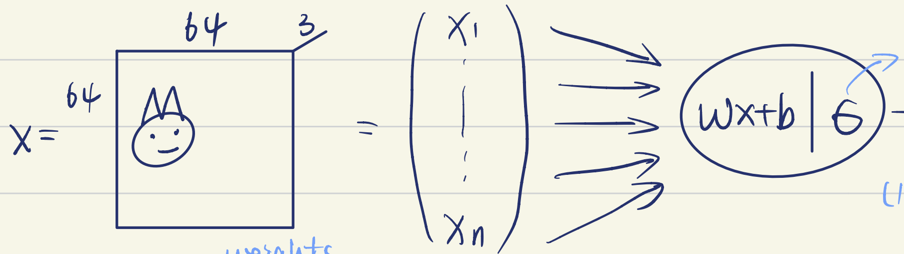
Lecture 1 | Intro to Neural Networks

1. From logistic regression to neural networks

<u>goal</u>: Find cats in images $\begin{cases} 1 \rightarrow \text{presence of a cat} \\ 0 \rightarrow \text{absence of a cat} \end{cases}$



$\hat{y} = \sigma(\theta^T x)$ (1,1)

$= \sigma(wx + b)$

w: (1, 12288) $\quad$ x: (64×64×3, 1), 64×64×3 = 12288

1) initialize w, b (weights, bias)

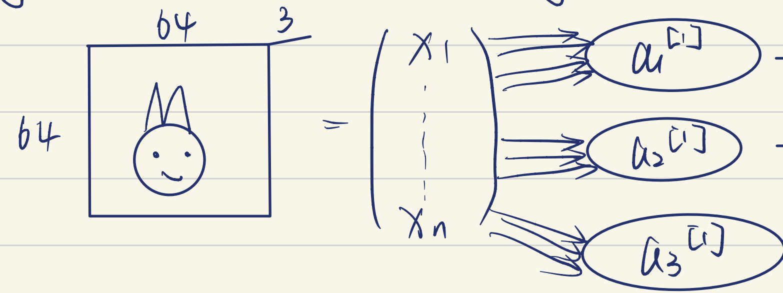
2) find the optimal w, b $\Rightarrow$ $L = -[y \log \hat{y} + (1-y)\log(1-\hat{y})]$

$\Rightarrow \begin{cases} w = w - \alpha \frac{\partial L}{\partial w} \\ b = b - \alpha \frac{\partial L}{\partial b} \end{cases}$

3) use $\hat{y} = \sigma(wx+b)$ to predict

- Equation 1: neuron = linear + activation
- Equation 2: model = architecture + parameters

goal 2.0: Find cat/lion/iguana in images



$\longrightarrow \hat{y}_1 = a_1^{[1]} = \sigma(w_1^{[1]} x + b_1^{[1]})$

$\longrightarrow \hat{y}_2 = a_2^{[1]} = \sigma(w_2^{[1]} x + b_2^{[1]})$

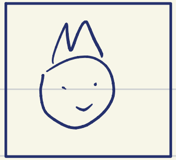
$\longrightarrow \hat{y}_3 = a_3^{[1]} = \sigma(w_3^{[1]} x + b_3^{[1]})$

$\Rightarrow$ but if the picture has both a cat and a lion:

the output $\Rightarrow \begin{array}{cl} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} & \begin{array}{l} p(cat) \\ p(lion) \\ p(iguana) \end{array} \end{array}$

goal 3.0: + constraint, unique animals on an image

softmax multi-class regression



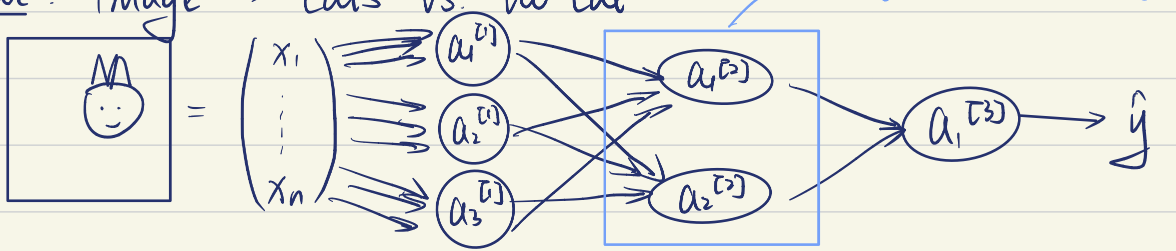
$$= \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \Rightarrow \begin{cases} z_1^{[1]} = e^{z_1^{[1]}} \big/ \sum_{k=1}^{3} e^{z_k^{[1]}} \\ z_2^{[1]} = e^{z_2^{[1]}} \big/ \sum_{k=1}^{3} e^{z_k^{[1]}} \\ z_3^{[1]} = e^{z_3^{[1]}} \big/ \sum_{k=1}^{3} e^{z_k^{[1]}} \end{cases}$$

} the sum equals 1.

cross-entropy loss: $\mathcal{L}_{CE} = -\sum_{k=1}^{3} y_k \log \hat{y}_k$

2. Neural Networks (end to end learning; black box model)

goal: image $\Rightarrow$ cats (1) vs. no cat (0)



parameters: $3n+3$ $\quad 2\times3+2$ $\quad 2\times1+1$

propagation equations:

$$\underset{(3,1)}{z^{[1]}} = \overset{(3,n)}{W^{[1]}}\,\overset{(n,1)}{x} + \underset{(3,1)}{b^{[1]}}$$

$$\underset{(3,1)}{a^{[1]}} = \sigma(z^{[1]})$$

$$\underset{(2,1)}{z^{[2]}} = \overset{(2,3)}{W^{[2]}}\,a^{[1]} + \underset{(2,1)}{b^{[2]}}$$

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

$$\underset{(1,1)}{z^{[3]}} = \overset{(1,2)}{W^{[3]}}\,a^{[2]} + \underset{(1,1)}{b^{[3]}}$$

$$\underset{(1,1)}{a^{[3]}} = \sigma(z^{[3]})$$

- What happened when an input batch of m examples?

$$X = \begin{pmatrix} | & | & & | \\ x^{(1)} & x^{(2)} & \cdots & x^{(m)} \\ | & | & & | \end{pmatrix} ; \quad \underset{(3,m)}{Z^{[1]}} = \overset{(3,n)}{W^{[1]}}\,\overset{(n,m)}{X} + \underset{(3,1)}{b^{[1]}} = \begin{bmatrix} | & | & & | \\ z^{[1](1)} & z^{[1](2)} & \cdots & z^{[1](m)} \\ | & | & & | \end{bmatrix}$$

broadcasting

$$\Rightarrow \tilde{b}^{[1]} = \underbrace{\begin{pmatrix} | & | & & | \\ b^{[1]} & b^{[1]} & \cdots & b^{[1]} \\ | & | & & | \end{pmatrix}}_{m}$$

- Optimizing $w^{[1]}, w^{[2]}, w^{[3]}, b^{[1]}, b^{[2]}, b^{[3]}$

  Define loss / cost function:

  1 example — multiple example

$$J(\hat{y}, y) = \frac{1}{m}\sum_{i=1}^{m} L^{(i)}, \text{ with } L^{(i)} = -\left[y^{(i)}\log \hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right]$$

- Backward Propagation

$$\forall l = 1,2,3: \begin{cases} w^{[l]} = w^{[l]} - \alpha \frac{\partial J}{\partial w^{[l]}} \\ b^{[l]} = b^{[l]} - \alpha \frac{\partial J}{\partial b^{[l]}} \end{cases}$$

$$\frac{\partial J}{\partial w^{[3]}} = \frac{\partial J}{\partial a^{[3]}} \cdot \frac{\partial a^{[3]}}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial w^{[3]}}$$

$$\frac{\partial J}{\partial w^{[2]}} = \frac{\partial J}{\partial z^{[3]}} \cdot \frac{\partial z^{[3]}}{\partial a^{[2]}} \cdot \frac{\partial a^{[2]}}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial w^{[2]}}$$

$$\frac{\partial J}{\partial w^{[1]}} = \frac{\partial J}{\partial z^{[2]}} \cdot \frac{\partial z^{[2]}}{\partial a^{[1]}} \cdot \frac{\partial a^{[1]}}{\partial z^{[1]}} \cdot \frac{\partial z^{[1]}}{\partial w^{[1]}}$$