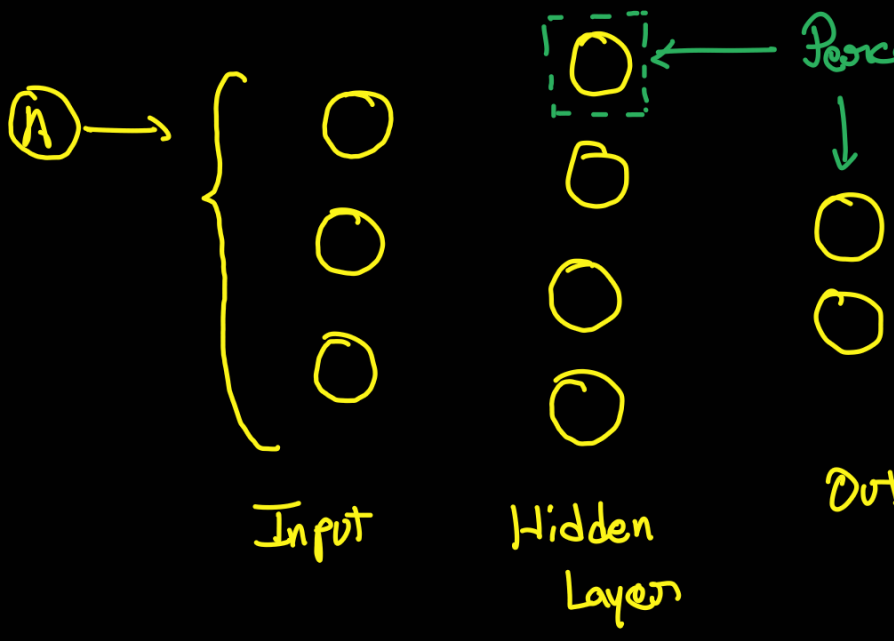
→ Image Processing

But what are neural networks?



→ There are different types of neural networks:-

① CNN ⟶ Convolutional Neural Network.

② LSTM ⟶ Long-Short Term memory.

---

a ⟶ (a = activation)

In image Processing, lets treat each pixel as the neuron with a certain pixel value (ranging b/w 0 to 1)

0 → black
1 → white

→ Let's understand this with the help of an example.

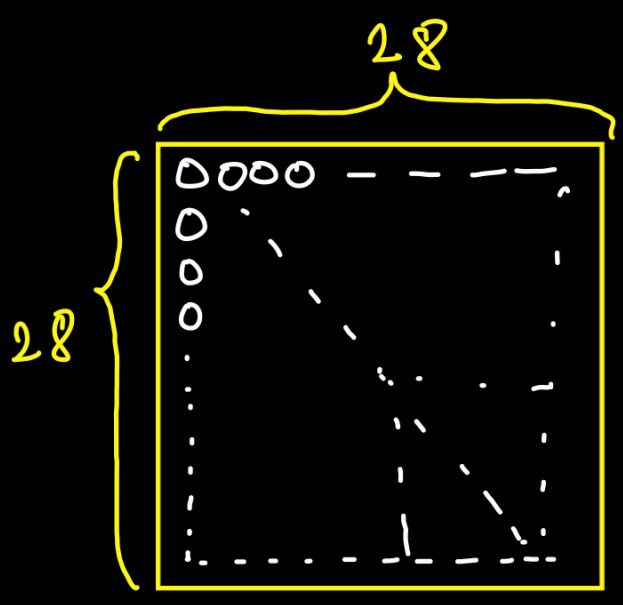
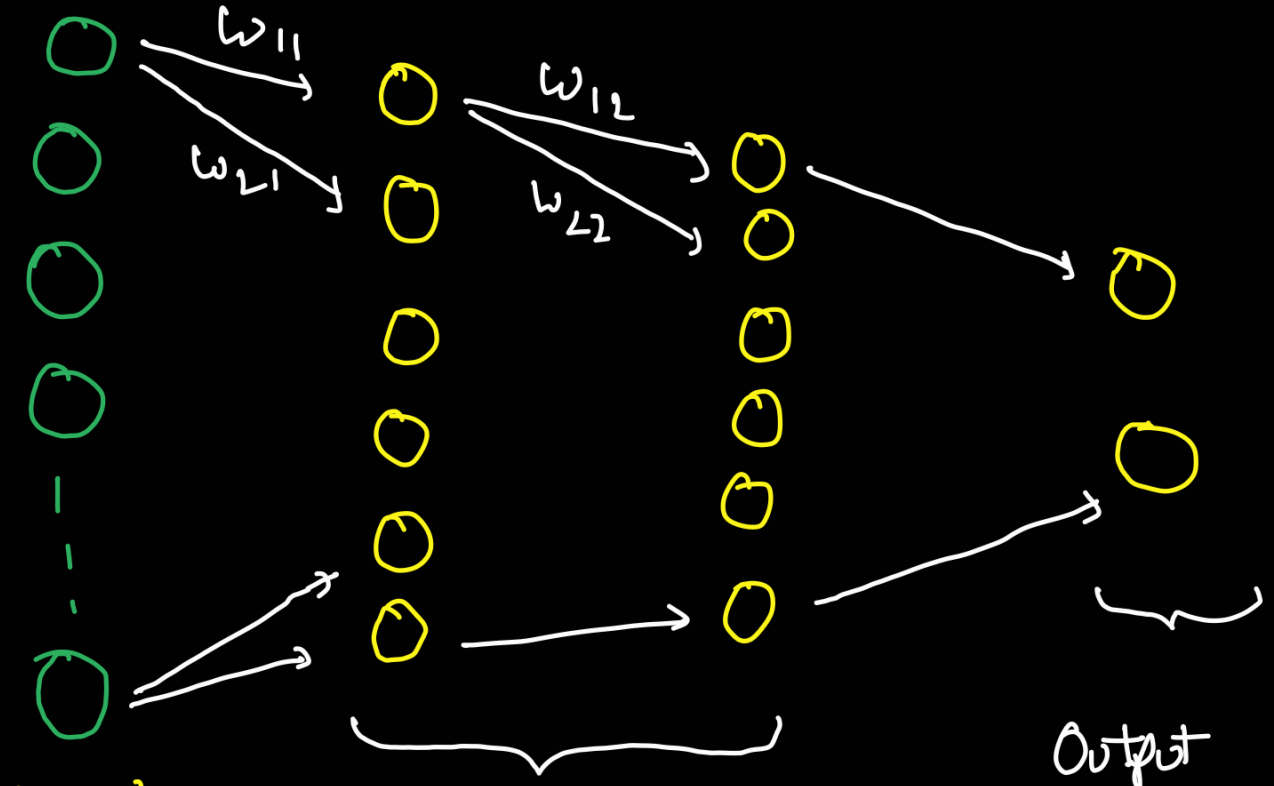


Image with 784 pixels having a certain activation.

Flatten

? 28×28

This is the input layer.



Input Layer

Hidden layers

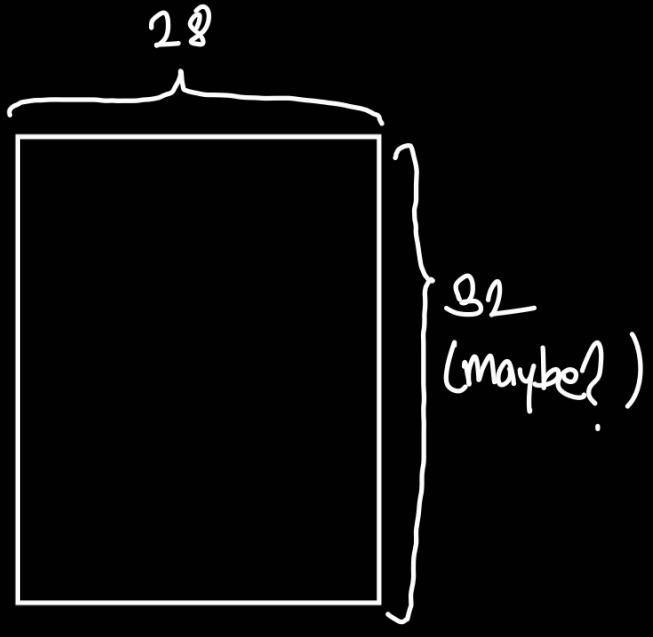
Output layer.

→ Each connection of a particular neuron in a layer to other has a certain weight attached to it which keeps on changing as the model trains.

→ But how?

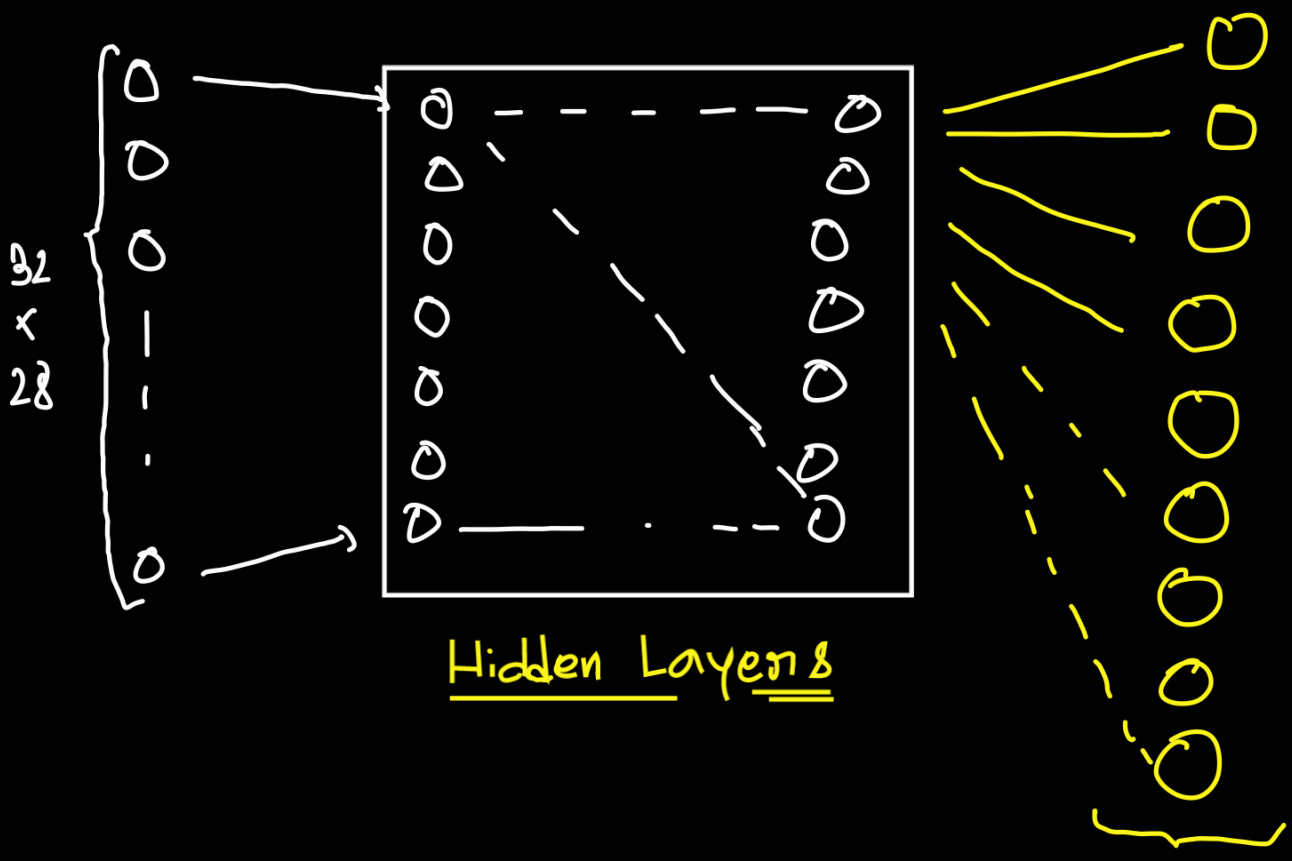
Remember Gradient Descent?

→ Assume the case of classification of hand-written digits.



This is the input image.

→ Let's flatten this!



→ Let's discuss about :-

① Edge Detection.
② Circle Detection

} Some similarities among different numbers which are filtered out through different layers.

* → 9 & 8 both have a circle shaped head hence, a layer might predict 8 & 9 as same, but adding another layer which checks for lower circle will Classify the numbers distinctly as 8 & 9.

→ Hmm Classification !

What is Classification ? Remember the initial class? (Predicts the outcome as finite distinct values unlike regression).

→ How does this work?

Remember the predicted value for linear regression?

$$y = mx + c$$

→ Let's assume the initial input pixels as features, (since the final prediction depends on this input).

Let the i^{th} pixel (neuron) in the j^{th} layer be :-

$a_{i,j}$, $w_{i,j}$ → Associated bias.

$$\therefore a_{i+1,j} = w_{0,j} a_{0,j} + w_{1,j} a_{1,j} - \cdots - w_{n,j} a_{n,j} + b$$

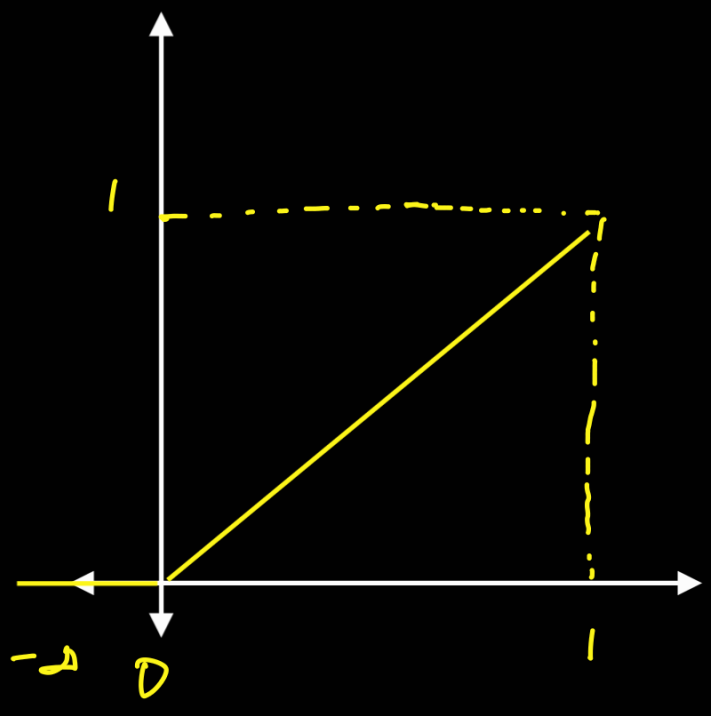
→ Right! How do we Classify?

→ There are different functions which help us, let us denote it as $\sigma(a_{ij})$

→ We set a certain value above which $\sigma(a_{ij})$ returns a "non-zero" value.

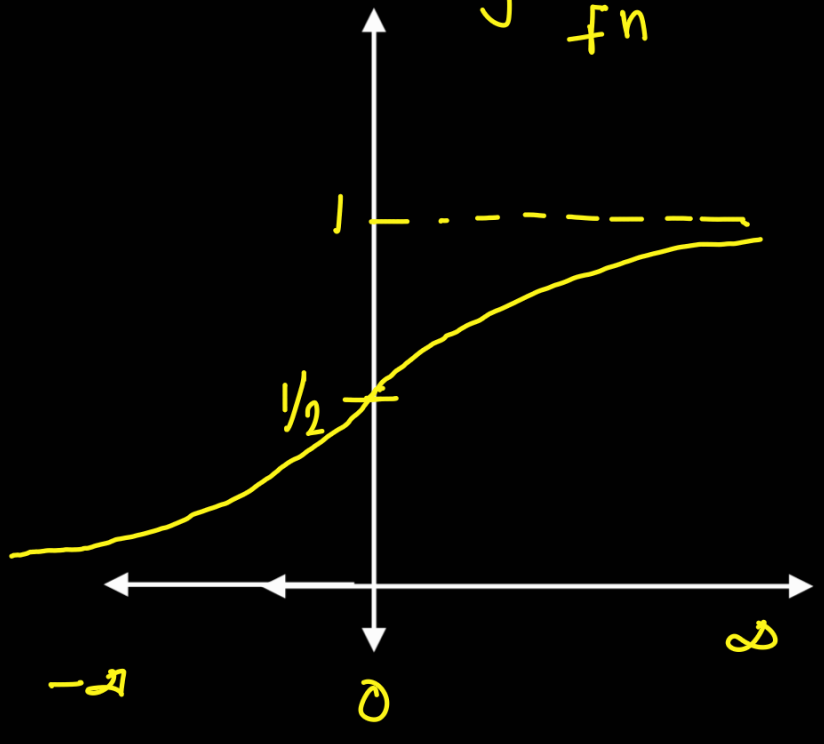
Examples :-

RELU



$\sigma(a_{i,j}) = \max(0, a_{ij})$

Sigmoid fn



$$\sigma(a_{i,j}) = \frac{1}{1+e^{-a_{ij}}}$$

→ Based on this classification technique & gradient descent learning algorithm, we classify things.

→ Summarising :-

$$\begin{bmatrix} a_{0,j+1} \\ \vdots \\ a_{n,j+1} \end{bmatrix} = \begin{bmatrix} w_{0,j,0} & \cdots & w_{n,j,0} \\ w_{0,j,1} & \cdots & w_{n,j,1} \\ \vdots & \ddots & \vdots \\ & \cdots & w_{n,j,n} \end{bmatrix} \begin{bmatrix} a_{0,j} \\ \vdots \\ a_{n,j} \end{bmatrix} + \begin{bmatrix} b_{0,j} \\ \vdots \\ b_{n,j} \end{bmatrix}$$

And values for the next layer are :-

$$\sigma\left(\begin{bmatrix} a_{0,j+1} \\ \vdots \\ a_{n,j+1} \end{bmatrix}\right)$$