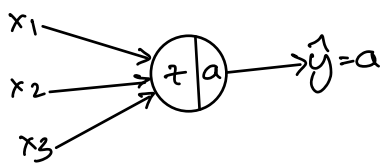
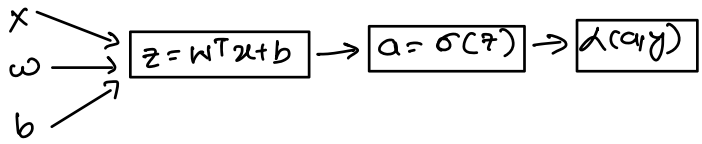
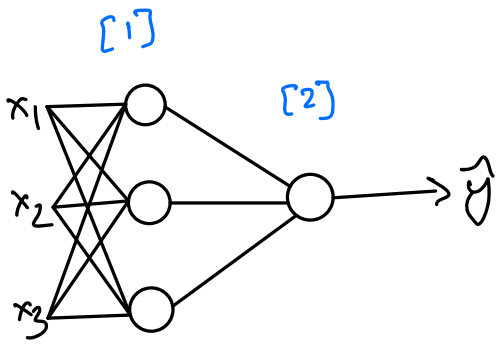
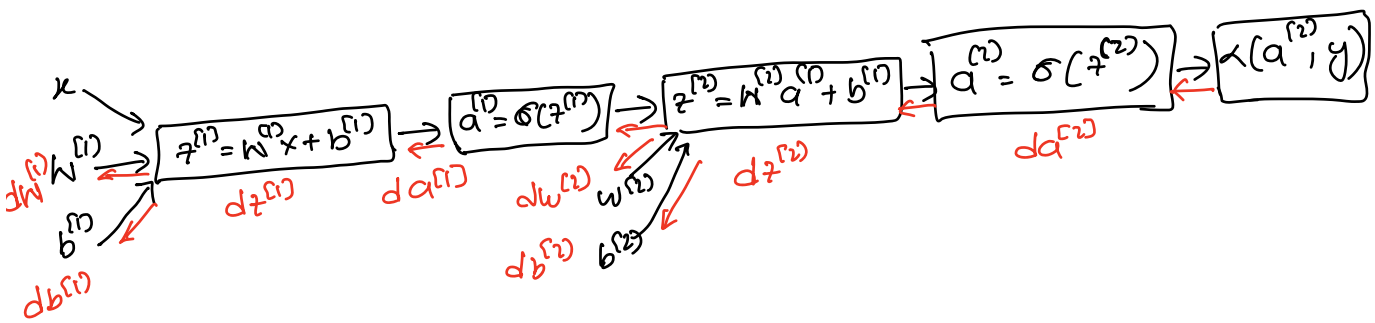
dec1: One hidden layer Network:-

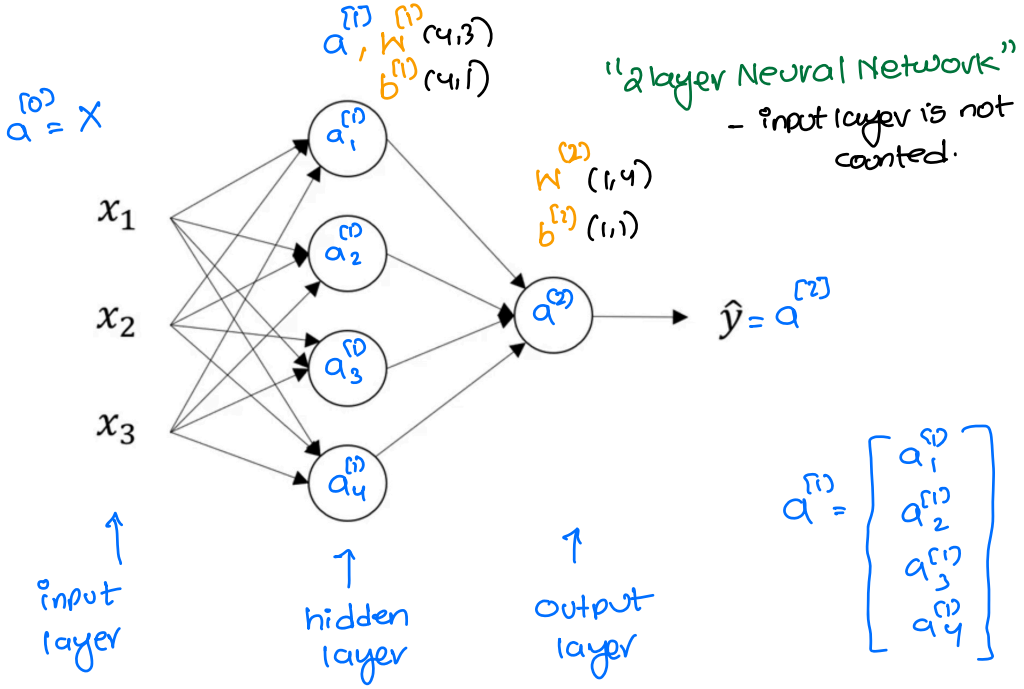






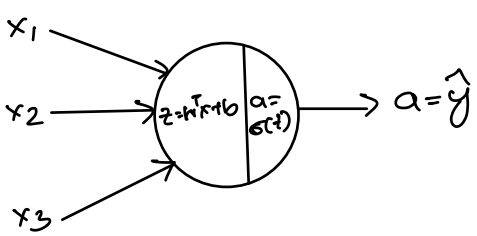


Lec 2: Neural Network Representation



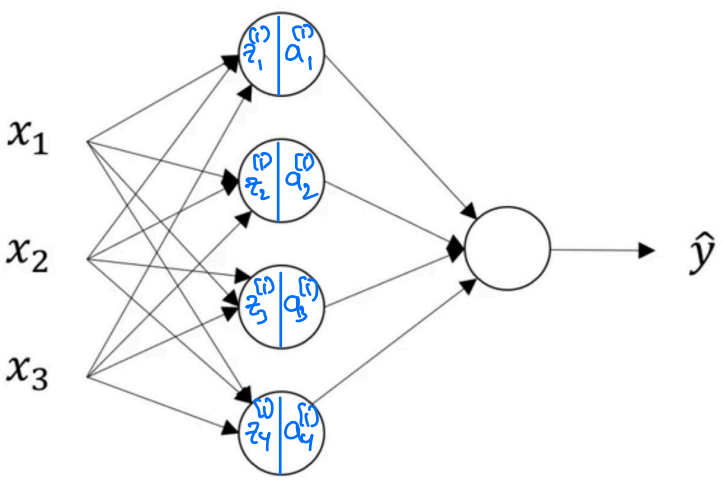
"2 layer Neural Network"
- input layer is not counted.

Lec 3: Computing a NN output!



$z = w^T x + b$

$a = \sigma(z)$



$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]} \qquad a_1^{[1]} = \sigma(z_1^{[1]})$$

$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]} \qquad a_2^{[1]} = \sigma(z_2^{[1]})$$

$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]} \qquad a_3^{[1]} = \sigma(z_3^{[1]})$$

$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]} \qquad a_4^{[1]} = \sigma(z_4^{[1]})$$

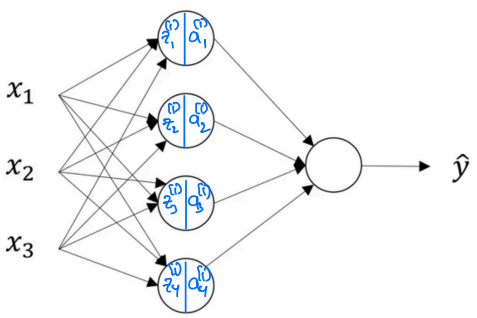
$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]} \qquad a_1^{[1]} = \sigma(z_1^{[1]})$$
$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]} \qquad a_2^{[1]} = \sigma(z_2^{[1]})$$
$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]} \qquad a_3^{[1]} = \sigma(z_3^{[1]})$$
$$z_4^{[1]} = w_4^{[1]T} x + b_4^{[1]} \qquad a_4^{[1]} = \sigma(z_4^{[1]})$$

vectorize

$$\begin{bmatrix} - & w_1^{[1]T} & - \\ - & w_2^{[1]T} & - \\ - & w_3^{[1]T} & - \\ - & w_4^{[1]T} & - \end{bmatrix}_{4\times 3} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3\times 1} + \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ b_3^{[1]} \\ b_4^{[1]} \end{bmatrix}_{[4,1]} = \begin{bmatrix} z_1^{[1]} \\ z_2^{[1]} \\ z_3^{[1]} \\ z_4^{[1]} \end{bmatrix} = z^{[1]}$$

$W^{[1]}$ $\quad x \quad$ $b^{[1]}$ $\quad z^{[1]}$

$$a^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ a_3^{[1]} \\ a_4^{[1]} \end{bmatrix} = \sigma(z^{[1]})$$

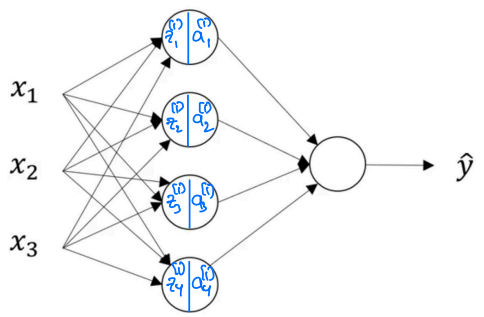


Final equations:

$$a^{[0]} = x_{(3,1)}$$

$$z^{[1]}_{(4,1)} = W^{[1]}_{(4,3)} a^{[0]}_{(3,1)} + b^{[1]}_{(4,1)}$$
$$a^{[1]}_{(4,1)} = \sigma(z^{[1]})_{(4,1)}$$
$$z^{[2]}_{(1,1)} = W^{[2]}_{(1,4)} a^{[1]}_{(4,1)} + b^{[2]}_{(1,1)}$$
$$a^{[2]}_{(1,1)} = \sigma(z^{[1]})_{(1,1)}$$

Deck:- Vectorizing across multiple examples:-



$a^{[0]} = x$

$z^{[1]} = W^{[1]} a^{[0]} + b^{[1]}$

$a^{[1]} = \sigma(z^{[1]})$

$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$

$a^{[2]} = \sigma(z^{[1]})$

} for single training example.

$x \longrightarrow a^{[2]} = \hat{y}$

$x^{(1)} \longrightarrow a^{[1](1)} = \hat{y}^{(1)}$

$x^{(2)} \longrightarrow a^{[2](2)} = \hat{y}^{(2)}$

$\vdots$

$x^{(m)} \longrightarrow a^{[2](m)} = \hat{y}^{(m)}$

$a^{[i](j)}$

i-th layer
j th example.

$$X = \begin{bmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{bmatrix}_{(n_x, m)}$$

for i = 1 to m:

$z^{[1](i)} = W^{[1]} x^{(i)} + b^{[1]}$

$a^{[1](i)} = \sigma(z^{[1](i)})$

$z^{[2](i)} = W^{[2]} a^{[1](i)} + b^{[2]}$

$a^{[2](i)} = \sigma(z^{[2](i)})$

Vectorized:-

$X = A^{[0]}$

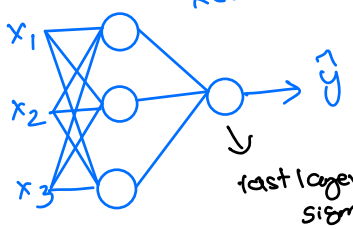
1
$$Z^{[1]}_{(4,m)} = W^{[1]}_{(4,n_x)} A^{[0]}_{(n_x,m)} + b^{[1]}_{(4,1)}$$
$$A^{[1]}_{(4,m)} = \sigma(Z^{[1]})_{(4,m)}$$

2
$$Z^{[2]}_{(1,m)} = W^{[2]}_{(1,4)} A^{[1]}_{(4,m)} + b^{[2]}_{(1,1)}$$
$$A^{[2]}_{(1,m)} = \sigma(Z^{[2]})_{(1,m)}$$

$$A^{[2]} = \begin{bmatrix} a^{[2](1)} & a^{[2](2)} & \cdots & a^{[2](m)} \end{bmatrix}_{(1,m)}$$

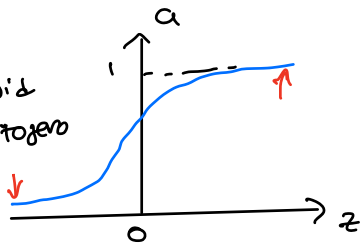
$$A^{[1]} = \begin{bmatrix} a_1^{[1](1)} & a_1^{[1](2)} & \cdots & a_1^{[1](m)} \\ \vdots & & & \vdots \\ a_4^{[1](1)} & a_4^{[1](2)} & \cdots & a_4^{[1](m)} \end{bmatrix}_{(4,m)}$$

Lecs:- Activation Function:-

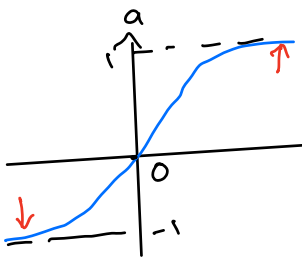


ReLU >> tanh(z) works better than sigmoid → centering of data to zero

last layer we can use sigmoid function $0 \le \hat{y} \le 1$



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



$a = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$

Given x

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

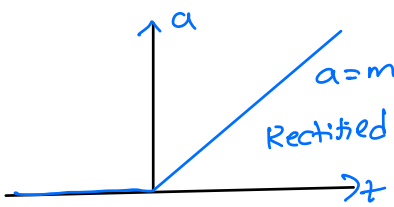
$a^{[1]} = \sigma(z^{[1]})$ $g^{[1]}(z^{[1]})$ tanh

$z^{[2]} = W^{[2]} a^{[1]} + b^{[2]}$

$a^{[2]} = \sigma(z^{[2]})$ $g^{[2]}(z^{[2]})$ sigmoid

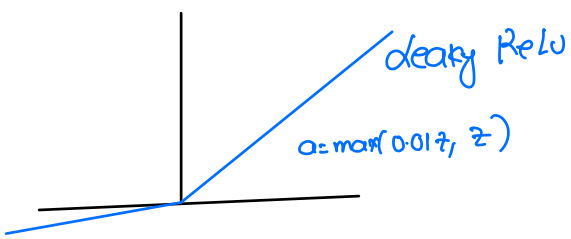
da is zero → gradient descent becomes slow.

ReLU

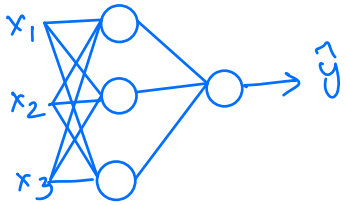


$a = \max(0, z)$

Rectified linear unit



dec 6:- Why do you need non linear activation function:-



assume that there is no activation function.

Given x:

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

$$a^{[1]} = z^{[1]} \quad \text{(linear activation)}$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

$$a^{[2]} = z^{[2]}$$

$$a^{[1]} = W^{[1]}x + b^{[1]}$$

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

$$= W^{[2]}\left[W^{[1]}x + b^{[1]}\right] + b^{[2]}$$

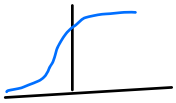
$$a^{[2]} = \left[W^{[2]}W^{[1]}\right]x + W^{[2]}b^{[1]} + b^{[2]}$$

$$a^{[2]} = W'x + b' \quad \text{(linear output)}$$

if $y \in \mathbb{R}$, we can use $g(z) = z$ linear activation, such as in output layer in regression.

Lec7: Derivatives of Activation Functions:-

Sigmoid $g(z) = \frac{1}{1+e^{-z}}$



$$\frac{dg(z)}{dz} = \frac{-1(-e^{-z})}{(1+e^{-z})^2} = \frac{e^{-z}}{1+e^{-z}} \cdot \frac{1}{1+e^{-z}} = (1-g(z))\, g(z)$$

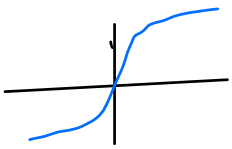
$g'(z) = g(z)(1-g(z))$ $\quad g'(z) = a(1-a)$

$z=10 \quad g(z) \simeq 1 \quad g'(z) \simeq 0$

$z=-10 \quad g(z) \simeq 0 \quad g'(z) \simeq 0$

$z=0 \quad g(z) = 0.5 \quad g'(z) = 0.25$

Tanh activation:-



$$g(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$$

$$g'(z) = \frac{(e^{z}+e^{-z})(e^{z}+e^{-z}) - (e^{z}-e^{-z})(e^{z}-e^{-z})}{(e^{z}+e^{-z})^2}$$

$$= 1 - \left(\frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}\right)^2$$

$g'(z) = 1 - (\tanh z)^2$

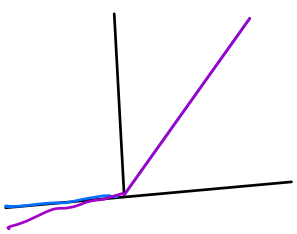
$a = \tanh z \quad g'(z) = 1-a^2$

$z=0 \quad \tanh(z) \simeq 1 \quad g'(z) \simeq 0$

$z=-10 \quad \tanh(z) \simeq -1 \quad g'(z) \simeq 0$

$z=0 \quad \tan(z) \simeq 0 \quad g'(z) = 1$

ReLU, Leaky



$g(z) = \max(0, z)$

$g'(z) = \begin{cases} 0 & \text{if } z<0 \\ 1 & \text{if } z \geq 0 \end{cases}$

$g(z) = \max(0.01z, z)$

$g'(z) = \begin{cases} 0.01 & \text{if } z<0 \\ 1 & \text{if } z \geq 0 \end{cases}$

Lec8:- Gradient Descent for single hidden layer:-

Parameters: $w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}$

$n_x = n^{[0]}$ - input features

$n^{[1]}$ - hidden units

$n^{[2]} = 1$ - output units

$W^{[1]} - (n^{[1]}, n^{[0]})$ $W^{[2]} - (n^{[2]}, n^{[1]})$

$b^{[1]} - [n^{[1]})$ $b^{[2]} - (n^{[2]})$

Cost function: $J(w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(y, \hat{y})$ $\hat{y} = a^{[2]}$

Initialize $w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}$

Gradient descent:

Repeat {

compute prediction $(\hat{y}^{(i)}, i = 1, \ldots m)$

$dw^{[1]}, db^{[1]}, dw^{[2]}, db^{[2]}$ $\frac{dJ}{dvar}$

update

$w^{[1]} = w^{[1]} - \alpha\, dw^{[1]}$

$b^{[1]} = b^{[1]} - \alpha\, db^{[1]}$

$w^{[2]} = w^{[2]} - \alpha\, dw^{[2]}$

$b^{[2]} = b^{[2]} - \alpha\, db^{[2]}$

}

Equations: deca:-

Forward propagation:

$$z^{[1]} = W^{[1]} x + b^{[1]}$$
$$A^{[1]} = g^{[1]}(z^{[1]})$$
$$z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$$
$$A^{[2]} = g^{[2]}(z^{[2]})$$

back propagation:-

$$Y = [y^{(1)} \dots y^{(m)}]_{(1,m)}$$

$$dz^{[2]} = A^{[2]} - Y$$
$$dw^{[2]} = \frac{1}{m} dz^{[2]} A^{[1]T}$$
$$db^{[2]} = \frac{1}{m} np.sum(dz^{[2]}, axis=1, keepdims=True) \quad (n^{[2]}, 1)$$
$$dz^{[1]} = w^{[2]T} dz^{[2]} * g^{[1]'}(z^{[1]}) \quad \text{(elementwise)}$$
$$(n^{[1]}, m) \quad (n^{[1]}, m) \quad (n^{[1]}, m)$$
$$dW^{[1]} = \frac{1}{m} dz^{[1]} x^T$$
$$db^{[1]} = \frac{1}{m} np.sum(dz^{[1]}, axis=1, keepdims=True)$$
$$(n^{[1]}, 1)$$

Summary of gradient descent

$dz^{[2]} = a^{[2]} - y$	$dZ^{[2]} = A^{[2]} - Y$
$dW^{[2]} = dz^{[2]} a^{[1]T}$	$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$
$db^{[2]} = dz^{[2]}$	$db^{[2]} = \frac{1}{m} np.sum(dZ^{[2]}, axis = 1, keepdims = True)$
$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]'}(z^{[1]})$	$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]'}(Z^{[1]})$
$dW^{[1]} = dz^{[1]} x^T$	$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$
$db^{[1]} = dz^{[1]}$	$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$

$$J(\dots) = \frac{1}{m} \sum_{i=1}^{n} \mathcal{L}(\hat{y}, y)$$

elementwise product

Summary of gradient descent (W3A1)

$$dz^{[2]} = a^{[2]} - y \qquad dZ^{[2]} = A^{[2]} - Y$$

$$J(\cdot) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}(\hat{y}, y)$$

$$dW^{[2]} = dz^{[2]} a^{[1]T} \qquad dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

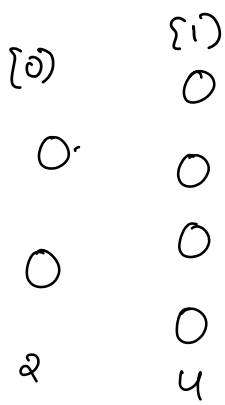
$$db^{[2]} = dz^{[2]} \qquad db^{[2]} = \frac{1}{m} np.sum(dZ^{[2]}, axis = 1, keepdims = True)$$

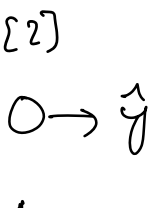
$$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]\prime}(z^{[1]}) \qquad dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]\prime}(Z^{[1]})$$

($n^{[1]}, 1$) ($n^{[1]}, m$) ($n^{[2]}, m$) ($n^{[1]}, m$) — element-wise product

$$dW^{[1]} = dz^{[1]} x^T \qquad dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

$$db^{[1]} = dz^{[1]} \qquad db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$$





$W_1 = (4,2)$ $b_1 = (4,1)$

$W_2 = (1,4)$ $b_2 = (1,1)$

$X = (2,m)$ $Y = (1,m)$

Forward propagation:

$$z^{[1]} = W^{[1]} X + b^{[1]}$$

$(4,m)$ $(4,2)$ $(2,m)$ $(4,1)$

$$A^{[1]} = g^{[1]}(z^{[1]})$$

$(4,m)$ $(4,m)$

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

$(1,m)$ $(1,4)$ $(4,m)$ $(1,1)$

$$A^{[2]} = g^{[2]}(z^{[2]})$$

$(1,m)$ $(1,m)$

$z^{[1]} - (4,m)$ $A^{[1]} - (4,m)$

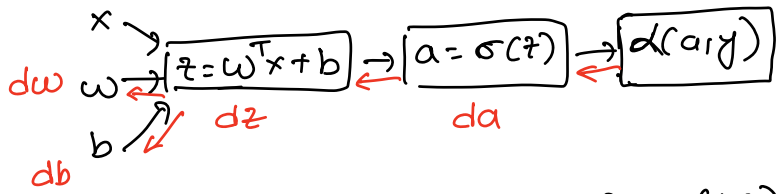
$z^{[2]} - (1,m)$ $A^{[2]} - (1,m)$

$dz^{[1]} - (4,m)$ $dA^{[1]} - (4,m)$

$dz^{[2]} - (1,m)$ $dA^{[2]} - (1,m)$

Lec 10: computing gradients:-

logistic regression:



$$\mathcal{L}(a, y) = -y \log a - (1-y) \log(1-a)$$

$$da = -\frac{y}{a} + \frac{(1-y)}{1-a}$$

$$dz = \frac{d\mathcal{L}}{dz} = \frac{d\mathcal{L}}{da} \frac{da}{dz}$$

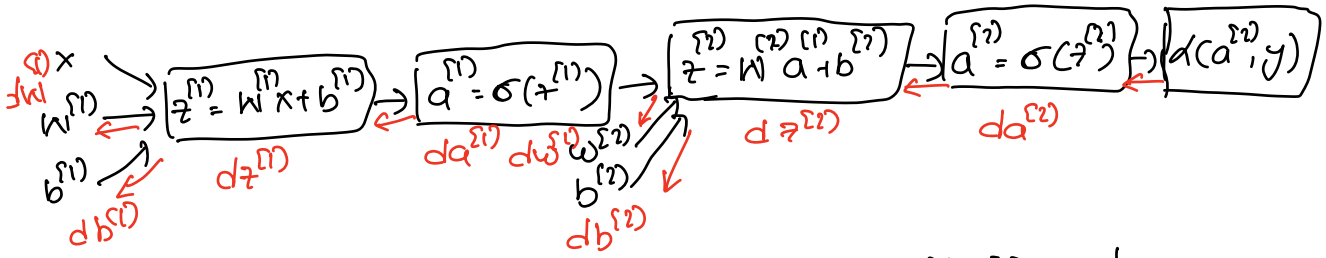
$$= da \frac{d\sigma(z)}{dz} = da\, a(1-a)$$

$$dz = \left(-\frac{y}{a} + \frac{(1-y)}{1-a}\right) a(1-a) = -y(1-a) + (1-y)a$$

$$dz = -y + ya + a - ya = a - y$$

$$\boxed{dz = a - y}$$

Gradient for one hidden layer: decll



$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]} a^{[1]T}$$

$$db^{[2]} = dz^{[2]}$$

$$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]\prime}(z^{[1]})$$

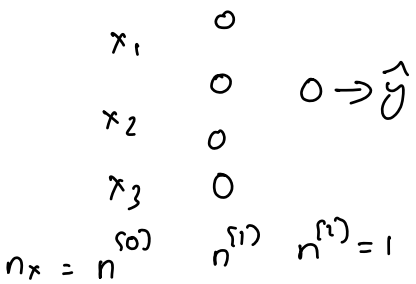
$$W^{[2]} = (n^{[2]}, n^{[1]})$$

$$z^{[2]}, dz^{[2]} = (n^{[2]}, 1) = (1, 1)$$

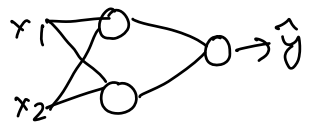
$$z^{[1]}, dz^{[1]} = (n^{[1]}, 1)$$

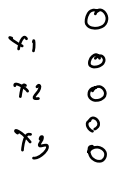
$$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]\prime}(dz^{[1]})$$

$$(n^{[1]}, 1) \quad (n^{[1]}, n^{[2]})(n^{[2]}, 1) \Rightarrow (n^{[1]}, 1) = (n^{[1]}, 1)$$



dec12: Random Initialization:-





(all hidden units computes same function)

$n^{(0)} = 2 \quad n^{(1)} = 2$

$$W^{[1]} = (2,2) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad b^{[1]} = (2,1) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

problem ?

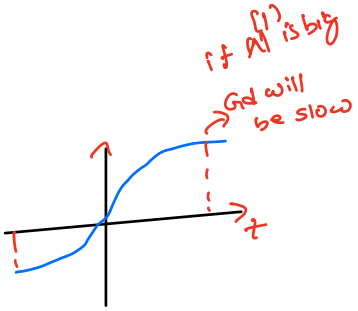
$$W^{[2]} = [0, 0]$$

$$a_1^{[1]} = a_2^{[1]} = \text{same}$$

$$dz_1^{[1]} = dz_2^{[1]} = \text{same}$$

$$dW = \begin{bmatrix} u & v \\ u & v \end{bmatrix}$$

$$W^{[1]} = W^{[1]} - \alpha dW^{[1]}$$



Sol:-

$$W^{[1]} = \text{np.random.randn}((2,2)) * 0.01$$

$$b^{[1]} = \text{np.zero}((2,1))$$

$$W^{[2]} = \text{np.random.randn}((1,2)) * 0.01$$

$$b^{[2]} = \text{np.zero}((1,1))$$