Chapter 1: Neural Network.

5. What are neural networks?

How do we (our brain) recognize hand-written digits without any effort that is written in many different ways?

ex.) 3, 3, 3 ⇒ recognized as 3.

Neural. network

1. What are the neurons?
2. How are they connected?

(Function)

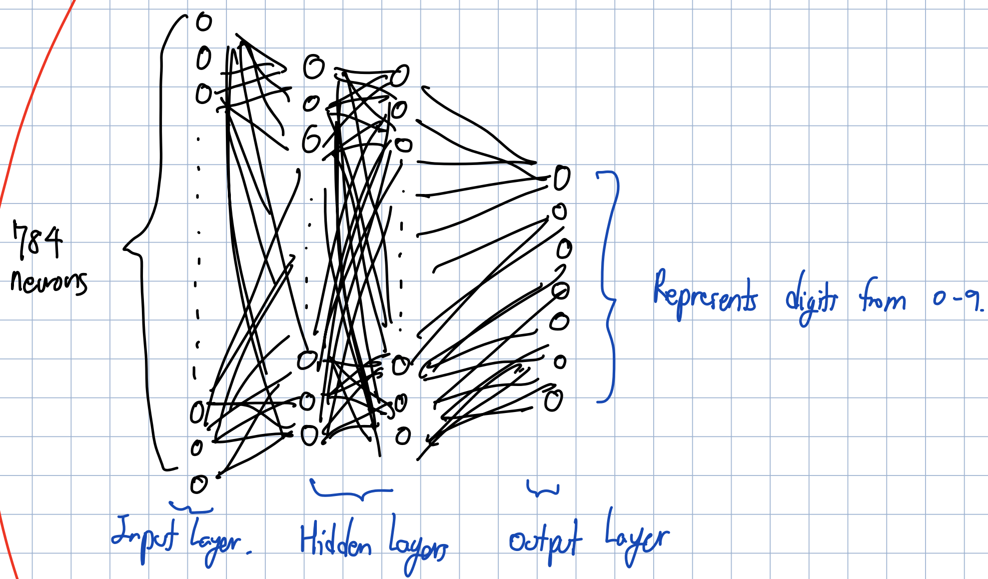
1. Neuron : Thing that holds a number

- 28 x 28 pixel = 784 neurons.
  - each pixel (neurons) holds a grey scaled values of the corresponding pixel ranging from 0-1.
    (black) (white)
  - Activation : values inside the neurons.
    - high activation = closer to 1 ⇒ whiter.
    - low activation : closer to 0 ⇒ darker.
  - 784 neurons make up the first layer
- last layer has 10 neurons. representing one of the digit. from 0-9.
  - the activation in the last layer corresponds how close the machine thinks the given digit is to the actual digit.
- "Hidden layer": layers between first and last layer.

ex.) 2 Hidden Layers : each with 16 neurons (arbitrary choice)

based on random choice or personal preference

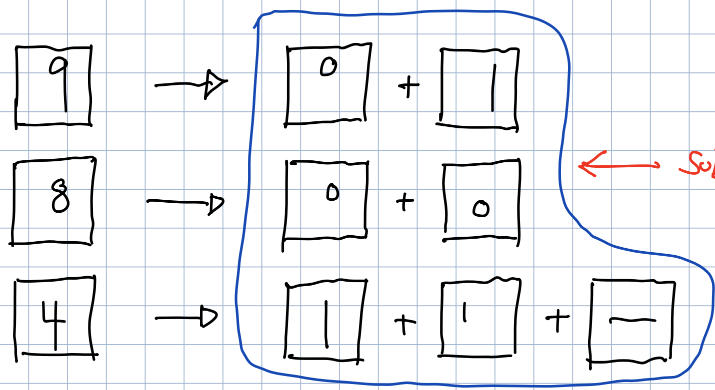
- activation in one layer determines the activation of the next layer.



If you feed in the Input layer, lighting up all 784 neurons with accordance of each of the brightness of the neurons, this pattern of activation causes very specific patterns in the next layer. ⟶ continuously until the output layer. The brightest neuron in the output layer is the network's choice for what image this represents.

* So why layered structure?

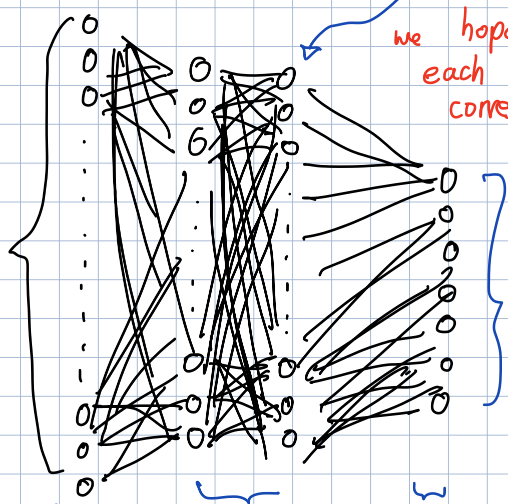
ex.) Think of digits.



← Subcomponent.

we hope that (perfect world) each neuron in the second to last layer corresponds with one of the sub components.

784 neurons



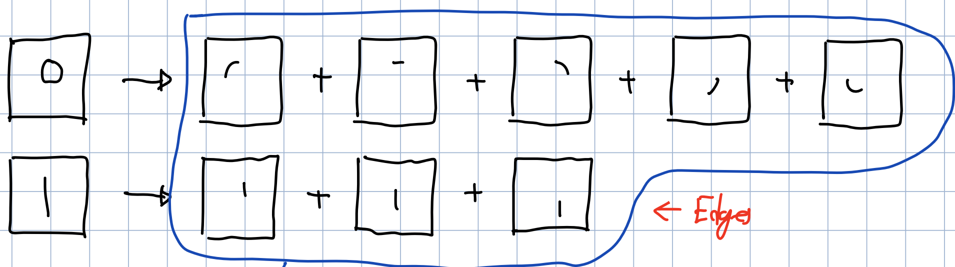
Represents digits from 0-9.

Input Layer. Hidden layers Output Layer

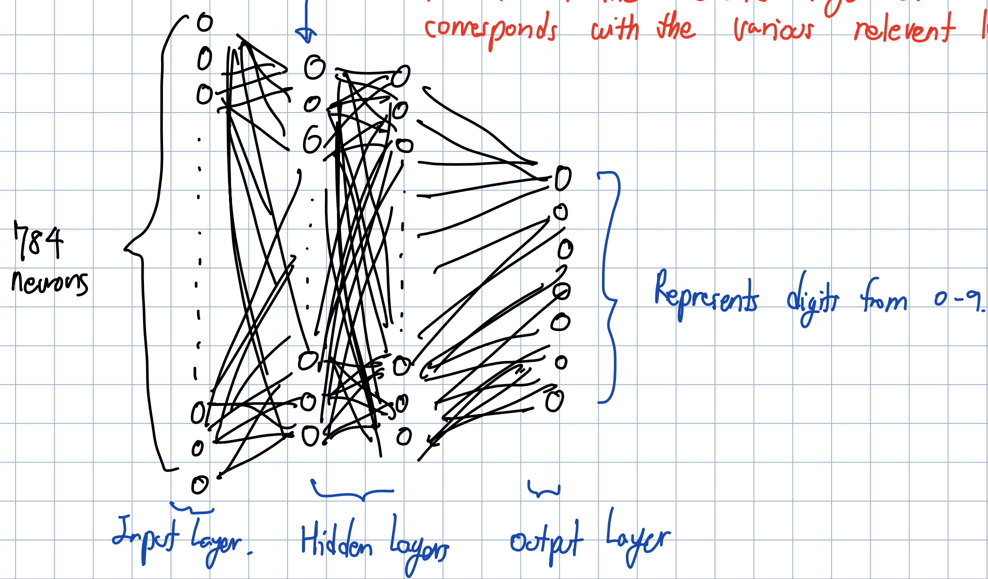
- In the output layer only require which combination of each subcomponents corresponds to which digits.

* But how do we recognize each subcomponents?

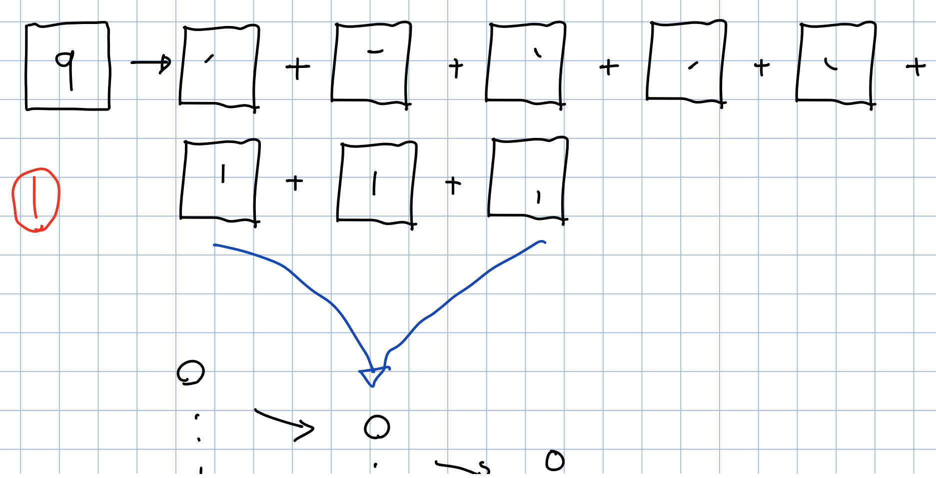
ex.) Edges of each subcomponent.

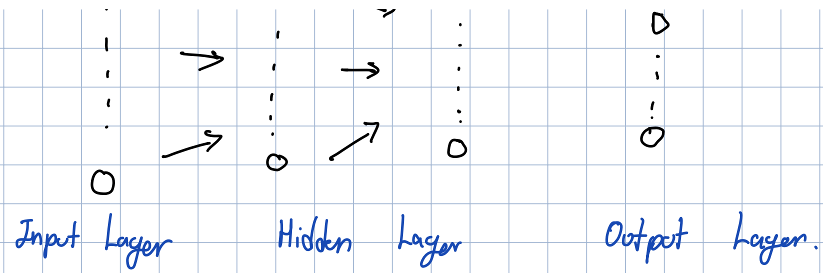


Our hope (perfect world) is that each neuron in the second layer of the network corresponds with the various relevent little edges.

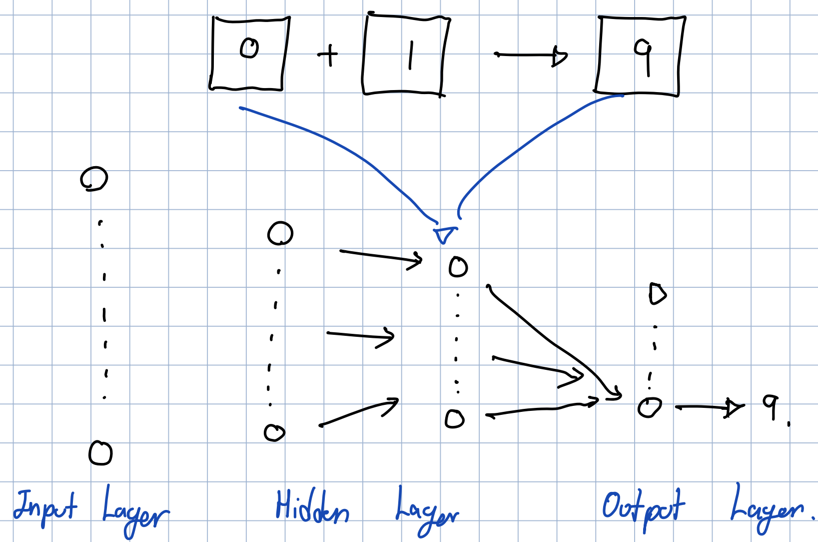


<u>Overall.</u>



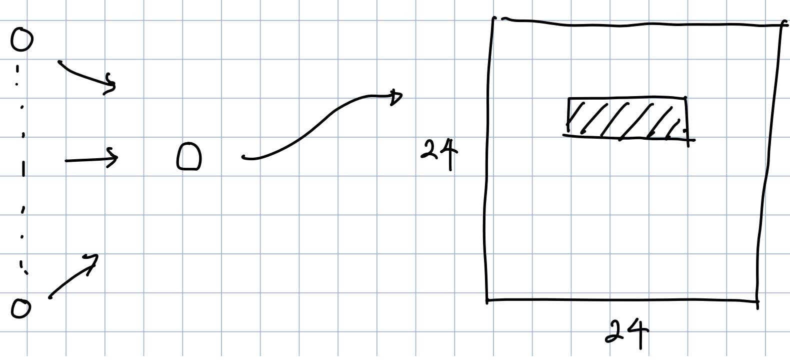


2



★ So how is it actually connected?

from 784 neurons



Q.) What Parameter should exist?

$p_1 : 0.00 \sim 1.00$
$p_2 : 0.00 \sim 1.00$
$\vdots$
$p_x : 0.00 \sim 1.00$

} so that it is expressive enough to potentially capture any pattern given in the image?

↳ We assign weights

w_1 : any number
w_2
$\vdots$
w_x :

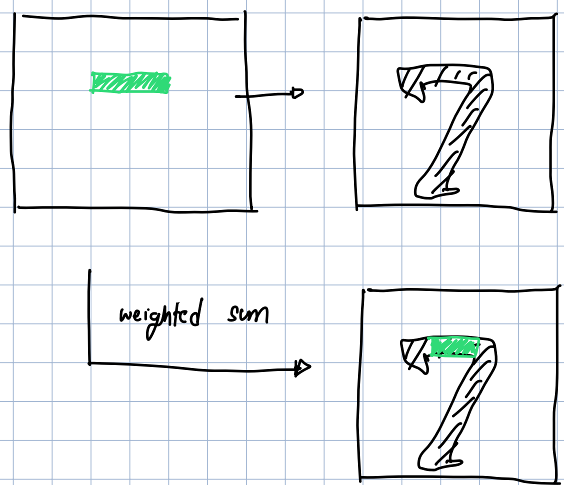
} could be positive or negative.

$\therefore w_1 a_1 + w_2 a_2 + w_3 a_3 + \dots + w_n a_n$

w = weights
a = activation.

Think of Grids. : make all the pixels value as zero, except for some positive weights in the region we care about

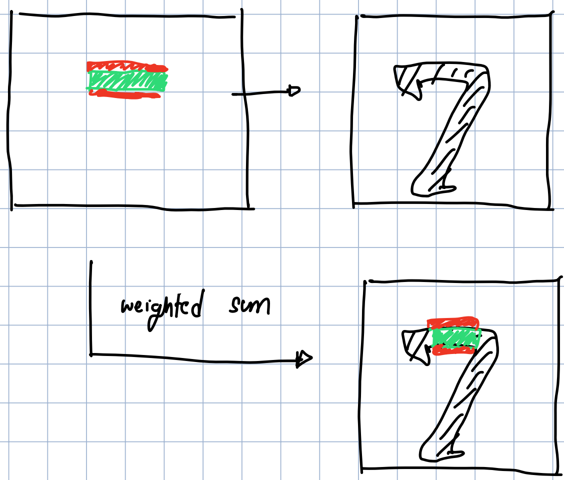
● positive weights.



and then taking the weighted sum of the pixel values amounts to adding up the values of the pixel just in the region we care about.

How to find edges?

- negative weights.

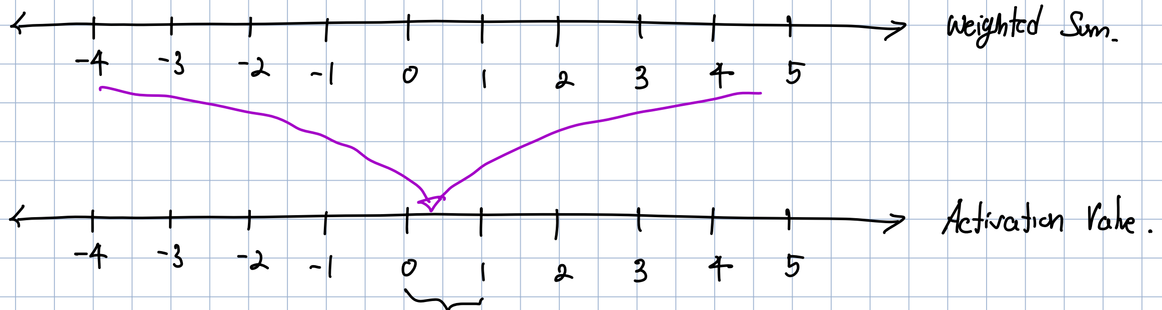


recall : high activation : closer to 1 (brightest)
low activation : closer to 0 (darkest)

Sum is largest (brightest) where the middle pixels are bright, but the surrounding pixels are darker.

$w_1 a_1 + w_2 a_2 + \dots + w_n a_n$

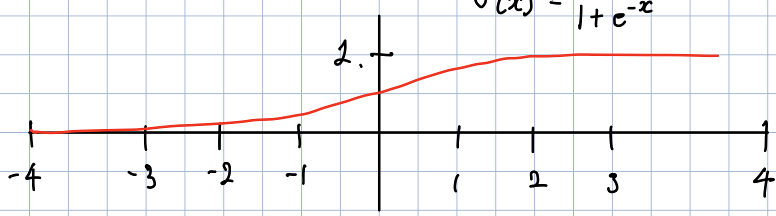
could be any value.



Activation should be within 0~1 range.

How? Sigmoid. aka : logistic (curve) regression

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



Using sigmoid : very negative → closer to 0 (low activation) darker.

very positive → closer to 1 (high activation) bright.

Sigmoid

How positive?

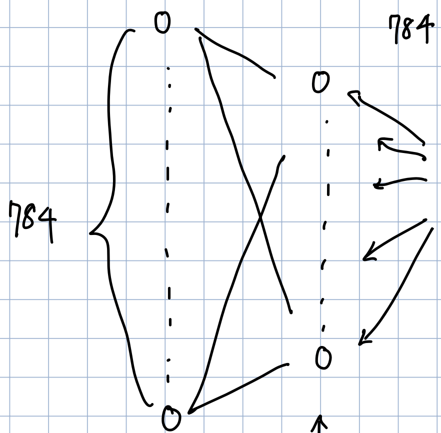
$$\therefore \quad \sigma(w_1 a_1 + w_2 a_2 + \dots + w_n a_n)$$

But! maybe you want the neuron to light up not when it is above 0

maybe you only want it to be active when > 10

Bias for inactivity.

$$\therefore \quad \sigma(w_1 a_1 + w_2 a_2 + \dots + w_n a_n \underbrace{-10}_{\text{"bias"}})$$



784 weights per neuron.

One bias for each

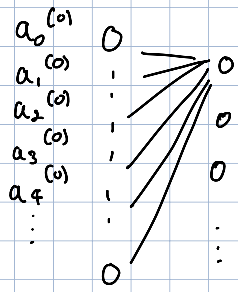
For a hidden layer with 16 neurons

→ 784 neurons × 16 weights & weights.

So! Input Layer → 2nd Layer.

$$\boxed{\begin{array}{l} 784 \times 16 + 16 \times 16 + 16 \times 10 \\ \text{weights.} \\ 16 + 16 + 10 \\ \text{bias.} \end{array}} = 13{,}002 \text{ total weights \& bias.}$$

Learing → Finding the right weights & biases.



$$\to a_0^{(1)} = \sigma\left(w_{0,0}\, a_0^{(0)} + w_{0,1}\, a_1^{(0)} + \dots + w_{0,n}\, a_n^{(0)} + b_0\right)$$

Watch Linear Algebra (3Blue1Brown)

$\Downarrow$ Notation

$$\underset{\text{Sigmoid}}{\sigma}\left(\underbrace{\begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,n} \\ w_{1,0} & w_{1,1} & w_{1,n} \\ \vdots & \vdots & \vdots \\ w_{k,0} & w_{k,1} & w_{k,n} \end{bmatrix}}_{\text{weights}} \underbrace{\begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix}}_{\text{Activation}} + \underbrace{\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_k \end{bmatrix}}_{\text{bias.}} \right)$$

Neuron = Function that holds a number

Network = connections of Functions.

$$f(a_0, \dots, a_{783}) = \begin{bmatrix} y_0 \\ \vdots \\ y_9 \end{bmatrix}$$