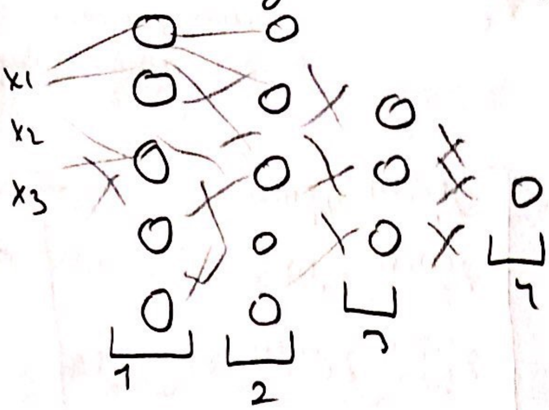
07.02.2019

Forward propagation in a deep network




$$z^{[l]} = w^{[l]} A^{[l-1]} + b^{[l]}$$

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

$x$: $z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$ $\xrightarrow{\text{vectorial}}$

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

$z^{[2]} = w^{[2]} a^{[1]} + b^{[2]}$ $[z^{[2](1)}, z^{[2](2)} \dots]$

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

$\vdots$

$z^{[4]} = w^{[4]} a^{[3]} + b^{[4]}$, $a^{[4]} = g^{[4]}(z^{[4]})$

$Z^{[1]} = W^{[1]} A^{[0]} + b^{[1]}$

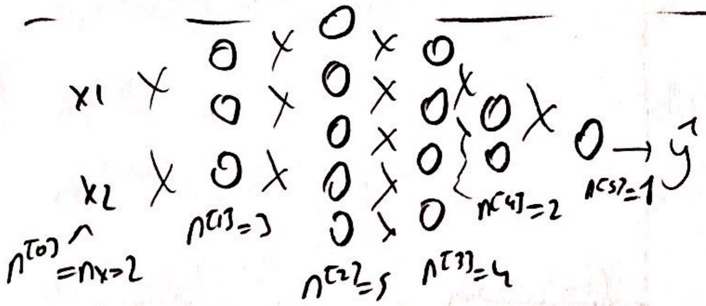
$A^{[1]} = g^{[1]}(Z^{[1]})$

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

$\vdots$

$\hat{Y} = g(Z^{[4]}) = A^{[4]}$

for $l = 1, 2, \dots 4$




$L = 5$

$w$ (hidden units, number of inputs)

$b$ (hidden units, 1)

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

$W^{[2]}$ : $(n^{[2]}, n^{[1]})$ $(5,3)$

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

$(5,1) = (5,3)\ (3,1) + (5,1)$

$W^{[3]} = (4,5)$

$W^{[4]} = (2,4)$

$W^{[5]} = (1,2)$

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

$(3,1)$ $(3,2)$ $(2,1)$ $(3,1)$

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

$$\begin{bmatrix} \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \end{bmatrix}$$

aynı dimension olmalı

$z^{[l]}, g^{[l]}(a^{[l]})$

$$\boxed{w^{[l]} : (n^{[l]}, n^{[l-1]})}$$

$b^{[l]} : (n^{[l]}, 1)$

$dw^{[l]} : (n^{[l]}, n^{[l-1]})$

$db^{[l]} : (n^{[l]}, 1)$

vectorized implementation

$z_t^{[n](i)}$

$n \Rightarrow$ layer number
$i \Rightarrow$ training number
$t \Rightarrow$ neuron number in layer

$$z^{[1]} = w^{[1]} \cdot x + b^{[1]}$$

$z^{[1]}$: $(n^{[1]}, 1)$  $w^{[1]}$: $(n^{[1]}, n^{[0]})$  $x$: $(n^{[0]}, 1)$  $b^{[1]}$: $(n^{[1]}, 1)$

$[\, z^{[1](1)}, z^{[1](2)}, z^{[2](3)} \cdots z^{[1](m)} \,]$ → training number

$$\begin{array}{c} \text{training} \\ \begin{bmatrix} z_1^{[1](1)} & z_1^{[1](2)} & \cdots m \\ z_2^{[1](1)} & z_1^{[1](2)} & \\ z_3^{[1](1)} & z_1^{[1](2)} & \end{bmatrix} \end{array}$$


$$z^{[1]} = w^{[1]} \cdot x + b^{[1]}$$

$z^{[1]}$: $(n^{[1]}, m)$  $w^{[1]}$: $(n^{[1]}, n^{[0]})$  $x$: $(n^{[0]}, m)$  $b^{[1]}$: $(n^{[1]}, 1)$ → broadcasting → $(n^{[1]}, m)$

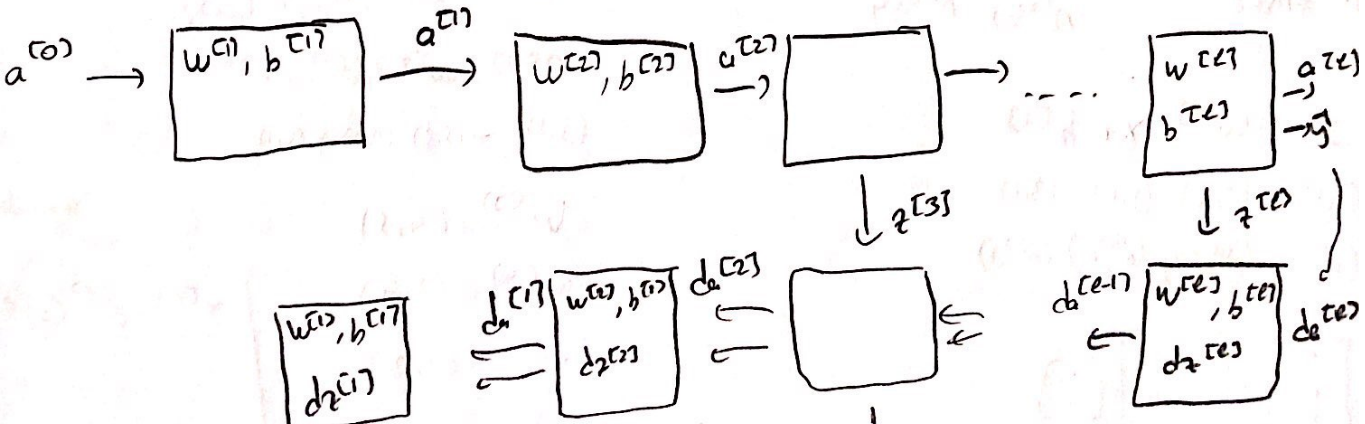
→ hidden unit → training size

$z^{[l]}, a^{[l]} : (n^{[l]}, 1)$

$Z^{[l]}, A^{[l]} : (n^{[l]}, m)$

$l=0 \quad A^{[0]} = X = (n^{[0]}, m)$

$dZ^{[l]}, dA^{[l]} : (n^{[l]}, m)$




$dw^{[1]}$
$db^{[1]}$

$dw^{[2]}$
$db^{[2]}$

$dw^{[3]}$
$db^{[3]}$

$$w^{[l]} := w^{[l]} - \alpha\, dw^{[l]}$$
$$b^{[l]} := b^{[l]} - \alpha\, db^{[l]}$$

Building blocks of deep neural networks 7.02.2019

Forward propagation for layer $l$

→ Input $a^{[l-1]}$

→ Output $a^{[l]}$, cache $(z^{[l]})$ → $W^{[l]}, b^{[l]}$

$$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$$
$$a^{[l]} = g^{[l]}(z^{[l]})$$

Vecto..

$$Z^{[l]} = W^{[l]} \cdot A^{[l-1]} + b^{[l]}$$
$$A^{[l]} = g^{[l]}(Z^{[l]})$$

Backward propagation for layer $l$

→ Input $da^{[l]}$

→ Output $da^{[l-1]}, dW^{[l]}, db^{[l]}$

$$dz^{[l]} = da^{[l]} * g^{[l]\prime}(z^{[l]})$$
$$dW^{[l]} = dz^{[l]} \cdot a^{[l-1]}$$
$$db^{[l]} = dz^{[l]}$$
$$da^{[l-1]} = \underbrace{W^{[l]T} \cdot dz^{[l]}}_{l(+1)}$$
$$dz^{[l]} = W^{[l+1]T} \cdot dz^{[l+1]} * g^{[l]\prime}(z^{[l]})$$

$$dZ^{[l]} = dA^{[l]} * g^{[l]\prime}(Z^{[l]})$$
$$dW^{[l]} = \frac{1}{m} dZ^{[l]} \cdot A^{[l-1]T}$$
$$db^{[l]} = \frac{1}{m} np.sum(dZ^{[l]}, axis=1, keepdims=True)$$
$$dA^{[l-1]} = W^{[l]T} \cdot dZ^{[l]}$$

---

What are hyperparameters?

Later course: Momentum, minibatch, regulariz..

Parameters: $W^{[1]}, b^{[1]}$ ...

Hyperparameters: learning rate $\alpha$

iterations

hidden layer $L$

hidden unit $n^{[1]}, n^{[2]}$

Choice of activation function