Neural network

what are the neurons?

how are they connected?

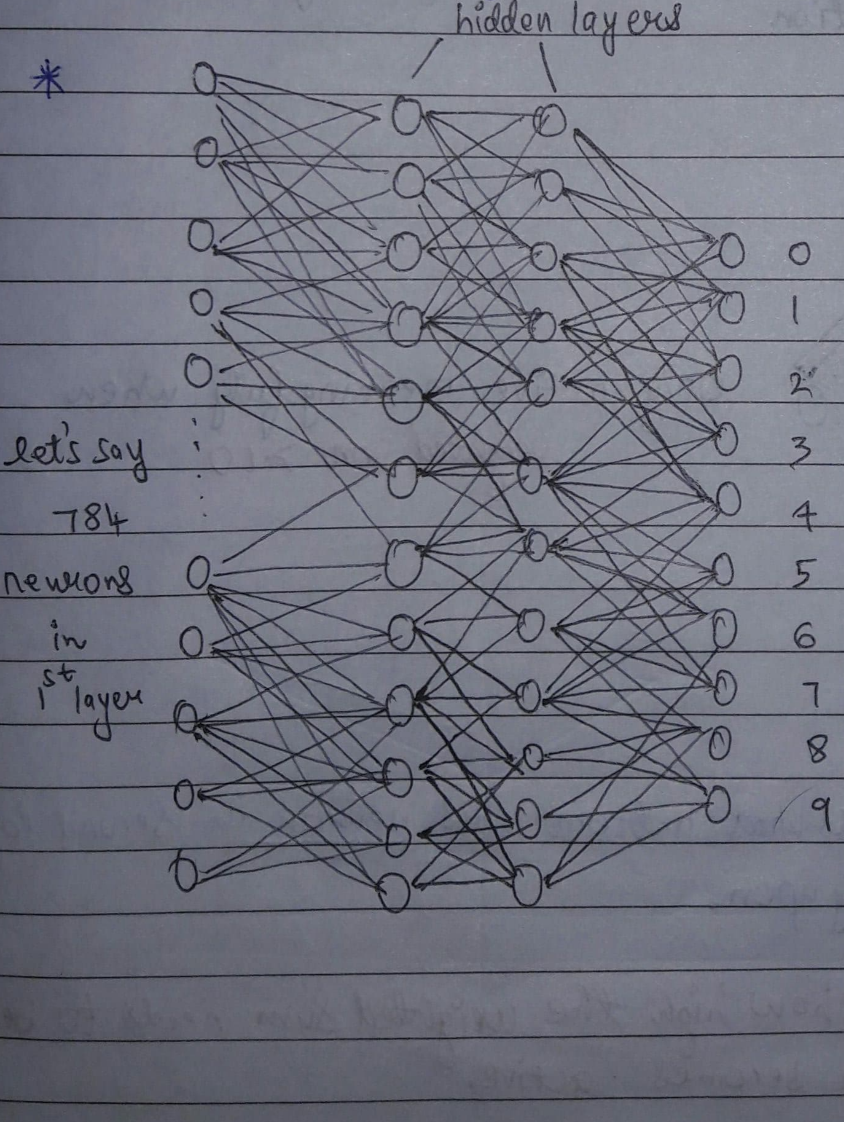
a → Thing that holds a number
Generally betn 0 & 1

black pixel

white pixel

number inside neuron is called activation & represent grey scale value of pixel.

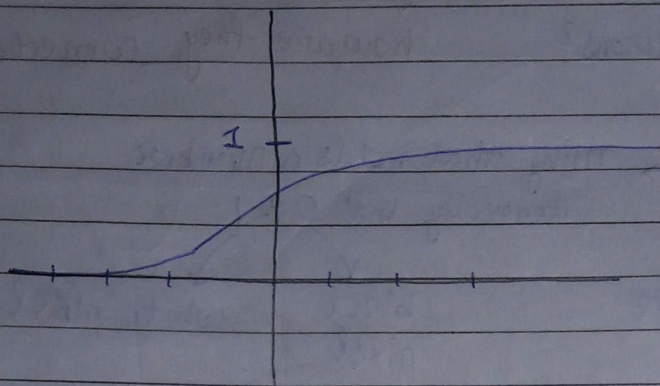
*



* Sigmoid

$$a(x) = \frac{1}{1+e^{-x}}$$

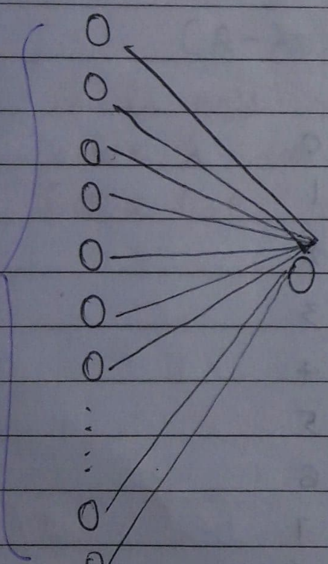
[we use this function as we need range of 0 to 1 for activation values]



$$a(w_1 a_1 + w_2 a_2 + w_3 a_3 + \cdots + w_n a_n \overset{\text{bias}}{-10} \cancel{+\ \text{bias}})$$

here $w_1, w_2 \cdots$ are weights of neurons, a is activation

784



only active meaningfully when weighted sum > 10

weights tells what pattern this neuron in second layer is picking upon.

Bias tells how high the weighted sum needs to be before neuron becomes active.

This is for one neuron.

1st & 2nd layer   2nd & 3rd   3rd & 4th

$$784 \times 16 + 16 \times 16 + 16 \times 10$$
weights

$$16 + 16 + 10$$
biases

Total 13,002 weights & biases.

$$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & & & \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}\right)$$

Each row represent connection betn one layer & particular neuron in next layer.

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

1st layer

Initial layer

Cost = $(0.43 - 0)^2 + (0.28 - 0)^2 + (0.8 - 1)^2 + \ldots$ actual values

(Should be as small as possible)

wrong values that we get

we consider average cost of all data's.

* Neural network function :-

Input :- 784 no's   Parameters :- 13,002 weights / biases
output :- 10 no's

Cost function

Input :- 13,002 weights / biases
output :- 1 no
Parameters :- multiple training examples.

"Gradient", the direction of steepest increase $\nabla C(x,y)$

$$\vec{W} = \begin{bmatrix} 2.43 \\ -1.12 \\ 1.47 \\ \vdots \\ 1.21 \end{bmatrix} \qquad -\nabla C(\vec{W}) = \begin{bmatrix} 0.18 \\ 0.45 \\ -0.51 \\ \vdots \\ -0.32 \\ 0.82 \end{bmatrix}$$

13,002 weights & biase

(This tells which changes to which weight matters most.)

How to nudge all weights & biases

$$a = \sigma(w_0 a_0 + w_1 a_1 + \cdots w_{n-1} a_{n-1} + b)$$

let's say
$a = 0.2$, To increase a,

i) increase b
ii) Increase w_i in proportion to a_i
[For more active neurons increase in w will have higher effect]

iii) Change a_i [you cannot change a_i]
[in proportion to w_i]

Now, when you have desired effects that you want bet^n last & 2^nd last layer, Same you can do for other previous layers, this is back propogation.

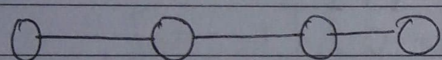
→ you do this for all training data,

	2	5	0	4	.....	Average
w_0	-0.08	0.02	-0.02	0.11		-0.08
w_1	-0.11	0.11	0.07	0.05		0.04
w_2	-0.04	0.02	0.05	0.72		0.34
⋮						
w_n	0.13	0.08	-0.06	0.09		0.76

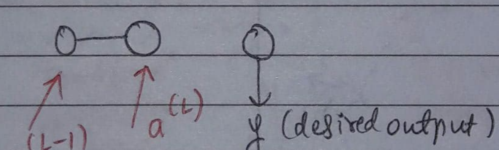
This collection is

$$-\nabla C(w_1, w_2, \cdots) = \begin{bmatrix} -0.08 \\ 0.04 \\ \cdots \\ 0.76 \end{bmatrix}$$

* Consider, 4 neurons



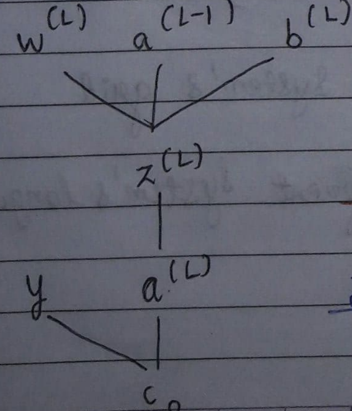
For now,
consider
last 2,



cost → $C_0 = (a^{(L)} - y)^2$

$a^{(L)} = w^{(L)} \cdot a^{(L-1)} + b^{(L)} = z^{(L)}$

$a^{(L)} = a(z^L)$



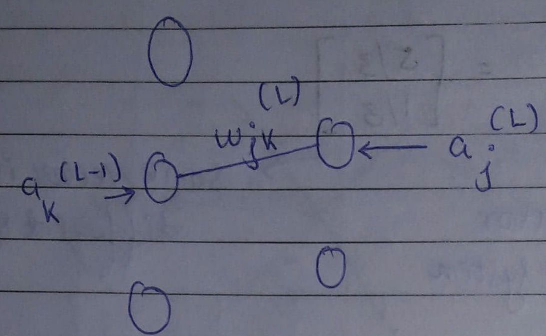
$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial z^{(L)}}{\partial w^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdot \frac{\partial C_0}{\partial a^{(L)}} = 2(a^{(L)} - y) \cdot a'(z^L) \cdot a^{(L-1)}$$

$$\frac{\partial C_0}{\partial a^{(L-1)}} = \frac{\partial z^{(L)}}{\partial a^{(L-1)}} \cdot \frac{\partial a^{(L)}}{\partial z^{(L)}} \cdot \frac{\partial C_0}{\partial a^{(L)}} = w^{(L)} \cdot a'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

$$\frac{\partial C_0}{\partial b^{(L)}} = 1 \cdot a'(z^{(L)}) \cdot 2(a^{(L)} - y)$$

Continued.
neural
networks

Repeat this backward to see how sensitive cost f^n is to previous weights, biases, activations.



$$C_0 = \sum_{j=0}^{n_L-1} (a_j^{(L)} - y_j)^2$$

Same

$$a_j^{(L)} = \sigma(z_j^{(L)})$$