(1) About Dataset.

* mNIST Dataset.
  Contain Image in (thousand).
  Low res - hand written digit ..

MATH

→ no. of Row
→ Row

784 ⇒ training pixel.
28 × 28

m training images.

784
28 × 28

Can Be Represented as a matrix.

→ transpose.

$$X = \begin{bmatrix} — & x^{(1)} & — \\ — & x^{(2)} & — \\ — & x^{(3)} & — \\ & \vdots & \\ — & x^{(n)} & — \end{bmatrix}^{T} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix}$$

↳ Row.

→ Goal

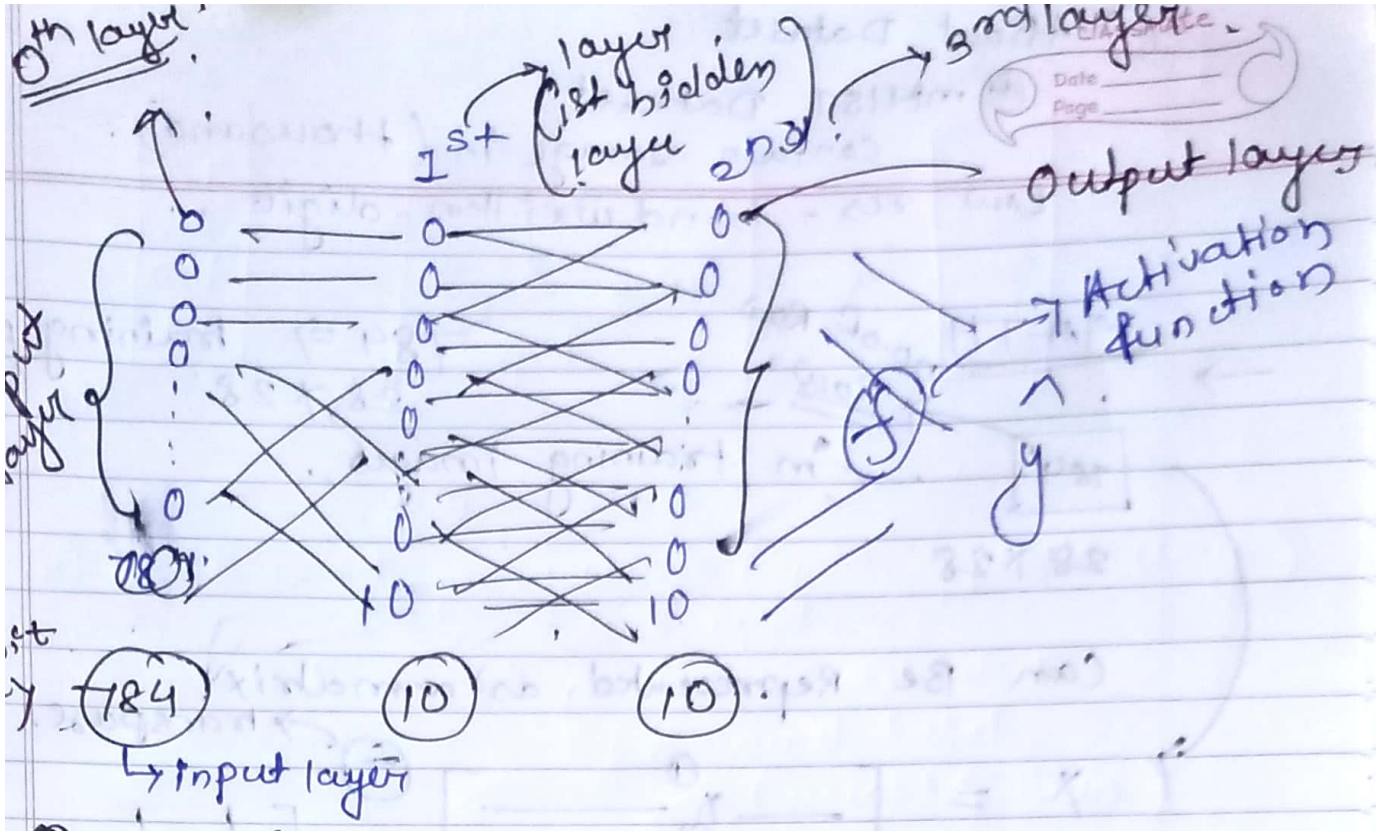
784 → process. → (predit what digit it is image represent).
28 × 28

!! will be Done By neural network.

About Neural network.

(2) layer

1st layer = 784 node → input layer
2nd layer = hidden layer → 10 unit
3rd layer = 10 unit → Output layer.



(4) Step involes in procss.

1. forward propogation.
2. Baward propogation.
3. Parameter Update.

① Forward propogation.

→ take a image run through network an compute. what your Output is going to Be.

$A^{[0]}$ → input layer (no processing going) $= X \quad (784 \times m)$.

$Z^{[1]}$ → un Activated 1st layer.

To get Z
Apply weight & bais matrix.

$= Z^{[1]} = W^{[1]} A^{[1]} + b^{[1]}$

$10 \times m \quad 10 \times 784 \quad 784 \times m \quad 10 \times m$

→ Liner Combination of input layer and 1st layer.

∴ we can say.

1st layer = liner Combination of input layer

2nd layer = liner Combination of 1st layer.

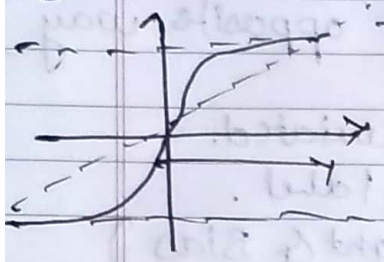
$A^{[1]} = g(f^{[1]}) = Relu(f^{[1]})$

Application of Activation Function.

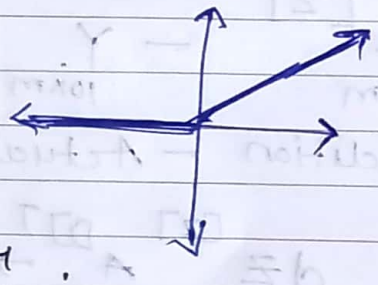
Note = without Activation function, It equation are just fancy Linear Regression...

Example of Activation function.

① tanh. ~~② sigmod.~~



→ We Are using ReLu. (rectified Linear unit).



$Relu(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$

Activated 2nd layer.

$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$

$10 \times m \quad 10 \times 10$

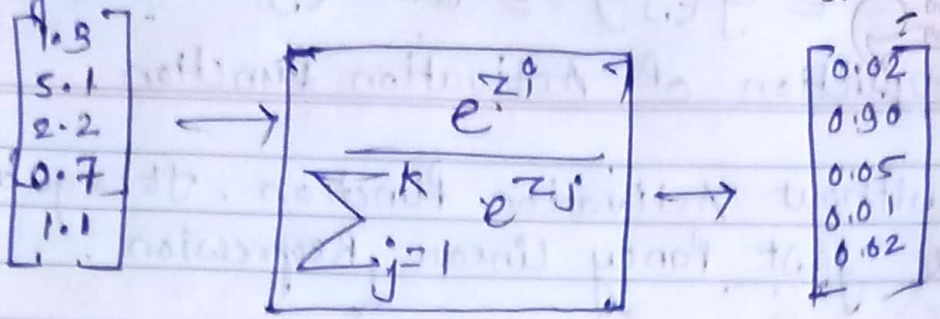
$A^{[2]} = Softmax(Z^{[2]})$

Reason to Apply Softmax Activation function

① Due the layer Being "Output layer" each of the 10 node correspond to each 10. Digit. (could Be recognize).

② Each Digit should have a probability

Output layer. → Softmax Activation function → Probability of occurring.



(11) Backward progation (goin opposite way) will find.

prediction. → prediction deviated. By Actual label. → (weight & Bias).

= Result in an error.

Therefore we will adjust weight and Biases

$dZ^{[2]}_{10\times m} = A^{[2]}_{10\times m} - Y_{10\times m}$ → error of 2nd layer.

prediction − Actual label.

$dw^{[2]}_{10\times 10} = \frac{1}{m} dZ^{[2]}_{10\times m} A^{[1]}_{m\times 10}$ → Contribution of weight to the error.

$db^{[2]} = \frac{1}{m} \sum dz^{[2]}$ → Contribution of Bias to error.

10×1 10×1

iii) for 1st hidden layer. ↗ derivation of Activation layer.

$$dz^{[1]} = w^{[2]T} \cdot dz^{[2]} * g'(z$$

$10 \times m \quad = \quad 10 \times 10 \quad 10 \times m$

Here error from 2nd layer Appling weights in reverse dot to the error of 1st layer.

$$dw^{[1]} = \frac{1}{m} dz^{[1]} x^T = \text{---}$$

$$db^{[2]} = \frac{1}{m} \sum dz^{[1]}$$

After all calculation we will know exactly how much the weight, Bias contribute for the error in each layer
∴ updating our parameter.

(iii) · Parameter Update.

$$w^{[1]} = w^{[1]} - \alpha\, dw^{[1]}$$

$$b^{[1]} = b^{[1]} - \alpha\, db^{[1]}.$$

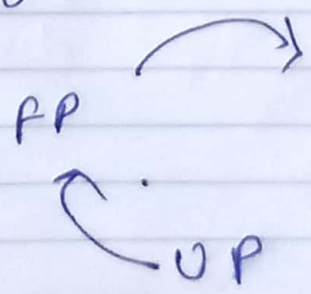
$$w^{[2]} = w^{[2]} - \alpha\, dw^{[2]}$$

$$b^{[2]} = b^{[2]} - \alpha\, db^{[2]}.$$

↗ hyper parameter

$\alpha \longrightarrow$ learning Rate.

So



FP → Forward progo——
BP → Backward prog ——
UP → Update parameter