CMI LAB

UNIVERSITY OF WOLVERHAMPTON
KNOWLEDGE ▪ INNOVATION ▪ ENTERPRISE

6CS012 – Artificial Intelligence and Machine Learning

Ahsan Adeel

Theme lead, Conscious Multisensory Integration (CMI) Lab

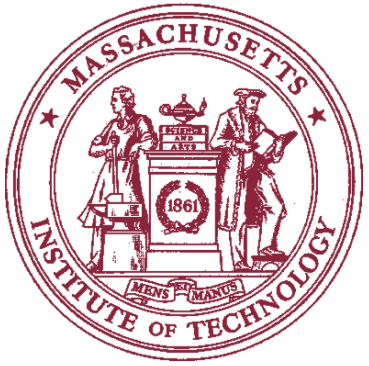
Fellow, MIT Synthetic Intelligence Lab, MIT

and Oxford Computational Neuroscience Lab, University of Oxford

Visiting EPSRC/MRC fellow, University of Stirling

http://www.cmilab.org/

http://www.cmilab.org/dr_ahsan_adeel.html#Home



NUFFIELD DEPARTMENT OF SURGICAL SCIENCES
Medical Sciences Division

Lecture 5 – Neural Network/MLP, Batch-Optimization, Stochastic Optimization, MNIST Classification

Lecture 4 Review

(Pay attention to the whiteboard)

Linear Regression

Logistic Regression

Analytical modelling

GD

WS4: Task 1

- **Use GD and build a classifier to detect the Iris-Virginica type: <u>Iris-Setosa (Y=0)</u> or <u>Iris-Versicolor (Y=1).</u>**

(i) Explain the whole procedure in your words

(ii) find the accuracy

- **Here is the pre-processing code:**

- from sklearn import datasets
- iris = datasets.load_iris()
- list(iris.keys())
- X = iris["data"] # petal width
- X=X[0:99]
- **X = normalize(X, norm='l2')**
- y = iris["target"]
- y=np.reshape(y, (1, 150))
- y=y.T
- y=y[0:99]
- **y = normalize(y, norm='l2')**

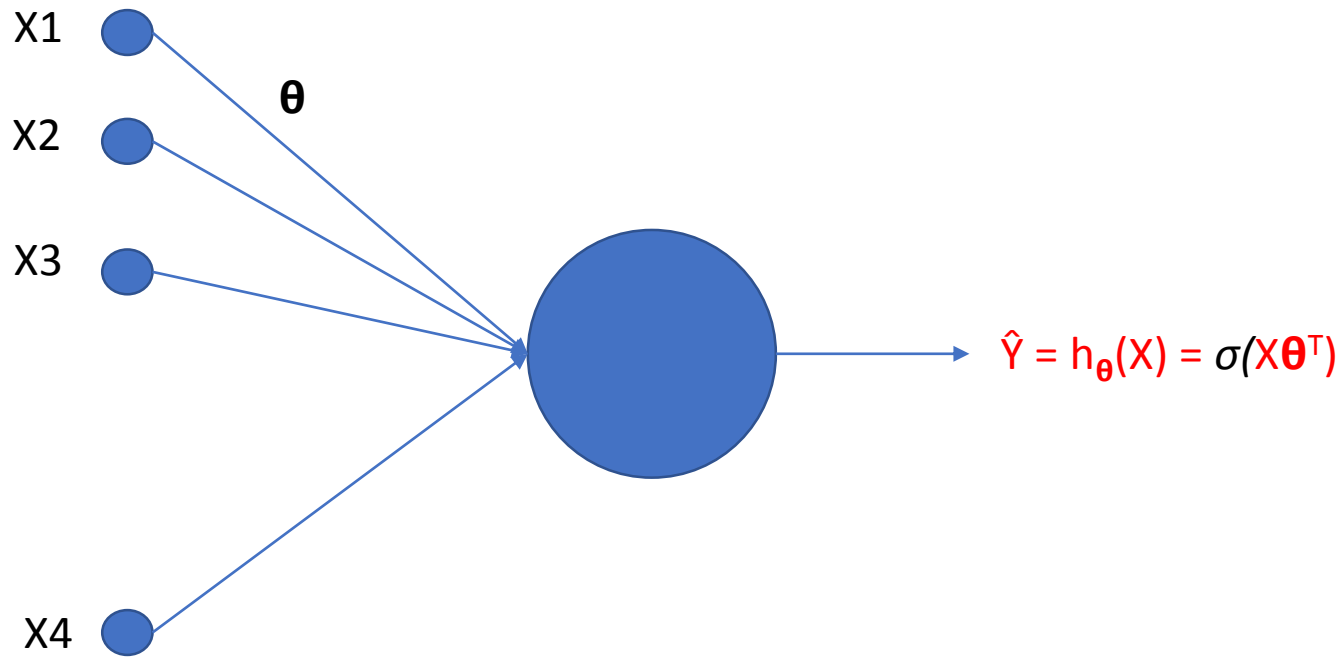
- **Hints:** $\hat{Y} = h_{\boldsymbol{\theta}}(X) = \sigma(X\boldsymbol{\theta}^T)$

$$\nabla_\theta J(\theta) = \sum_i x^{(i)}(h_\theta(x^{(i)}) - y^{(i)})$$

$$\boldsymbol{\theta}^{(\text{next step})} = \boldsymbol{\theta} - \eta \nabla_\theta J(\theta)$$

- **Hints:**

WS4: Task 1



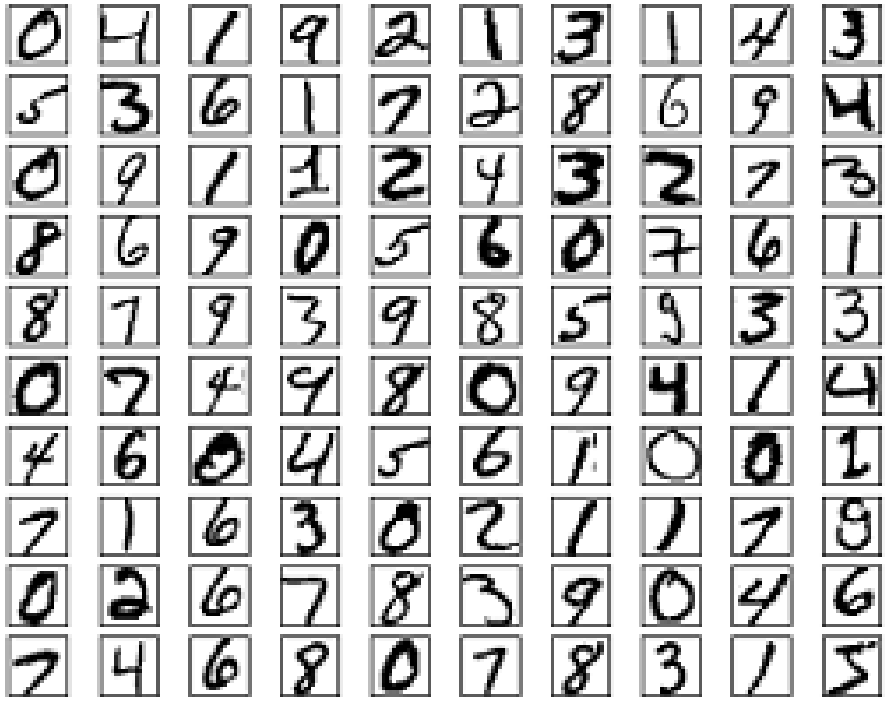
WS4: Task 2

- **Use GD and build a classifier to detect the Iris-Virginica type: Iris-Setosa or Iris-Versicolor or Iris-Virginica.**
- **(i) Explain the whole procedure in your words**
- **(ii) find the accuracy**

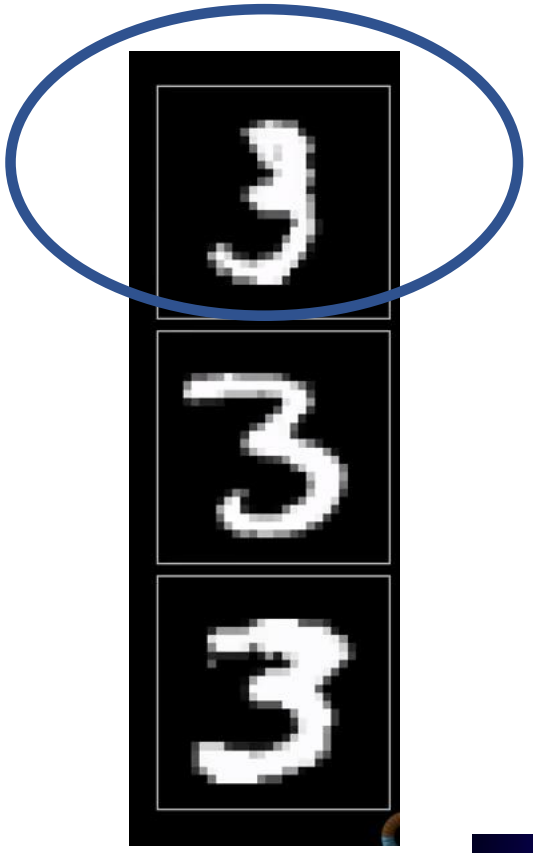
Here is the pre-processing code

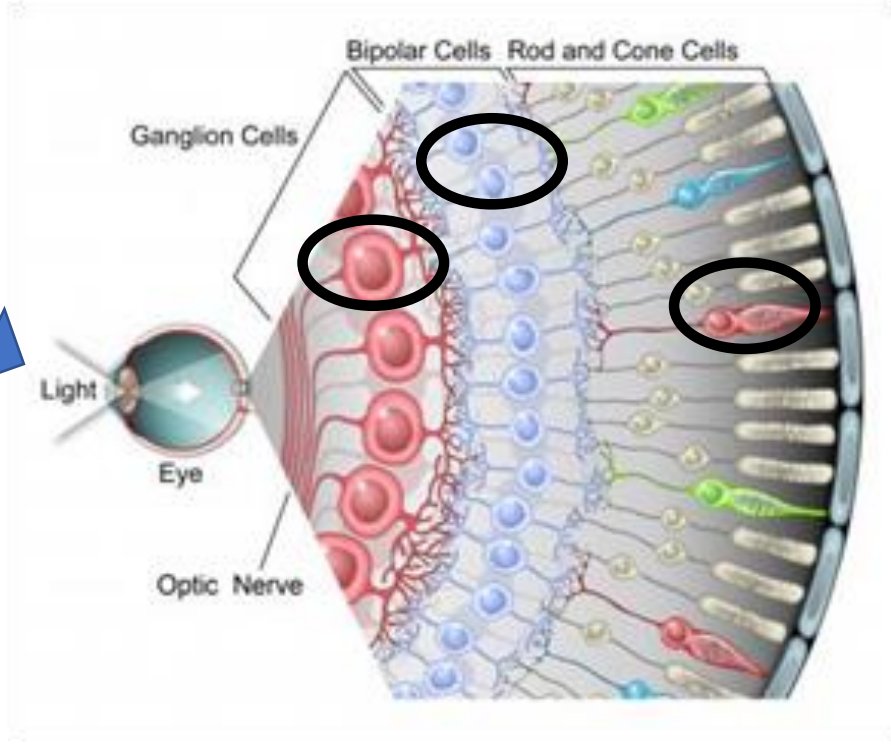
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import normalize
import scipy.sparse
from sklearn import datasets
iris = datasets.load_iris()
list(iris.keys())
X = iris["data"]

X = normalize(X, norm='l2')
y = iris["target"]



def oneHotIt(Y):
    m = Y.shape[0]
    #Y = Y[:,0]
    OHX = scipy.sparse.csr_matrix((np.ones(m),

(Y, np.array(range(m)))))
    OHX = np.array(OHX.todense()).T
    return OHX

y_mat = oneHotIt(y)

  y=y_mat
```

Recognizing handwritten digits



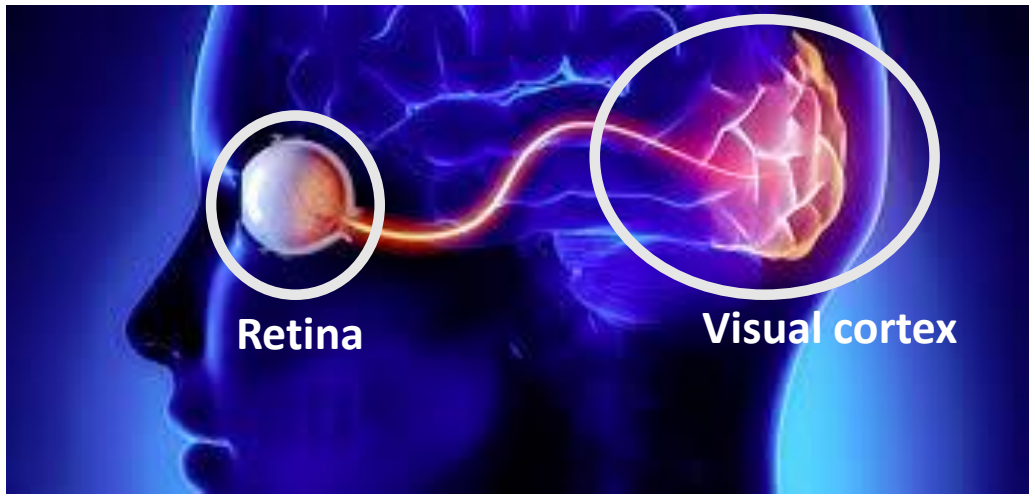
Biological Analogy





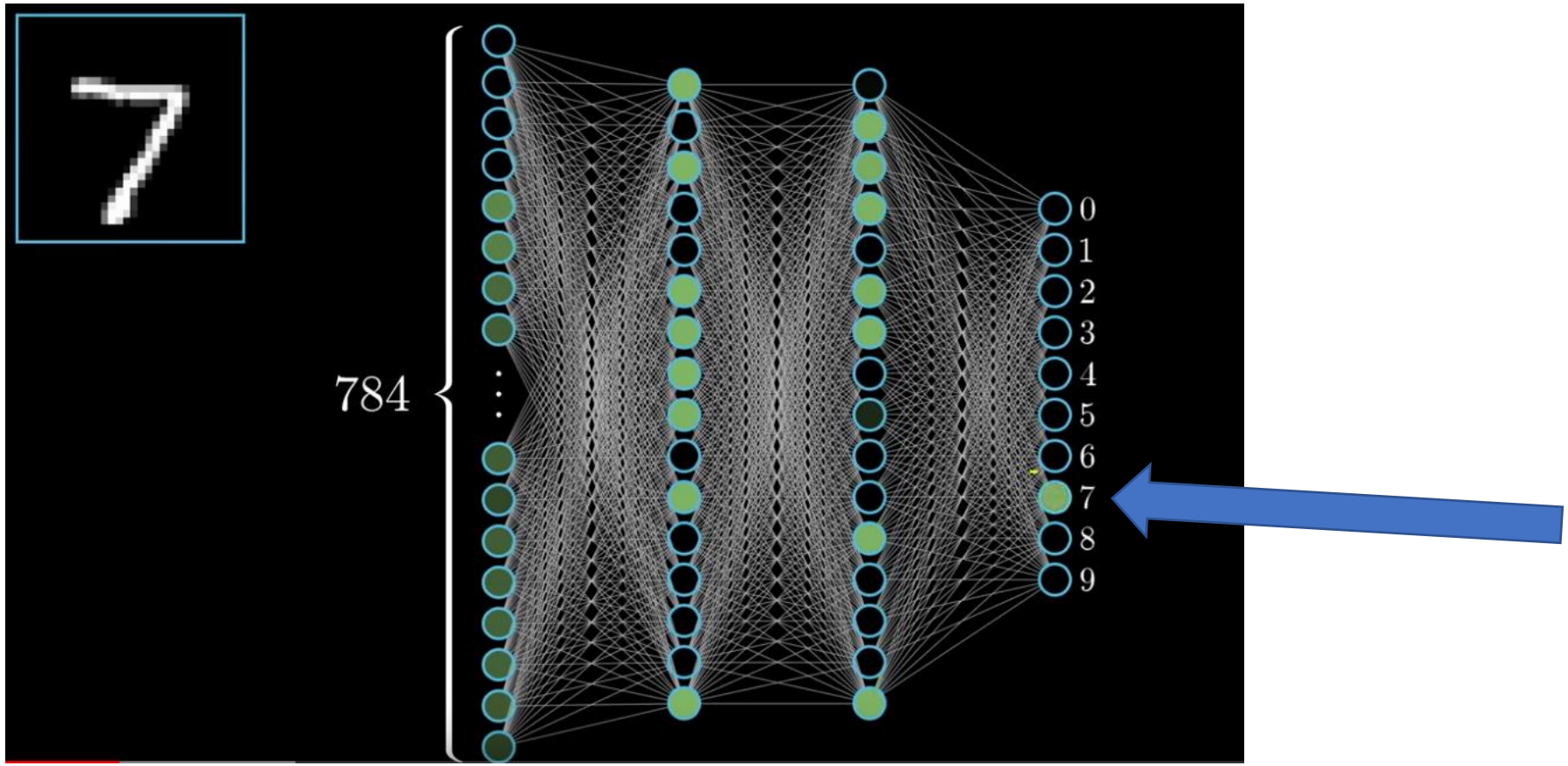
Particular light sensitive cells are firing

The retina is the layer of nerve cells responsible for sensing light and transmit signals to the brain so you can see or perceive accurately.



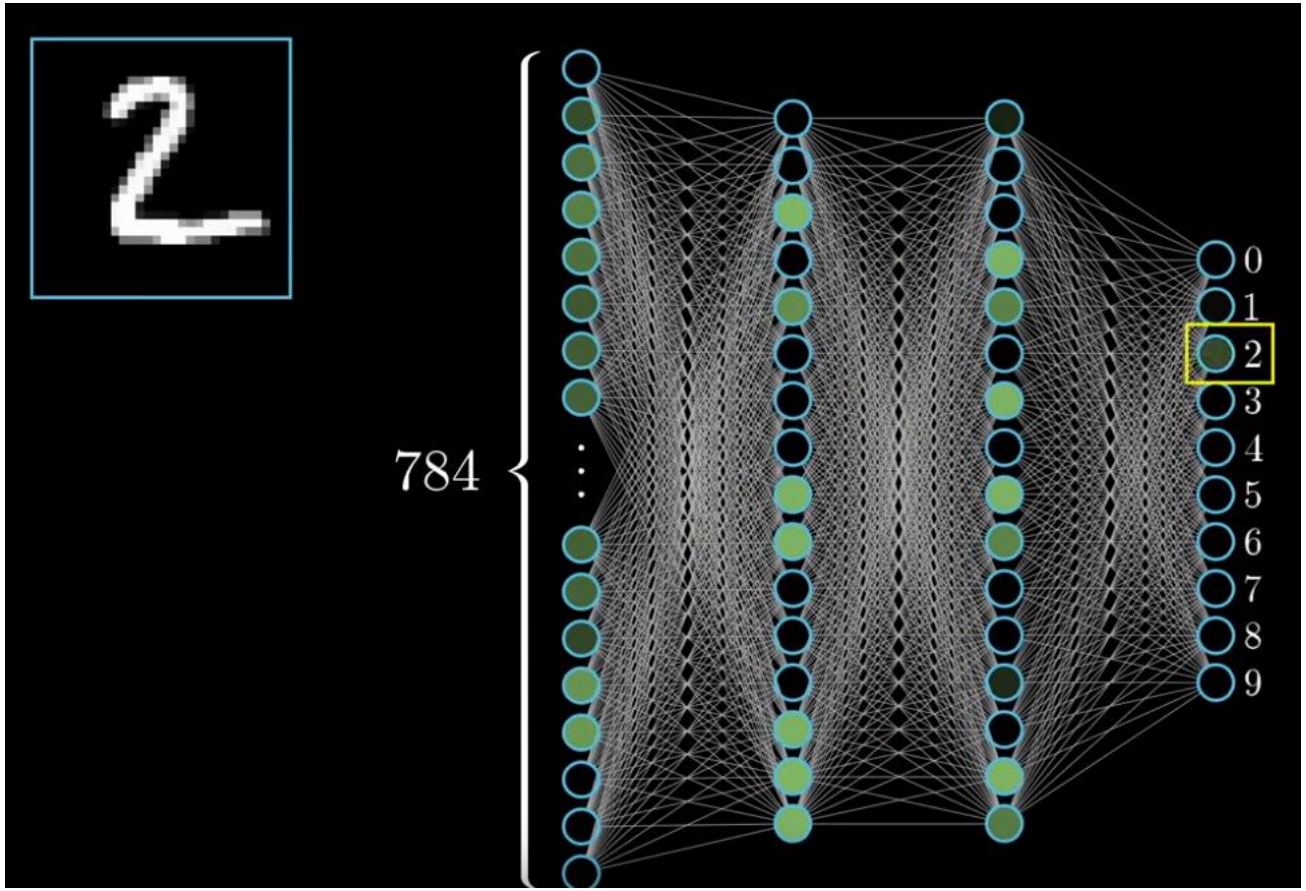
Visual cortex, a region of the brain responsible for receiving, integrating, and processing visual information relayed from the retina.

ANN/MLP recognizing handwritten digits



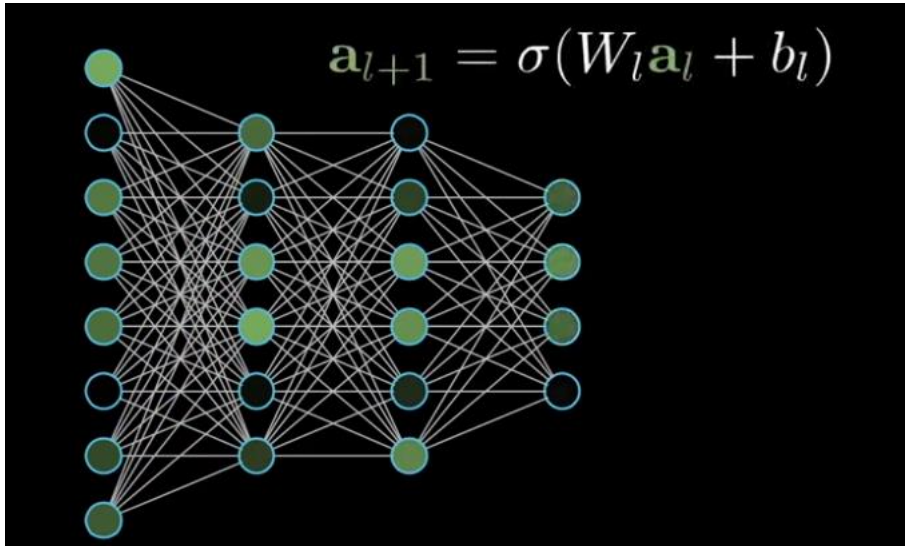
Let’s try and visualize how our brain probably works

ANN/MLP recognizing handwritten digits



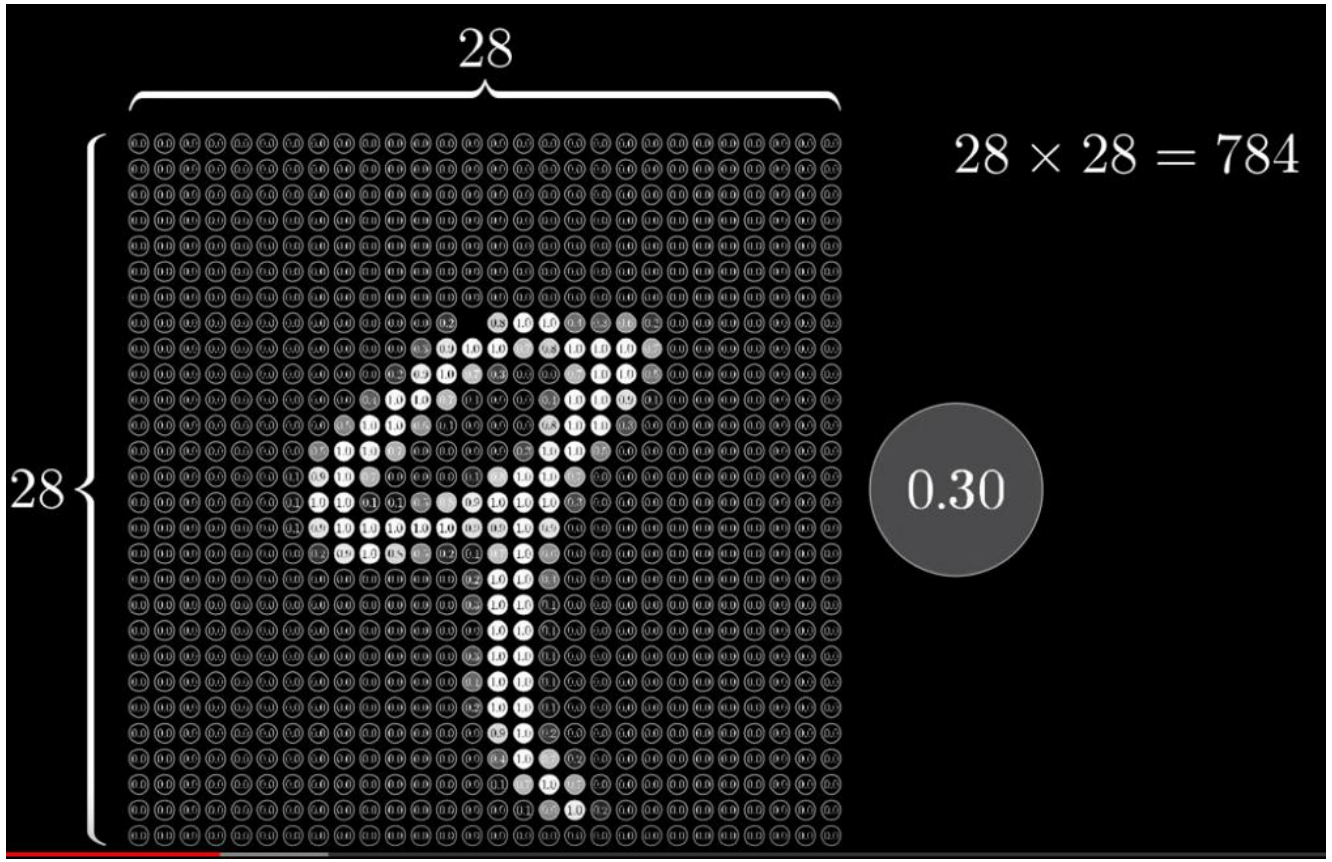
Let’s try and visualize how our brain probably works

ANN/MLP recognizing handwritten digits



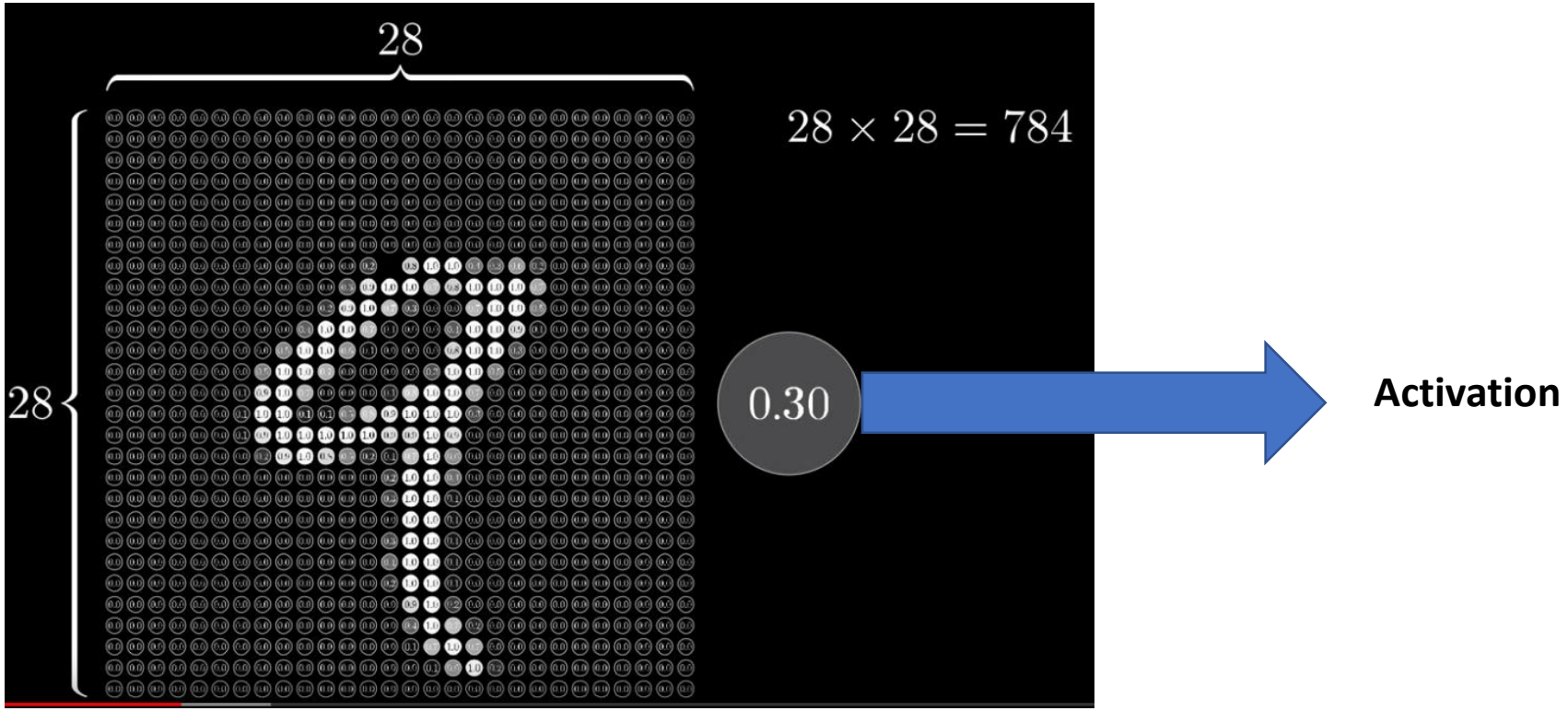
- Different artificial neural networks such as MLP, CNN, CLSTM etc., have the same basic concept
- If you have a good understanding of an MLP, rest is all just an addition to the same concept

ANN/MLP recognizing handwritten digits



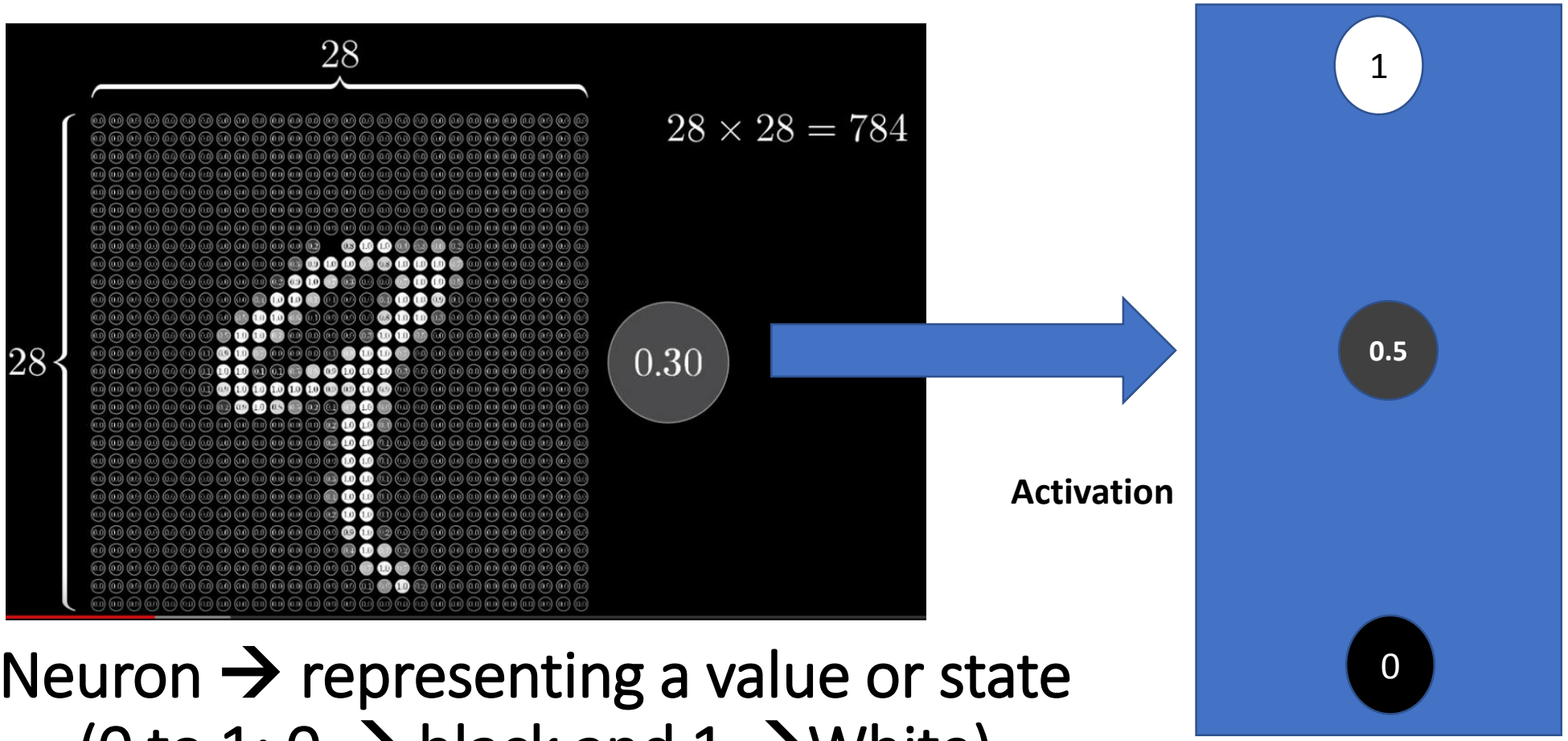
Neuron → representing a value or state
(0 to 1: 0 → black and 1 →White)

ANN/MLP recognizing handwritten digits



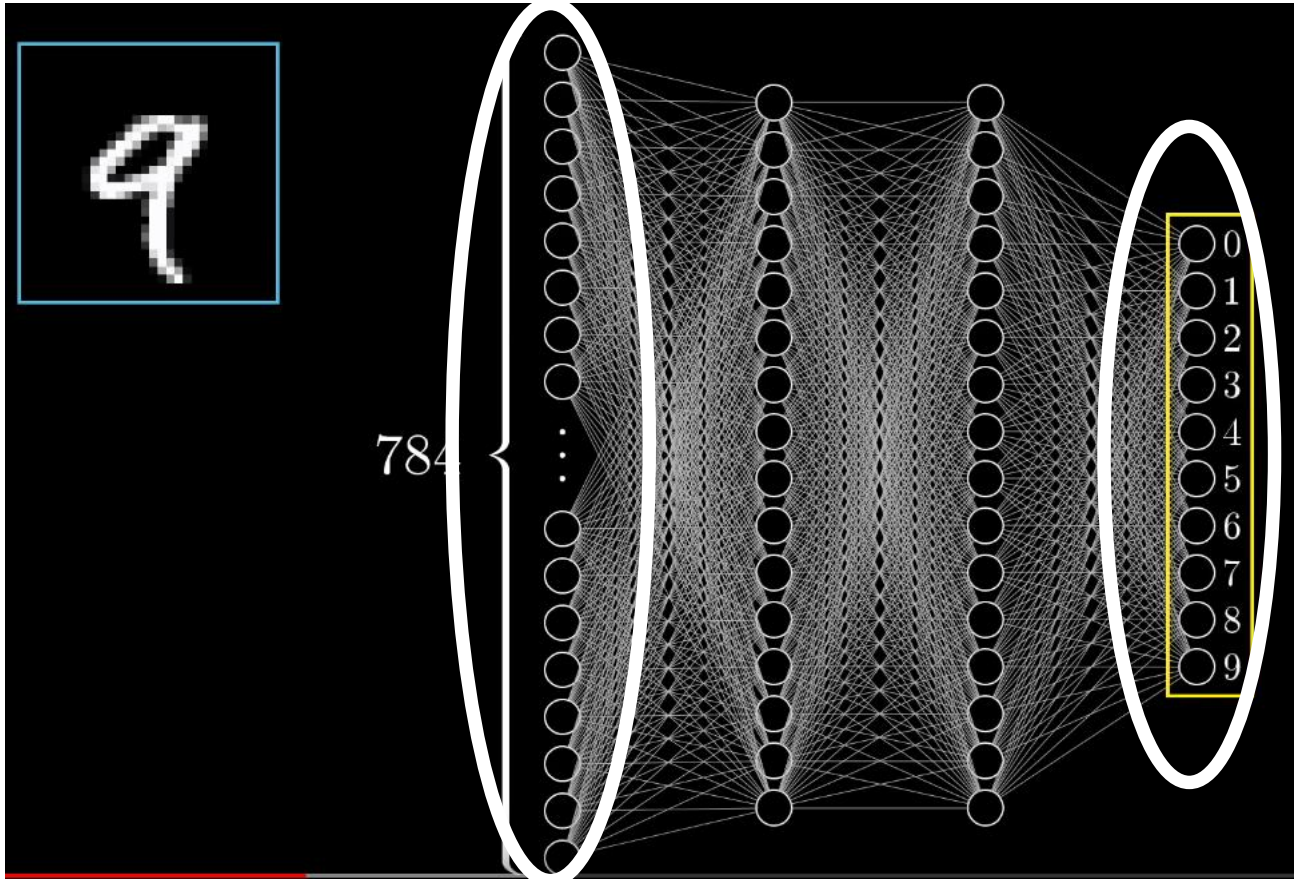
Neuron → representing a value or state
(0 to 1: 0 → black and 1 →White)

ANN/MLP recognizing handwritten digits



Neuron → representing a value or state
(0 to 1: 0 → black and 1 →White)

Artificial intelligence – Artificial Neural Networks

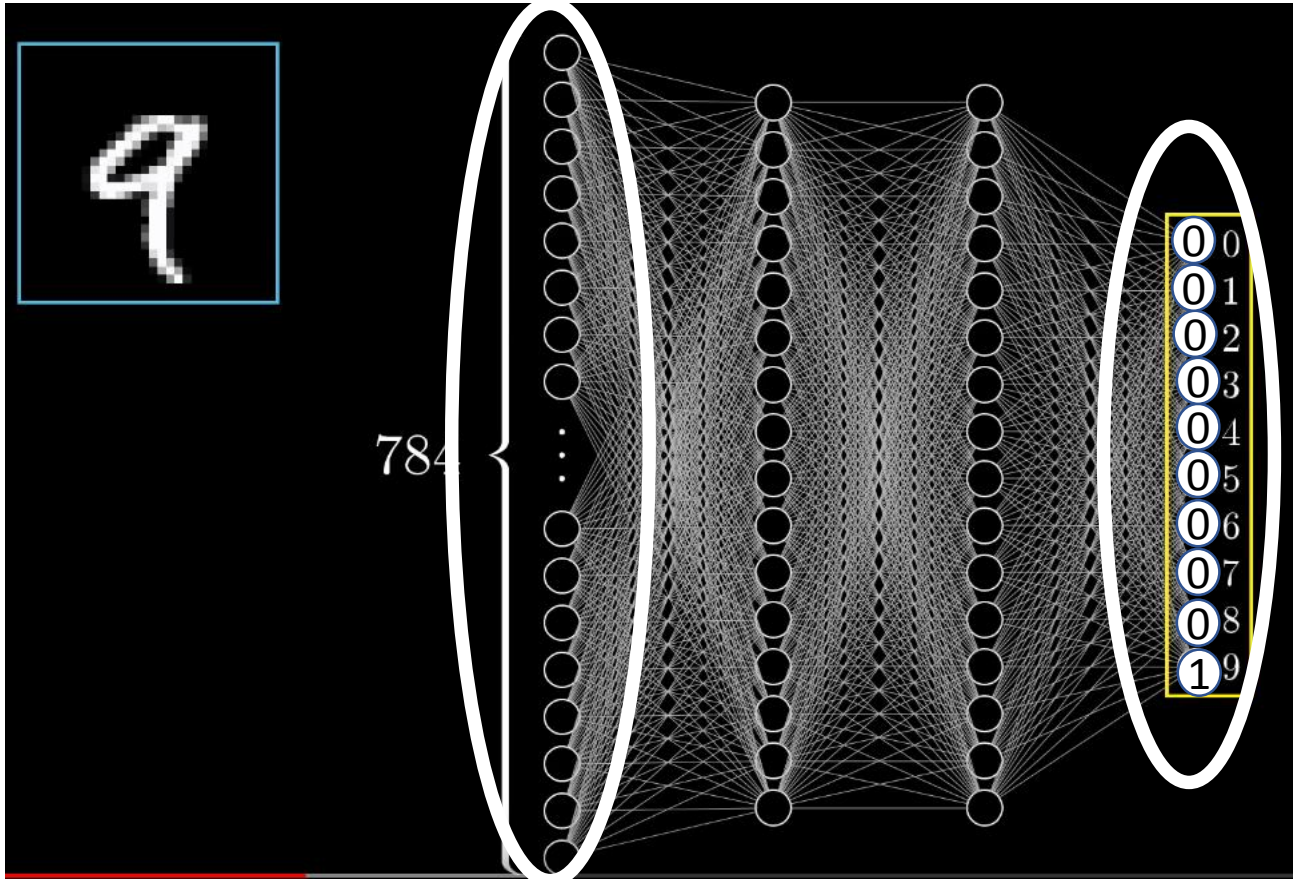


Remember each input neuron is representing a specific value

Input becomes the first or the input layer of the neural network

Neurons in the last layer become the output layer of the neural network

Artificial intelligence – Artificial Neural Networks



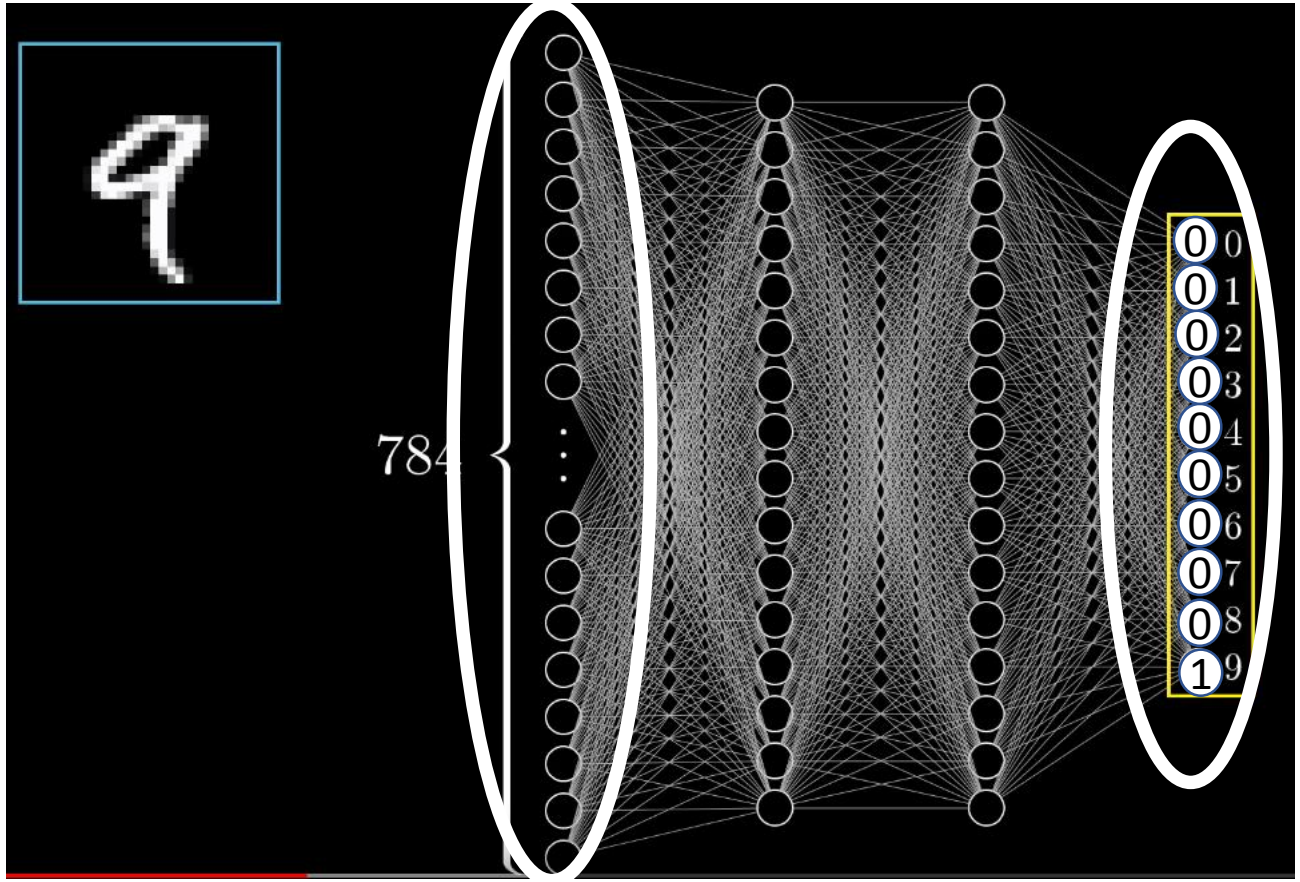
If the network is well trained

The 10th neuron in the output layer will have the max value (ideally 1)

Input becomes the first or the input layer of the neural network

Neurons in the last layer become the output layer of the neural network

Artificial intelligence – Artificial Neural Networks

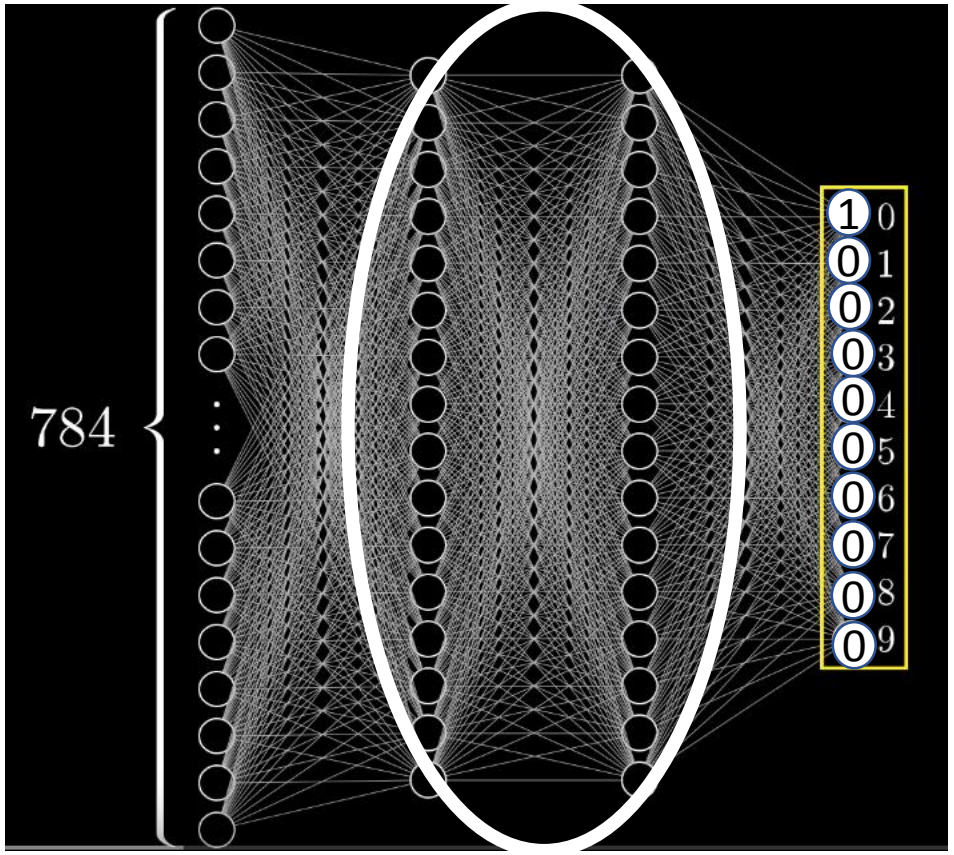


They can also have other values too e.g. 0.8 for the last neuron 0.1 or <0.8 values for other first nine neurons.

Input becomes the first or the input layer of the neural network

Neurons in the last layer become the output layer of the neural network

Artificial intelligence – Artificial Neural Networks



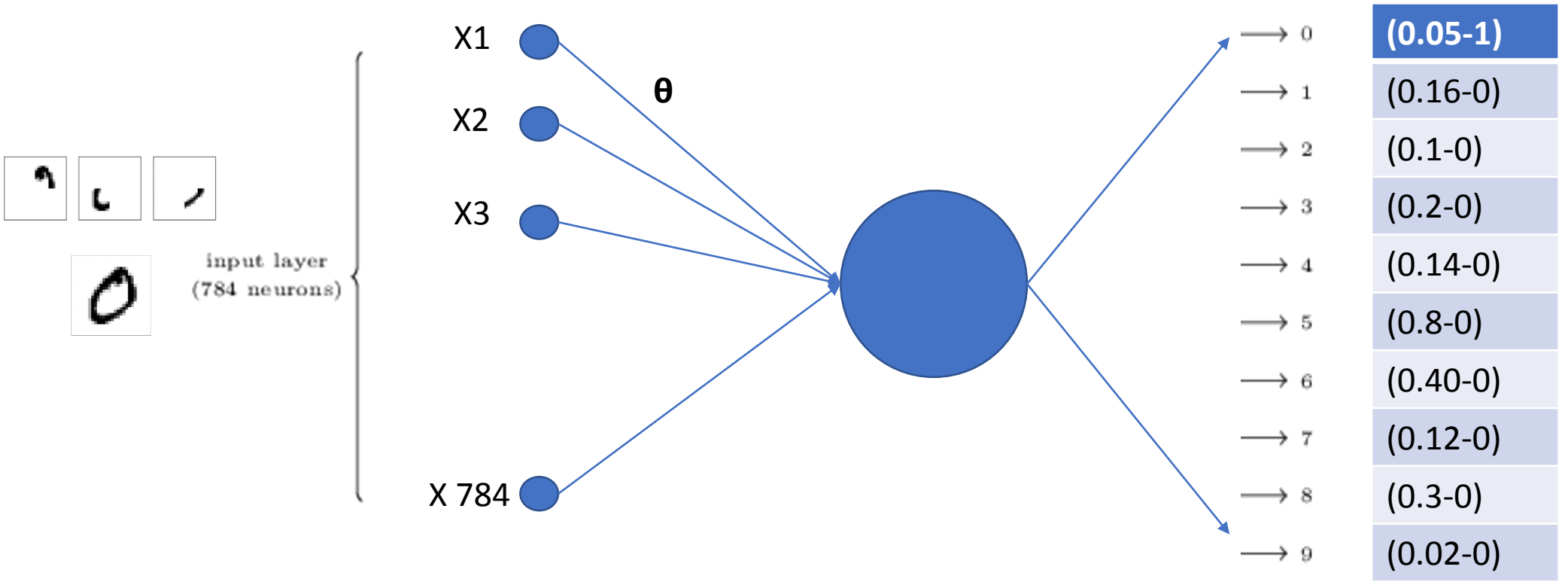
Hidden layers?

Implement MLP-ANN to classify digits

- MNIST:
  - 60, 000 training images → split into 50, 000 (training and testing)
  - 10, 000 (validation, will be used later for hyper-parameters optimization)

Let's first use a Single Neuron to recognize handwritten digits

Recall WS4 task 2



(0.05-1)
(0.16-0)
(0.1-0)
(0.2-0)
(0.14-0)
(0.8-0)
(0.40-0)
(0.12-0)
(0.3-0)
(0.02-0)

WS5 – Task 1
(MNIST classification using Batch GD (BGD) driven softmax regression)

- **Download mnist folder from Canvas**
- **The provided code will extract and pre-process the dataset**
- **The provided code also does the post-processing including thresholding, accuracy etc.**
- **For your ease, following lines are left empty for you to fill.**

err =
gradients =
theta =
y_predict =

Note that batch gradient descent computes the gradient using the whole dataset

1.1. Use BGD and complete the model
1.2. Explain in detail (step by step) the whole process in your own words, including:
- **raw data**
- **pre-processing (both X and Y (onehot encoded))**
- **Number of iterations/epochs**
- **Calculation of above 4 lines**

1.3. Test the model on test data and comment on the results in detail

WS5 – Task 2
(MNIST classification using Stochastic GD (SGD) driven softmax regression)

- **Download mnist.pkl.gz from Canvas**
- **The provided code will extract and pre-process the dataset**
- **The provided code also does the post-processing including thresholding, accuracy etc.**
- **For your ease, following lines are left empty for you to fill.**

err =
gradients =
Updated theta =
y_predict =

Note that stochastic gradient descent computes the gradient using small batches

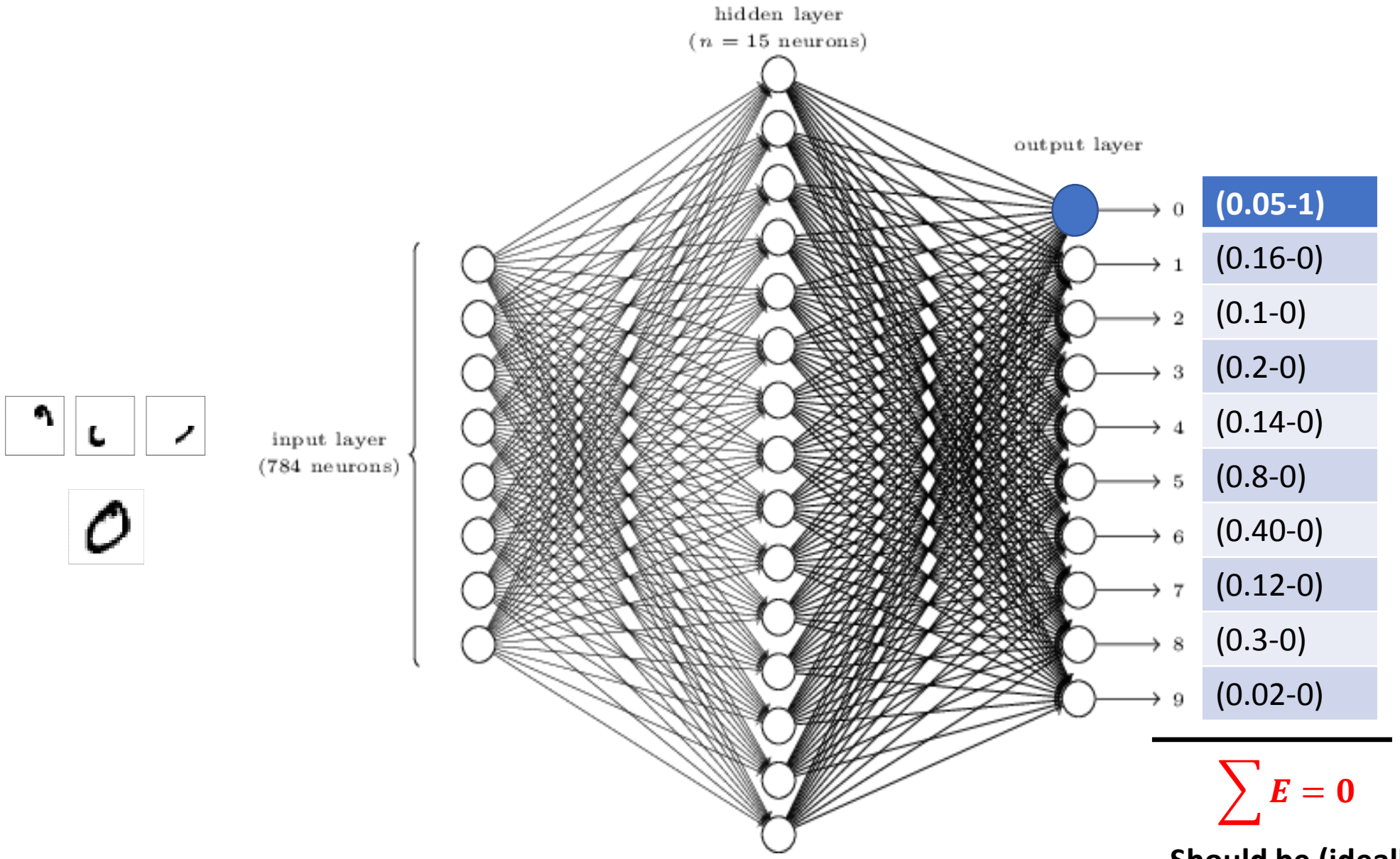
2.1. Use SGD and complete the model

2.2. Explain in detail (step by step) the whole process in your own words, including:

- **raw data**
- **Number of iterations/epochs**
- **pre-processing (mini-batches)**
- **Calculation of above 4 lines**
- **difference between task 1 and 2, in terms of accuracy**

2.3. Test the model on test data and comment on the results in detail

Now let's Use Multiple Neurons to recognize handwritten digits

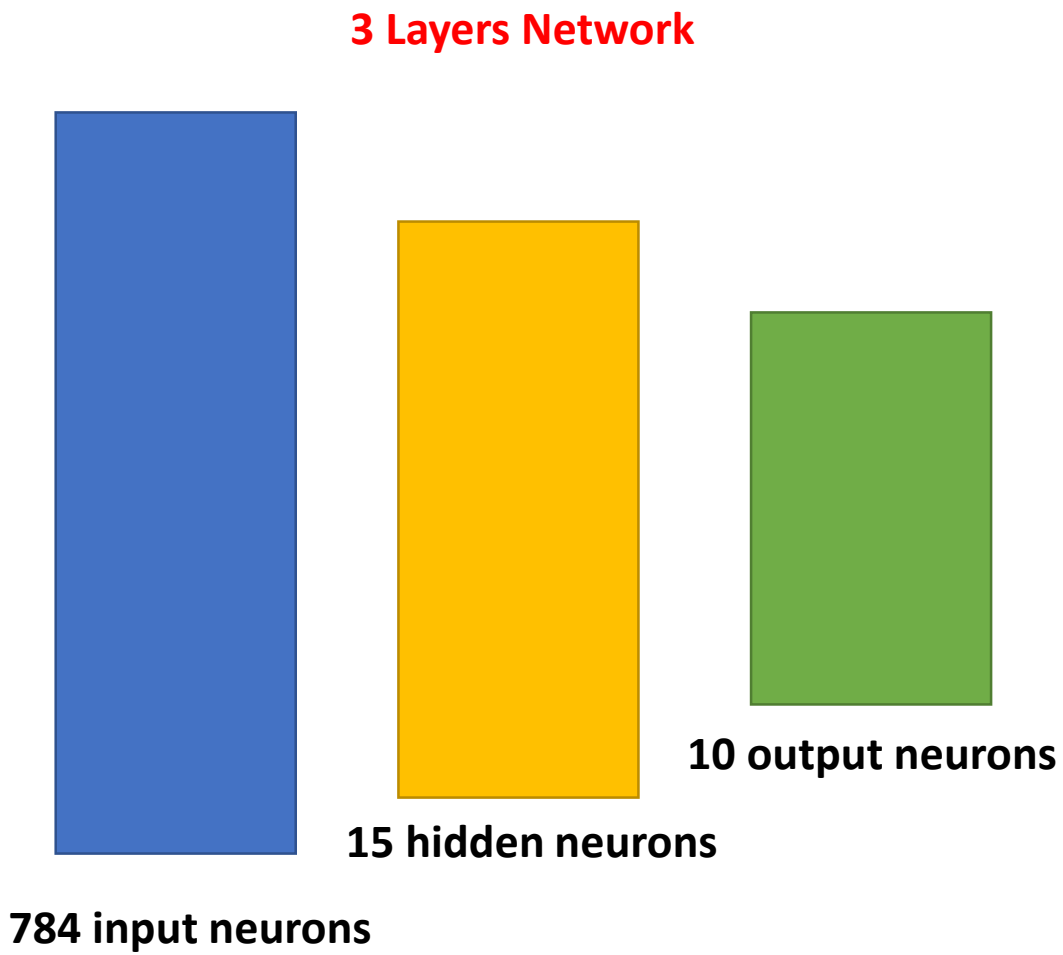


0	**(0.05-1)**
1	(0.16-0)
2	(0.1-0)
3	(0.2-0)
4	(0.14-0)
5	(0.8-0)
6	(0.40-0)
7	(0.12-0)
8	(0.3-0)
9	(0.02-0)

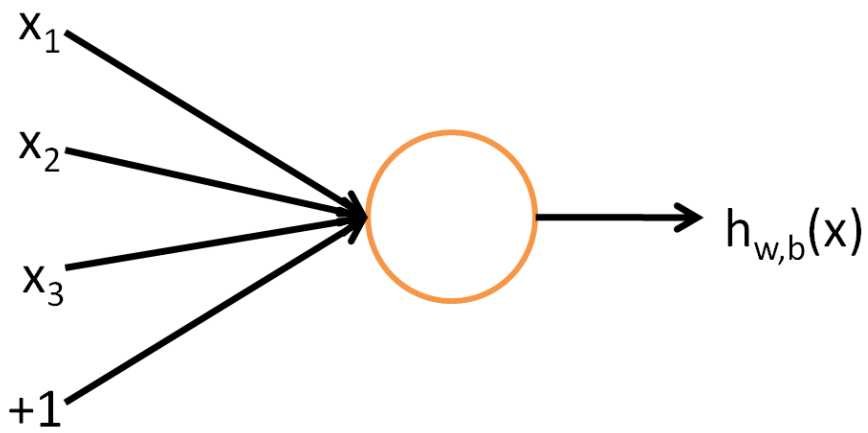
$$\sum \boldsymbol{E} = \mathbf{0}$$

Should be (ideally)

Now let’s Use Multiple Neurons to recognize handwritten digits



Multi-Layer Neural Network – A formal overview

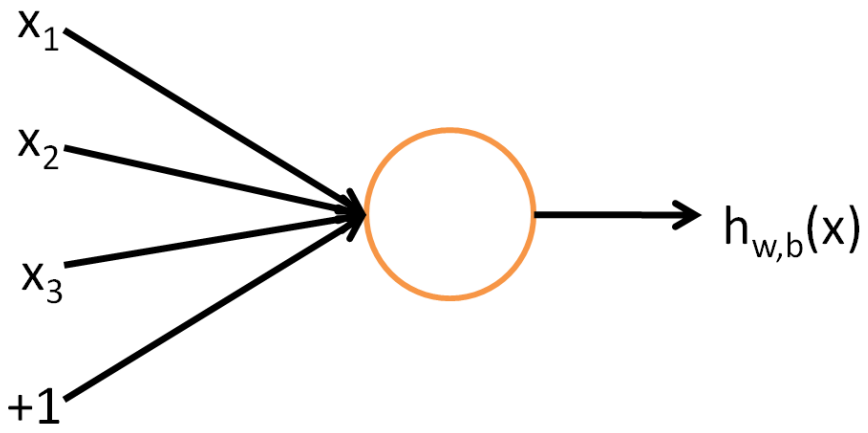


The “neuron” is a computational unit that takes as input x1,x2,x3 (and a +1 intercept term), and outputs:

$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b), \text{ where } f : \Re \mapsto \Re \text{ is called the } \textbf{activation function.}$$

$$f(z) = \frac{1}{1 + \exp(-z)}.$$

Multi-Layer Neural Network – A formal overview



Can you recall the **bias** term from Lecture 1 or 2?

Is it different from linear and logistic regression?

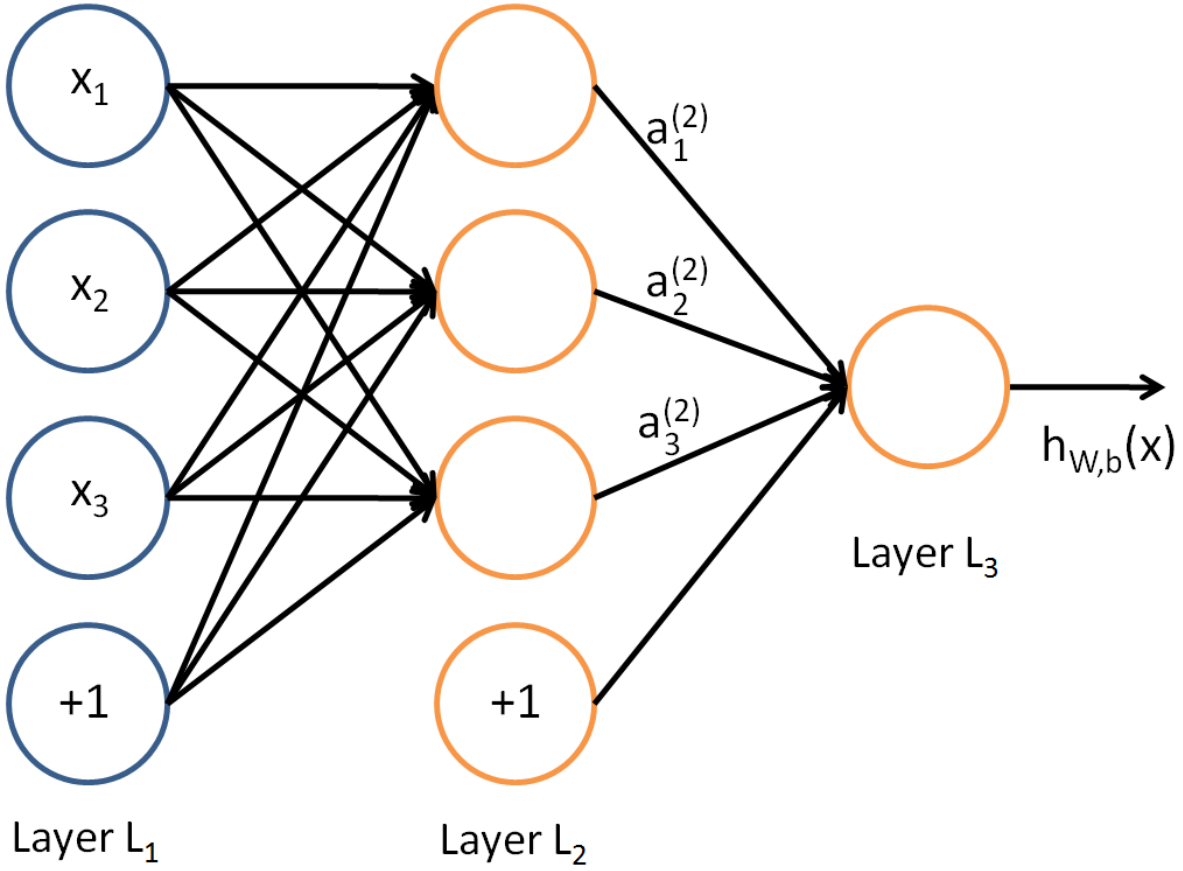
The "neuron" is a computational unit that takes as input x1,x2,x3 (and a +1 intercept term), and outputs:

$$h_{W,b}(x) = f(W^T x) = f(\sum_{i=1}^{3} W_i x_i + b), \text{ where } f : \Re \mapsto \Re \text{ is called the } \textbf{activation function.}$$

$$f(z) = \frac{1}{1 + \exp(-z)}.$$

Multi-Layer Neural Network – A formal overview

A neural network is put together by hooking together many of our simple “neurons,” so that the output of a neuron can be the input of another.



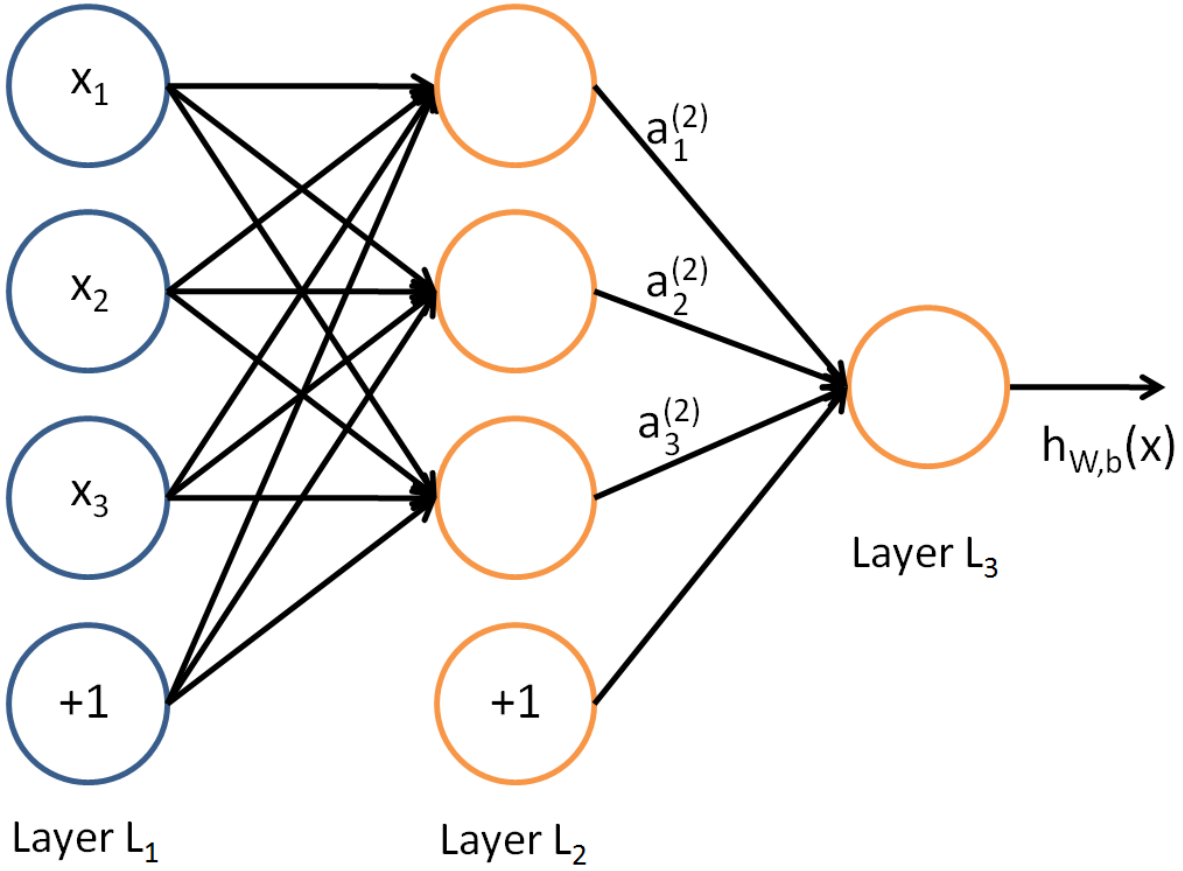
Multi-Layer Neural Network – A formal overview

"+1" are called **bias units**

The leftmost layer is called the **input layer**

The rightmost layer is the **output layer**

The middle layer is called the **hidden layer**, why?



Lecture 6:
MLP - **Feedforward/ forward propagation**
Backpropagation, and
MNIST classification

References:

- http://deeplearning.stanford.edu/tutorial
- Huff, Trevor, and Scott C. Dulebohn. "Neuroanatomy, Visual Cortex." (2017).
- Ian Goodfellow and Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016, url: http://www.deeplearningbook.org
- https://www.khanacademy.org/math/statistics-probabilit
- 3blue1brown: https://www.3blue1brown.com/
- http://neuralnetworksanddeeplearning.com/index.html