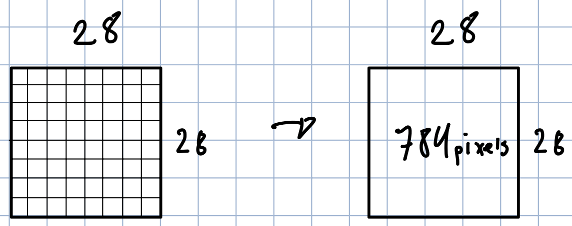



= 1 training img

we will have m of these.

Training Data Matrix:

$$
\begin{array}{c}
\\
\# \text{ of images } (m)
\end{array}
\overbrace{\begin{bmatrix}
0 & 1 & 2 & \cdots & 784 \\
0 & 1 & 2 & \cdots & 784 \\
0 & 1 & 2 & \cdots & 784 \\
 & & \downarrow & & \\
0 & 1 & 2 & \cdots & 784 \\
0 & 1 & 2 & \cdots & 784
\end{bmatrix}}^{\text{pixels : 784 columns}}{}^{T}
=
\overbrace{\begin{bmatrix}
0 & 0 & & 0 & 0 \\
1 & 1 & & 1 & 1 \\
2 & 2 & \rightarrow & 2 & 2 \\
\vdots & \vdots & & \vdots & \vdots \\
784 & 784 & & 784 & 784
\end{bmatrix}}^{m \text{ images}}
$$

Transpose...

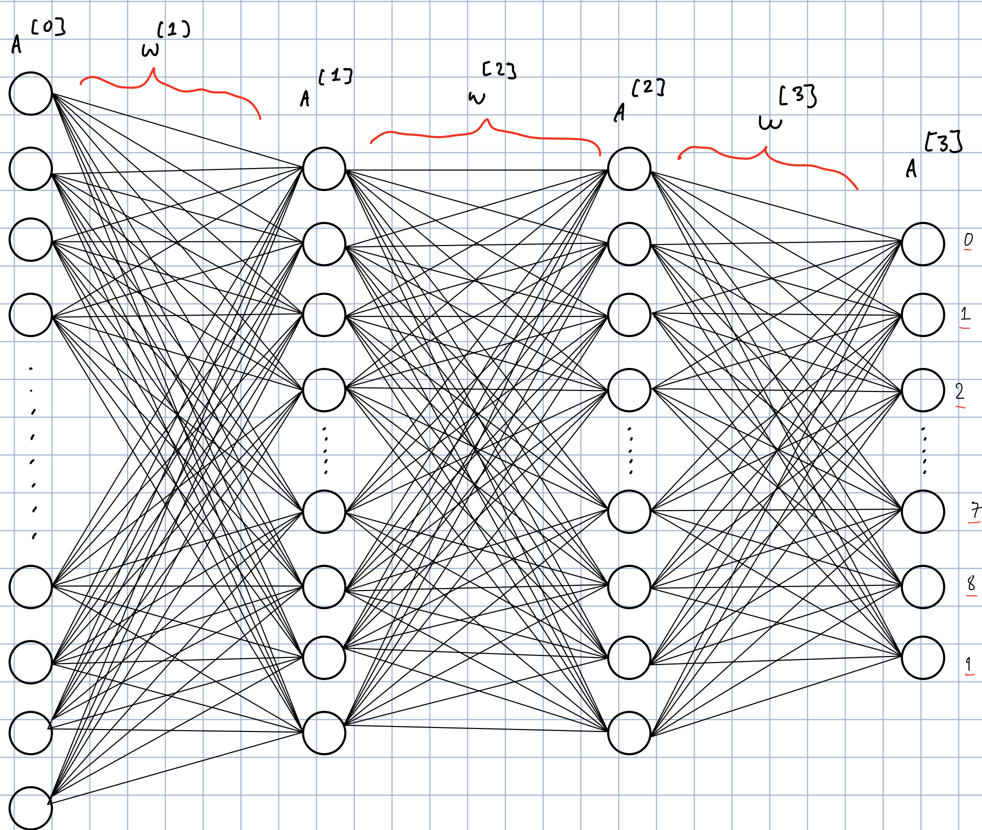
#pixels

pixels

# of pixels (each will hold a # from 0-1 indicating activation)

Training Matrix will be called "T" (784 x m)

## My Neural Network

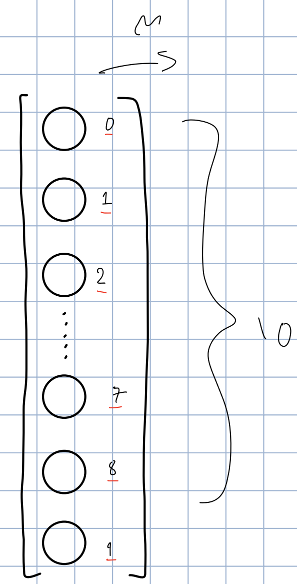



$L_0$ (784 neurons) Input Layer

$L_1$ (16 neurons) Hidden Layer 1

$L_2$ (16 neurons) Hidden Layer 2

$L_3$ (10 neurons) Output Layer




comparison / error checking vector

$\vec{Y}$

10 x m

# Forward Prop

"Run the image through the network"

## LAYER 0-1

layer 0, Input layer.

$$A^{[0]} = \underbrace{T}_{784 \times M}$$

Training Data

Unactivated layer 1...

$$Z^{[1]} = W^{[1]} A^{[0]} + B^{[1]}$$

$Z^{[1]}$: $16 \times M$; $\underbrace{W^{[1]}\, A^{[0]}}_{16 \times M}$: $16 \times 784$, $784 \times M$; $B^{[1]}$: $16 \times 1$

Weights for each connection L0 -> L1
16 connections, 784 times

Input layer neuron activation values.
784 neurons, M training steps

Bias for each weighted sum.
16 biases,
M training steps

Activated layer 1

$$\underbrace{A^{[1]}}_{16 \times M} = \sigma[\underbrace{Z}_{16 \times M}] = \mathrm{ReLu}(Z^{[1]})$$

16 neurons from 0-1 determined by L0 & spread throughout all the training steps

Sigmoid squish the weighted sum for values 0-1

## LAYER 1-2

$$\underbrace{Z^{[2]}}_{16 \times M} = \underbrace{\underbrace{W^{[2]}}_{16 \times 16} \underbrace{A^{[1]}}_{16 \times M}}_{16 \times M} + \underbrace{B^{[2]}}_{16 \times 1}$$

$$\underbrace{A^{[2]}}_{16 \times M} = \underbrace{\sigma(Z^{[2]})}_{16 \times M}$$

# BACK Prop

## LAYER 3-2

$$\underbrace{\delta Z^{[3]}}_{10 \times M} = \underbrace{A^{[3]}}_{10 \times M} - \underbrace{Y}_{10 \times M}$$

How far off are the final layer outputs from the expected outputs?

final layer activations for M training steps...

Expected final layer activations for M trainings steps...

$$\underbrace{\delta W^{[3]}}_{10 \times 16} = \frac{1}{M} \underbrace{\delta Z^{[3]}}_{10 \times M} \underbrace{A^{[2]T}}_{M \times 16}$$

How much do weights need to be nudged between $L_3$ & $L_2$? 10 nudges 16 times...

total # of training data

$$\underbrace{\delta B^{[3]}}_{10 \times 1} = \frac{1}{M} \sum \underbrace{\delta Z^{[3]}}_{10 \times 1}$$

Nudges we need to make to the biases between $L_3$ & $L_2$

## LAYER 2-1

$$\underbrace{\delta Z^{[2]}}_{16 \times M} = \underbrace{\underbrace{W^{[3]T}}_{16 \times 10} \underbrace{\delta Z^{[3]}}_{10 \times M}}_{16 \times M} \cdot g'(\underbrace{Z^{[2]}}_{16 \times M})$$

How far off are L2's activations from L3's expected activations?

$$\underbrace{\delta W^{[2]}}_{16 \times 16} = \frac{1}{m} \underbrace{\delta Z^{[2]}}_{16 \times M} \underbrace{A^{[1]T}}_{M \times 16}$$

$$\underbrace{\delta B^{[2]}}_{16 \times 1} = \frac{1}{m} \sum \underbrace{\delta Z^{[2]}}_{16 \times 1}$$

## LAYER 1-0

$$\delta Z^{[1]} = W^{[2]T} \delta Z^{[2]} \cdot g'(Z^{[1]})$$

$$\underbrace{\ }_{16\times M} \quad \underbrace{\underbrace{\ }_{16\times 16}\ \underbrace{\ }_{16\times M}}_{16\times M} \quad \underbrace{\ }_{16\times M}$$

$$\underbrace{\delta W^{[1]}}_{16\times 784} = \frac{1}{m}\,\underbrace{\delta Z^{[1]}}_{16\times m}\,\underbrace{T^{T}}_{m\times 784}$$

$$\underbrace{\delta B^{[1]}}_{16\times 1} = \frac{1}{m}\sum \underbrace{\delta Z^{[1]}}_{16\times 1}$$

LAYER 2->3

$$\underbrace{Z^{[3]}}_{10\times m} = \underbrace{\underbrace{W^{[3]}}_{10\times 16}\,\underbrace{A^{[2]}}_{16\times m}}_{10\times m} + \underbrace{B^{[3]}}_{10\times 1}$$

$$\underbrace{A^{[3]}}_{10\times m} = \underbrace{\sigma\left(Z^{[3]}\right)}_{10\times m}$$

use softmax if this causes issues...

# UPDATE PARAMS

## LAYER 0-1

$$\underbrace{W^{[1]}}_{16\times 784} = \underbrace{W^{[1]}}_{16\times 784} - \alpha\,\underbrace{\delta W^{[1]}}_{16\times 784}$$

$$\underbrace{B^{[1]}}_{16\times m} = \underbrace{B^{[1]}}_{16\times m} - \underbrace{\alpha\,\delta B^{[1]}}_{16\times 1}$$

## LAYER 1-2

$$\underbrace{W^{[2]}}_{16\times 16} = \underbrace{W^{[2]}}_{16\times 16} - \alpha\,\underbrace{\delta W^{[2]}}_{16\times 16}$$

$$\underbrace{B^{[2]}}_{16\times m} = \underbrace{B^{[2]}}_{16\times m} - \alpha\,\underbrace{\delta B^{[2]}}_{16\times 1}$$

LAYER 2-3

$$\underbrace{W^{[3]}}_{10\times16} = \underbrace{W^{[3]}}_{10\times16} - \alpha\, \underbrace{\delta W^{[3]}}_{10\times16}$$

$$\underbrace{B^{[3]}}_{10\times m} = \underbrace{B^{[3]}}_{10\times m} - \alpha\, \underbrace{\delta B^{[3]}}_{10\times1}$$

$\alpha$ = Learning Ratio

[ Hyper parameter (not trained by model) we set this ratio...

$i=0 \quad i<4$

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \end{array}$$

$[A^0, A^1, A^2, A^3]$ len = 4...

$[W_1, W_2, W_3]$ len = 3

$[\delta W_1, \delta W_2, \delta W_3]$ len = 3

$[B_1, B_2, B_3]$

$[\delta B_1, \delta B_2, \delta B_3]$

$[Z_1, Z_2, Z_3]$

$[\delta Z_1, \delta Z_2, \delta Z_3]$