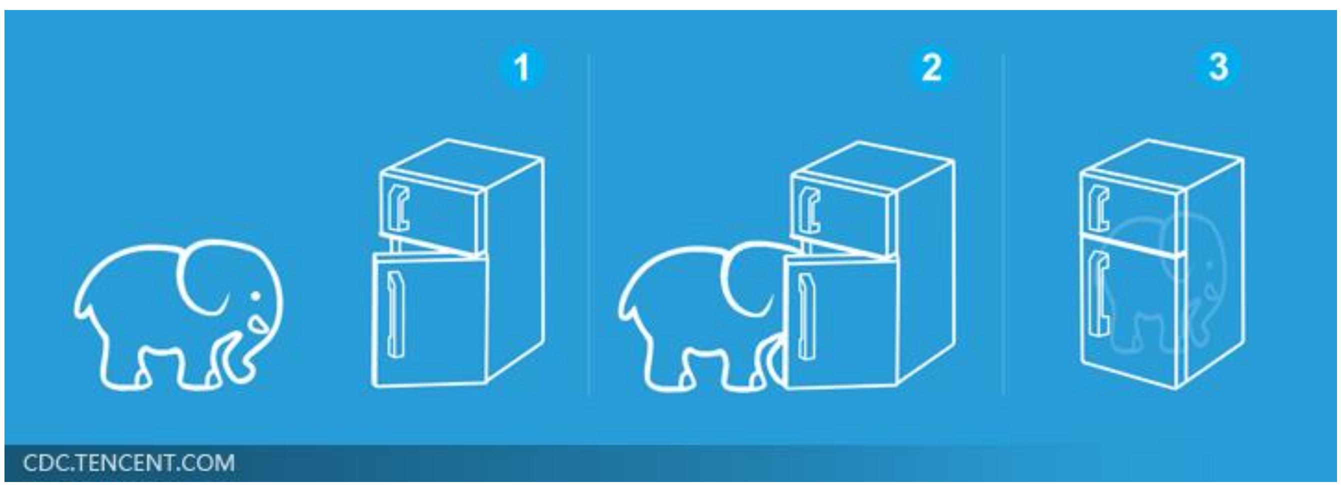
# Deep Learning

Hung-yi Lee
李宏毅

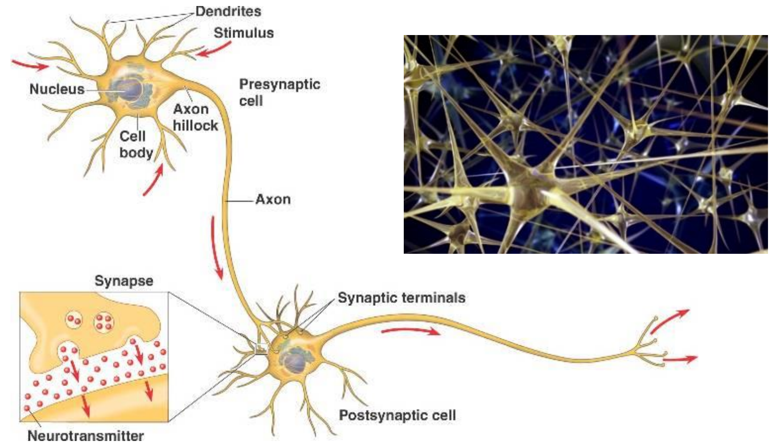
# Three Steps for Deep Learning

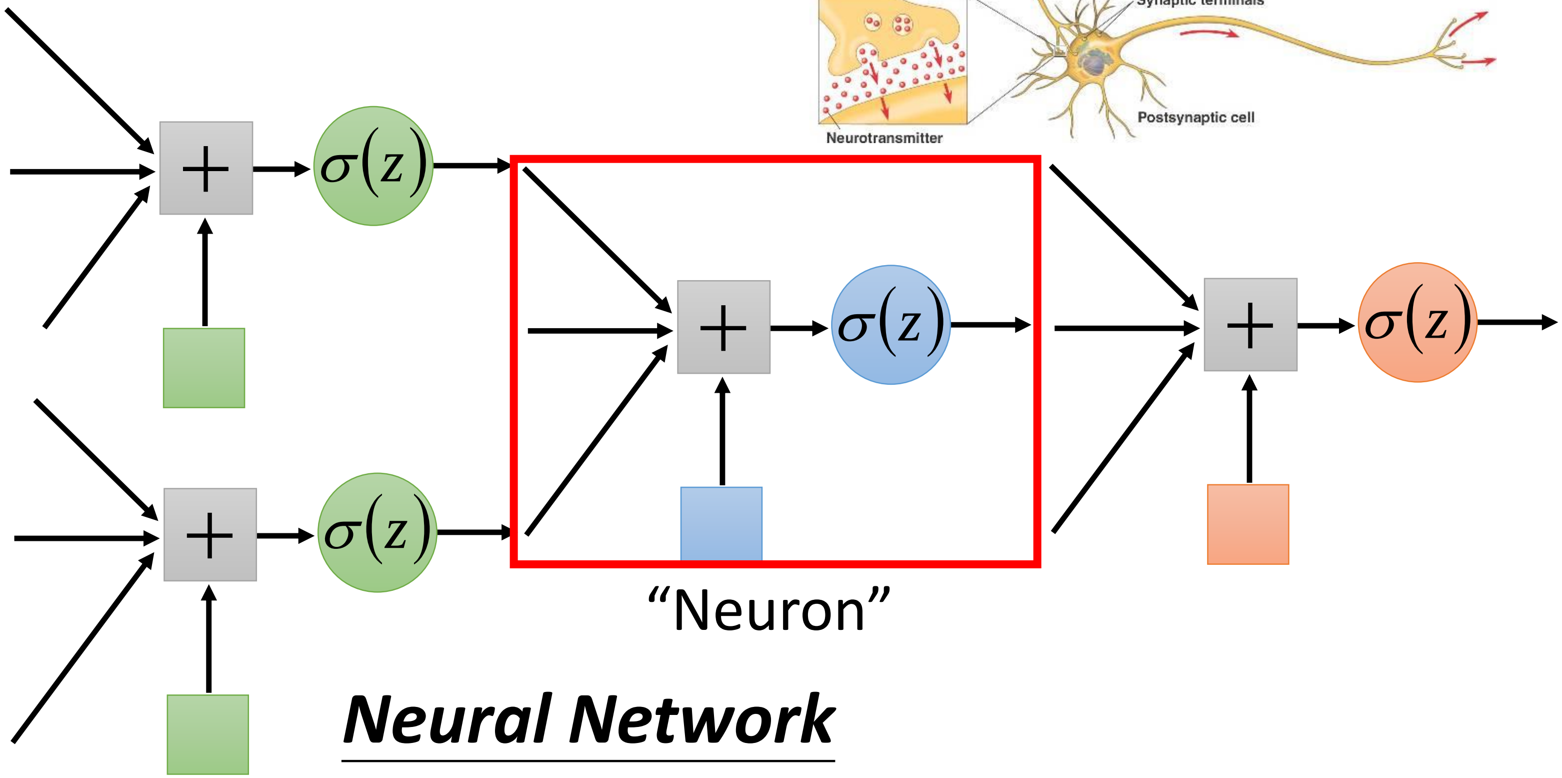
Step 1: Neural Network → Step 2: goodness of function → Step 3: pick the best function

Deep Learning is so simple ……



# Neural Network



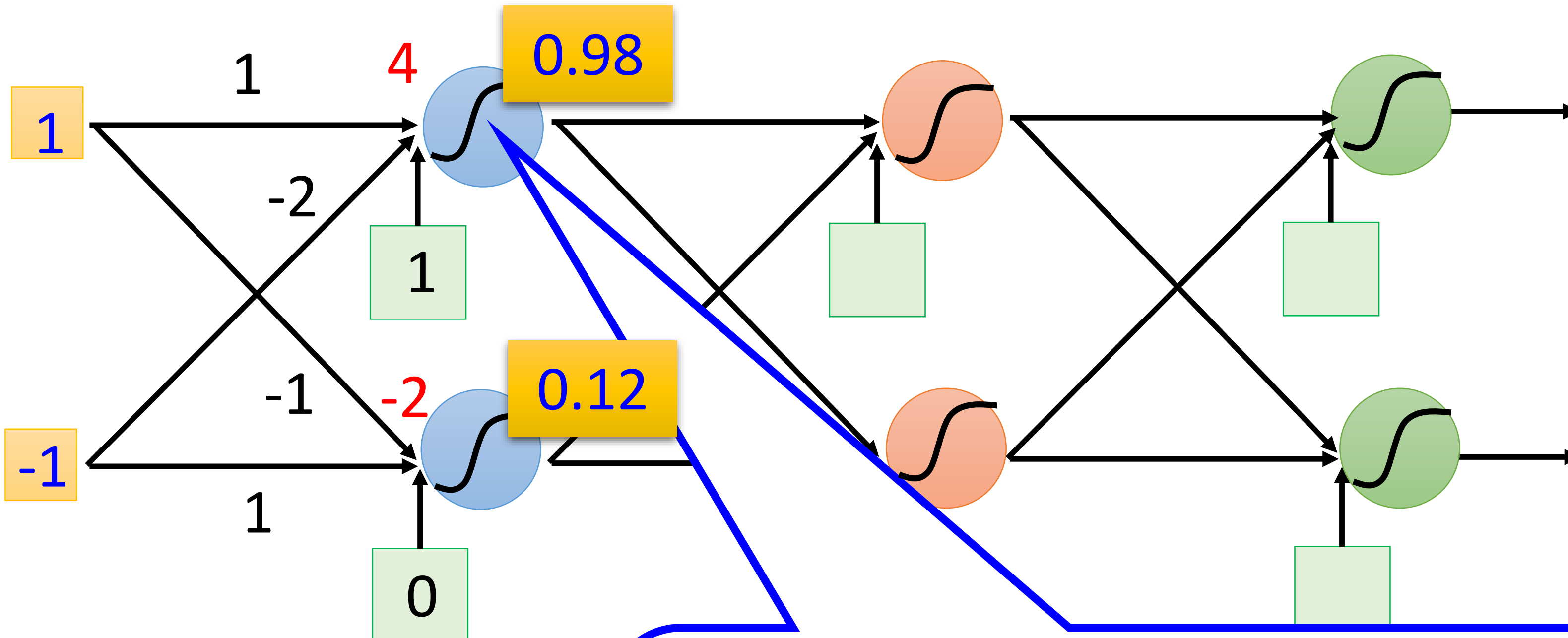




***Neural Network***

Different connection leads to different network structures

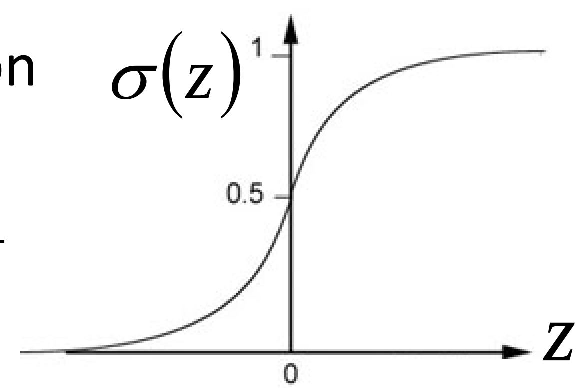
Network parameter $\theta$: all the weights and biases in the "neurons"

# Fully Connect Feedforward Network

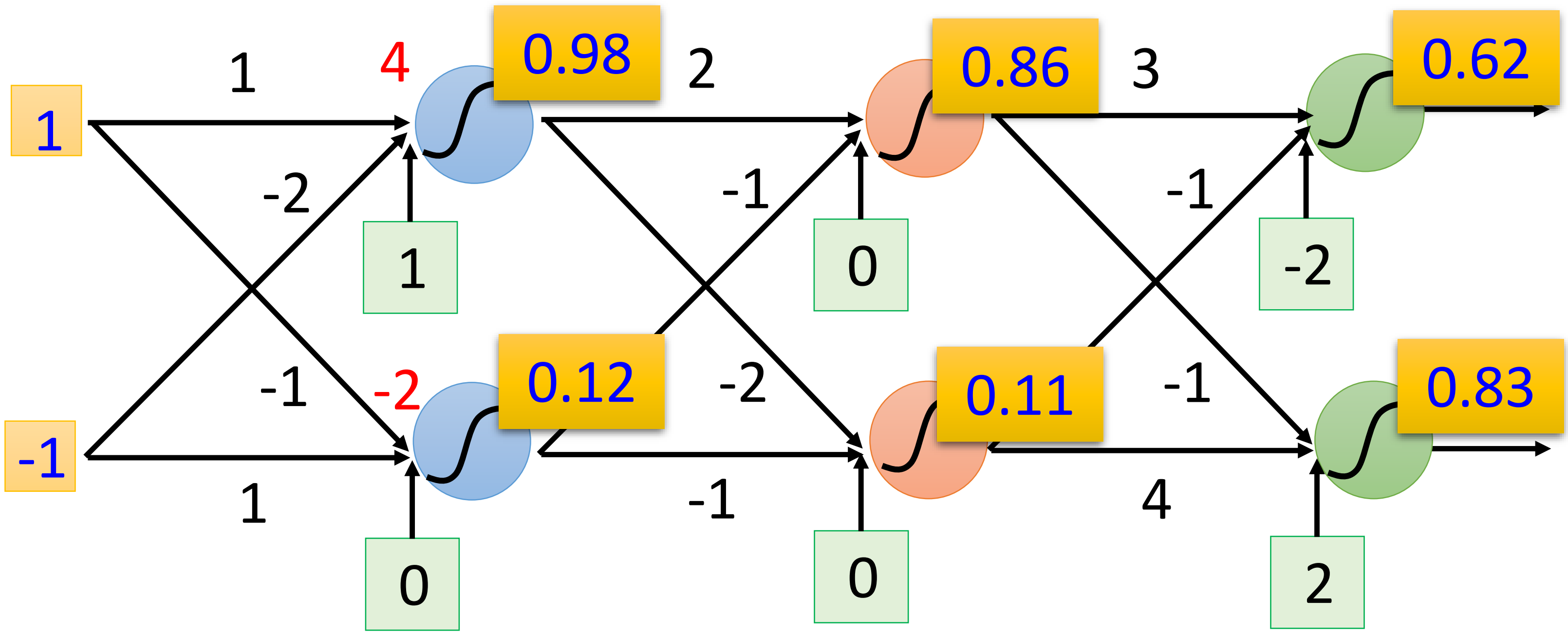



Sigmoid Function

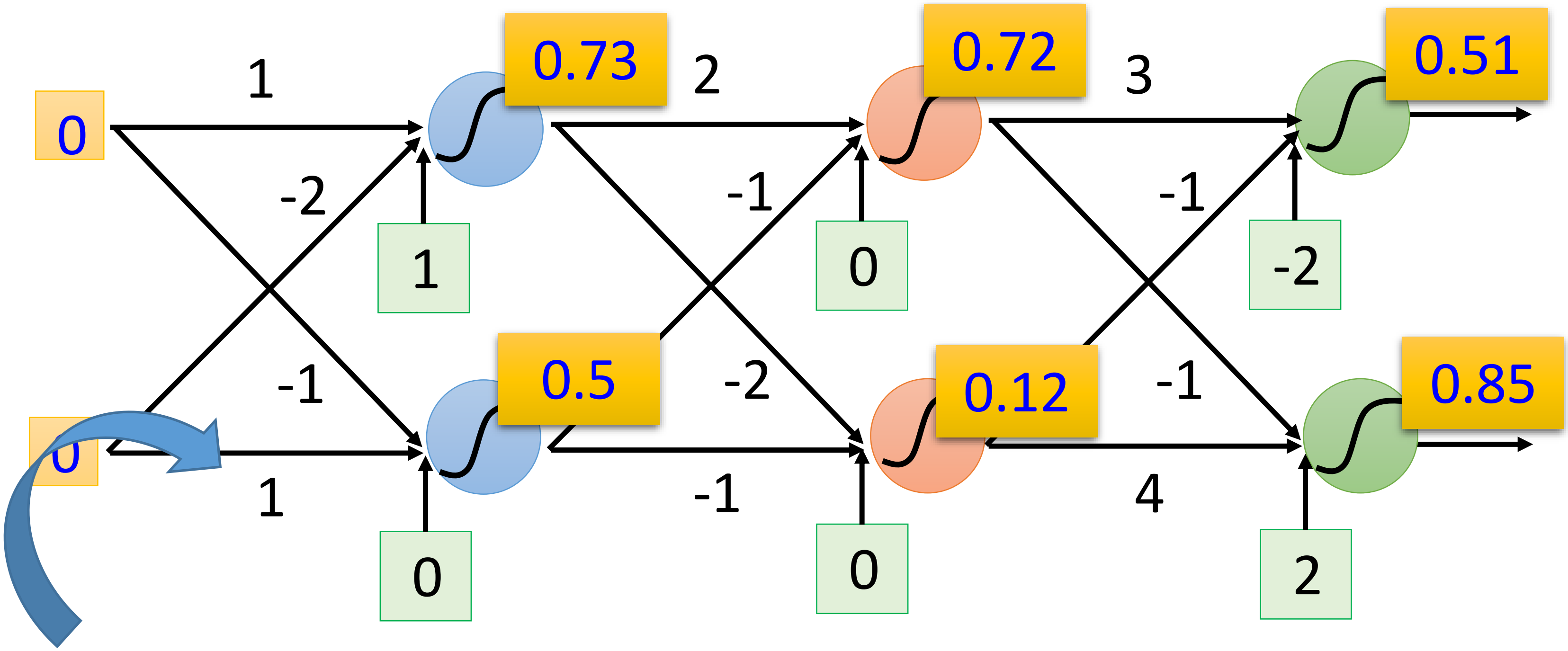
$$\sigma(z) = \frac{1}{1+e^{-z}}$$



# Fully Connect Feedforward Network



# Fully Connect Feedforward Network



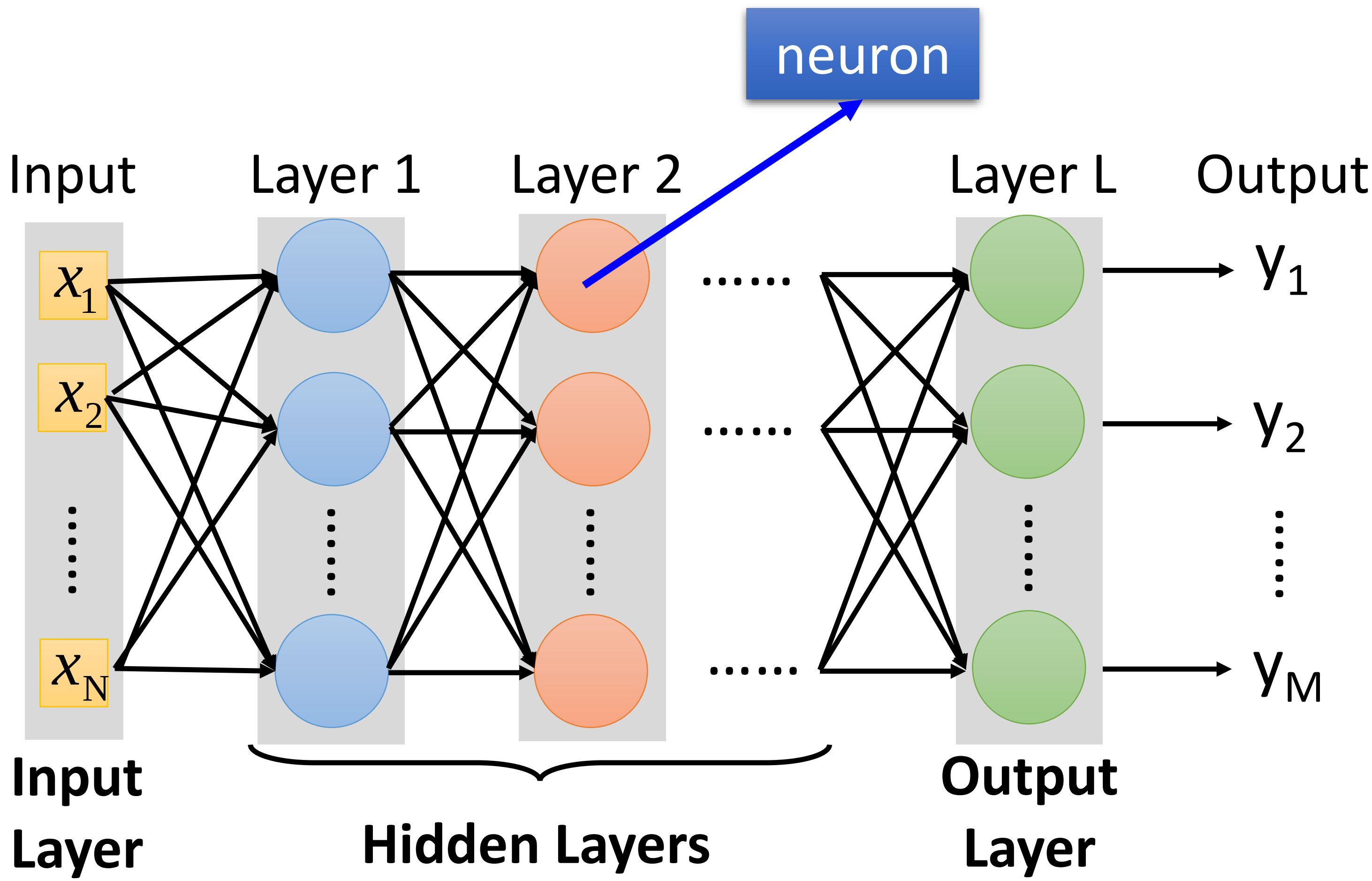

This is a function.

Input vector, output vector

$$f\left(\begin{bmatrix}1\\-1\end{bmatrix}\right)=\begin{bmatrix}0.62\\0.83\end{bmatrix} \quad f\left(\begin{bmatrix}0\\0\end{bmatrix}\right)=\begin{bmatrix}0.51\\0.85\end{bmatrix}$$

Given network structure, define ***a function set***

# Fully Connect Feedforward Network

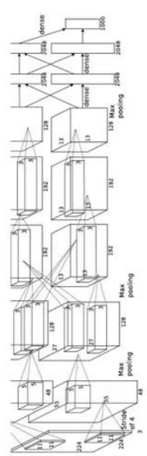


# Deep = Many hidden layers

http://cs231n.stanford.edu/slides/winter1516_lecture8.pdf

8 layers

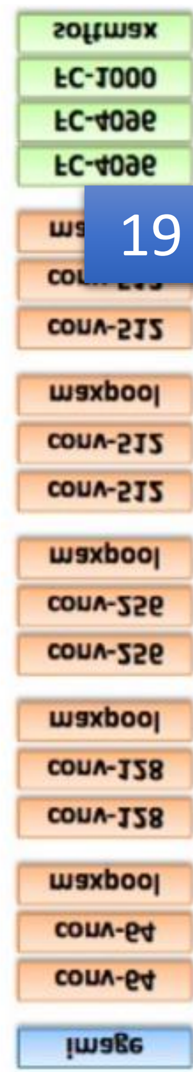
16.4%



AlexNet (2012)

19 layers

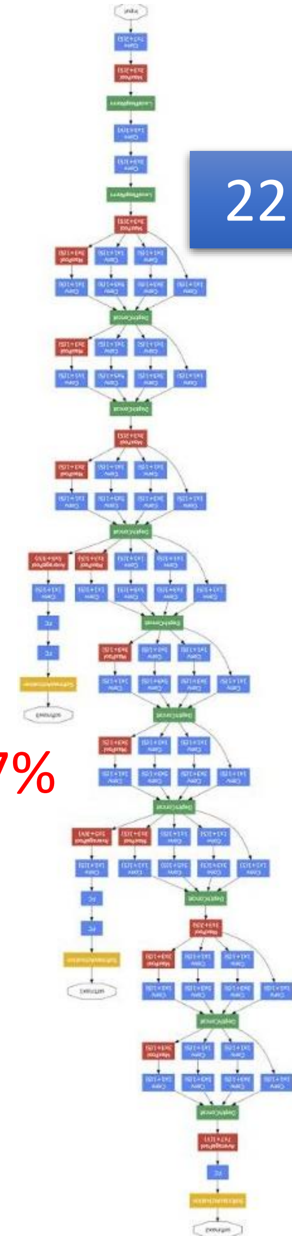
7.3%




VGG (2014)

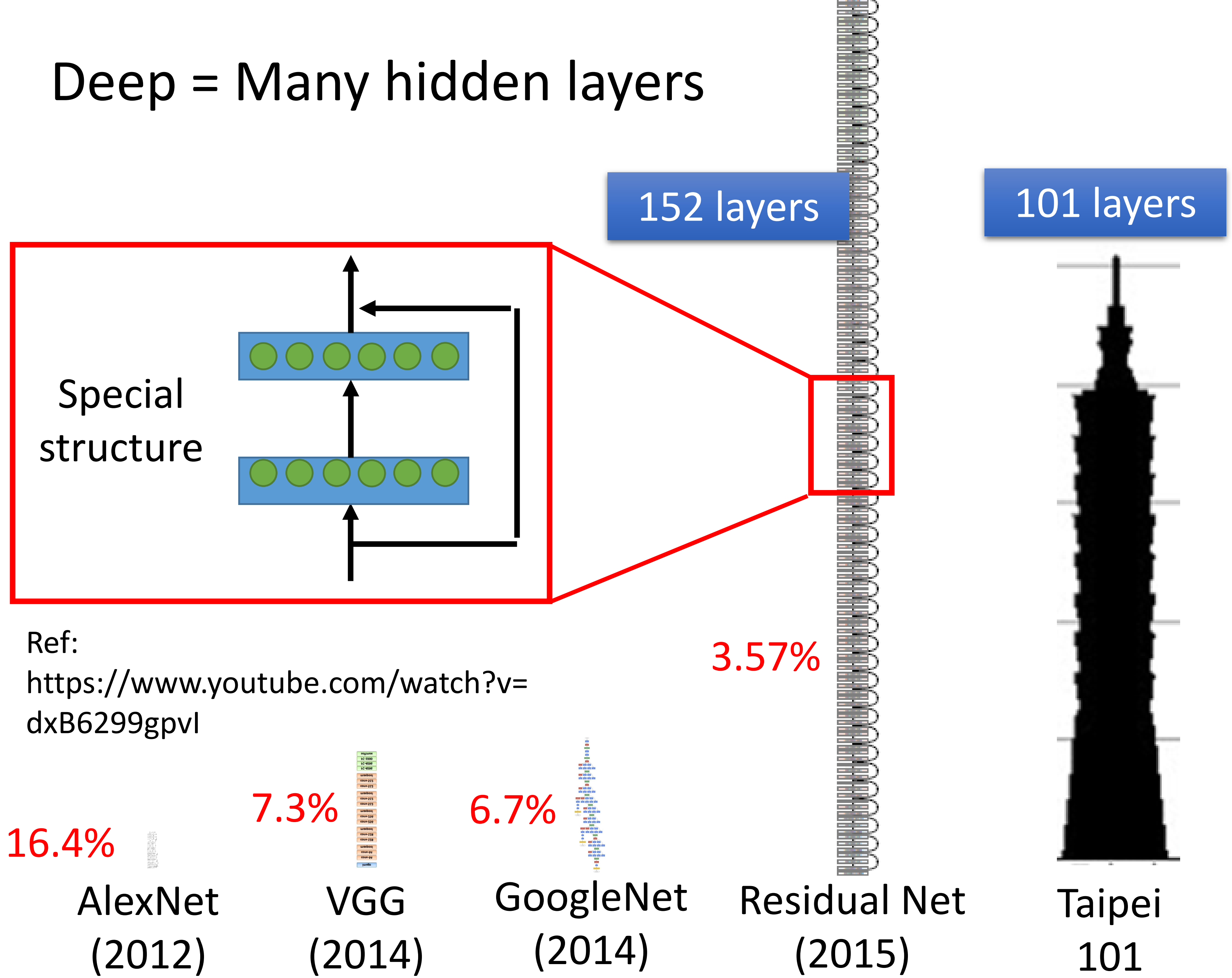
22 layers

6.7%



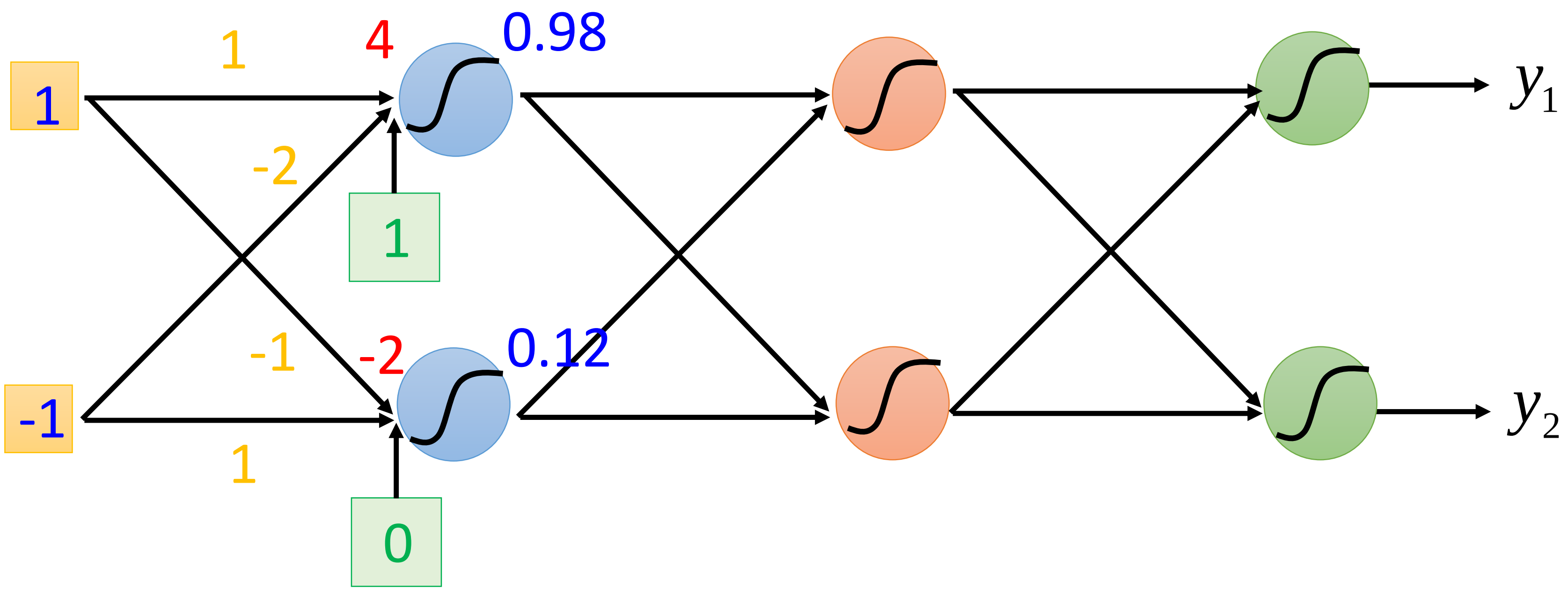
GoogleNet (2014)

# Deep = Many hidden layers



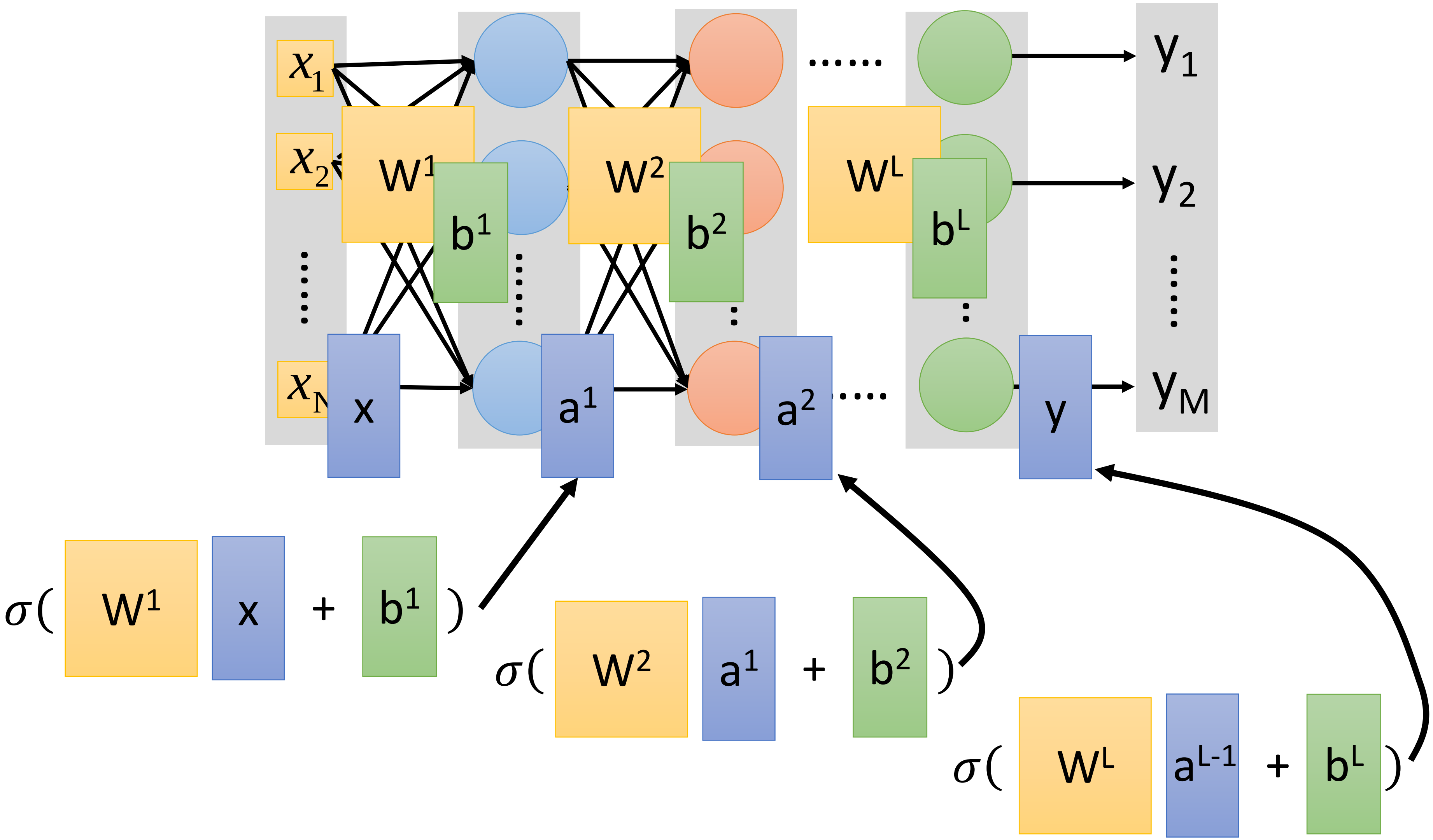

Ref:
https://www.youtube.com/watch?v=dxB6299gpvI

# Matrix Operation

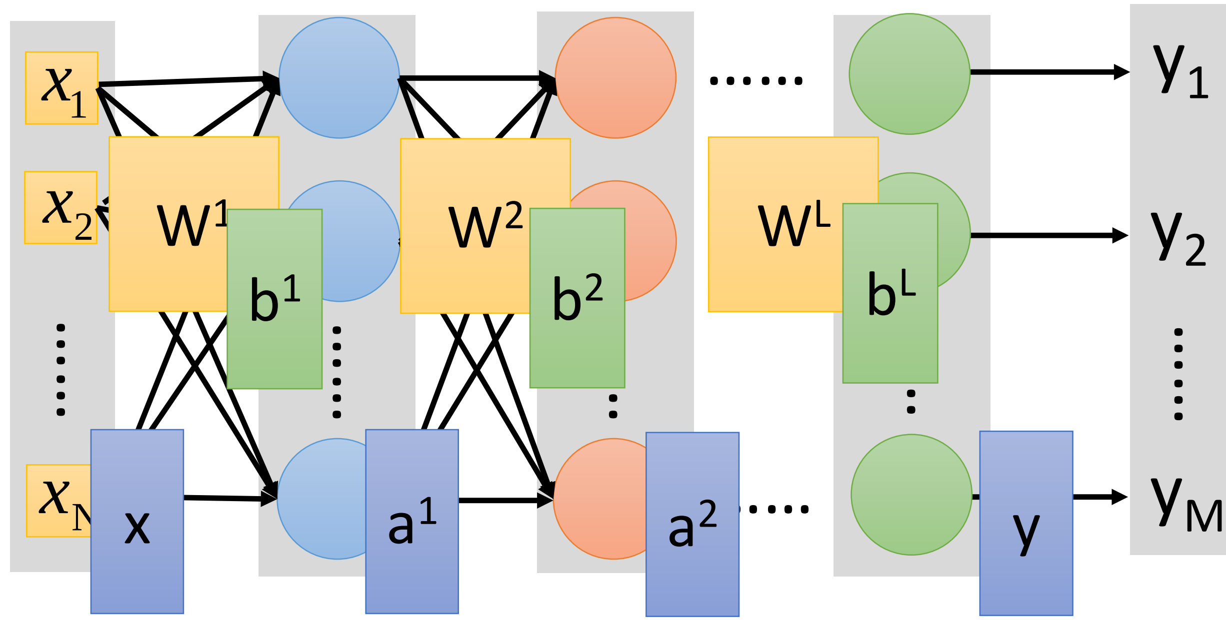



$$\sigma\left(\underbrace{\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\begin{bmatrix} 4 \\ -2 \end{bmatrix}}\right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

# Neural Network



# Neural Network

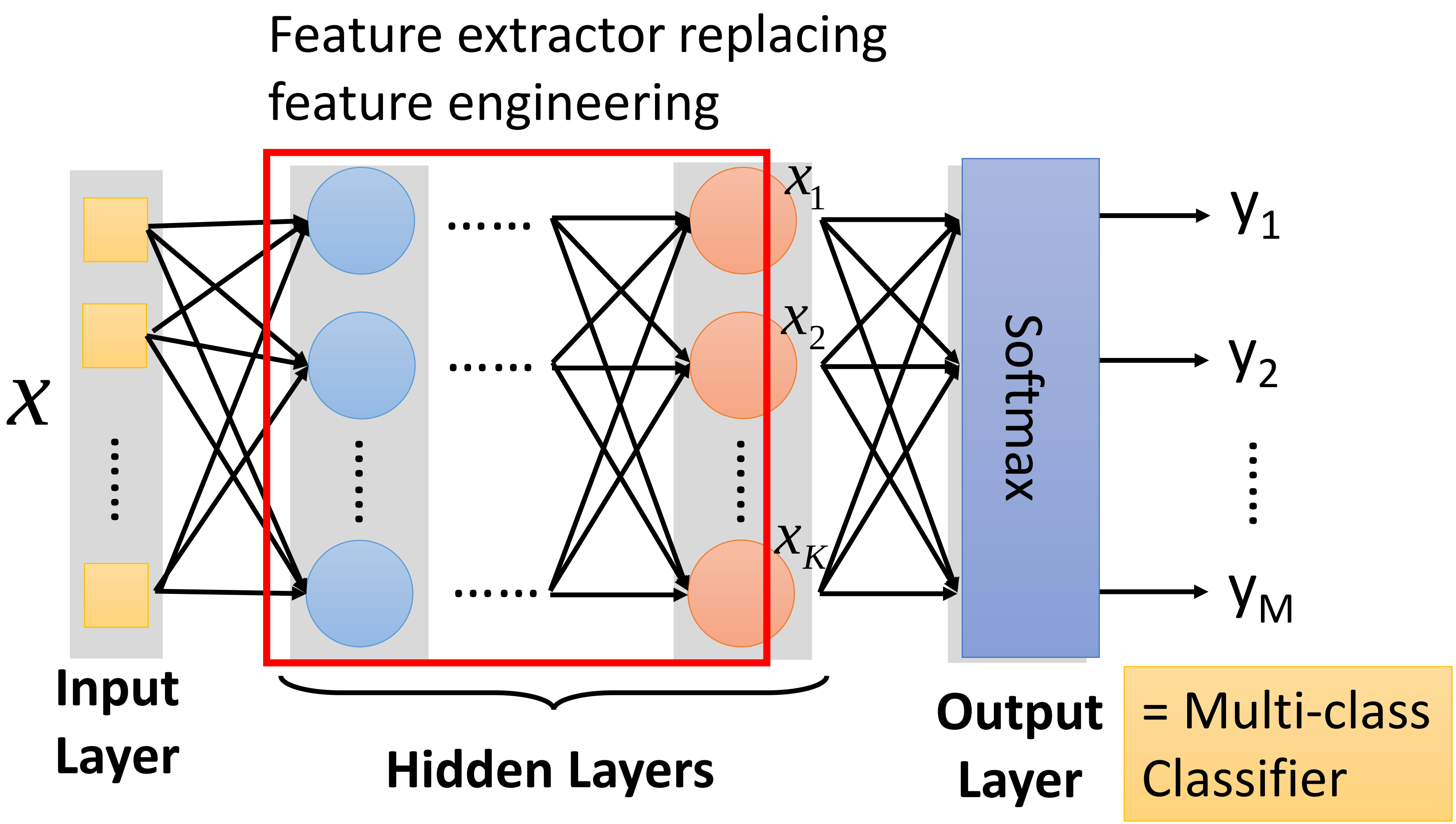



$$y = f(x)$$

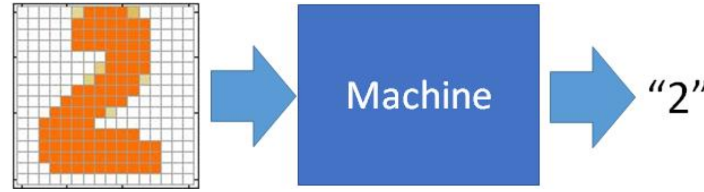
Using parallel computing techniques to speed up matrix operation

$$= \sigma(W^L \cdots \sigma(W^2 \sigma(W^1 x + b^1) + b^2) \cdots + b^L)$$

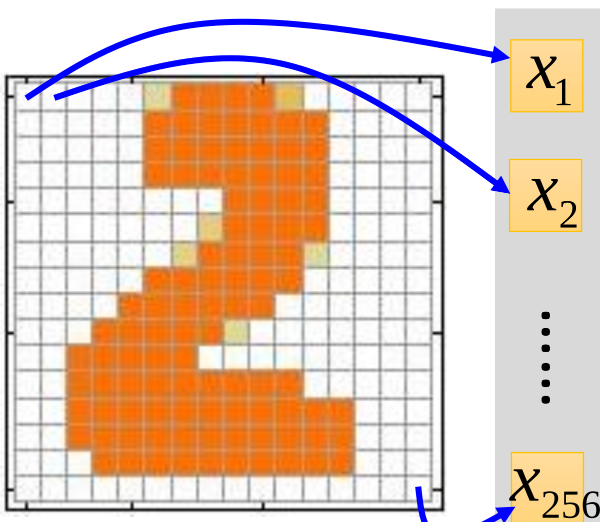
# Output Layer as Multi-Class Classifier



# Example Application




## Input

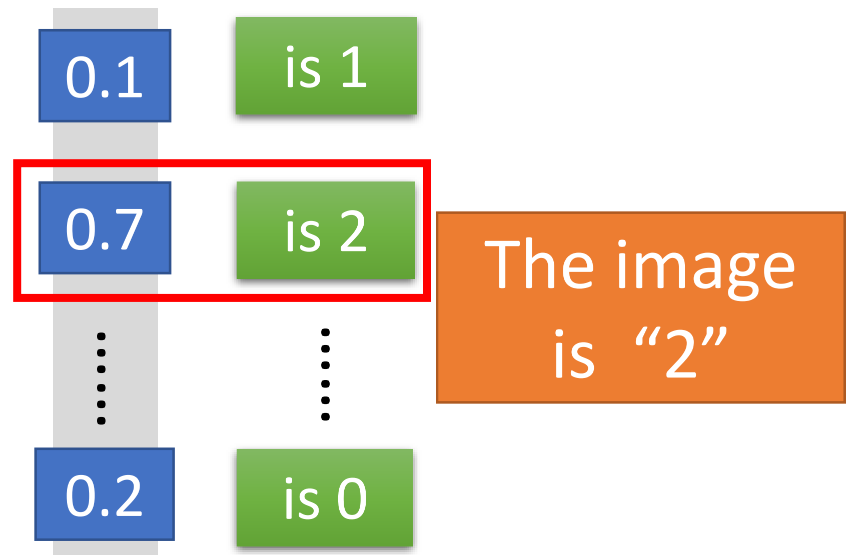



16 x 16 = 256

Ink → 1

No ink → 0

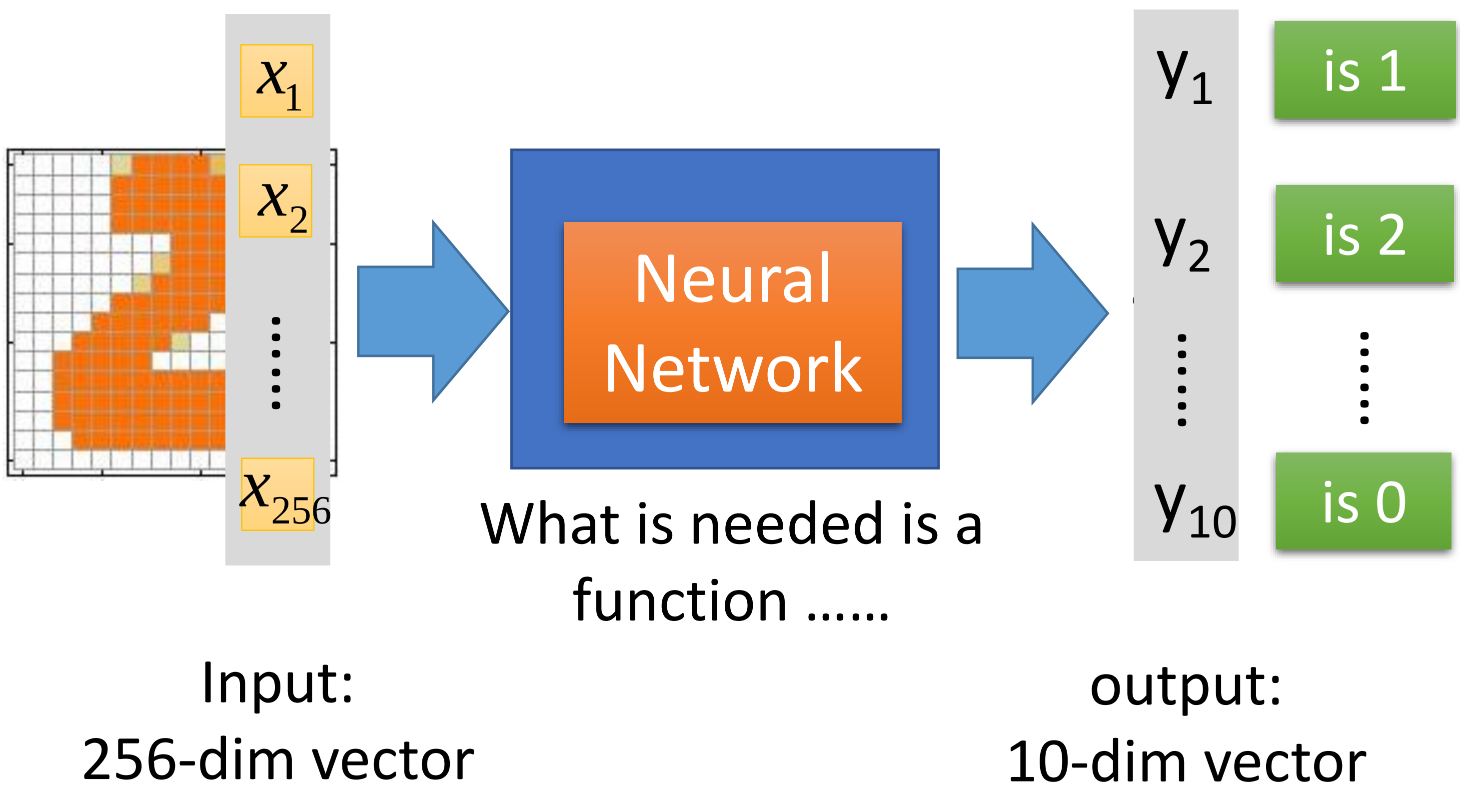
## Output



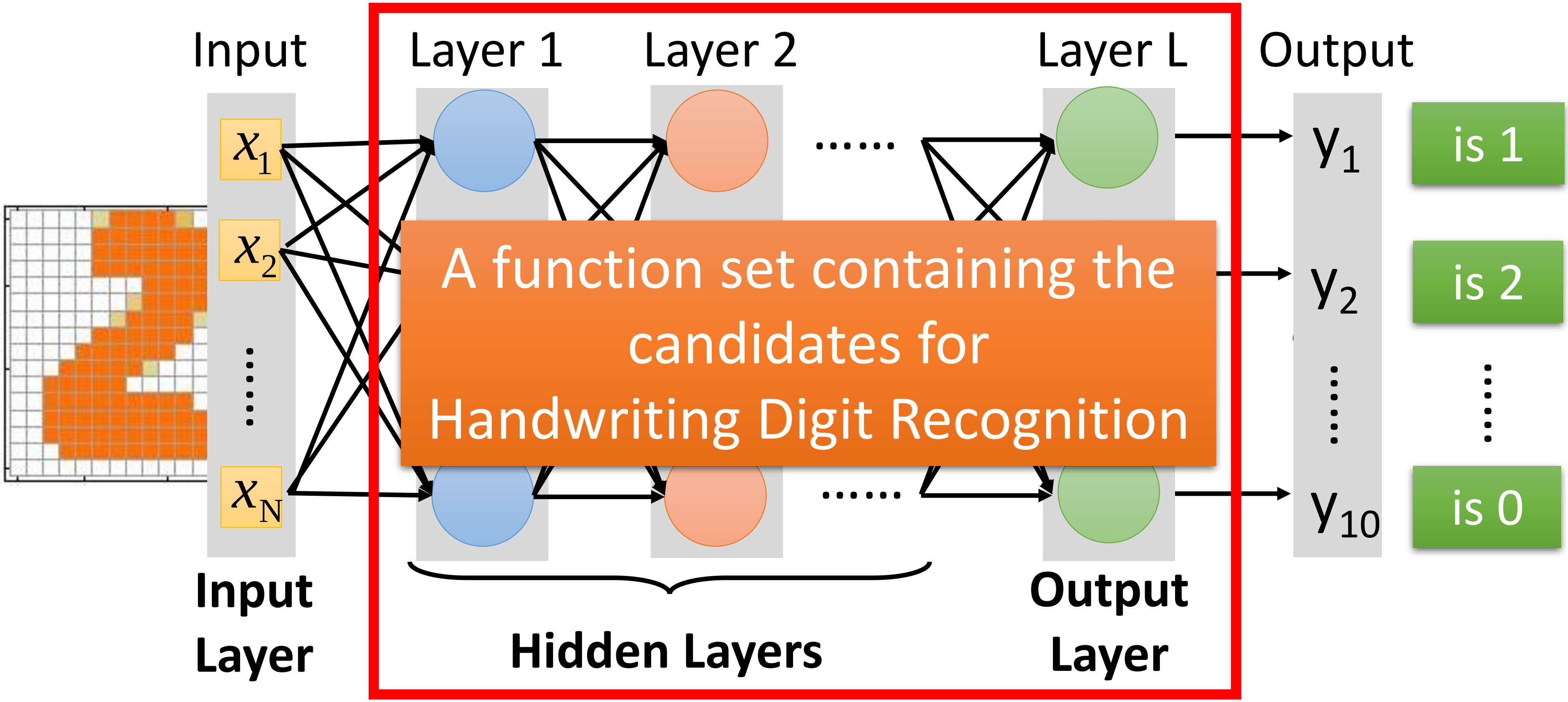

Each dimension represents the confidence of a digit.

# Example Application

- Handwriting Digit Recognition

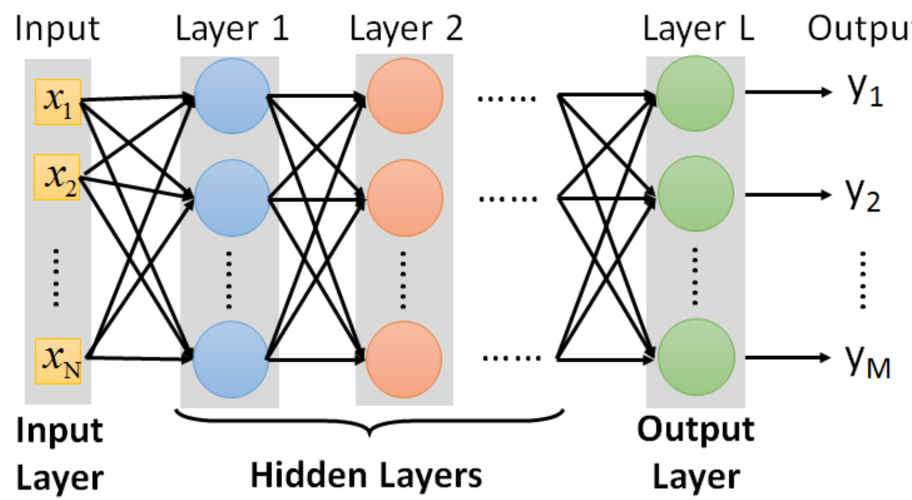


# Example Application




You need to decide the network structure to let a good function in your function set.

# FAQ




- Q: How many layers? How many neurons for each layer?

Trial and Error + Intuition

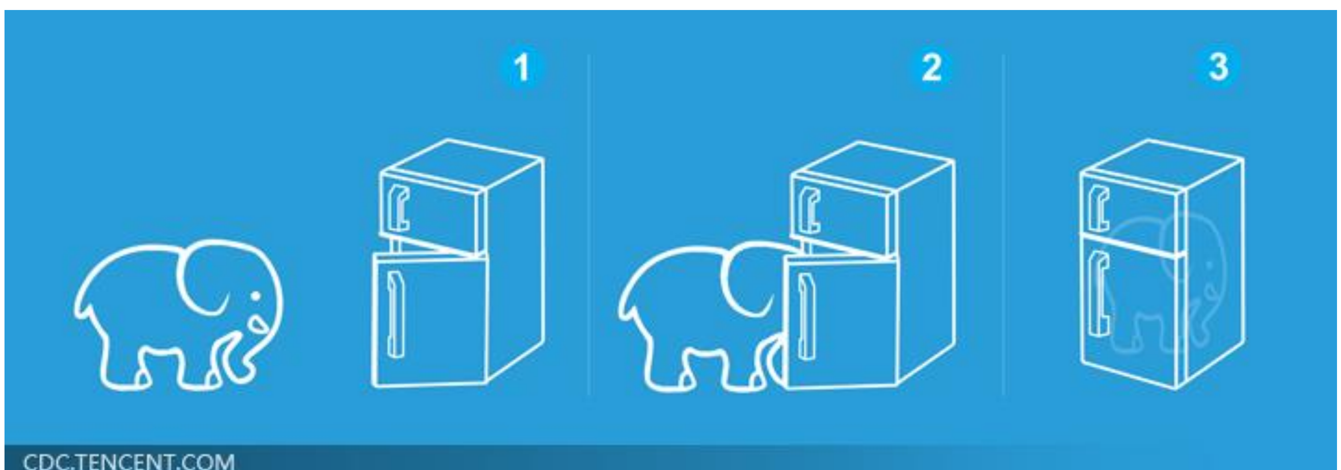
- Q: Can the structure be automatically determined?
  - E.g. Evolutionary Artificial Neural Networks
- Q: Can we design the network structure?

Convolutional Neural Network (CNN)

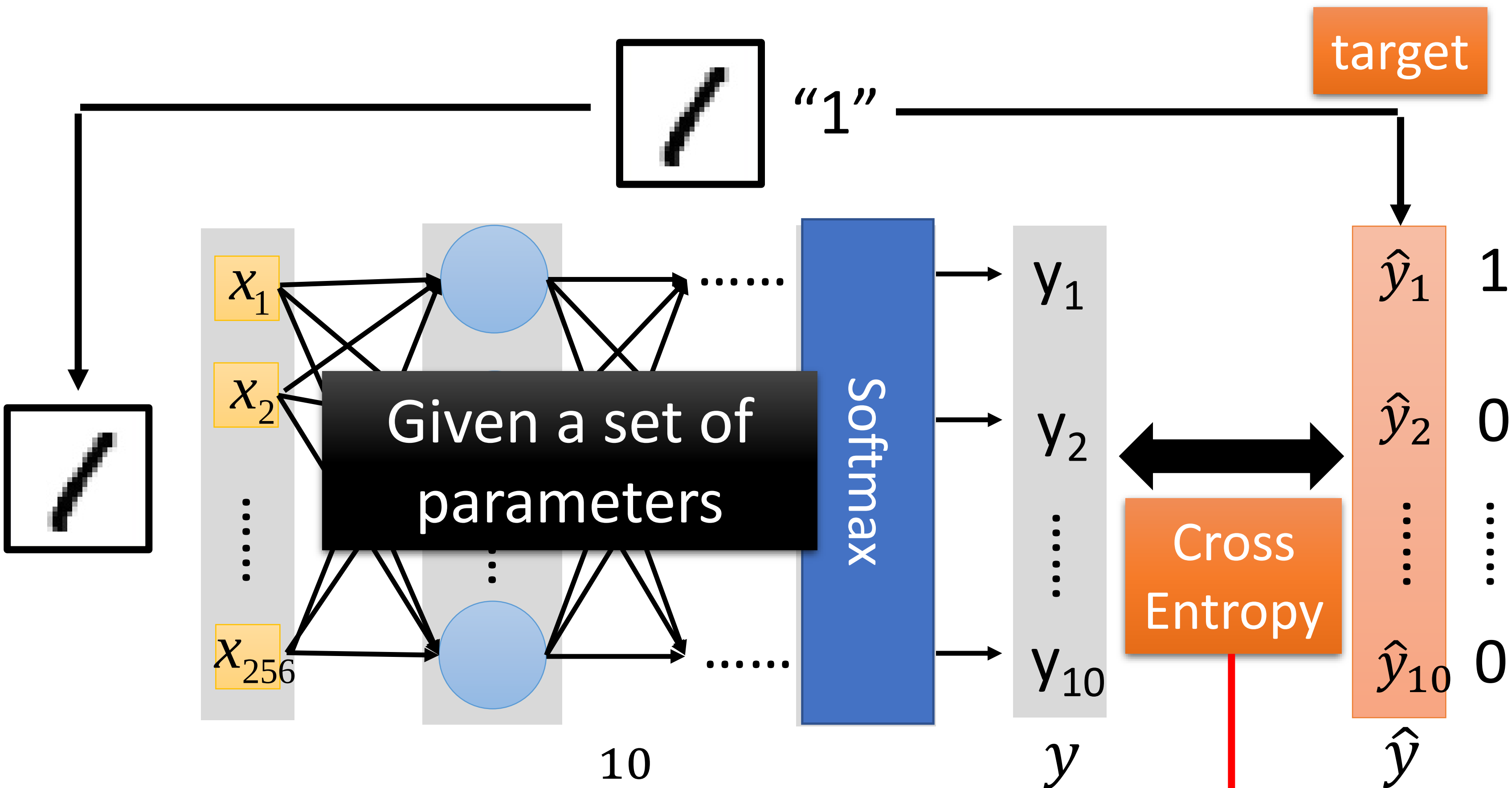
# Three Steps for Deep Learning

Step 1: Neural Network → Step 2: goodness of function → Step 3: pick the best function

Deep Learning is so simple ……



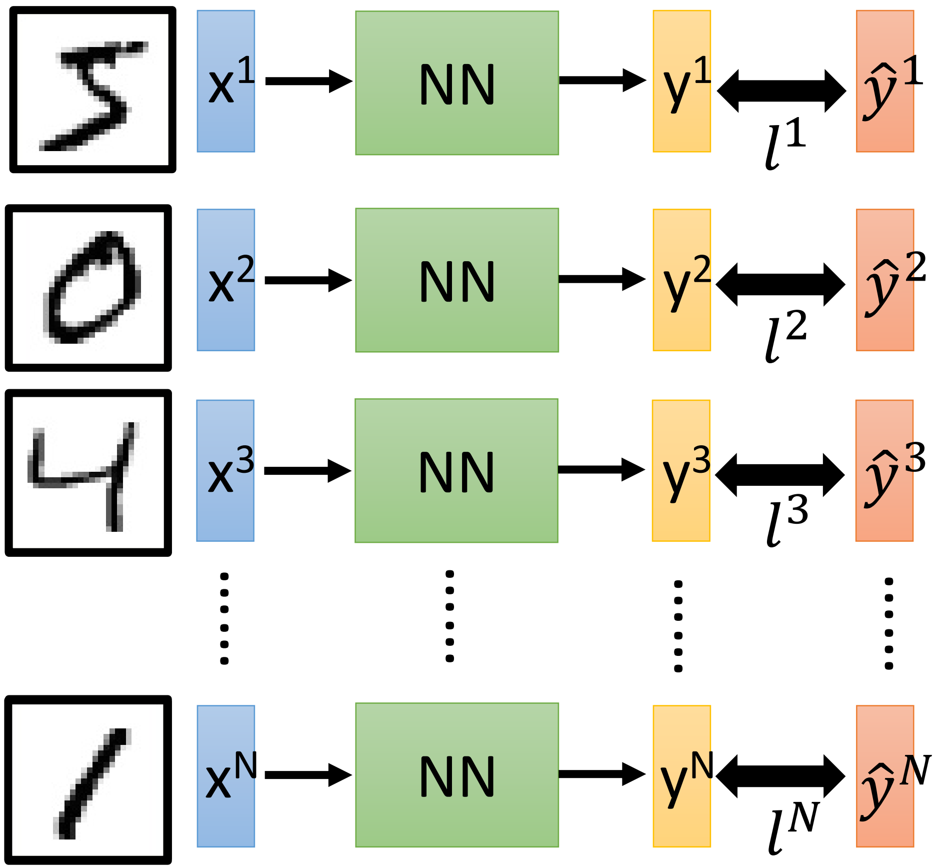
# Loss for an Example




$$l(y, \hat{y}) = -\sum_{i=1}^{10} \hat{y}_i \ln y_i$$

# Total Loss

For all training data …




Total Loss:

$$\boxed{L = \sum_{n=1}^{N} l^n}$$

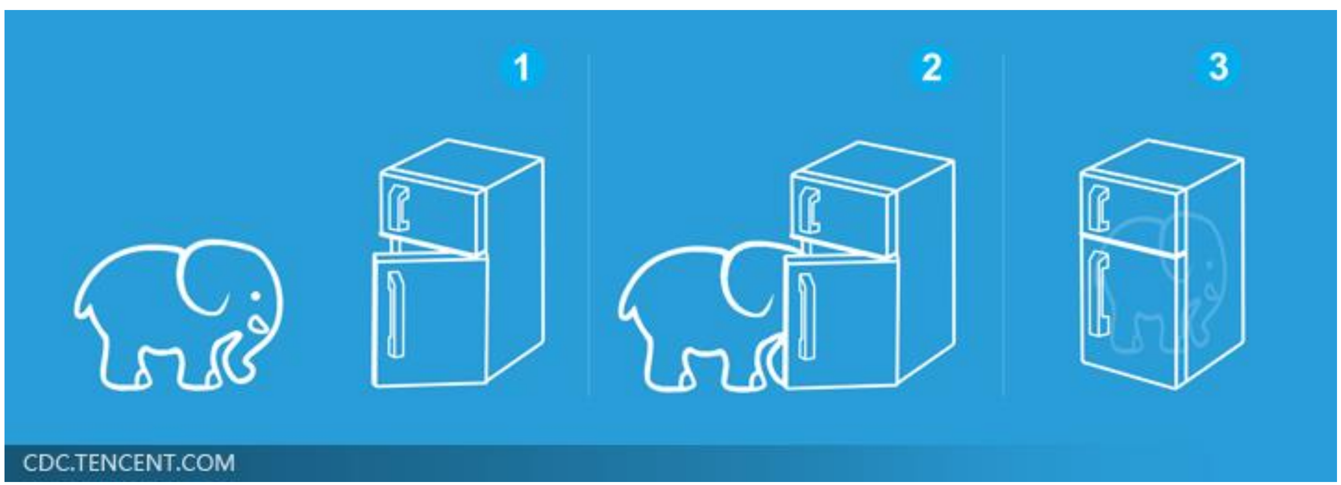
Find ***a function in function set*** that minimizes total loss L

Find ***the network parameters $\theta^*$*** that minimize total loss L

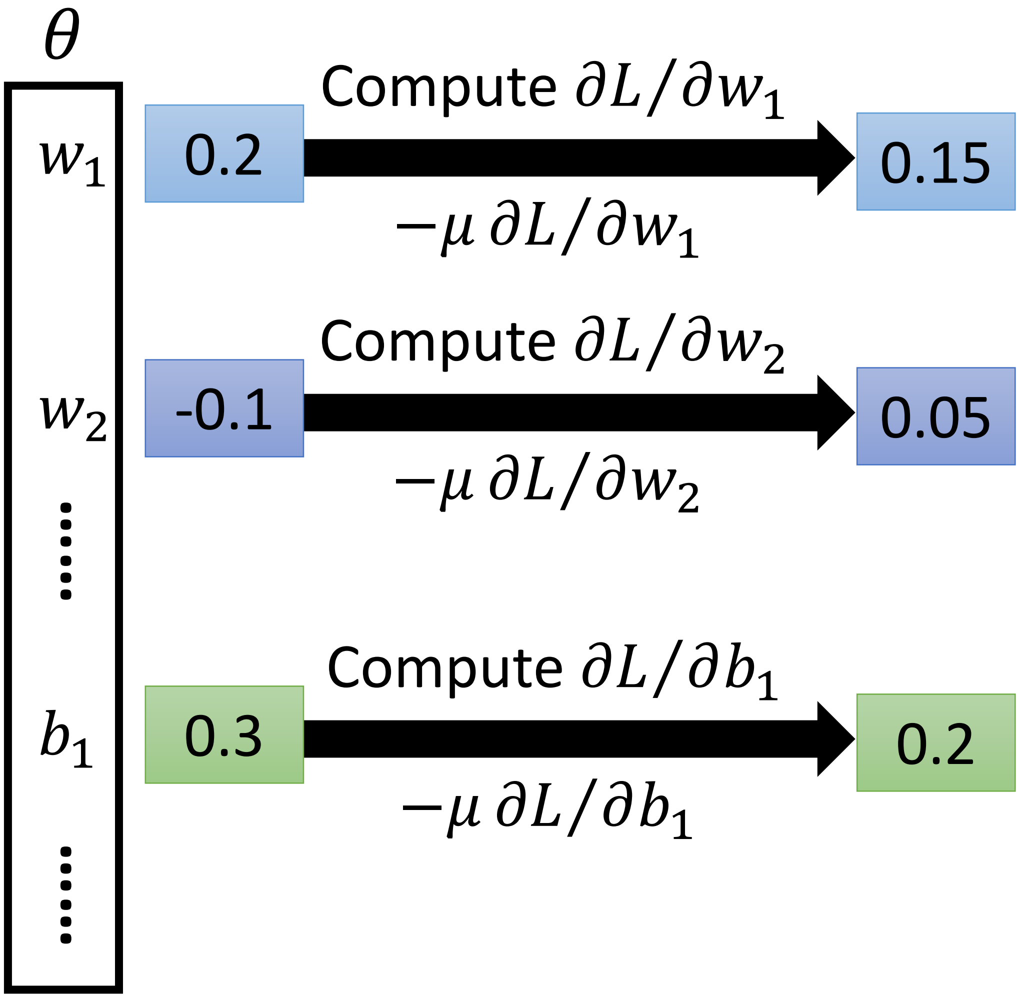
# Three Steps for Deep Learning

Step 1: Neural Network → Step 2: goodness of function → Step 3: pick the best function

Deep Learning is so simple ……



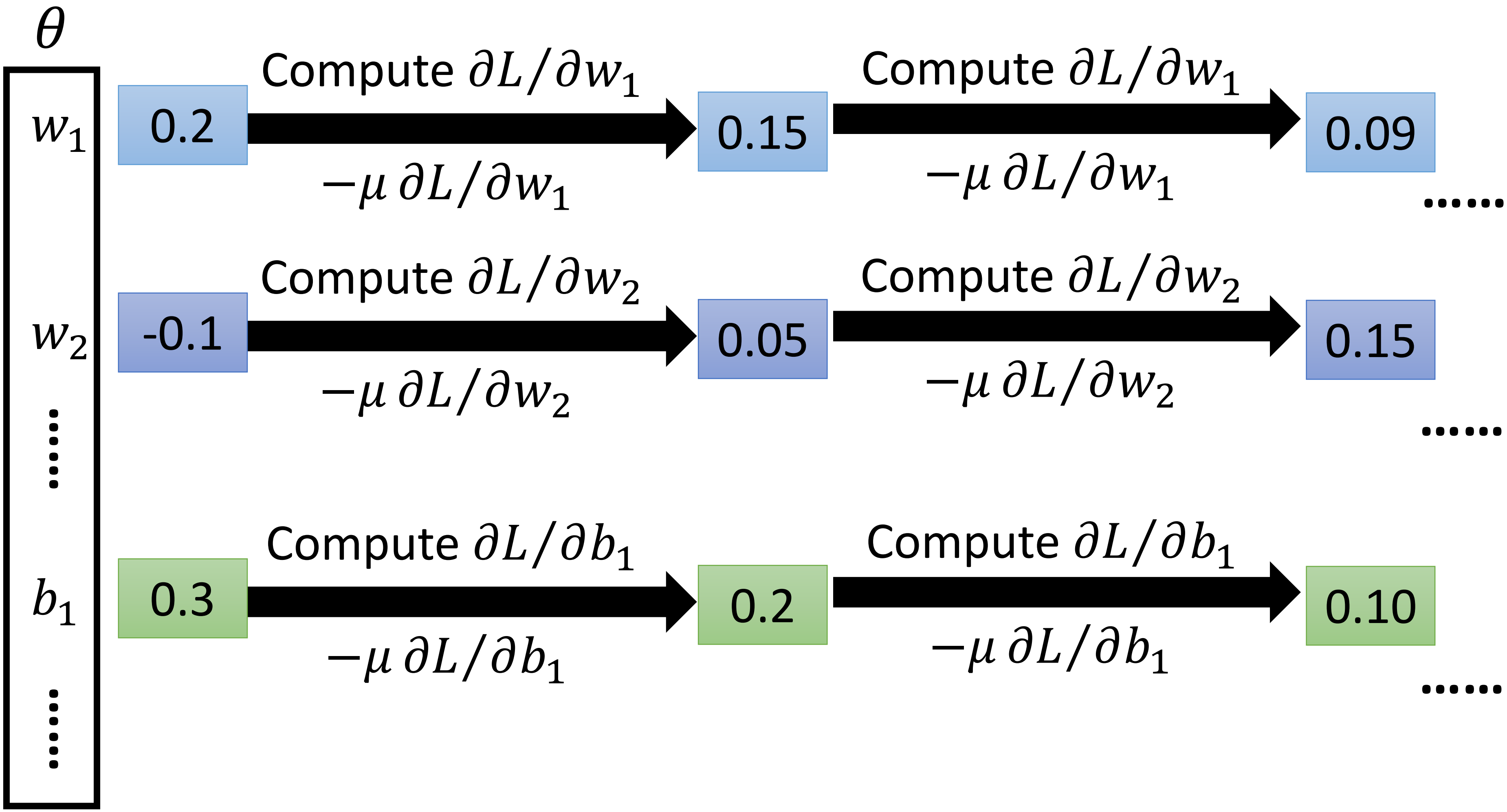
# Gradient Descent




$$\nabla L = \begin{bmatrix} \dfrac{\partial L}{\partial w_1} \\ \dfrac{\partial L}{\partial w_2} \\ \vdots \\ \dfrac{\partial L}{\partial b_1} \\ \vdots \end{bmatrix}$$

gradient

# Gradient Descent

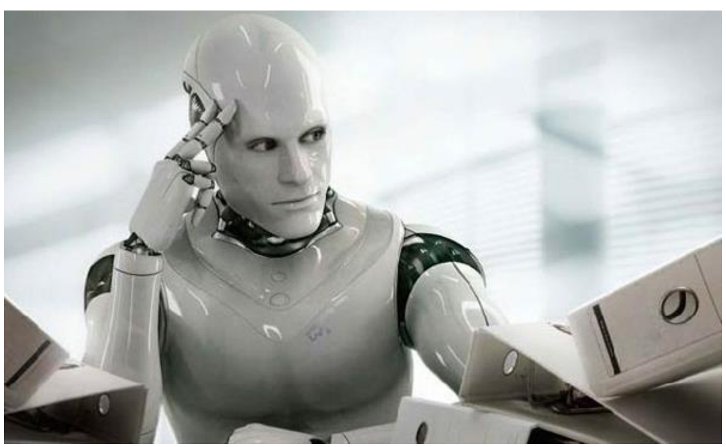


# Gradient Descent

This is the "learning" of machines in deep learning ......

➡ Even alpha go using this approach.

People image ......



Actually .....


I hope you are not too disappointed :p

# Backpropagation

- Backpropagation: an efficient way to compute $\partial L / \partial w$ in neural network

TensorFlow

torch

theano

Caffe

Microsoft CNTK

Chainer

Deep Learning library produced by Amazon
DSSTNE

mxnet

libdnn
台大周伯威
同學開發

Ref:
http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/DNN%20backprop.ecm.mp4/index.html

# Three Steps for Deep Learning

Step 1: Neural Network → Step 2: goodness of function → Step 3: pick the best function

Deep Learning is so simple ……