} \cdot \mathbf{w} + b$$

$$\text{output'} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \cdot \begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} + b$$

since **the sigmoid function** is defined as follows:

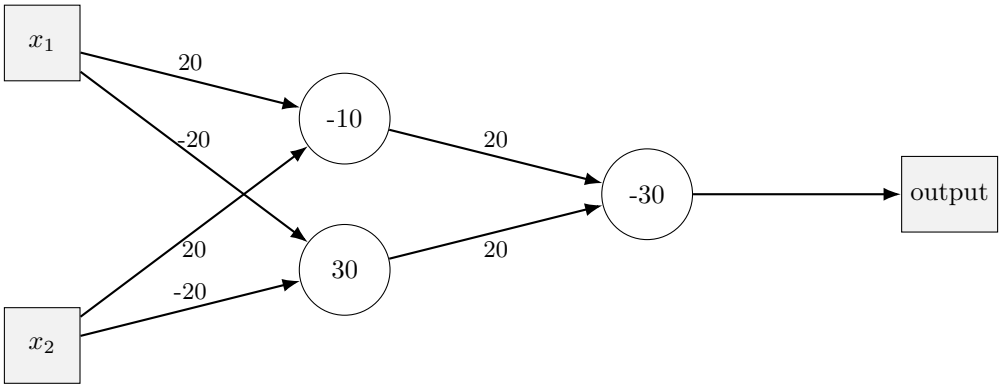
$$f(x) = \frac{1}{1 + e^{-x}}$$

$$\Rightarrow \quad \text{output} = f(\mathbf{x} \cdot \mathbf{w} + b)$$



$$\sigma(\begin{bmatrix} w_{1,1} & w_{2,1} & w_{3,1} \\ w_{1,2} & w_{2,2} & w_{3,2} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}) = \begin{bmatrix} output_1 \\ output_2 \end{bmatrix}$$

XOR example



1.1 Back Propagation

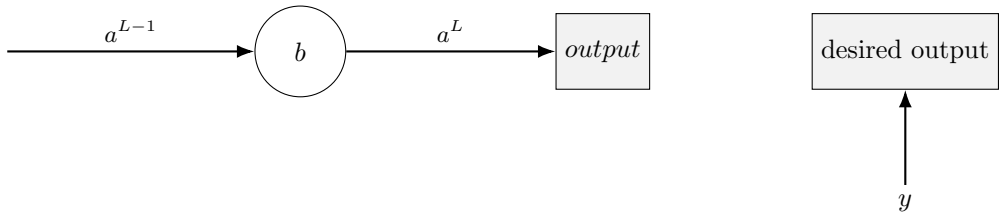
What is the **cost** of a single training example?

$$C(\text{one training instance}) = (\text{output} - \text{target})^2$$

$\Rightarrow$ cost of the entire network

$$\text{target}' = \begin{bmatrix} \text{target output for training data 0} \\ \text{target output for training data 1} \\ \text{target output for training data 2} \\ \text{target output for training data 3} \\ \vdots \end{bmatrix}$$

$$C(\mathbf{w}, \mathbf{b}, \text{target}') = \frac{1}{n} \sum_{i=1}^{n} C_i(\mathbf{w}, \mathbf{b}, \text{target}'_i)$$



$C_i \Rightarrow$ means for only one training data

$$\begin{aligned} C_0 &= (a^L - y)^2 \\ a^L &= \sigma(w^L \cdot a^{L-1} + b^L) \\ z^L &= w^L \cdot a^{L-1} + b^L \end{aligned}$$

$$\begin{aligned} \frac{\partial C_0}{\partial w^L} &= \frac{\partial}{\partial w^L}(a^L - y)^2 \\ &= 2(a^L - y) \cdot \frac{\partial}{\partial w^L}(a^L - y) \\ &= 2(a^L - y) \cdot \frac{\partial a^L}{\partial w^L} \quad (\text{since } \frac{\partial y}{\partial w^L} = 0) \end{aligned}$$

$$\begin{aligned} \frac{\partial a^L}{\partial w^L} &= \frac{\partial}{\partial w^L}\sigma(z^L) \\ &= \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial w^L} \\ &= \sigma(z^L)(1 - \sigma(z^L)) \cdot \frac{\partial z^L}{\partial w^L} \\ &\quad (\text{because } \sigma'(x) = \sigma(x)(1 - \sigma(x))) \end{aligned}$$

$$\begin{aligned} \frac{\partial z^L}{\partial w^L} &= \frac{\partial}{\partial w^L}(w^L \cdot a^{L-1} + b^L) \\ &= a^{L-1} \quad (\text{since } \frac{\partial b^L}{\partial w^L} = 0) \end{aligned}$$

$$\begin{aligned} \implies \frac{\partial C_0}{\partial w^L} &= 2(a^L - y) \cdot \sigma(z^L)(1 - \sigma(z^L)) \cdot a^{L-1} \\ &= 2(a^L - y) \cdot a^L(1 - a^L) \cdot a^{L-1} \\ \implies \frac{\partial C_0}{\partial b^L} &= 2(a^L - y) \cdot \sigma(z^L)(1 - \sigma(z^L)) \\ &= 2(a^L - y) \cdot a^L(1 - a^L) \end{aligned}$$

1.2 Backpropagation Generalization

From Previous

$$\implies \frac{\partial C_0}{\partial w^L} = 2(a^L - y) \cdot a^L(1 - a^L) \cdot a^{L-1}$$
$$\implies \frac{\partial C_0}{\partial b^L} = 2(a^L - y) \cdot a^L(1 - a^L)$$

Biases

$$\frac{\partial C}{\partial b^L} = 2a^L \odot \left(\mathbf{1} - a^L\right) \odot \left(a^L - y\right)$$

$$\begin{bmatrix} \frac{\partial C}{\partial b_0} \\ \frac{\partial C}{\partial b_1} \\ \frac{\partial C}{\partial b_2} \\ \frac{\partial C}{\partial b_3} \\ \vdots \end{bmatrix} = 2 \begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} \odot \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix} - \begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} \right) \odot \left(\begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} - \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \end{bmatrix} \right)$$

Activation of Previous Layer

$$\frac{\partial C}{\partial a^{L-1}} = \left(W^L\right)^\top \left[2a^L \odot (1 - a^L) \odot (a^L - y)\right]$$

$$\begin{bmatrix} \frac{\partial C}{\partial a_0^{L-1}} \\ \frac{\partial C}{\partial a_1^{L-1}} \\ \frac{\partial C}{\partial a_2^{L-1}} \\ \frac{\partial C}{\partial a_3^{L-1}} \\ \vdots \end{bmatrix} = \begin{bmatrix} W^L_{(N\times M)} & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots \end{bmatrix}^\top \cdot \left(2 \begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} \odot \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix} - \begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} \right) \odot \left(\begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} - \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \end{bmatrix} \right) \right)$$

Weights

$$\frac{\partial C}{\partial W^L} = \left[2a^L \odot (1 - a^L) \odot (a^L - y)\right] \left(a^{L-1}\right)^\top$$

$$\begin{bmatrix} \frac{\partial C}{\partial W^L}^L{}_{(N\times M)} & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots \end{bmatrix} = \left(2 \begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} \odot \left(\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \end{bmatrix} - \begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} \right) \odot \left(\begin{bmatrix} a_0^L \\ a_1^L \\ a_2^L \\ \vdots \end{bmatrix} - \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \end{bmatrix} \right) \right) \cdot \begin{bmatrix} a_0^{L-1} \\ a_1^{L-1} \\ a_2^{L-1} \\ \vdots \end{bmatrix}^\top$$

Next Layer

$$\textbf{Biases: } \frac{\partial C}{\partial b^{L-1}} = \frac{\partial C}{\partial a^{L-1}} \odot a^{L-1} \odot (1 - a^{L-1})$$

$$\textbf{Weights: } \frac{\partial C}{\partial W^{L-1}} = \frac{\partial C}{\partial b^{L-1}} \cdot \left(a^{L-2}\right)^{\top}$$

$$\textbf{Previous Activations: } \frac{\partial C}{\partial a^{L-2}} = \left(W^{L-1}\right)^{\top} \cdot \frac{\partial C}{\partial b^{L-1}}$$

Algorithm 1 Training Loop with Mini-batch Gradient Descent

```
for epoch = 1 to E do
    shuffle(training_data)
    for each batch ∈ mini_batches(training_data, B) do
        zero_gradients()
        for each (x, y) ∈ batch do
            forward(x)
            backward(y)
            accumulate_gradients()
        end for
        average_gradients(B)
        update_parameters(learning_rate)
    end for
end for
```