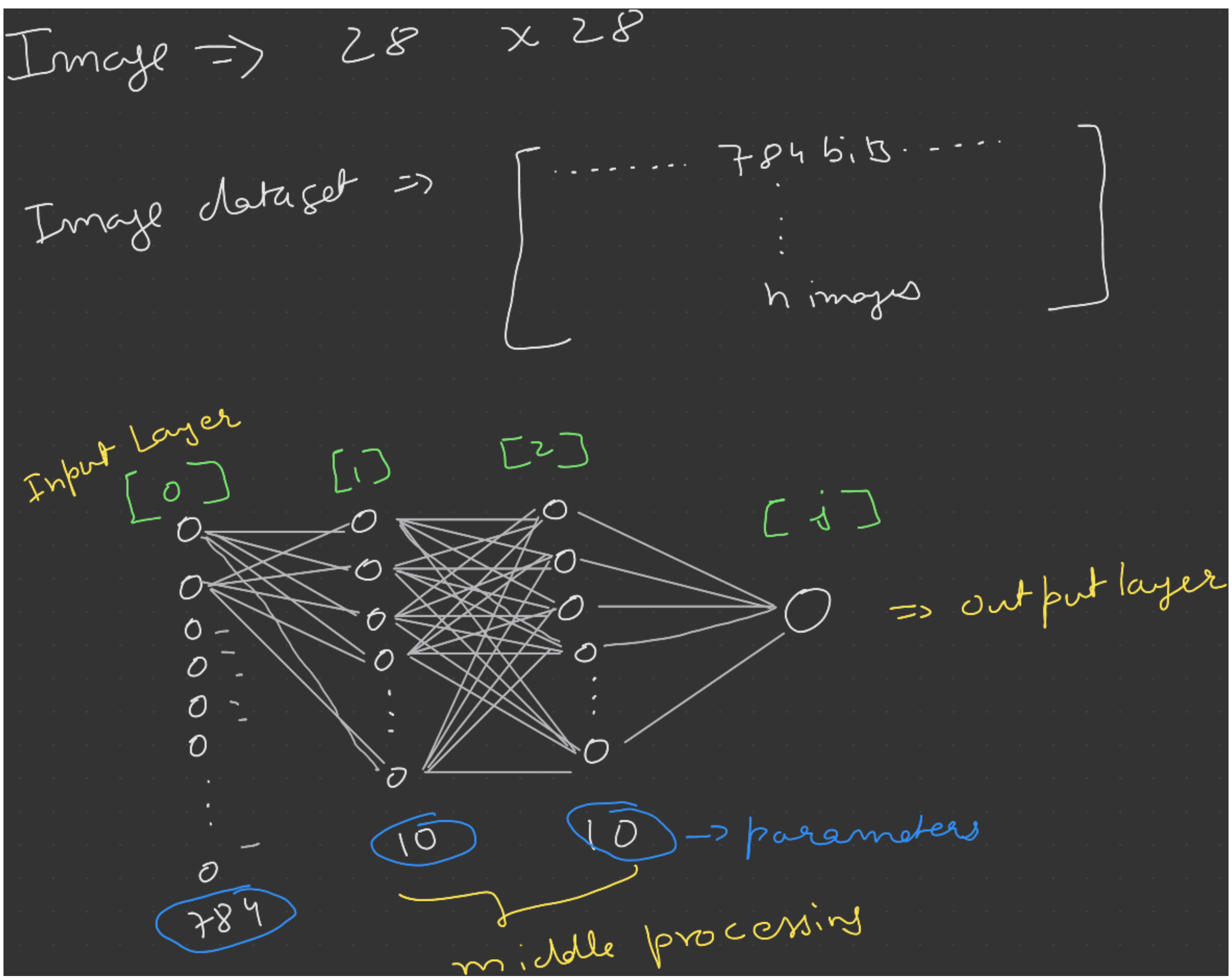
MNIST Dataset, Custom NN

Overview of Image and Neural Net

The image data will undergo preprocessing by being transformed into a 28 x 28 format, resulting in an image output represented by 784 bits. The dataset will be organized as a matrix of size 784 x n, where n is the number of images. Subsequently, n features will be fed into a neural network with 784 parameters, comprising an initial/input layer (a0) with 784 nodes, two hidden layers (z1 and z2) each containing 10 nodes, and a final output layer.



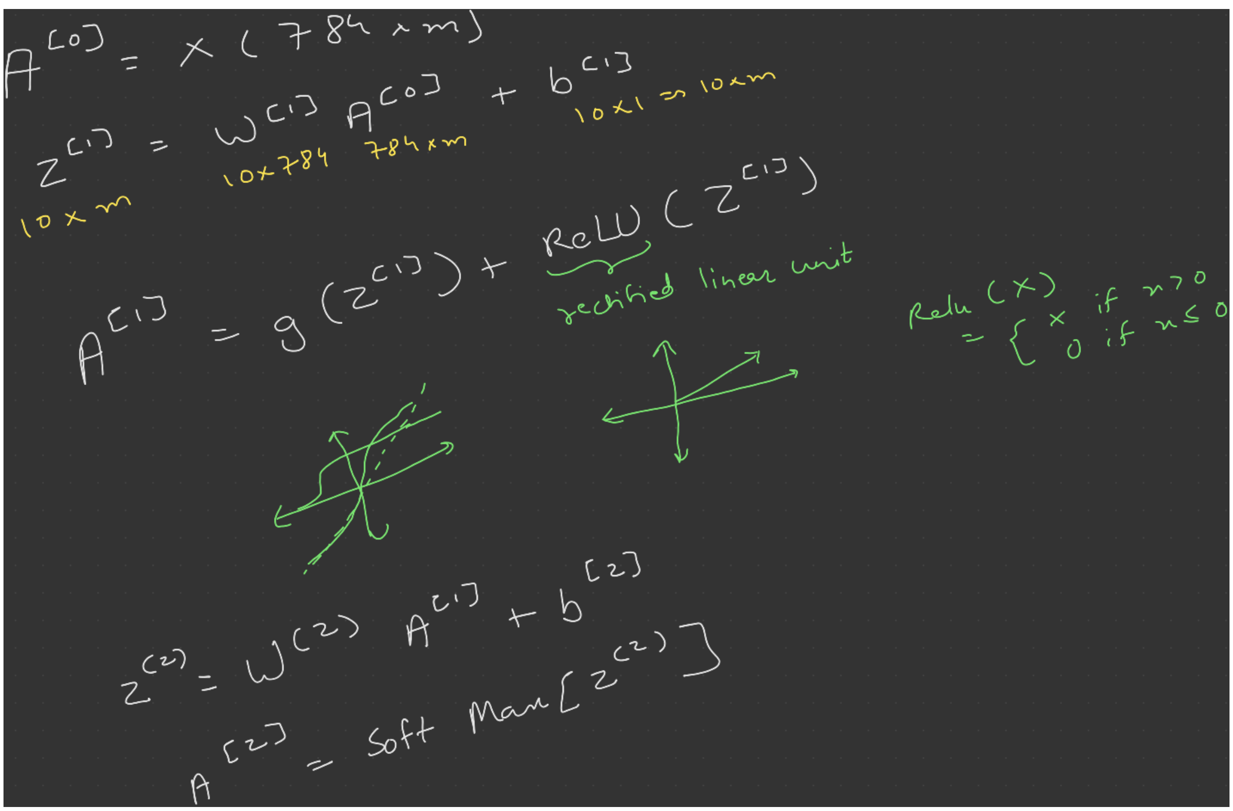
How Processing will occur

- Initial/Input Layer (a0): The input layer consists of 784 nodes, with each node corresponding to a bit in the image representation.

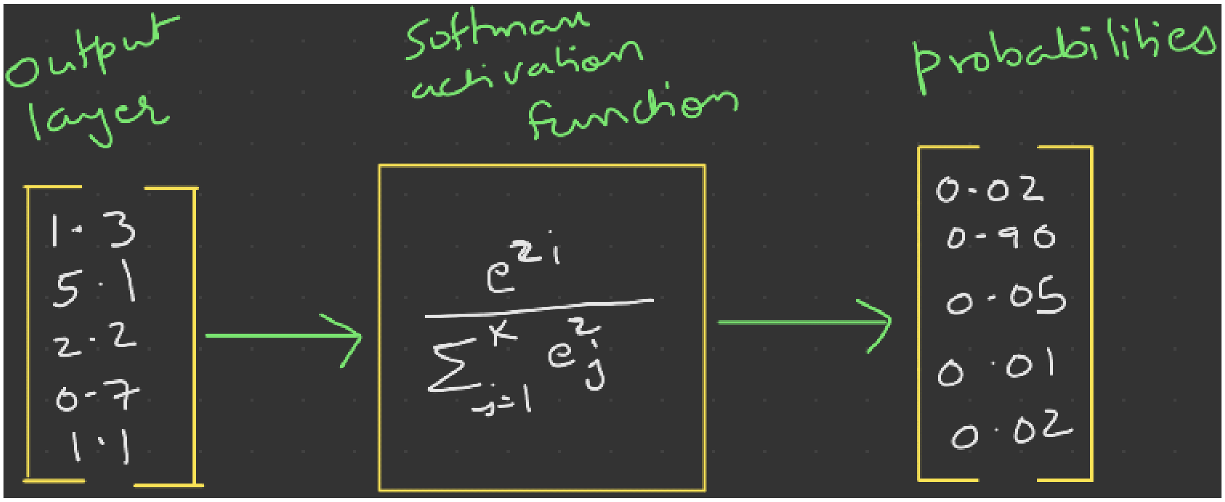
- First Hidden Layer (z1): This layer involves a weighted sum operation (z1 = w1 * a0 + b), where w1 represents the weights associated with each node, and b denotes the bias term.

- Second Hidden Layer (z2): Similar to the first hidden layer, this step computes a weighted sum (z2 = w2 * a1 + b), where a1 is the output from the first hidden layer.

- Softmax Activation: After processing through the hidden layers, the resulting values are passed through the softmax activation function. This function converts the raw output into probability scores, indicating the likelihood of the input image representing a particular number.



Softmax working:



Doing Correction:

2nd layer

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} d^{[2]} A^{[1]T}$$

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

challenge: Take error from 2, backpropagate to 1, resolve it.

1st layer

$$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g'(Z^{[1]})$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

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

$$W^{[1]} = W^{[1]} - \alpha dW^{[1]}$$

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

$$W^{[2]} = W^{[2]} - \alpha dW^{[2]}$$

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

$\alpha \Rightarrow$ learning rate

Keep doint it again and again until accuracy is good