May 27

Code: model = tf.keras.Sequential([

flatten(input_shape=(28,28)),

dense(128, "relu"),

dropout(0.2),

dense(10, "softmax")

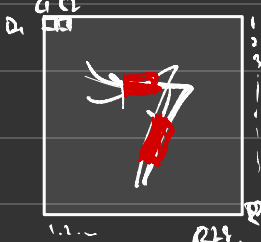
])

Drop 20% of 128

(60k, 28, 28)

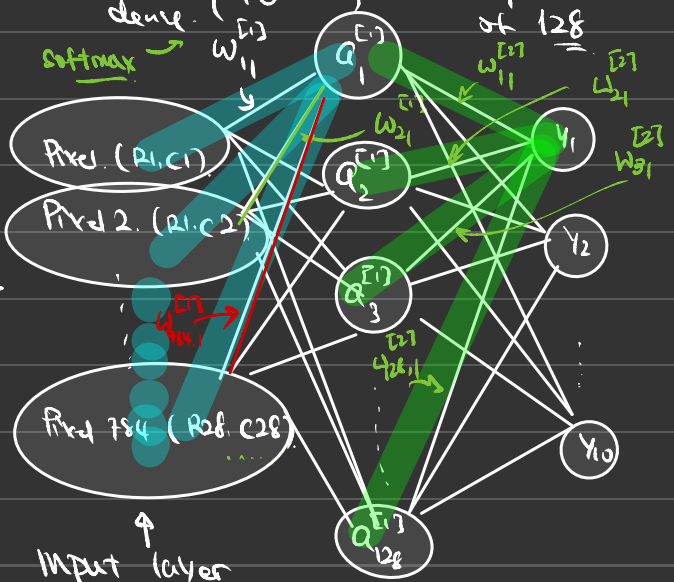
X_train.shape

MNIST Data



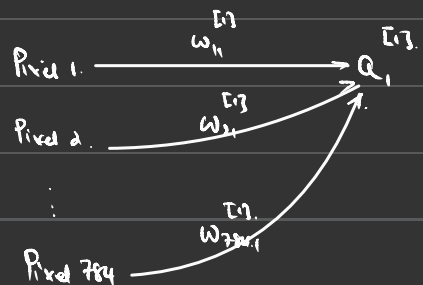
flatten

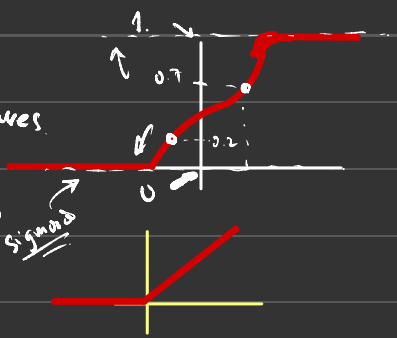
dim: 28 x 28



fully connected:

" each neuron from previous layer map to each neuron from current layer. "





variables / features

$a_1^{[1]} = $ a function of $\{$ Pixel 1, Pixel 2 ... Pixel 784 $\}$

and $\{ w_{1,1}^{[1]}, \dots w_{784,1}^{[1]} \}$

↑ Parameters or weights

choose sigmoid as activation function ↪

$$= 1/(1+\exp(-\sum_{j=1}^{784} w_{j,1}^{[1]} \cdot \text{Pixel}_j))$$

much easier to train

Remark: If I choose "relu", then $a_1^{[1]} := \max\left(\sum_{j=1}^{784} w_{j,1}^{[1]} \cdot \text{Pixel}_j, 0\right)$

new variable ← y_1 = a fct of $\{ a_1^{[1]}, \dots a_{128}^{[1]} \}$ and $\{ w_{1,1}^{[2]} \dots w_{128,1}^{[2]} \}$ → new weight

choose sigmoid $= 1/(1+\exp(-\sum_{j=1}^{128} w_{j,1}^{[1]} \cdot a_j^{[2]}))$

Then compute $y_2, y_3, \dots, y_{10}$. ← 10 classes because there are 10 digits called labels.

10-Class Classification: Given a picture, you are allowed to guess amongst 10 classes.

Y_1 ← digit 0 : Class 1.

Y_2 ← digit 1 : Class 2.

⋮

Y_{10} ← digit 9 : Class 10

$\frac{\exp(-Y_1)}{\sum_{j=1}^{10} \exp(-Y_j)}$ ≈ the probability an obs. falls in Class 1.

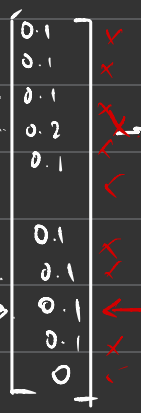
$\frac{\exp(-Y_2)}{\sum_{j=1}^{10} \exp(-Y_j)}$ ≈ Same Prob. an obs. in Class 2.

⋮

$\frac{\exp(-Y_{10})}{\sum_{j=1}^{10} \exp(-Y_j)}$ ≈ ... Class 10.

One picture.

→ NN →



→ this means the picture is predicted as digit "3", ie. Class 4.

Label: 7

← Correct answer ✓

Sum to 1

Softmax: math form $\frac{\exp(\text{target})}{\text{Sum of } \exp(\text{all possibility})}$

I make educated guess of which class the obs. you give me falls in.