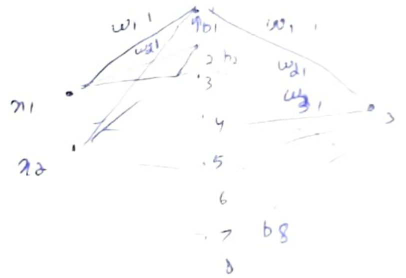
Building a neuron from scratch

we ~~have~~ have binary classification dataset

x_1 x_2 y

size 2000×2

after ~~batching~~ splitting $x_{train} = 1340 \times 2$



$n_1 = 2$

$n_2 = 8$

$n_3 = 1$

$W_1 = (n_2, n_1) = \cancel{8 \times 2}$ 8×2

$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{21} \\ w_{13} & \\ & \\ w_{81} & w_{28} \end{bmatrix}$$

$W_2 = (n_3, n_2) = 1 \times 8$ $[w_{21}\ \ w_{22}\ \ w_{23}\ \ \ \ w_8]$

$b_1 = (n_2, 1) = 8 \times 1$

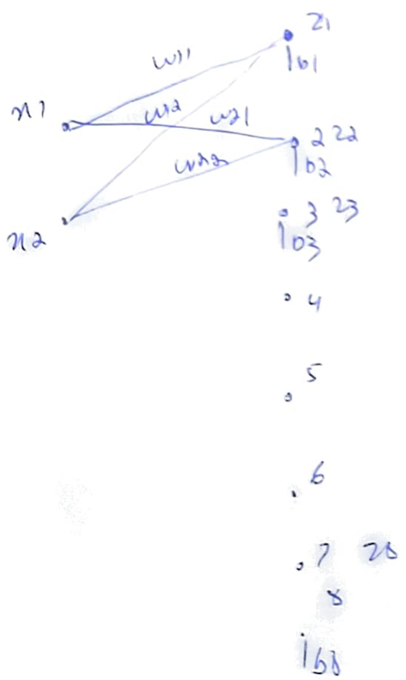
$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_8 \end{bmatrix}$$

$b_2 = (n_3 \times 1) = 1 \times 1 = [b_2]$

For batch normalization we will put

$$W_1 = W_1 \times \sqrt{\frac{1}{n_1 + n_2}}$$

$$W_2 = W_2 \times \sqrt{\frac{1}{n_2 + n_3}}$$



$$Z_1^{[1]} = w_{11}x_1 + w_{21}x_2 + b_1$$

$$Z_2 = w_{12}x_1 + w_{22}x_2 + b_1$$

$$\vdots$$

$$\cancel{w_8}\ Z_8 = w_{18}x_1 + w_{28}x_2 + b_8$$

$$Z^{[1]} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ \vdots & \\ w_{18} & w_{28} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_8 \end{bmatrix}$$

but this x is for one sample

x_1 x_2 ... 1 ... 5

we need to compute everything

so put x = training data^T

$x_{old} = 1340 \times 2$

$x_{new} = 2 \times 1340$

so $z^{[1]} = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \\ \vdots & \\ w_{18} & w_{28} \end{bmatrix}_{8\times 2} \times \begin{bmatrix} x_{11} & x_{12} & \cdots \\ x_{21} & x_{22} & \cdots \end{bmatrix}_{2\times 1340} + \begin{bmatrix} b_1 \\ \vdots \\ b_8 \end{bmatrix}_{8\times 1}$

here $W^{[1]}x$ will give 8×1340, usually we won't be able to generate an addition for this irregular shapes, but numpy modifies it i.e

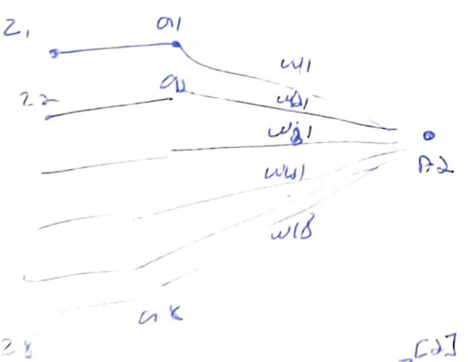
$b_1 \rightarrow \begin{bmatrix} b_1 \\ \vdots \\ b_8 \end{bmatrix}_{8\times 1} \Rightarrow \begin{bmatrix} b_1 & b_1 & \cdots \\ b_2 & b_2 & \cdots \\ \vdots & \vdots & \\ b_8 & b_8 & \cdots \end{bmatrix}_{8\times 1340}$

Forward propagation

so $z^{[1]} = W^{[1]}x + b_1$

$8\times 1340 + 8\times 1340$

so $z^{[1]} = 8\times 1340$



$z_1 \rightarrow a_1$ we use ReLU function

ReLU = max(0, z)

so $A^{[1]} = \text{ReLU}(z^{[1]})$

now for second layer $z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$

$1\times 8 \times 8\times 1340 + [1]$

$z^{[2]} = 1\times 1340 + 1\times 1340$ → after numpy

$z^{[2]} = 1\times 1340$

here A^2 is our output, so for output (due since binary classification)

we use sigmoid function $\frac{1}{1+e^{-x}}$

backward propagation

- we are using cross entropy as loss function

$L = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$

here $\hat{y} = A_2$ = sigmoid function

we will be finding how each weights & biases are affecting the loss function & try to update the weights & biases to reduce the cost / loss function

1) $\frac{dL}{dz_2}$

since $L = -y\log(A_2) - (1-y)\log(1-A_2)$

$\frac{\partial L}{\partial A_2} = -\frac{y}{A_2} - \frac{1-y}{1-A_2} \times -1 = \frac{-y}{A_2} + \frac{1-y}{1-A_2}$

$\frac{dL}{dz_2} \Rightarrow L = f(A_2)$

$A_2 = f(z_2)$

$\cancel{z_2 = f(}$

A_2 = sigmoid function $= \frac{1}{1+e^{-z}}$

so $\frac{dL}{dz_2} = \frac{\partial L}{\partial A_2} \times \frac{\partial A_2}{\partial z_2}$

$\frac{\partial A}{\partial z} = A_2' = f(z)(1-f(z))$

$f(z) = A_2$

$A_2' = A_2(1-A_2)$

$\frac{dL}{dz_2} = \left(\frac{-y}{A_2} + \frac{1-y}{1-A_2}\right) A_2(1-A_2)$

$= -y(1-A_2) + A_2 \cancel{\Rightarrow}(1-y)$

$= A^{[2]} - y$

$= -y + A^{[2]}\cancel{y} + A^{[2]} - \cancel{A^{[2]}y}$

$= A^{[2]} - y$

so $\frac{dL}{dz_2} = A^{[2]} - y$ = we will notate it as dz_2

$\boxed{dz_2 = A^{[2]} - y}$

now $dw_2 = \frac{dL}{dw_2} \Rightarrow$

$L = f(A_2)$

$A_2 = f(z_2)$

$z_2 = w^{[2]}A^{[1]} + b^{[2]}$

so by chain rule

$dw_2 = \frac{dL}{dw_2} = \underbrace{\frac{\partial L}{\partial A_2}\frac{\partial A_2}{\partial z_2}}_{dz_2} \cdot \frac{\partial z_2}{\partial w_2} \sim \cancel{w^{[2]}}$ $A^{[1]}$

$= dz^{2} \cdot A^{[1]}$

$1 \times 1340 \quad 18 \times 1340$

so use Transpose for $A^{[1]}$ & since it is a matrix multiplication

& divide it by m (no of samples)

so $\boxed{dw_2 = \frac{dz_2 \cdot A^{[1]T}}{m}}$

Now $\frac{\partial L}{\partial z_1} = dz_1$

$L = f(A_2)$

$A_2 = f(z_2)$

$z_2 = f(A_1) = z_2 = w^{[2]} A^{[1]} + b^{[2]}$

$A^{[1]} = f(z_1)$

so $dz_1 = \frac{dL}{dz_1} = \underbrace{\frac{\partial L}{\partial A_2} \times \frac{\partial A_2}{\partial z_2}}_{dz_2} \times \frac{\partial z_2}{\partial A_1} \times \frac{\partial A_1}{\partial z_1}$

$z_2 = w^{[2]} A^{[1]} + b^{[2]}$

$\frac{\partial z_2}{\partial A^{[1]}} = w^{[2]}$

$A_1 =$ ReLU function

$\frac{\partial A_1}{\partial z_1} =$ ReLU D function

so $dz_1 = \frac{dL}{dz_1} = \underbrace{dz_2 \cdot w^{[2]}}_{8^T \times 1340} * \text{reluD}(z_1)$

1×1340, 8×1, ~~8×1~~ 1×1340 — 8×134

8×1340

$8^T \times 1340$ * → Do element wise multiplication

$\boxed{dz_1 = w_2^T \cdot dz_2 * \text{relud}(z_1)}$

$dw_1 = \frac{dL}{dw_1} = \underbrace{\frac{\partial L}{\partial A_2} \cdot \frac{\partial A_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial A_1} \times \frac{\partial A_1}{\partial z_1}}_{dz_1} \times \frac{\partial z_1}{\partial w_1}$

$z_1 = w^{[1]} x + b^{[1]}$

$\frac{\partial z_1}{\partial w_1} = x$

$\boxed{dw_1 = \frac{dz_1 \cdot x^T}{m}}$

$dz_2 = A^{[2]} - y$

$dw_2 = \frac{dz_2 \cdot A^{[1]T}}{m}$

$dz_1 = w_2^T \cdot dz_2 * \text{relud}(z_1)$

$dw_1 = \frac{dz_1 \cdot x^T}{m}$

$w_1 = w_1 -$ learning rate $\times dw_1$

$w_2 = w_2 -$ learning rate $\times dw_2$

$X = 2 \times 1340$

$Y = 1 \times 1340$

$A^{[1]} = 8 \times 1340$

$A^{[2]} = 1 \times 1340$

$dz^{[2]} = 1 \times 1340$

$dw^{[2]} = 1 \times 8$

$w^{[1]} = 8 \times 2$

$w^{[2]} = 1 \times 8$

$dw^{[1]} = 8 \times 2$

$z^{[1]} = 8 \times 1340$

$z^{[2]} = 1 \times 1340$

$b^{[1]} = 8 \times 1$

$b^{[2]} = (1, 1)$

similarly for b_2 & b_1

$L = -y \log(A^{[2]}) - (1-y)\log(1-A^{[2]})$

$db_2 = \frac{\partial L}{\partial b_2} = \underbrace{\frac{\partial L}{\partial A^{[2]}} \cdot \frac{\partial A^{[2]}}{\partial Z^{[2]}}}_{dZ2} \times \frac{\partial Z^{[2]}}{\partial b^{[2]}}$

$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$

$\frac{\partial Z^{[2]}}{\partial b_2} = 1$

$L = f(A_2)$
$A_2 = f(Z_2)$
$Z_2 = f(A_1)$
$A^{(1)} = f(Z_1)$
$Z_1 = f(b_1)$

so $db_2 = dZ2 \times 1$

$db_2 = dZ^{[2]}$ 1×1340

$db_1 = \frac{\partial L}{\partial b_1} = \underbrace{\frac{\partial L}{\partial A2} \cdot \frac{\partial A2}{\partial Z2} \times \frac{\partial Z2}{\partial A1} \times \frac{\partial A1}{\partial Z1}}_{dZ^{[1]}} \times \frac{\partial Z1}{\partial b1}$

$\frac{\partial Z1}{\partial b_1}$ $\frac{\partial}{\partial b_1}[W^{[1]} X + b^{[1]}]$

$= 1$

so $db_1 = \frac{\partial L}{\partial b_1} = dZ^{[1]}$

~~but we~~ here $dZ^{[1]}$ & $dZ^{[2]}$ both are 1×1340

but we want the average

$$\boxed{\begin{aligned} \text{so}\quad db_1 &= \frac{\text{sum}(dZ^{[1]})}{m} \\ db_2 &= \frac{\text{sum}(dZ^{[2]})}{m} \\ b_2 &= b_2 - \alpha\, db_2 \\ b_1 &= b_1 - \alpha\, db_1 \end{aligned}}$$

Now Loss function

$L = -y \log(\hat{y}) - (1-y)\log(1-\hat{y})$

since we are using sigmoid function in output $\hat{y}$, sigmoid ranges (0,1), so if the value coming from sigmoid is very less ~~compared to~~, the computer will automatically round up this to 0, which is not desired so

$L = -y \log(\text{np.maximum}(\hat{y}, 1e{-}14)) - (1-y)\log(\text{np.maximum}(1-\hat{y}, 1e{-}14))$

$\text{cost} = \frac{\text{sum of loss functions}}{m}$

$m = X.\text{shape}[1]$

don't forget to reshape ~~Y~~ Ytest & Ytrain because it will be initially $(m,)$ → convert it to $(m,1)$