Feed-forward neural network

<u>binary classification</u>

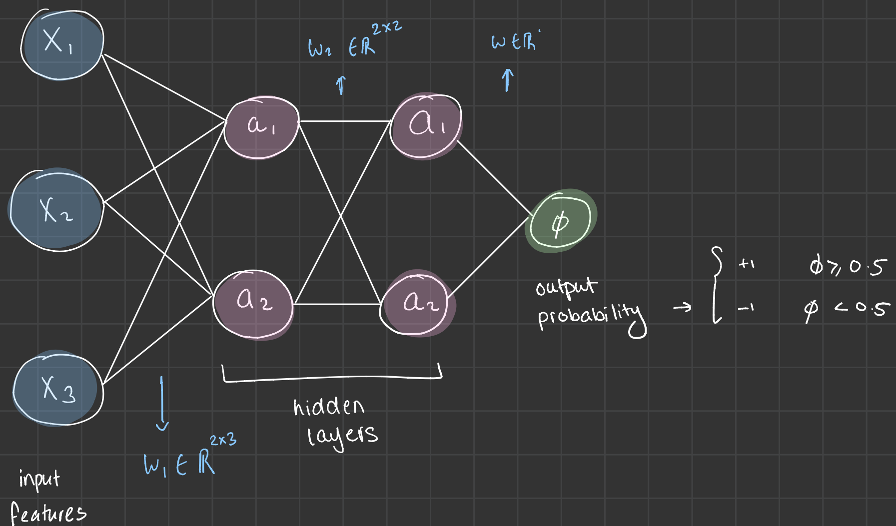
$X \in \mathbb{R}^{3\times1}$ is a datapoint w/ 3 features : $X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

$W \in \mathbb{R}^{1\times3}$ is a weight vector representing node connections

$b \in \mathbb{R}$ is a bias term

activation function

$$a = f\left(Wx + b \right) \in \mathbb{R}$$



$X = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$ $\quad W_1 = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \end{bmatrix}$ $\quad b_1 = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$

$$W_2 = \begin{bmatrix} 0.7 & 0.8 \\ 0.9 & 1 \end{bmatrix} \qquad b_2 = \begin{bmatrix} 0.3 \\ 0.4 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 1.1 & 1.2 \end{bmatrix} \qquad b_3 = 0.5$$

$$f(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$ → sigmoid activation

hidden layer #1

P.S. i did not bother calculating, GPT did. so if theres any mistakes, not my fault fr.

$$W_1X + b_1 = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 1.4 \\ 3.2 \end{bmatrix} + \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 3.4 \end{bmatrix}$$

$$f(W_1X + b_1) = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sigma(1.5) \\ \sigma(3.4) \end{bmatrix} = \begin{bmatrix} 0.818 \\ 0.968 \end{bmatrix}$$

hidden layer #2

"new" $X = \begin{bmatrix} 0.818 \\ 0.968 \end{bmatrix}$

$$W_2X + b_2 = \begin{bmatrix} 0.7 & 0.8 \\ 0.9 & 1 \end{bmatrix} \begin{bmatrix} 0.818 \\ 0.968 \end{bmatrix} + \begin{bmatrix} 0.3 \\ 0.4 \end{bmatrix}$$

$$= \begin{bmatrix} 2.226 \\ 2.734 \end{bmatrix}$$

$$f(w_2 x + b_2) = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sigma(2.226) \\ \sigma(2.734) \end{bmatrix} = \begin{bmatrix} 0.903 \\ 0.939 \end{bmatrix}$$

output layer

$$W_3 X + b_3 = \begin{bmatrix} 1.1 & 1.2 \end{bmatrix} \begin{bmatrix} 0.903 \\ 0.939 \end{bmatrix} + 0.5$$

$$= 2.1195 + 0.5$$

$$= 2.6195$$

$$f(W_3 X + b_3) = \hat{y} = \sigma(2.6195) = 0.932 \geq 0.5$$

$$\Rightarrow y_{pred} = +1$$

back propagation

loss function

goal: $\min\limits_{W} L(y, \hat{y})$

minimize w.r.t. W

actual label

predicted label

- stochastic gradient descent

update rules

$$w_j := w_j - \alpha \frac{\partial}{\partial w_j} L(y, \hat{y})$$

$\hat{y} = f(wx + b)$

$$b_j := b_j - \alpha \frac{\partial}{\partial b_j} L(y, \hat{y})$$

learning rate

assume we're working w/ a regression problem:

$$L(y, \hat{y}) = \frac{1}{2\underbrace{n}_{\text{number of samples}}} \sum_{i=1}^{n} (\underbrace{y_n}_{\text{true}} - \underbrace{\hat{y}_n}_{\text{predicted}})^2$$

let's just do this for one datapoint:

$$L(y, \hat{y}) = \frac{1}{2}(y - \underbrace{f(z)}_{\hat{y}})^2 \qquad \hat{y} = f(z) = e^z \qquad z = wx + b$$

$$\Rightarrow \frac{d}{dw} L(y, \hat{y}) = \frac{dL}{d\underbrace{f(z)}_{\text{activation function}}} \cdot \frac{df(z)}{d\underbrace{z}_{wx+b}} \cdot \frac{dz}{dw}$$

$$\frac{dL}{df(z)} = (y - f(z)) \qquad \frac{df(z)}{dz} = e^z \qquad \frac{dz}{dw} = x$$

$$\Rightarrow \frac{dL}{dw} = (y - f(z)) \cdot e^z \cdot x$$

$$= (y - (w\underbrace{x}_{\text{layer input, whatever it is}} + b)) \cdot e^{wx+b} \cdot x$$

- SGD with momentum : $w_t := w_t - \alpha \underbrace{V_t}_{\text{velocity}}$ → accumulates prev. gradients

$$v_t = \beta \underbrace{v_{t-1}}_{\text{velocity at previous time}} + (1 - \underbrace{\beta}_{\text{momentum}}) \cdot \frac{dL}{dw_{t-1}}$$

- Adam → uses momentum & adaptive learning rate