Matrix Application Neural Network

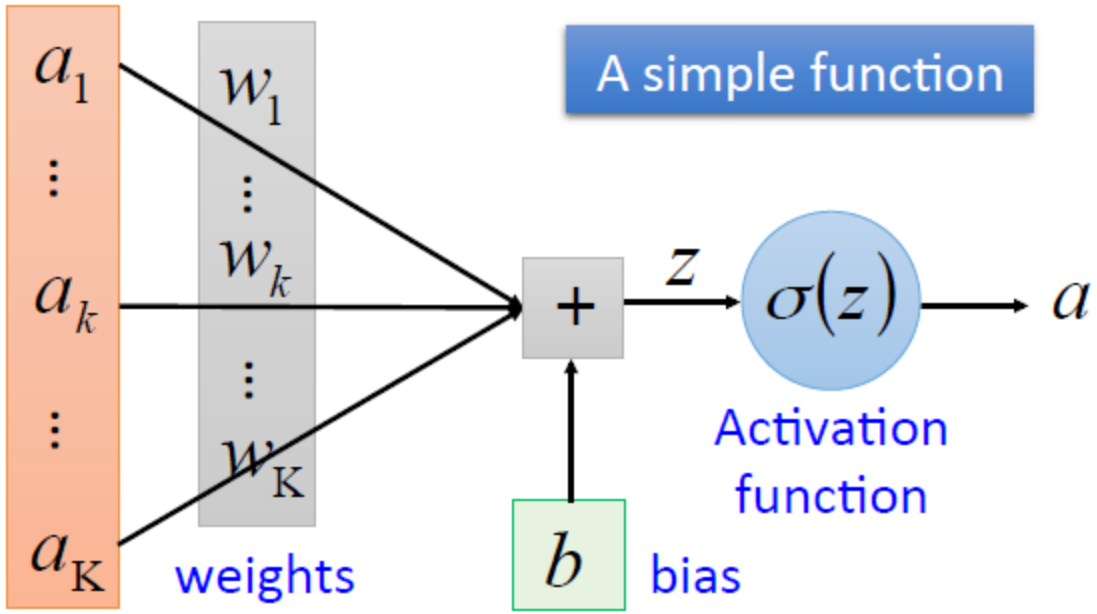
Ms. Liu

Neuron

Neural Network

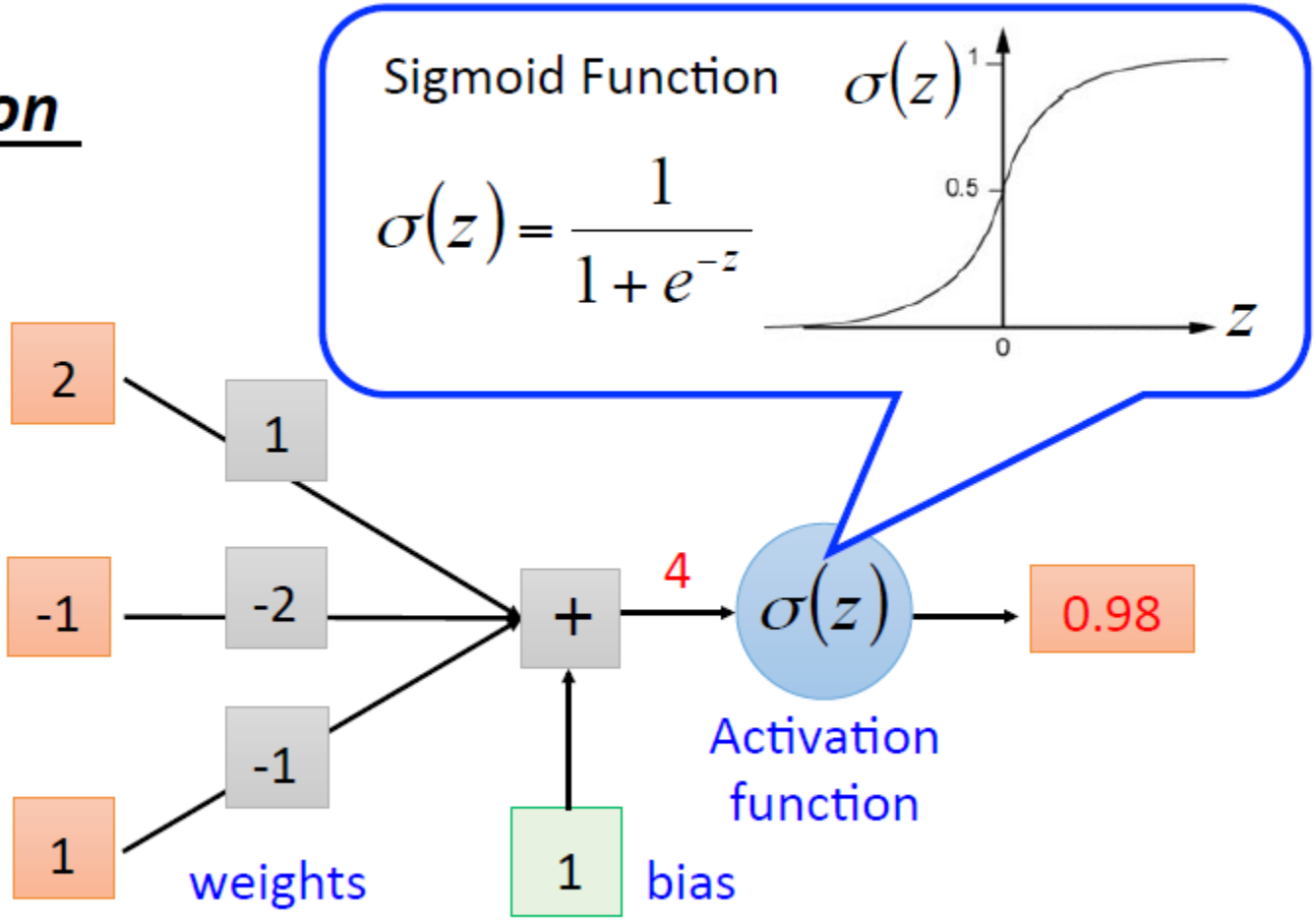
Neuron (a building block)

$$z = a_1 w_1 + \cdots + a_k w_k + \cdots + a_K w_K + b$$



Deep Learning Tutorial by Hung-yi Lee
http://www.slideshare.net/tw_dsconf/ss-62245351

Neural Network

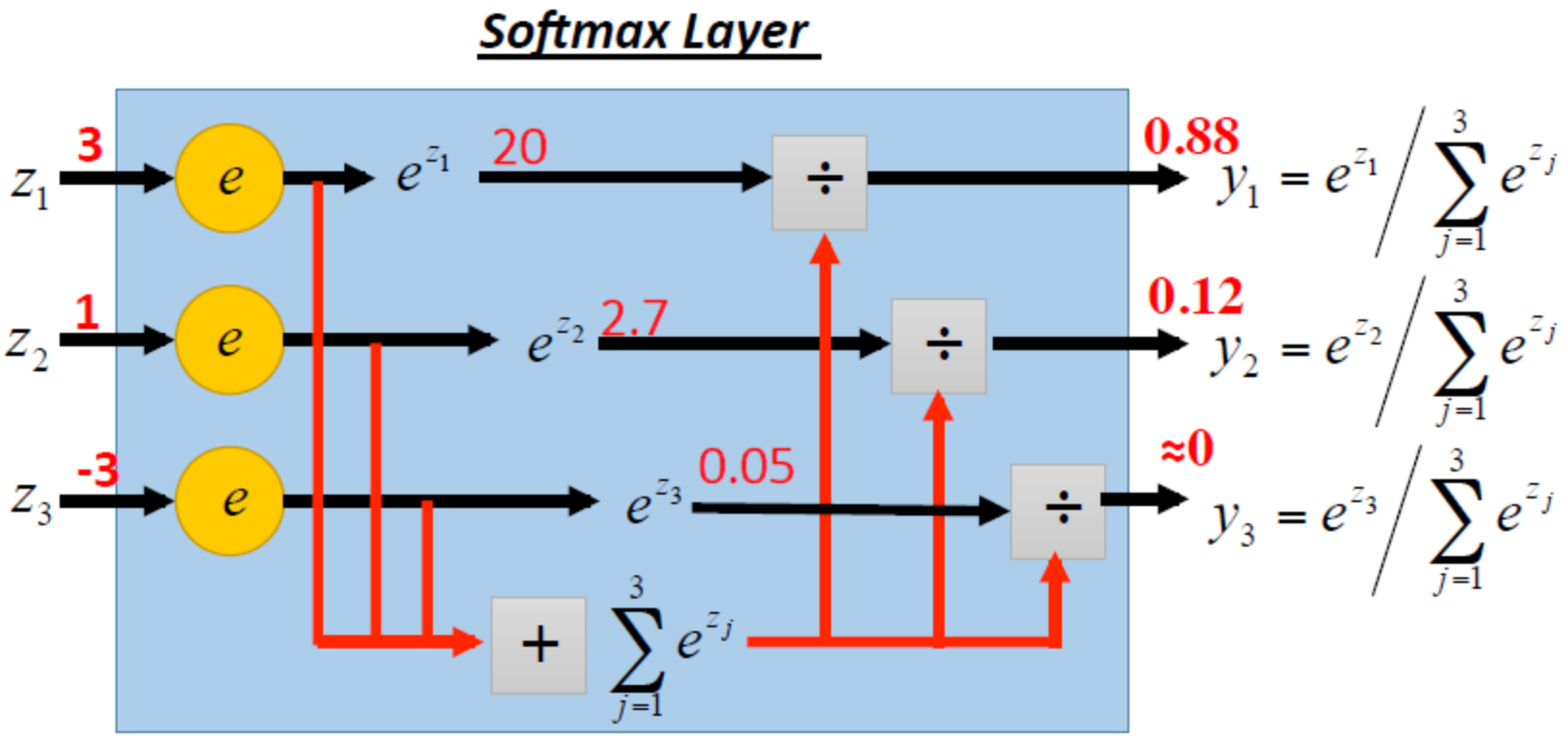


Output Layer

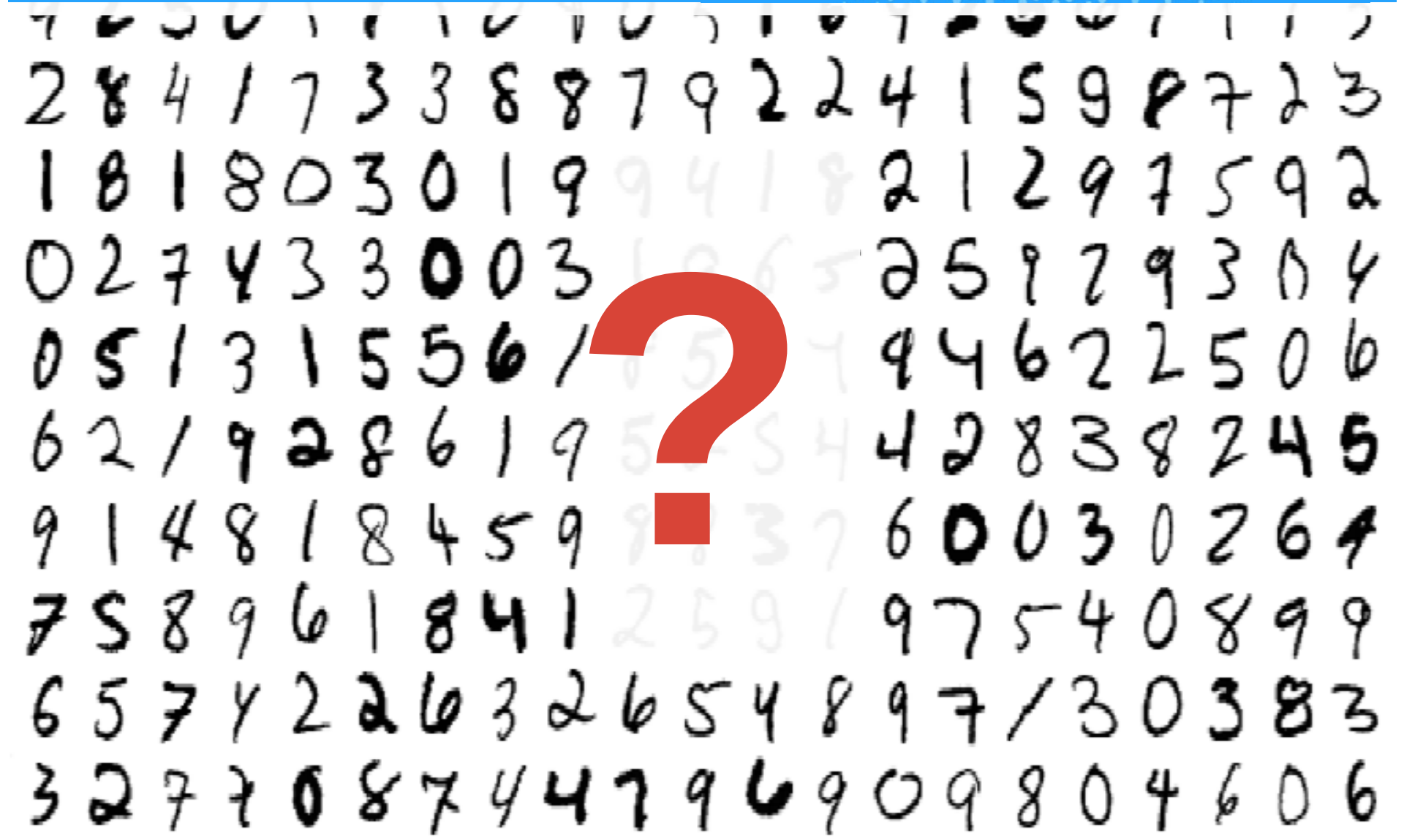
- Softmax layer as the output layer

Probability:

- $0<y_i<1$
- $\text{sum}_i\, y_i = 1$

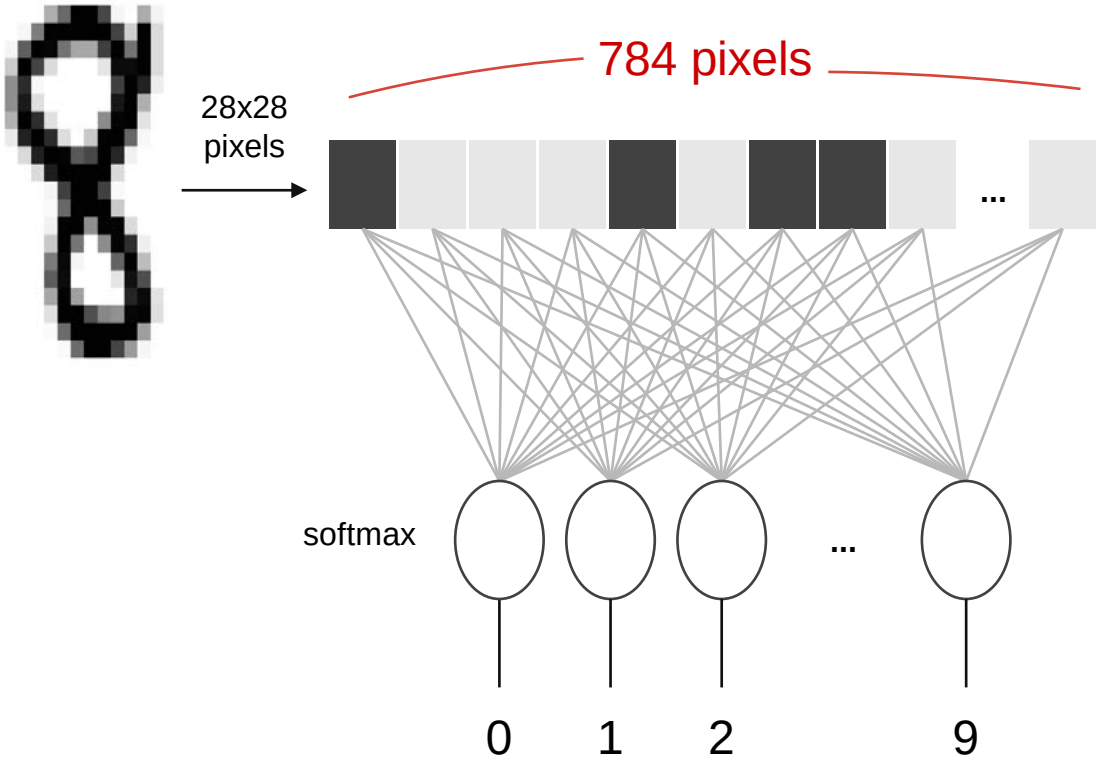


Hello World: handwritten digits classification - MNIST



MNIST = Mixed National Institute of Standards and Technology - Download the dataset at http://yann.lecun.com/exdb/mnist/

Very simple model: softmax classification



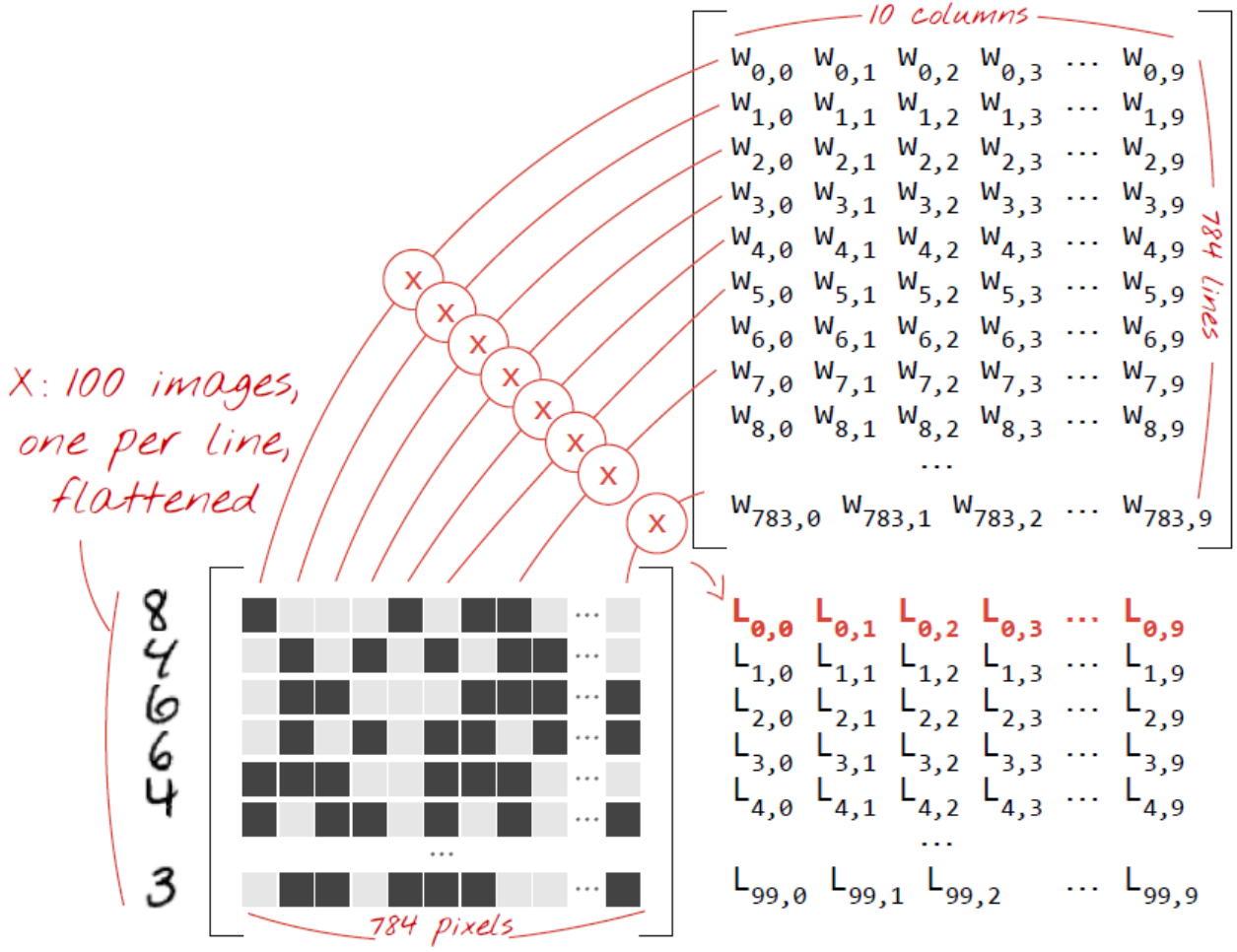
weighted sum of all pixels + bias

$$softmax(L_n) = \frac{e^{L_n}}{||e^{L}||}$$

neuron outputs

from *TensorFlow and Deep Learning*, Martin Gorner

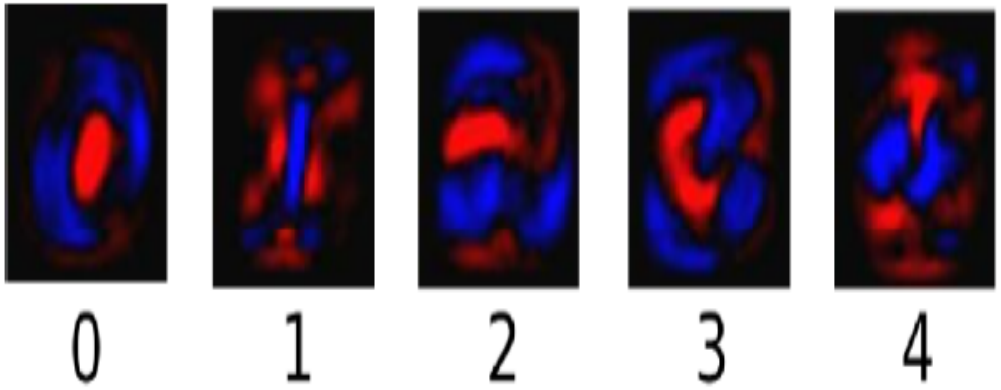
In matrix notation, 100 images at a time



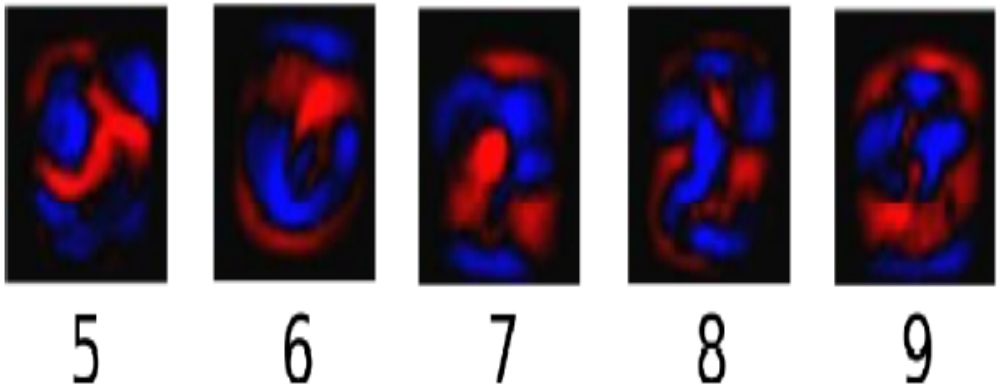
$$L = X.W + b$$

What is *W* (learned parameter)?

algorithm $y = Wx + b$



W_{ij} in W in R^{10x784}
learned by a model



$W_{ij} > 0$ (blue) for i

$W_{ij} < 0$ (red) aganist i

Jinn-Liang Liu

Softmax, on a batch of images

Predictions Y[100, 10]

Images X[100, 784]

Weights W[784,10]

Biases b[10]

$$Y = softmax(X.W + b)$$

applied line by line

matrix multiply

broadcast on all lines

tensor shapes in [ ]

Now in TensorFlow (Python)

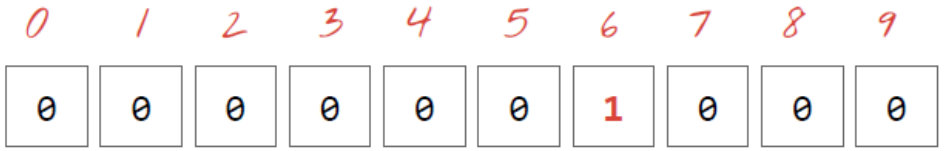
tensor shapes: X[100, 784] W[748,10] b[10]

```
Y = tf.nn.softmax(tf.matmul(X, W) +
b)
```

matrix multiply

broadcast
on all lines

Success ?

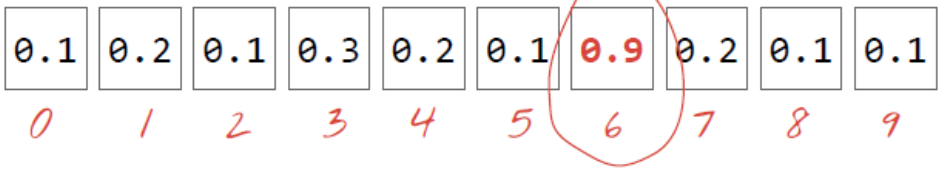


actual probabilities, "one-hot" encoded

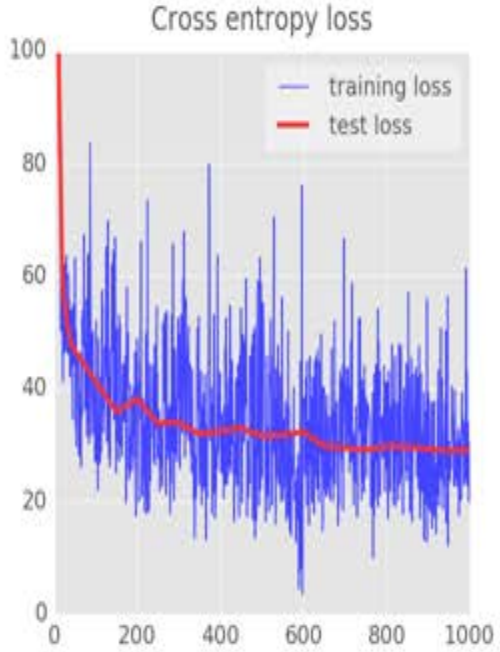
Cross entropy: $-\sum Y_i' . log(Y_i)$

computed probabilities

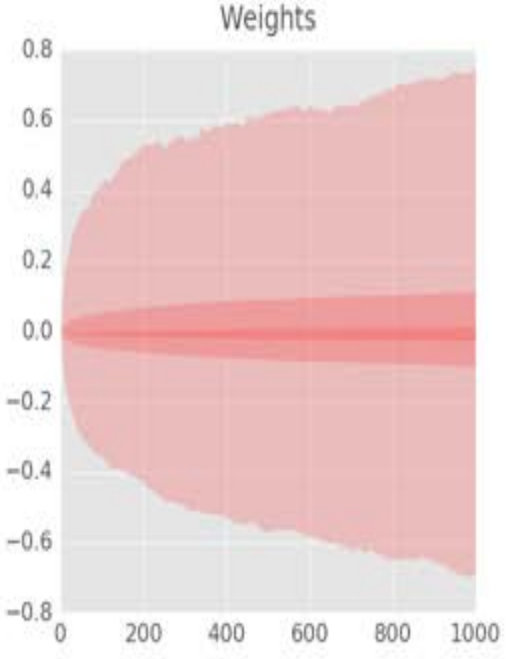
this is a "6"

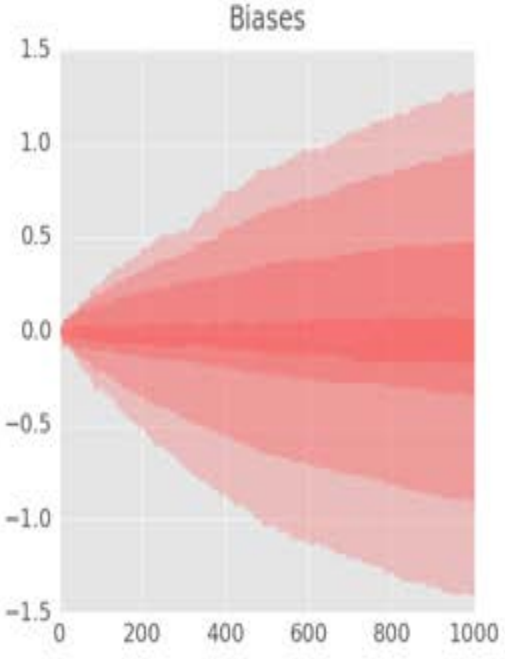



Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
training accuracy
test accuracy


Cross entropy loss
100
80
60
40
20
0
0
200
400
600
800
1000
training loss
test loss


Training digits


Weights
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0
200
400
600
800
1000


Biases
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0
200
400
600
800
1000


Test digits
100%
98%
96%
94%
92%
90%

92 %