Intros

Structure: Weekly meetings, more like office hours.
↳ also, show me what has been done in the code.

Go over rough idea:

To do:

- ▷ Watch 3b1b videos

- ▷ Find meeting time for next week
  (July 3rd only day I am available)

- ▷ Code implementation
  (probably best to use python. Google collab .. ?)

- ▷ Data processing:
  - Use Mnist or USPS ?

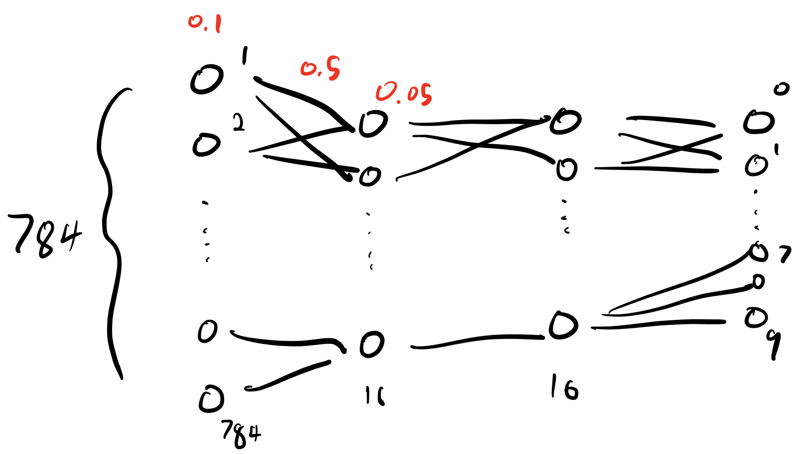
- ▷ Implementation: no hidden layers first ?

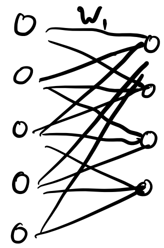
- ▷ **Recommended**: Students meet once per week for a couple hours to try to code the neural network.

$28\left\{\underbrace{\boxed{7}}_{28}\right.$ $\longleftrightarrow$ vector w/ $28 \cdot 28 = \underline{784}$ components.

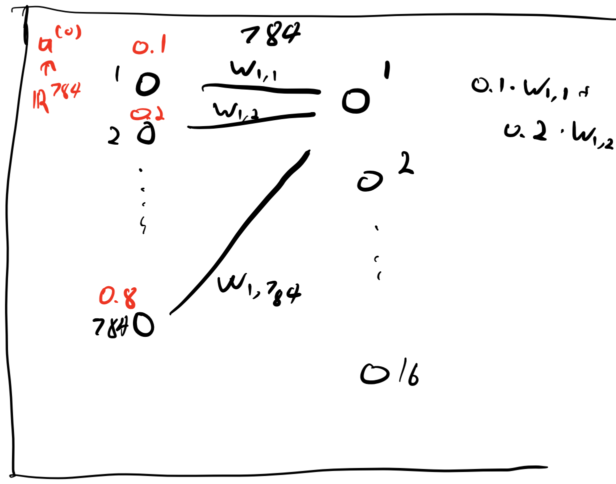
$\updownarrow$

$\mathbb{R}^{784}$





(multilayer perceptron)



$0.1 \cdot w_{1,1} + 0.2 \cdot w_{1,2} + \cdots + 0.8 \cdot w_{1,784}$

First layer: 784 neurons.

input space: $[0,1]^{784}$

Final layer: 10 nevrons

2 inner layers, 16 nevrons each.

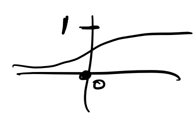
Layer 0 Layer 1

$$\mathbb{R}^{784} \longrightarrow \mathbb{R}^{16}$$

$$a^{(0)} \longmapsto \sigma(W^{(1)}a^{(0)} + b^{(1)})$$

or ReLu

$$\underbrace{\sigma}_{\text{Sigmoid.}}(x) = \frac{1}{1+e^{-x}}$$



$$W^{(1)} = \begin{pmatrix} w_{1,1} & \cdots & w_{1,784} \\ w_{2,1} & \cdots & w_{2,784} \\ & \vdots & \\ w_{16,1} & \cdots & w_{16,784} \end{pmatrix} \qquad b^{(1)} = \begin{pmatrix} b_1 \\ \vdots \\ b_{16} \end{pmatrix} \qquad a^{(0)} \begin{pmatrix} a_1^{(0)} \\ \vdots \\ a_{784}^{(0)} \end{pmatrix}$$

$W_1 \in M_{16\times 784}(\mathbb{R})$ weight matrix.

$b_1 \in M_{16\times 16}(\mathbb{R})$ bias matrix.

So $a^{(1)} = \sigma(W^{(1)}a^{(0)} + b^{(1)})$

$$a_i^{(1)} = \sigma\left(\left(\sum_{j=0}^{784} w_{i,j}a_j^{(0)}\right) + b_i^{(1)}\right)$$

Layer 0 Layer 1 Layer 2 Layer 3

$$\begin{tikzcd}
\mathbb{R}^{784} \arrow[r, "A_1"] \arrow[rrr, bend left, "A"] & \mathbb{R}^{16} \arrow[r, "A_2"] & \mathbb{R}^{16} \arrow[r, "A_3"] & \mathbb{R}^{10}
\end{tikzcd}$$

$$\sigma(W^{(1)}\cdot + b^{(1)}) \qquad \sigma(W^{(2)}\cdot + b^{(2)}) \qquad \sigma(W^{(3)}\cdot + b^{(3)})$$

$$16\times 784 \quad 16\times 1 \qquad 16\times 16 \quad 16\times 1 \qquad 10\times 16 \quad 10\times 1$$

Overall activation

$$A = \sigma\left(W^{(3)}\cdot\sigma\left(W^{(2)}\cdot\sigma\left(W^{(1)}(\bullet) + b^{(1)}\right) + b^{(2)}\right) + b^{(3)}\right)$$

$$A_3 \circ A_2 \circ A_1(\bullet)$$

$\boxed{\text{activation function:}}$

$$A : \mathbb{R}^{784} \longrightarrow \mathbb{R}^{10} \qquad A = A_{W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}, W^{(3)}, b^{(3)}}$$

(notation: $A_{\underline{W}, \underline{b}}$)

Choices		# parameters:	Choices	# parameters
$W^{(1)}$	$\in M_{784\times 16}(\mathbb{R})$	$784\times 16 = 12{,}544$	$b^{(1)} \in \mathbb{R}^{16}$	16
$W^{(2)}$	$\in M_{16\times 16}(\mathbb{R})$	$16\times 16 = 256$	$b^{(2)} \in \mathbb{R}^{16}$	16
$W^{(3)}$	$\in M_{16\times 10}(\mathbb{R})$	$16\times 10 = 160$	$b^{(3)} \in \mathbb{R}^{10}$	10

total # parameters: 13,002.

Training:

Cost function built from a training set.

Collection of pairs

$$TS = \{ (I_1, L_1), (I_2, L_2), \ldots, (I_n, L_n) \}$$

where $I_k \in \mathbb{R}^{784}$ is an input, $L_i \in \mathbb{R}^{10}$ is the correct output.

$$C = C_{TS} : \mathbb{R}^{13,002} \longrightarrow \mathbb{R},$$

really

$$C : M_{784\times 16} \times \mathbb{R}^{16} \times M_{16\times 16} \times \mathbb{R}^{16} \times M_{16\times 10} \times \mathbb{R}^{10} \longrightarrow \mathbb{R}$$

$$(W^{(1)}, b^{(1)}, W^{(2)}, b^{(2)}, W^{(3)}, b^{(3)}) \longmapsto \underbrace{\frac{1}{n} \sum_{i=1}^{n}}_{\text{average over all images in the training data.}} \underbrace{\| A_{\underline{W}, \underline{b}}(I_i) - L_i \|^2}_{\text{distance of the guess } (A_{\underline{W},\underline{b}}(I_i)) \text{ from the answer } L_i}$$

Intuition of Cost function :

$C(\underline{W}, \underline{b})$ is small $\Longleftrightarrow$ the weights $\underline{W}, \underline{b}$ do a good job guessing the correct answer.

$\underline{W} = W^{(1)}, W^{(2)}, W^{(3)}$, $\quad \underline{b} = b^{(1)}, b^{(2)}, b^{(3)}$

$\Longrightarrow$ $\boxed{\text{Goal}}$ should be to finds weights $\underline{W}, \underline{b}$ minimizing C.

Gradient Descent

Idea: ① Start with Random inputs $(\underline{W}_0, \underline{b}_0)$ to C.

② Find the direction of $(\underline{W}_0, \underline{b}_0)$ that leads the the lages decrease in C.

③ Change $(\underline{W}_0, \underline{b}_0)$ to $(\underline{W}_1, \underline{b}_1)$ according to the direction form ②.

More precisely:

For 2, use $-\nabla C(\underline{w}, \underline{b})$.

$$C: \mathbb{R}^n \to \mathbb{R}, \qquad \nabla C: \mathbb{R}^n \to \mathbb{R}$$

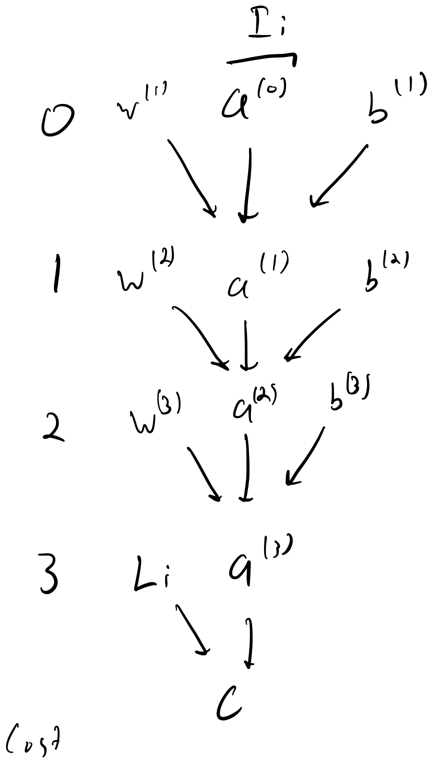
$$\nabla C = \begin{pmatrix} \frac{\partial C}{\partial x_1} \\ \vdots \\ \frac{\partial C}{\partial x_n} \end{pmatrix}$$

$$\left(\text{e.g.} \quad \begin{array}{l} C: \mathbb{R}^2 \to \mathbb{R} \\ (x,y) \mapsto x^2 y + y \end{array}, \quad \nabla C = \begin{pmatrix} \frac{\partial C}{\partial x} \\ \frac{\partial C}{\partial y} \end{pmatrix} = \begin{pmatrix} 2xy \\ x^2+1 \end{pmatrix} \right)$$

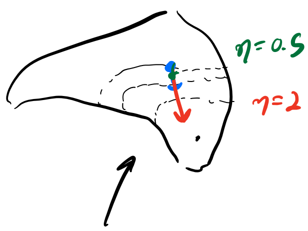
. . . . - - -

☆ To do: details for Gradient descent ☆

$$= \frac{1}{n} \sum_{i=1}^{n} \| \underbrace{A_3 A_2 A_1 (a^{(0)})}_{a^L} - L$$

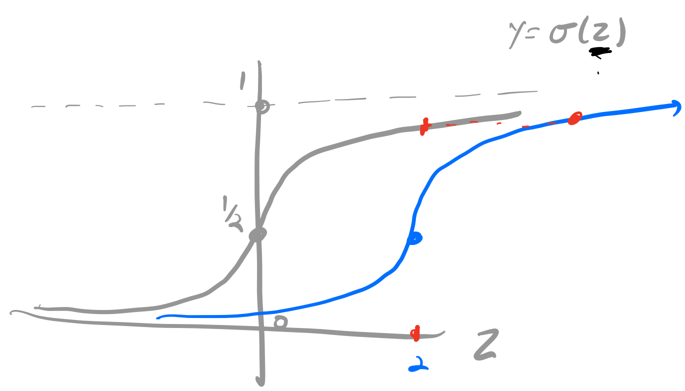


$$\boxed{\text{July } 3^{rd}}$$



η

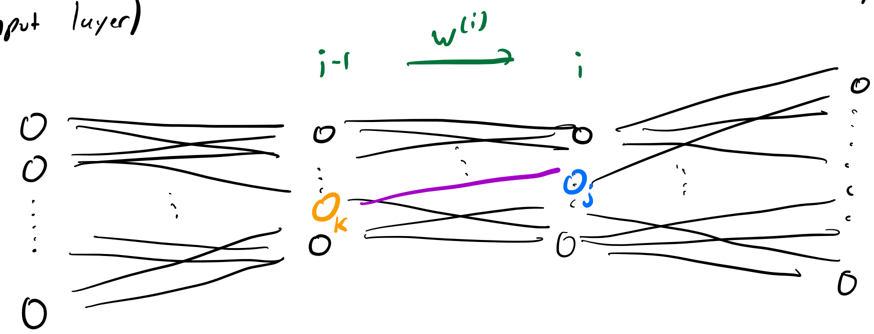
$$\underline{W}_0 - \underline{\eta}\, \nabla \mathcal{L}(\underline{w}_0)$$



$\sigma(z-2)$

Recalling activation

Layer 0 (input layer) | Layer 1 | Layer 2 | Layer 3 (output layer)



# neurons	784	16	16	10

Notation: $w_{jk}^{(i)}$

is weight by which neuron k in layer $i-1$ influences neuron j in layer i

activations:

From Layer 0 to layer 1:

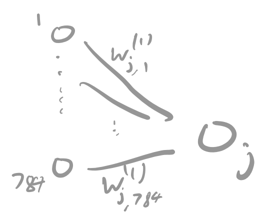
initial activation: $a^{(0)} = \begin{pmatrix} a_1^{(0)} \\ \vdots \\ a_{784}^{(0)} \end{pmatrix} \in \mathbb{R}^{784}$.

(input to layer zero)

activation of j^{th} neuron in 1^{st} layer: $(j = 1, \dots, 16)$

$$a_j^{(1)} = \sigma\Big(\underbrace{\Big(\sum_{k=1}^{784} w_{jk}^{(1)} a_k^{(0)}\Big) + b_j^{(1)}}\Big)$$

define: $z_j^{(1)} =$



Keep track of all neurons at once using vectors:

$$a^{(1)} := \begin{pmatrix} a_1^{(1)} \\ \vdots \\ a_{16}^{(1)} \end{pmatrix} \in \mathbb{R}^{16} \quad \& \quad z^{(1)} := \begin{pmatrix} z_1^{(1)} \\ \vdots \\ z_{16}^{(1)} \end{pmatrix} \in \mathbb{R}^{16}$$

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

j,kth entry $W_{jk}^{(1)}$

All together: $\underbrace{a^{(1)}}_{\substack{16\times 1\\ \text{vector}}} = \sigma(z^{(1)})$, $\underbrace{z^{(1)}}_{\substack{16\times 1\\ \text{vector}}} = \underbrace{W^{(1)}}_{\substack{16\times 784\\ \text{matrix}}} \cdot \underbrace{a^{(0)}}_{\substack{784\times 1\\ \text{vector}}} + \underbrace{b^{(1)}}_{\substack{16\times 1\\ \text{vector}}}$

Layer 1 to layer 2

$$\underbrace{a^{(2)}}_{16\times 1} = \sigma(z^{(2)}), \quad \underbrace{z^{(2)}}_{16\times 1} = \underbrace{W^{(2)}}_{16\times 16} \cdot \underbrace{a^{(1)}}_{16\times 1} + \underbrace{b^{(2)}}_{16\times 1}$$

From layer 2 to layer 3

$$\underbrace{a^{(3)}}_{10\times 1} = \sigma(z^{(3)}), \quad \underbrace{z^{(3)}}_{10\times 1} = \underbrace{W^{(3)}}_{10\times 16} \cdot \underbrace{a^{(2)}}_{16\times 1} + \underbrace{b^{(3)}}_{10\times 1}$$

Final cost:

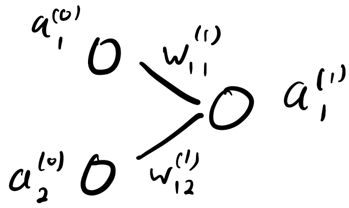
$$C = \frac{1}{\#\, x \text{ in training data}} \sum_{x \text{ in training data}} C_x,$$

$x \in \mathbb{R}^{784}$

$$C_x = \| \underbrace{\overbrace{A_{\underline{w},\underline{b}}(x)}^{10\times 1}}_{\text{guess based on given weights \& biases.}} - \overbrace{\text{actual value}(x)}^{10\times 1} \|^2$$

need to find $\nabla C = \frac{1}{\#\cdots} \nabla \cdot C_x$

Simple example:



$$a^{(0)} = \begin{pmatrix} a_1^{(0)} \\ a_2^{(0)} \end{pmatrix}, \qquad a^{(1)} = \sigma(z^{(1)}),$$

$$z^{(1)} = \underbrace{\begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} \end{pmatrix}}_{1\times 2} \underbrace{\begin{pmatrix} a_1^{(0)} \\ a_2^{(0)} \end{pmatrix}}_{2\times 1} + \underbrace{b_1^{(1)}}_{1\times 1}$$

$$= w_{11}^{(1)} a_1^{(0)} + w_{12}^{(1)} a_2^{(0)} + b_1^{(1)}$$

variables in cost function

Exercise: If $w^{(1)} = \begin{pmatrix} 2 & \ln(2) \end{pmatrix}$ and $a^{(0)} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, & $b^{(1)} = 0$,

find the activation of the neuron in the output layer.

Suppose our training data is one single data point:

$$\left(\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \quad 1 \right)$$

(This means the input of $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ to the first layer should output the "vector" 1 (rember there is only one neuron).

Cost function:

$$\| \sigma (w_{11}^{(1)} \cdot 1 + w_{12}^{(1)} \cdot 2 + b_1^{(1)}) - 1 \|^2$$

$w_1 = w_{11}^{(1)}$

$w_2 = w_{12}^{(1)}$

$b = b_1^{(1)}$

Cost function

$$C(w_1, w_2, b) := \left(\sigma(w_1 + 2w_2 - b) - 1 \right)^2$$

Exercise: Find the Gradient of the cost function.

$$\frac{\partial C}{\partial w_1} = 2\left(\sigma(w_1 + 2w_2 - b) - 1\right) \cdot \sigma'(w_1 + 2w_2 - b) \cdot \underbrace{1}_{\frac{\partial w_1}{\partial w_1} = 1}$$

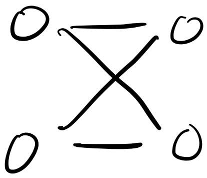
$$\frac{\partial C}{\partial w_2} = 2\left(\sigma(w_1 + 2w_2 - b) - 1\right) \cdot \sigma'(w_1 + 2w_2 - b) \cdot \underbrace{2}_{\frac{\partial (w_1 + 2w_2 - b)}{\partial w_2} = 2.}$$

$$\frac{\partial C}{\partial b} = 2\left(\sigma(w_1 + 2w_2 - b - 1\right) \cdot \sigma'(w_1 + 2w_2 - b) \cdot (-1)$$

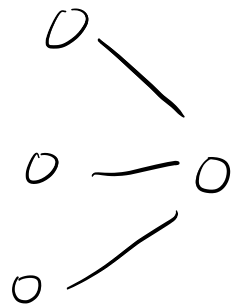
$$\nabla C = \begin{pmatrix} \frac{\partial C}{\partial w_1} \\ \frac{\partial C}{\partial w_2} \\ \partial C / \partial b \end{pmatrix}$$

Exercise: Go through chapter 2 in the book, and for all the formulas displayed there, write what they are in this simple case.

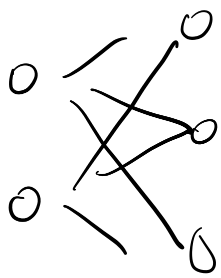
Exercise: Do the same for other simple neural networks, e.g.



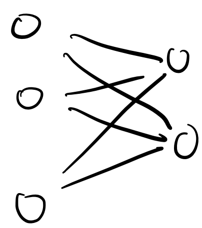
or



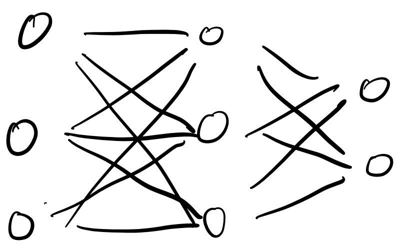
or



or



or



"**Assignment**:"

- **Reading**: Ch1 & Ch2 of the book.

  (More theoretical)
  math!

- **Programming**: (Recommended as group work!)
  - Run the code in the book for MNIST digits
    (experiment with using different structures of neural networks, e.g. of shape [784, 16, 16, 10])
  - Run the code in the book for USPS digits.
    experiment to find good parameters.
  - If there is some other data set you want to analyze, go for it!

Next meeting w/ me: July 17 – 21

(in person or zoom, ..)

Students only meeting next week encouraged!