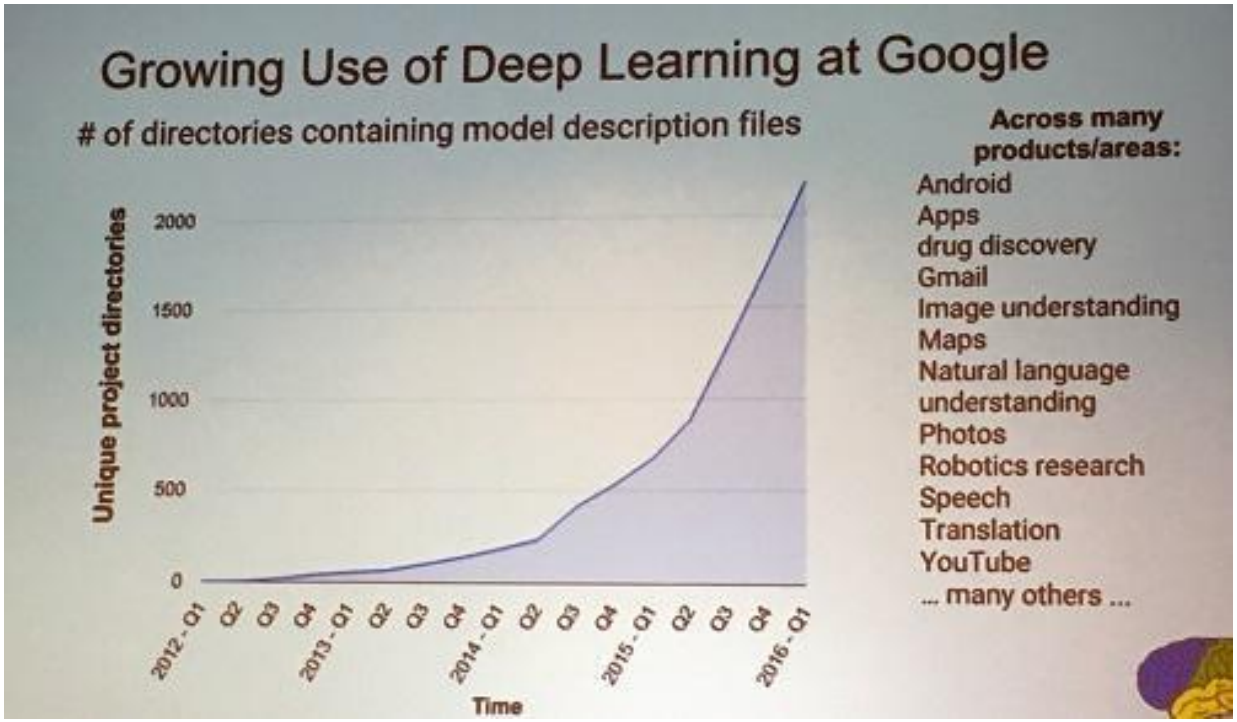
Deep Learning

Hung-yi Lee

李宏毅

Deep learning attracts lots of attention.

- I believe you have seen lots of exciting results before.



Deep learning trends at Google. Source: SIGMOD 2016/Jeff Dean

Ups and downs of Deep Learning

- 1958: Perceptron (linear model)
- 1969: Perceptron has limitation
- 1980s: Multi-layer perceptron
  - Do not have significant difference from DNN today
- 1986: Backpropagation
  - Usually more than 3 hidden layers is not helpful
- 1989: 1 hidden layer is "good enough", why deep?
- 2006: RBM initialization
- 2009: GPU
- 2011: Start to be popular in speech recognition
- 2012: win ILSVRC image competition
- 2015.2: Image recognition surpassing human-level performance
- 2016.3: Alpha GO beats Lee Sedol
- 2016.10: Speech recognition system as good as humans

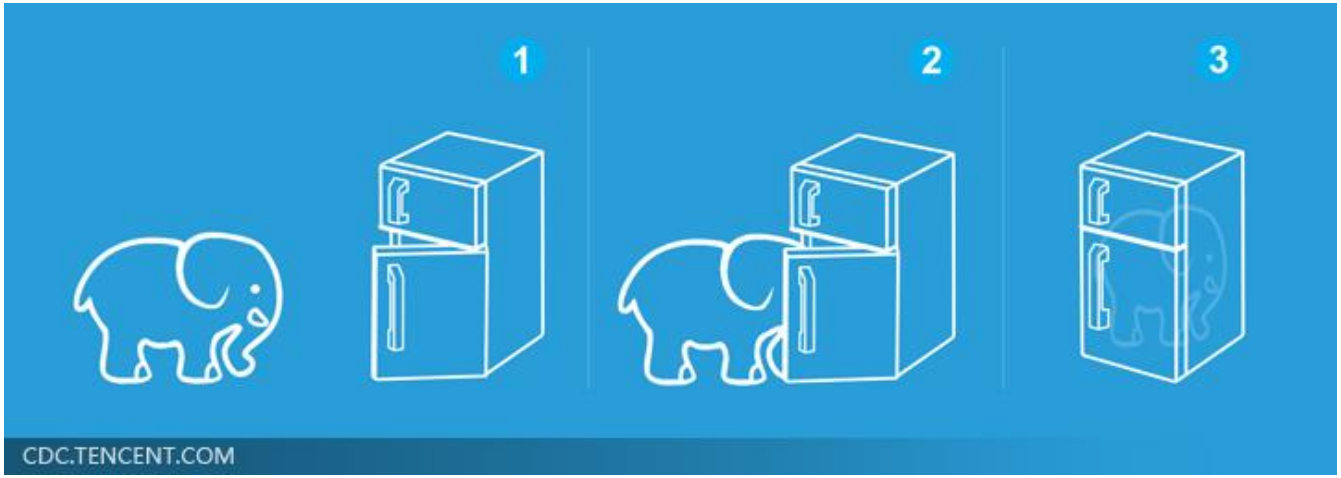
Three Steps for Deep Learning

Step 1:
Neural Network

Step 2: goodness of function

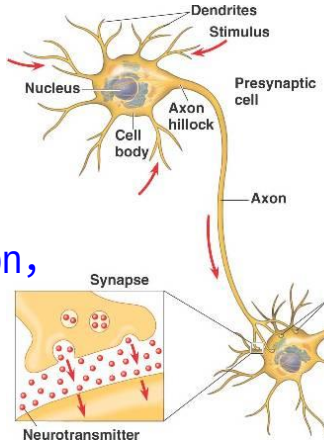
Step 3: pick the best function

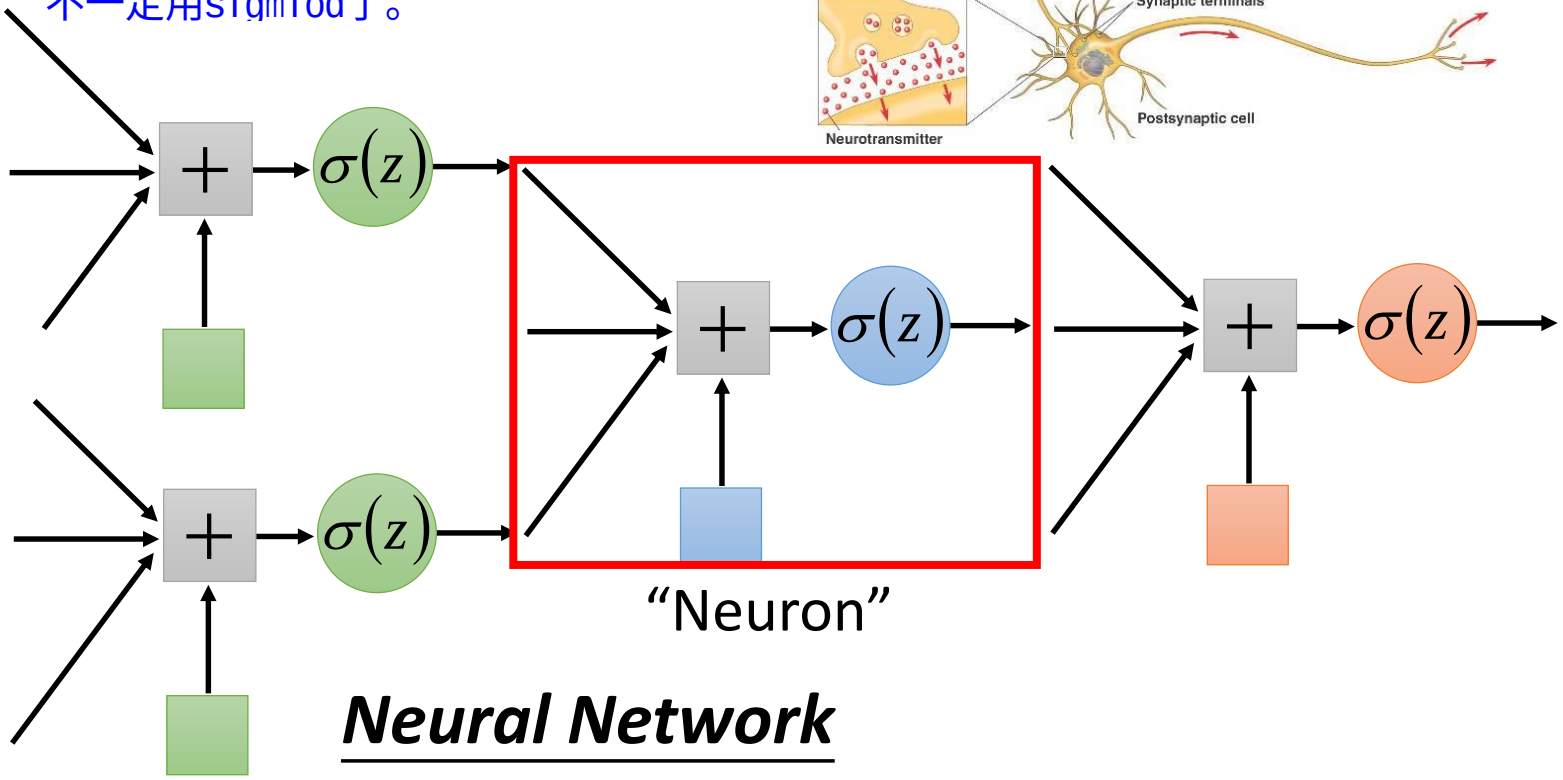
Deep Learning is so simple ……



Neural Network

每一个neural都是一个activation function，
不一定用sigmiod了。





"Neuron"

Neural Network

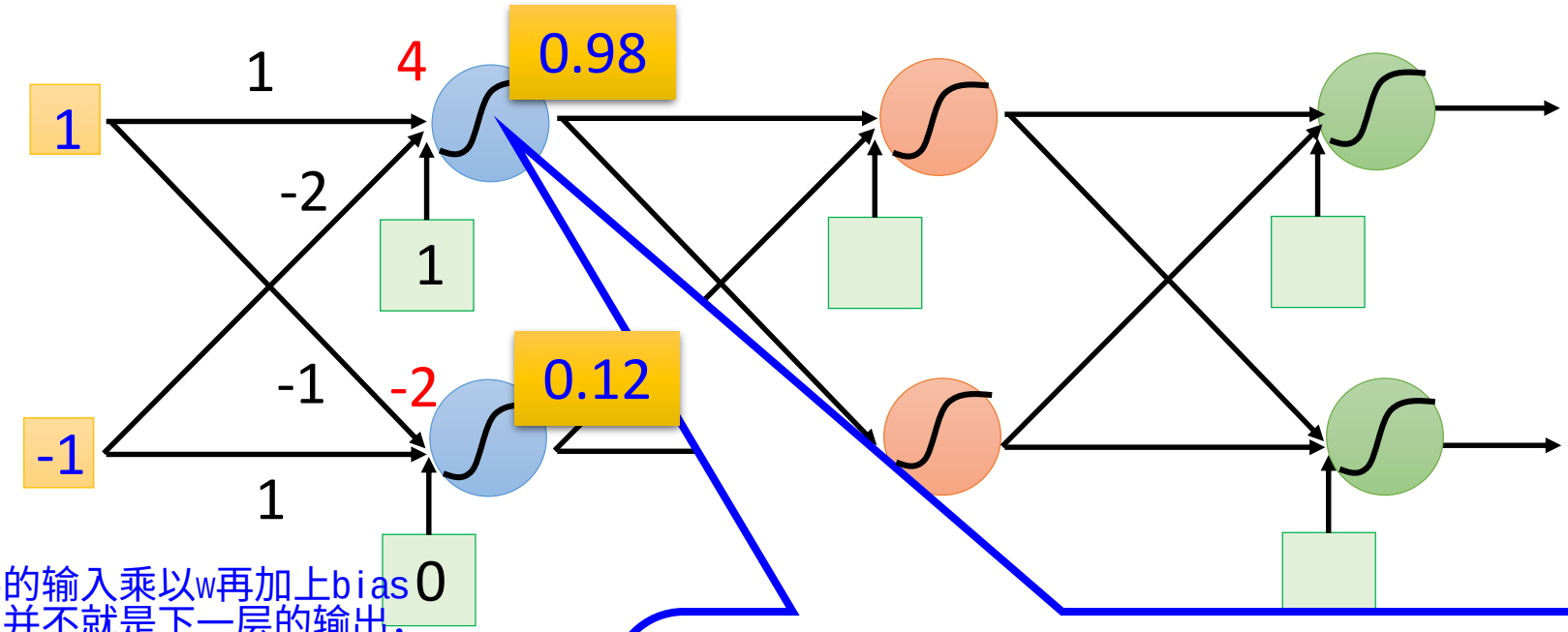
Different connection leads to different network structures

不一样的链接方式就形成了不一样的结构，就是一个不一样的model(fs),然后进行训练，得到每个neural的w和b

Network parameter θ: all the weights and biases in the "neurons"

Fully Connect Feedforward Network

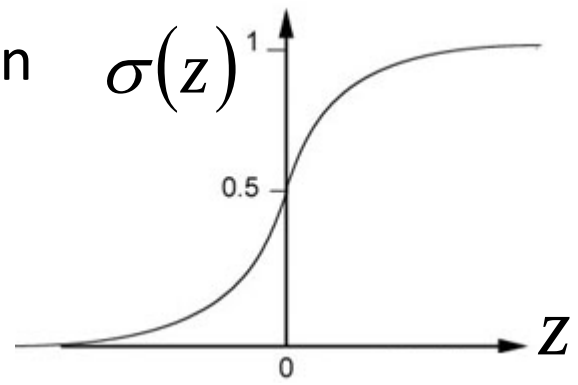
全连接前向反馈网络是指：在这个网络中每两层的neural之间凉凉相连，且因为数据是从前先后传播的，所以叫前向反馈。



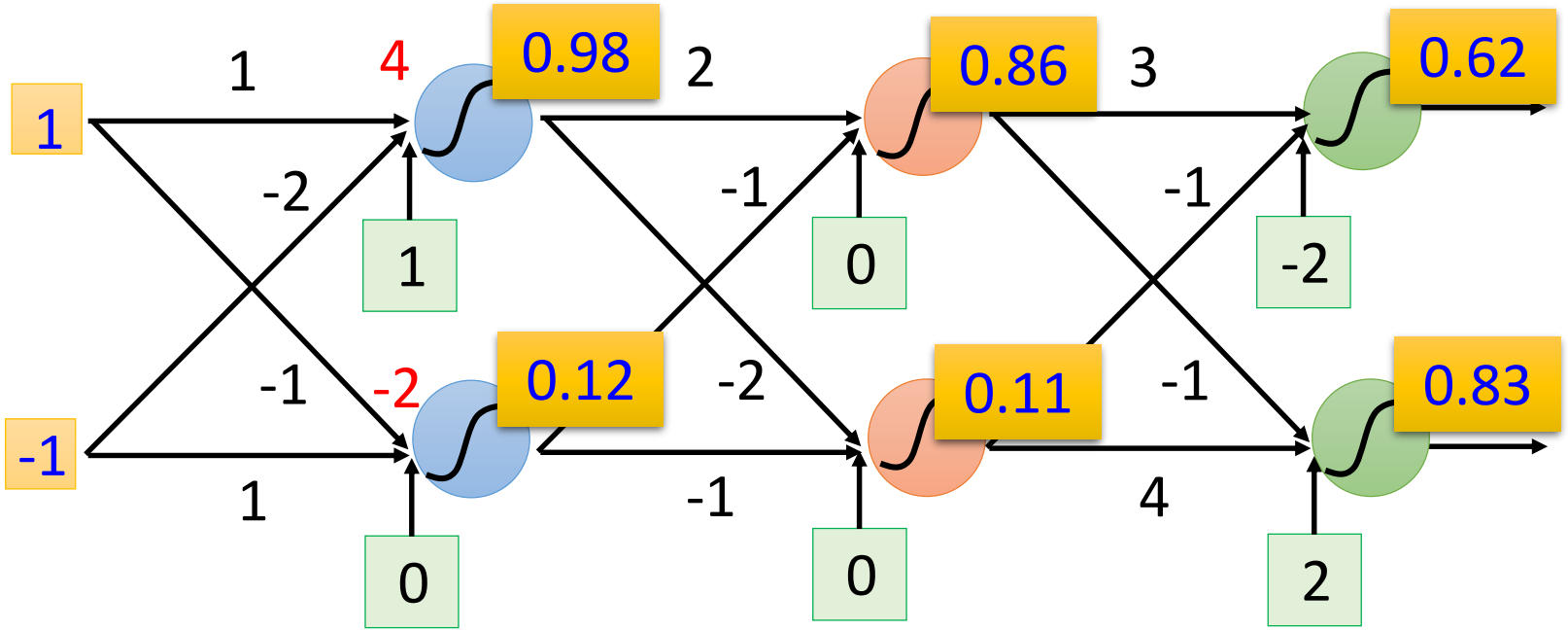
前一层的输入乘以w再加上bias得到的并不就是下一层的输出，而是要经过activation function的计算。如果af取sigmoid函数，wx+b就相当于z。

Sigmoid Function

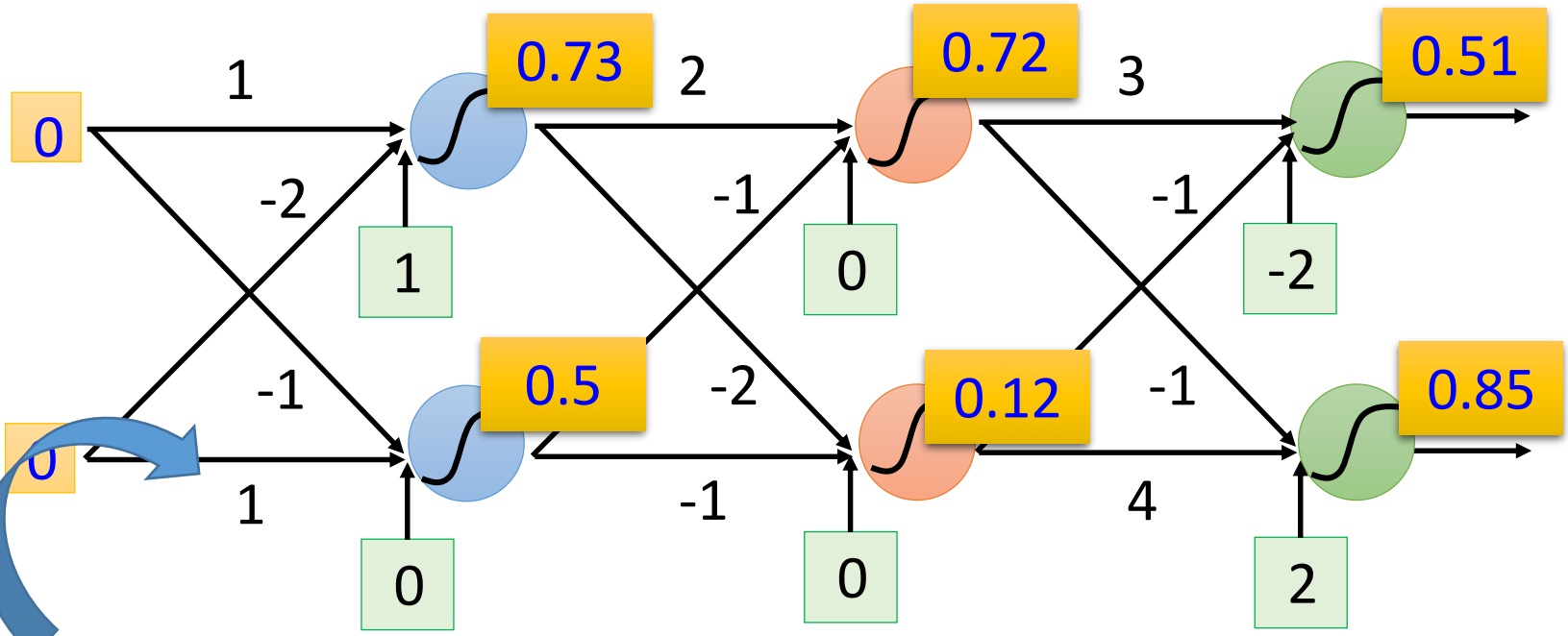
$$\sigma(z)=\frac{1}{1+e^{-z}}$$



Fully Connect Feedforward Network



Fully Connect Feedforward Network



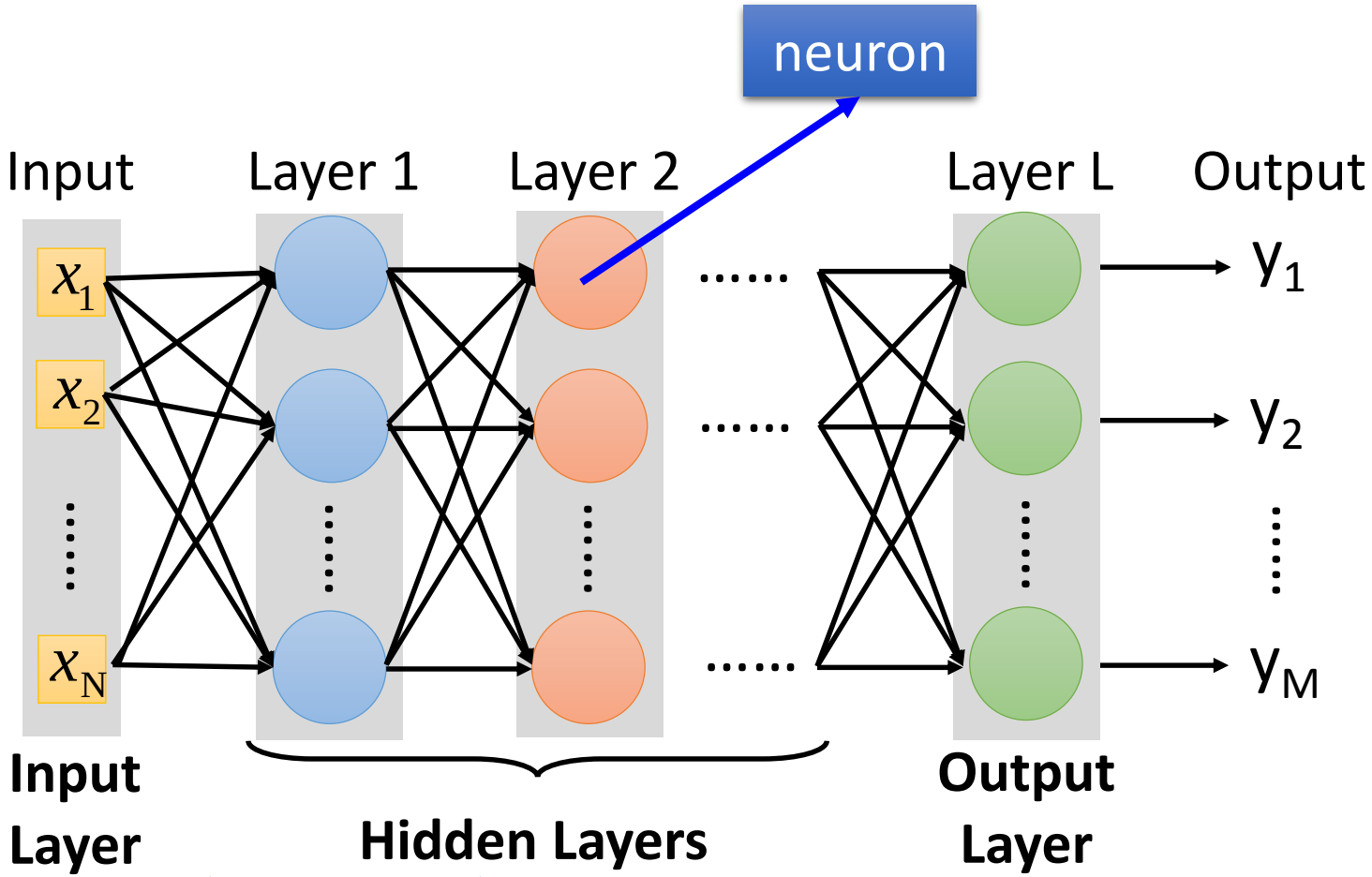
This is a function.

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

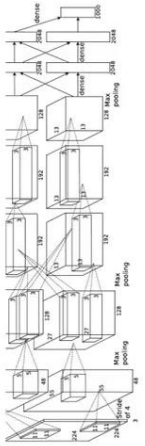


把输入的向量佳作输入层（并不是神经元层），最后一层神经元叫做输出层，中间所有的都叫做隐藏层。

Deep = Many hidden layers

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

8 layers



16.4%

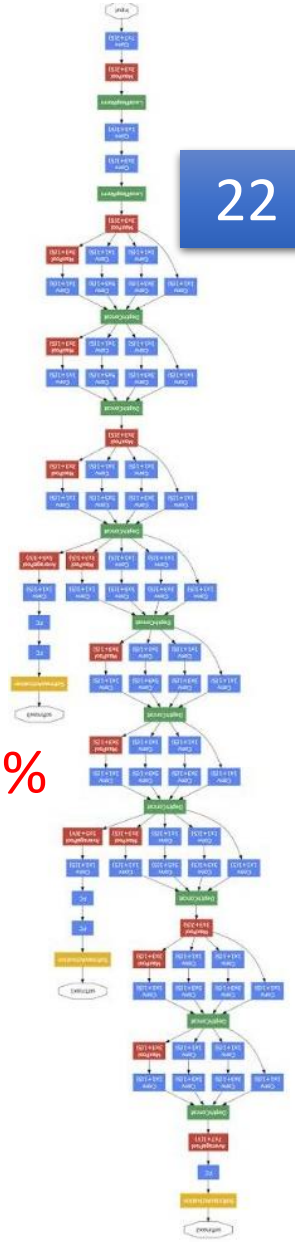
AlexNet (2012)

19 layers

7.3%

VGG (2014)

22 layers



6.7%

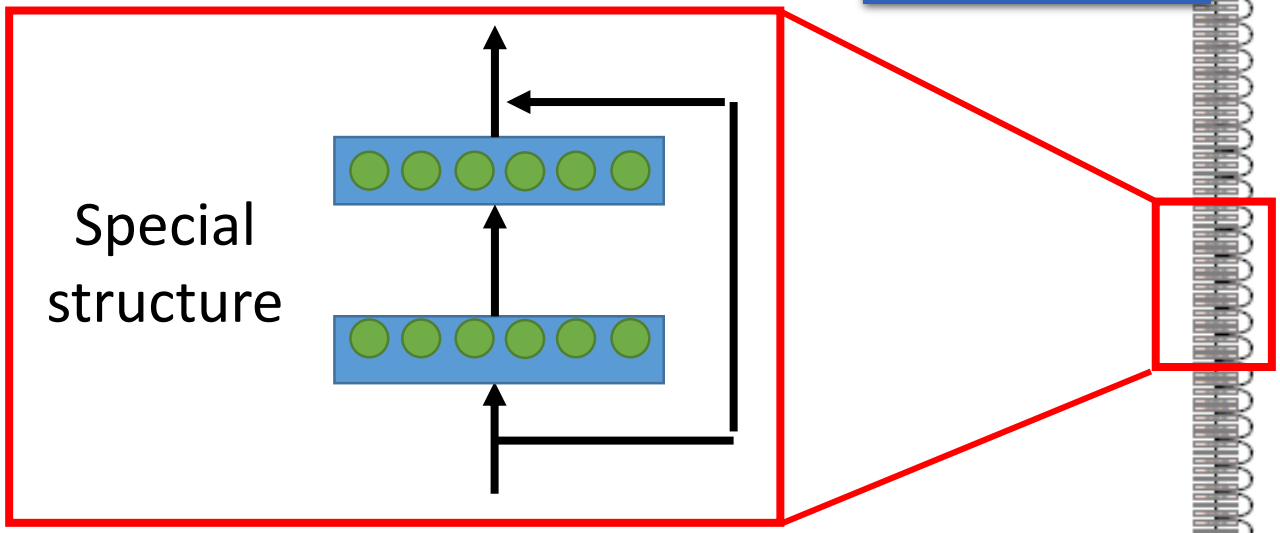
GoogleNet (2014)

Deep = Many hidden layers

152 layers

101 layers

Special structure



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

3.57%

16.4%

AlexNet (2012)

7.3%

VGG (2014)

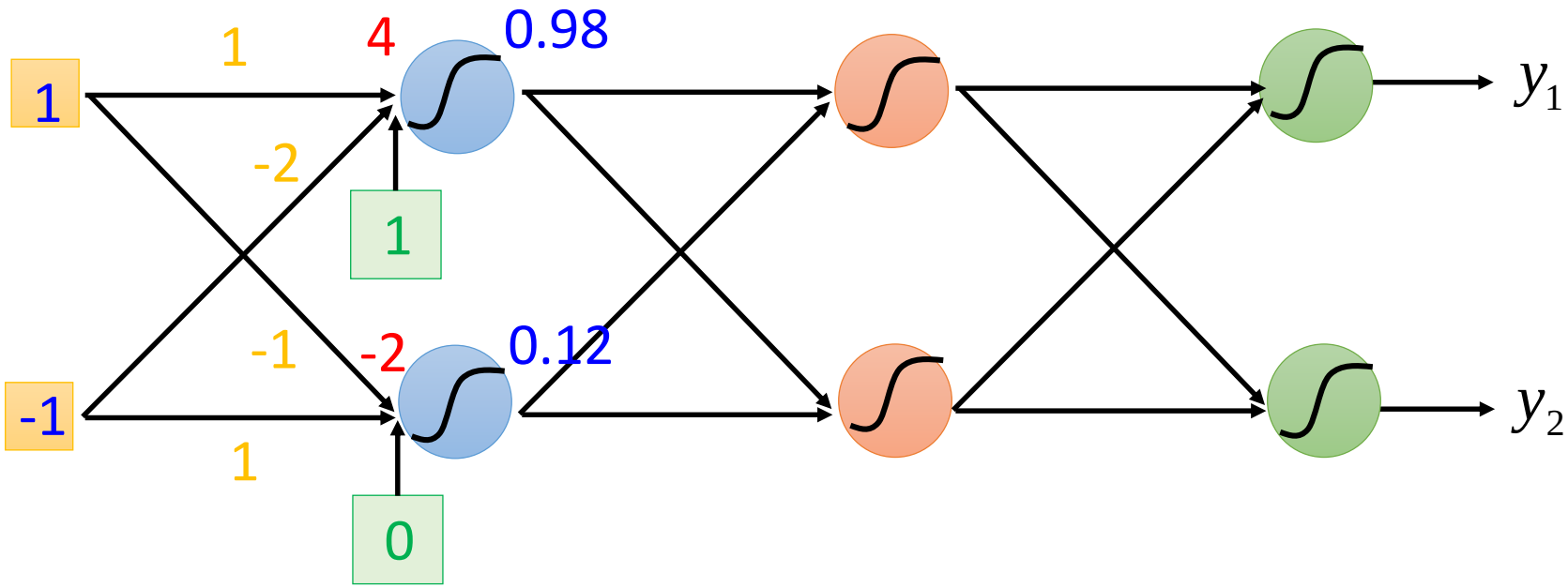
6.7%

GoogleNet (2014)

Residual Net (2015)

Taipei 101

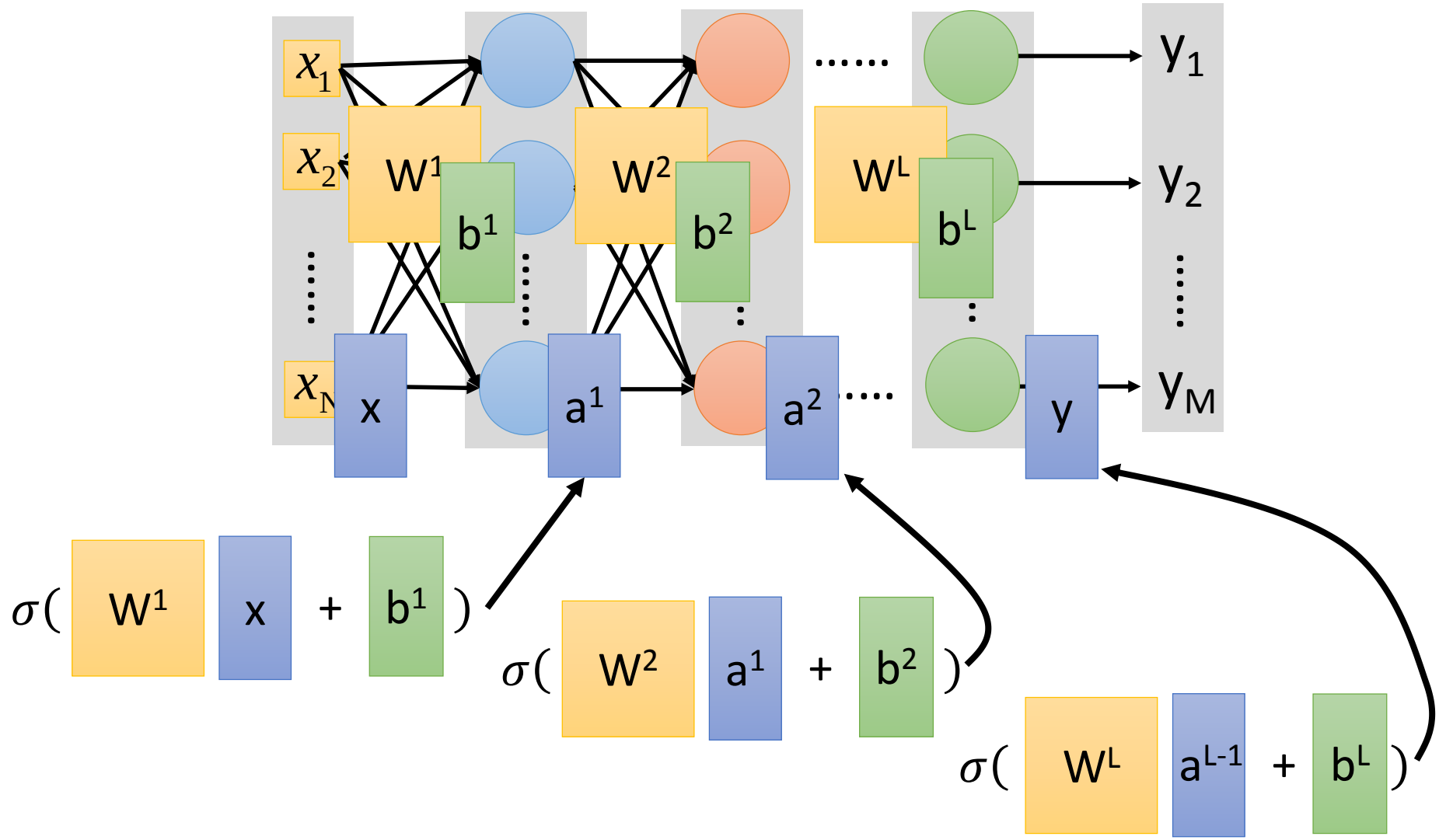
Matrix Operation



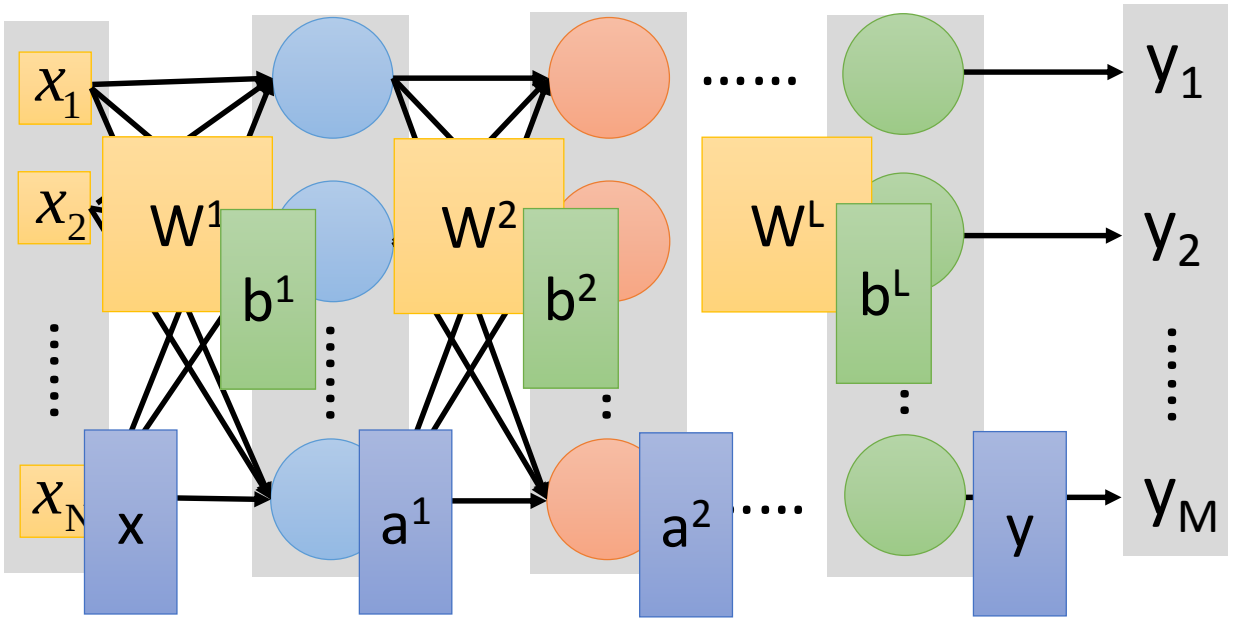
$$\sigma\left(\begin{bmatrix}1 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}+\begin{bmatrix}1\\ 0\end{bmatrix}\right)=\begin{bmatrix}0.98\\ 0.12\end{bmatrix}$$

$$\begin{bmatrix}4\\ -2\end{bmatrix}$$

Neural Network



Neural Network

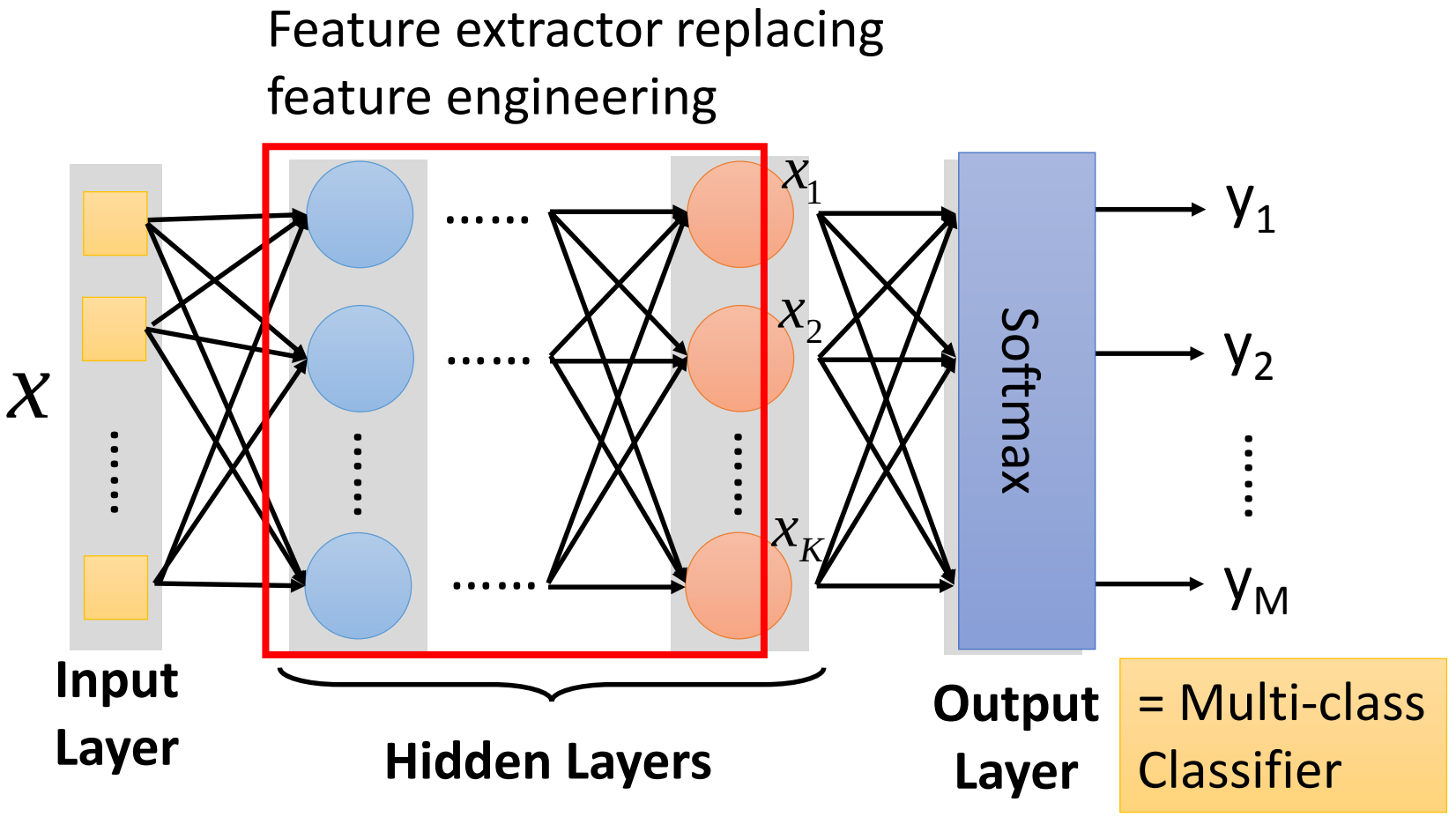


$$\mathbf{y} = f(\mathbf{x})$$

Using parallel computing techniques to speed up matrix operation

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

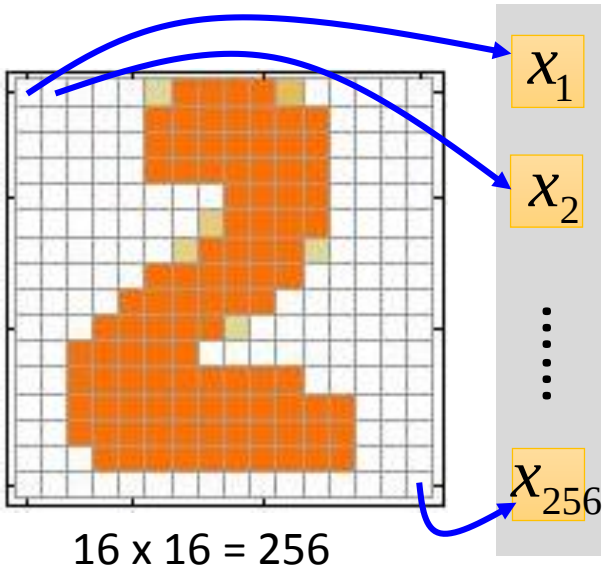
Output Layer as Multi-Class Classifier



将输出层看作一个multi-class classfier，经过softmax的作用，得到了第一营每一个数字的概率，那个数字的概率最大，便认为是那个数字

Example Application

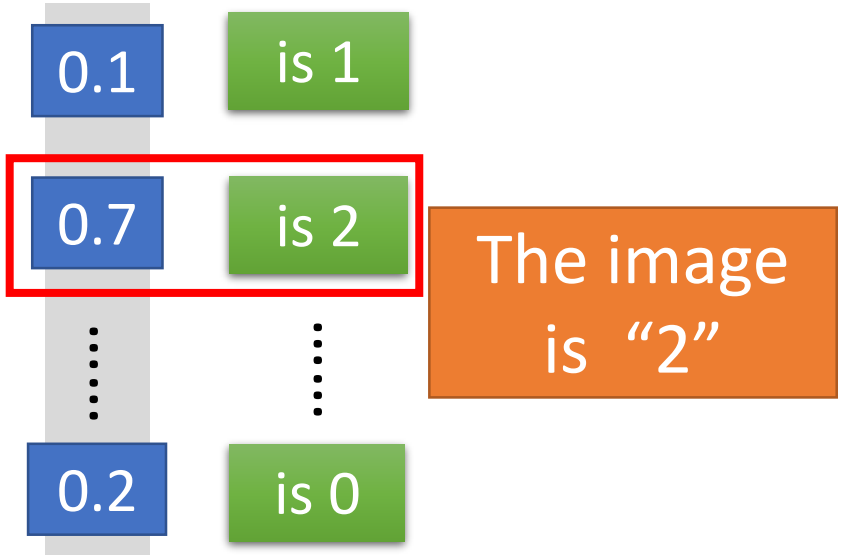
Input



Ink → 1

No ink → 0

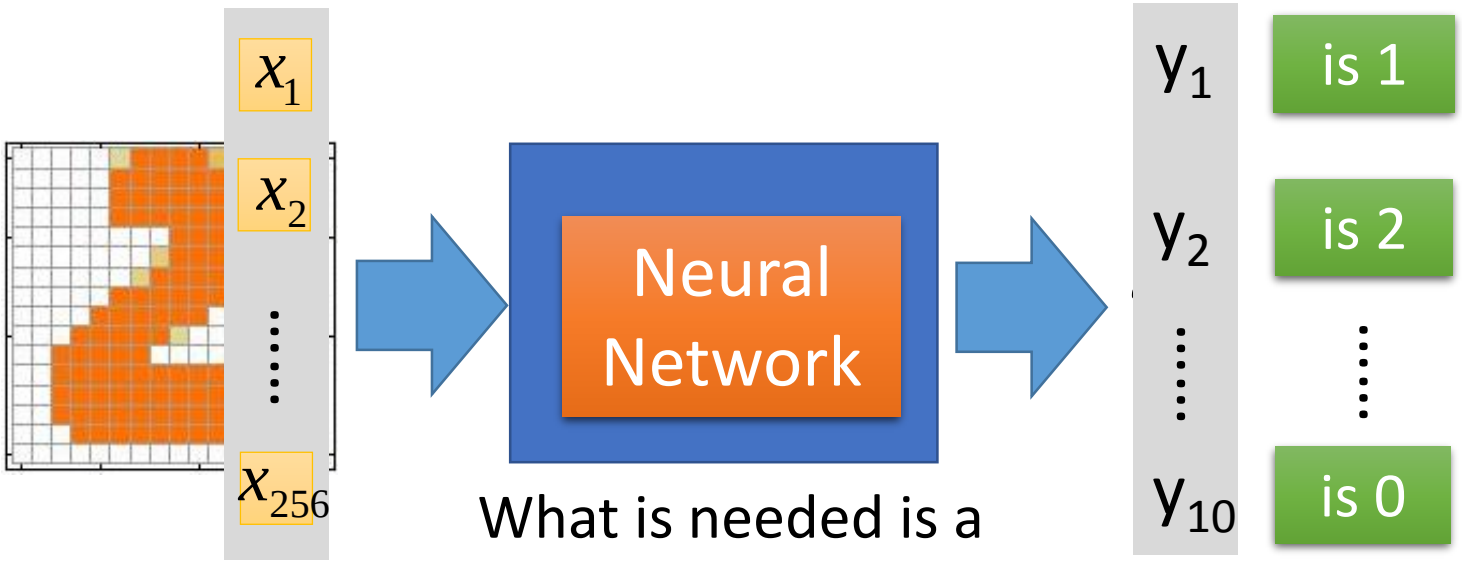
Output



Each dimension represents the confidence of a digit.

Example Application

- Handwriting Digit Recognition

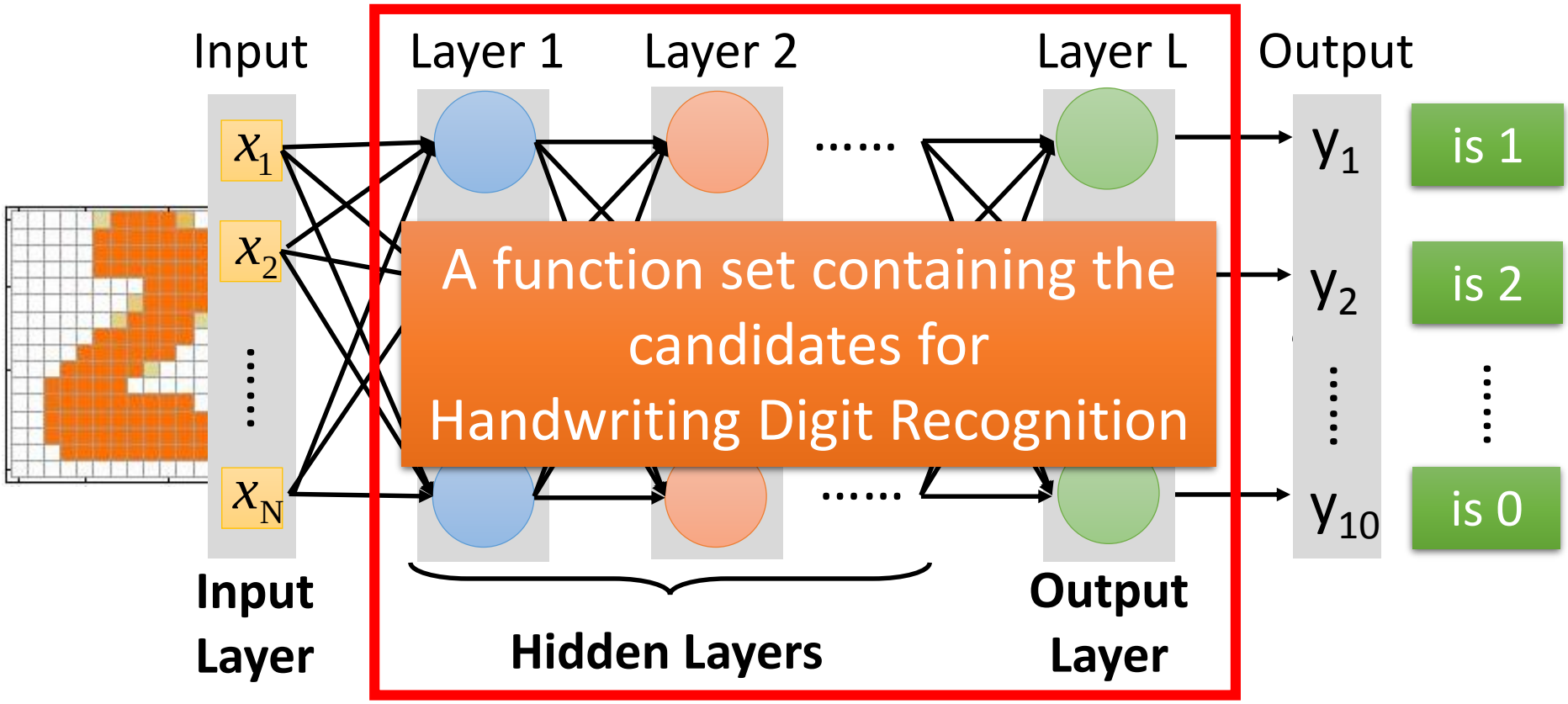


What is needed is a function ……

Input:
256-dim vector

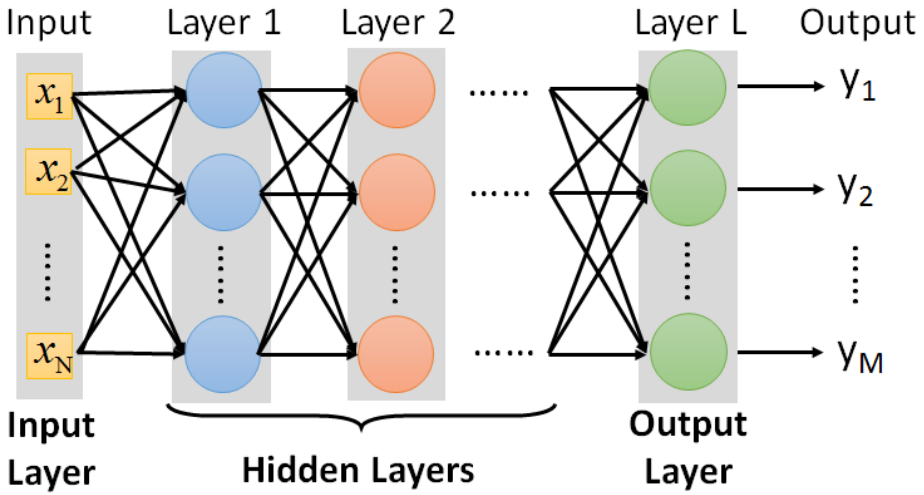
output:
10-dim vector

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)

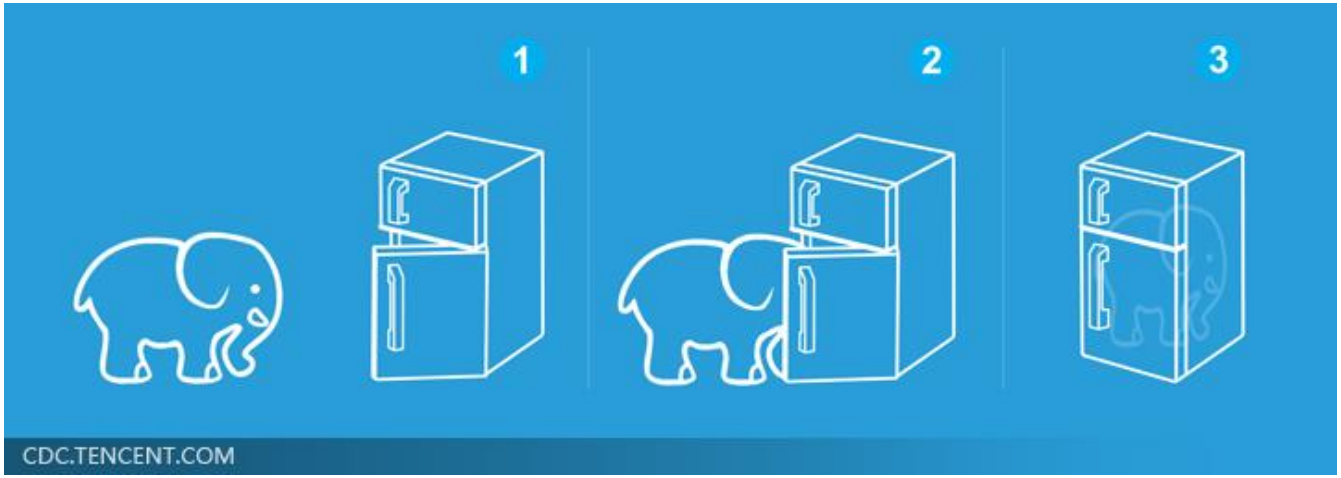
Three Steps for Deep Learning

Step 1: Neural Network

Step 2: goodness of function

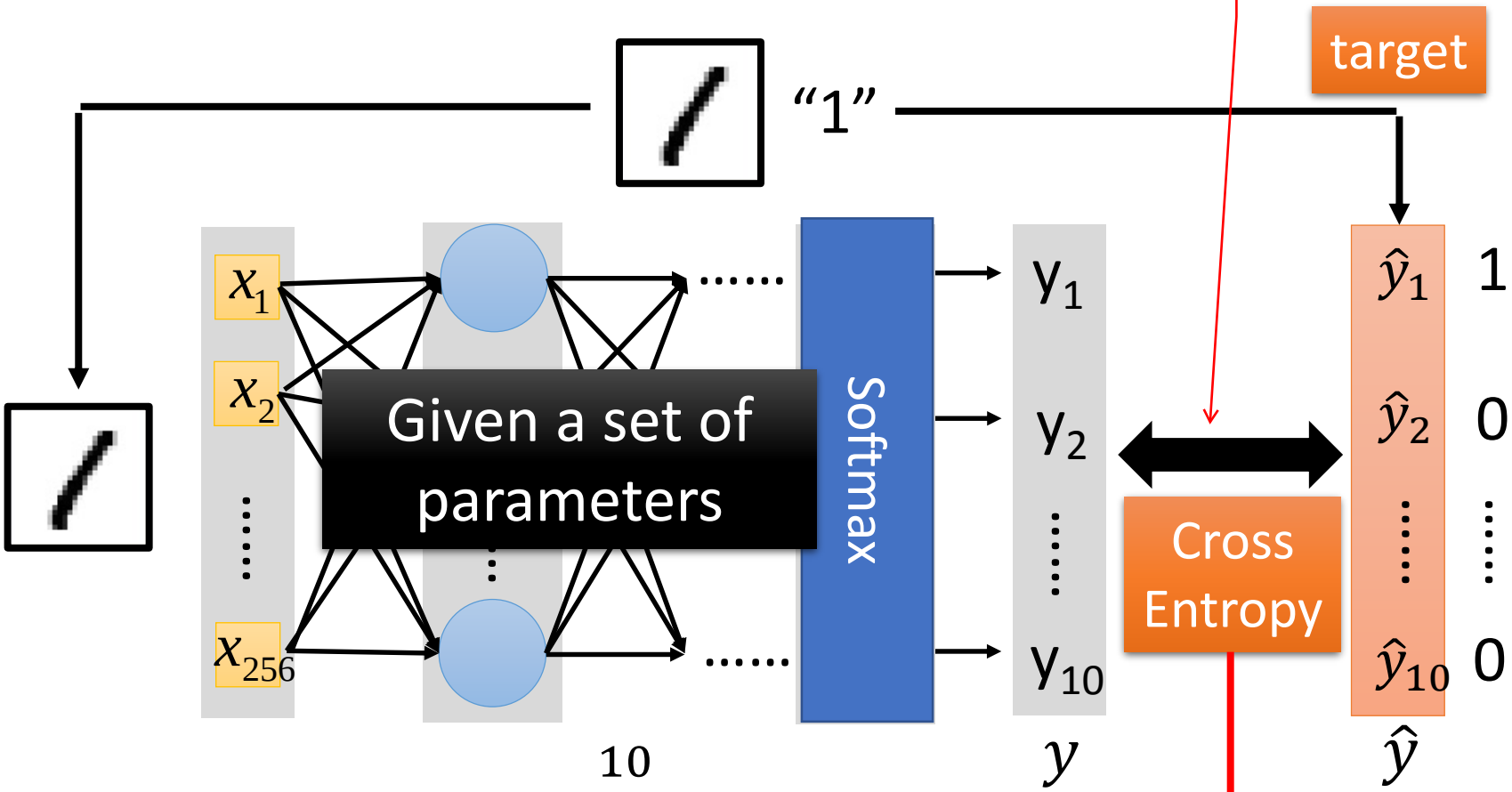
Step 3: pick the best function

Deep Learning is so simple ……



Loss for an Example

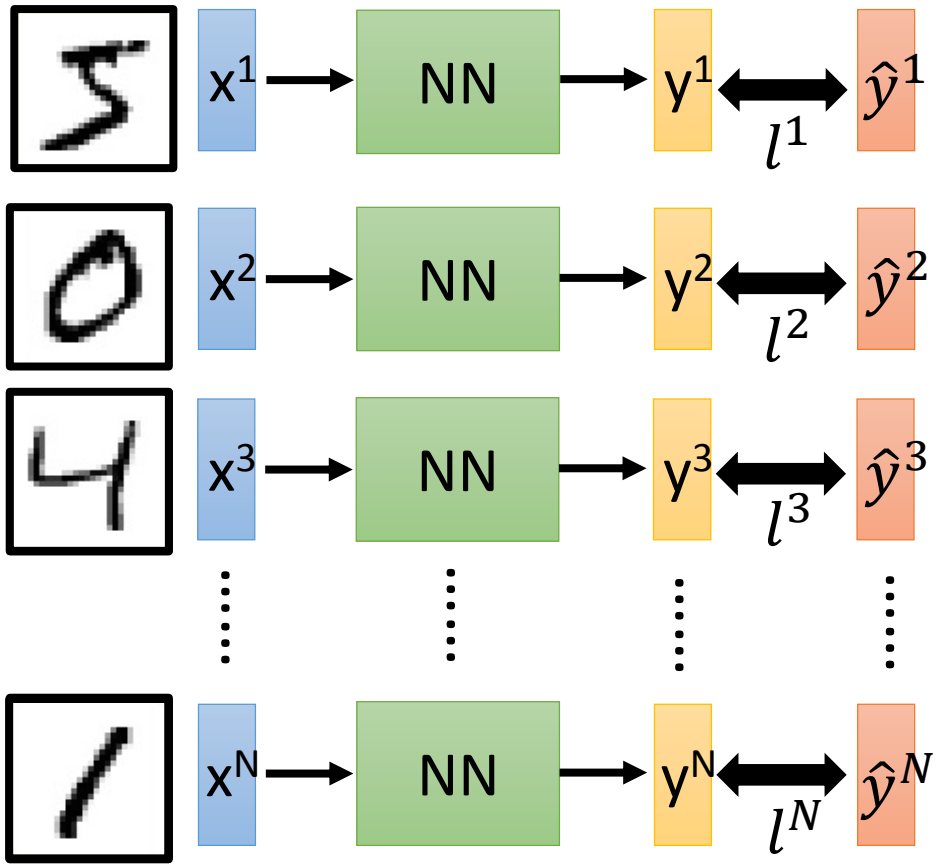
在回归问题中，常用均方差来衡量预测值与实际值之间的距离。通常用cross entropy来描述两个概率分布（预测的概率与实际的概率）之间的距离。



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

Total Loss

For all training data …



Total Loss:

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

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

Find ***the network parameters*** θ^* that minimize total loss L

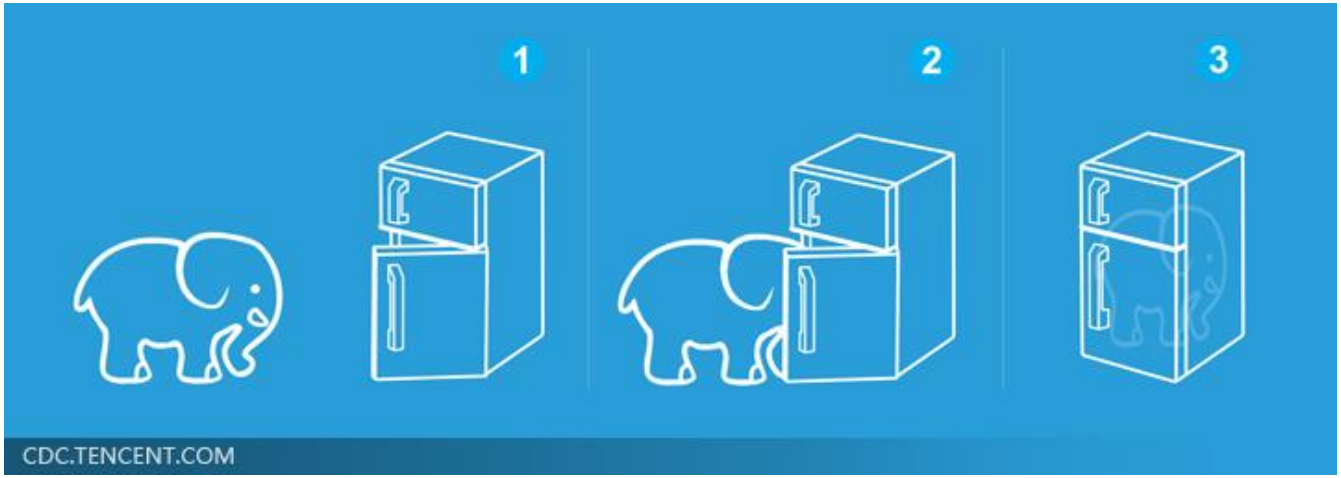
Three Steps for Deep Learning

Step 1: Neural Network

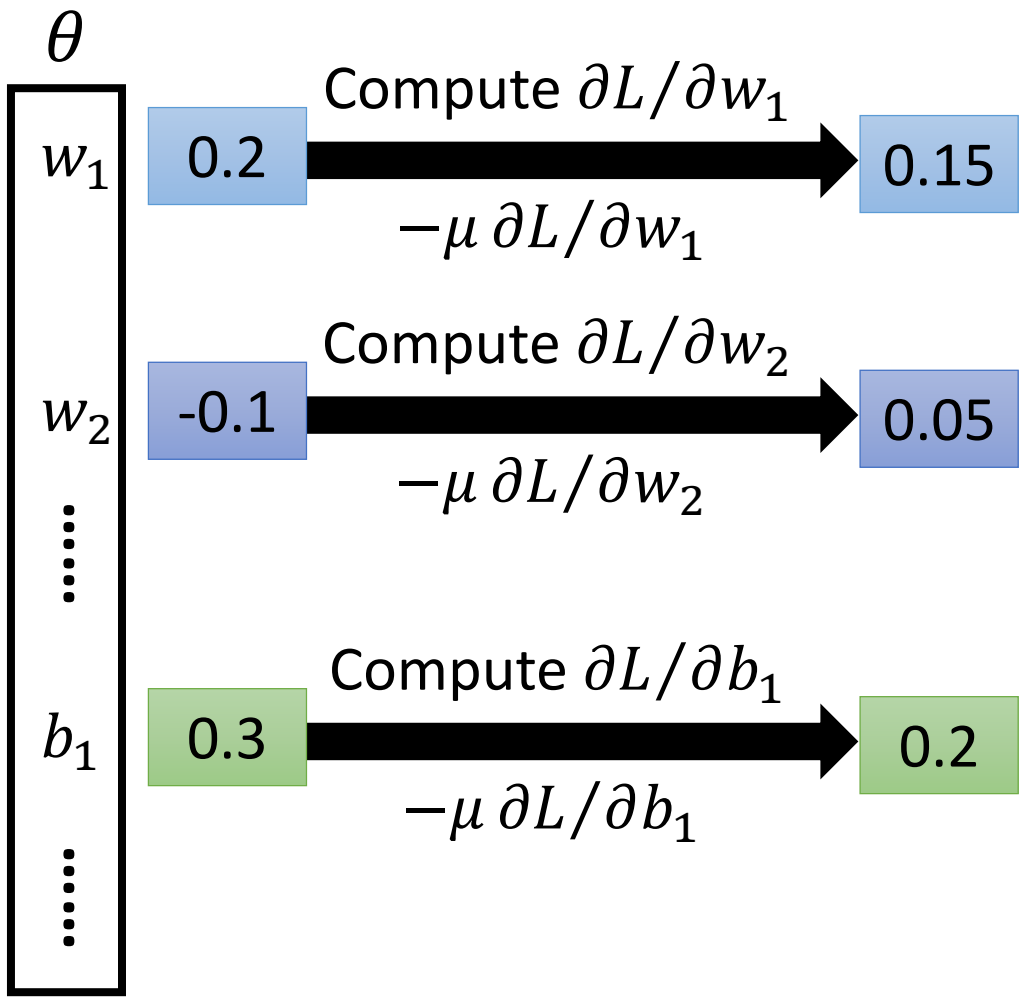
Step 2: goodness of function

Step 3: pick the best function

Deep Learning is so simple ……



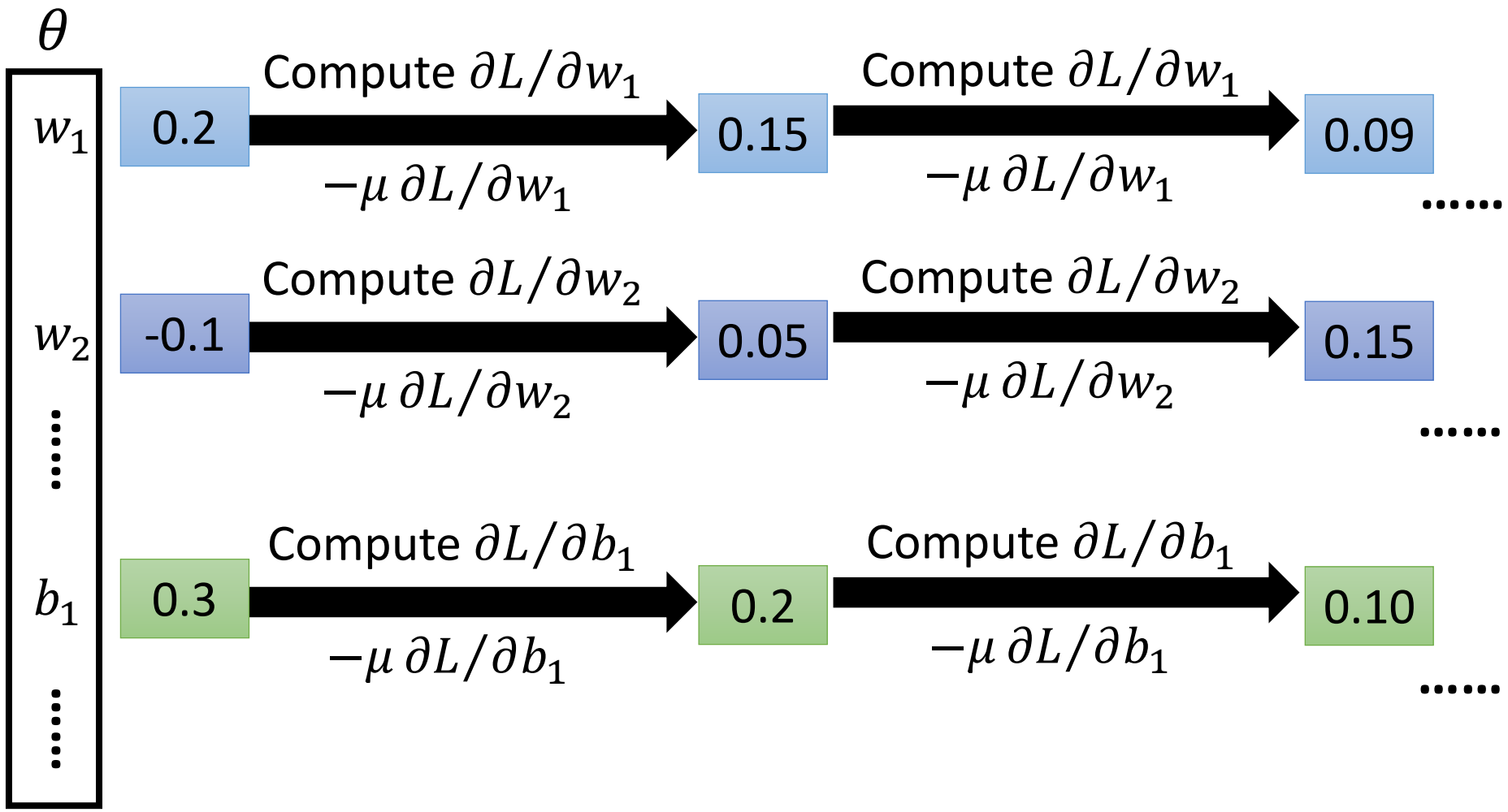
Gradient Descent



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

gradient

Gradient Descent

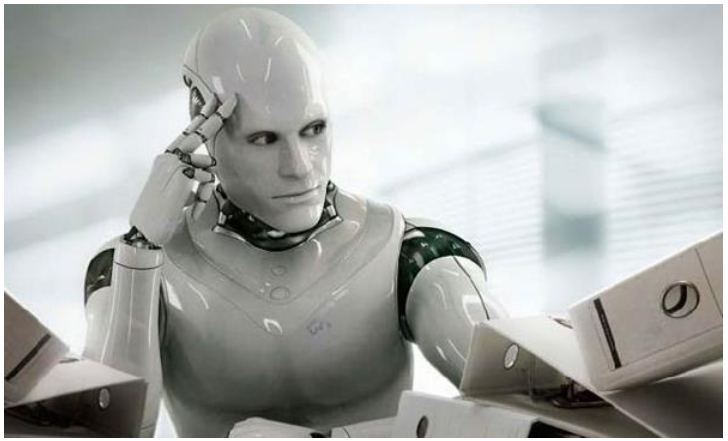


Gradient Descent

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

theano

Caffe

Microsoft
CNTK

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

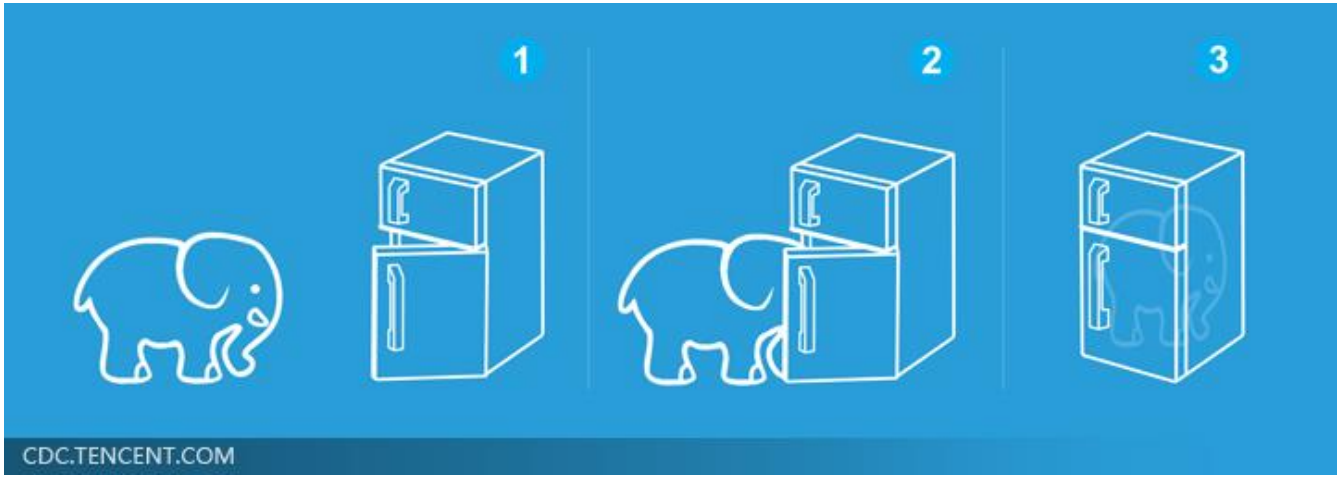
Three Steps for Deep Learning

Step 1: Neural Network

Step 2: goodness of function

Step 3: pick the best function

Deep Learning is so simple ……



Acknowledgment

- 感謝 Victor Chen 發現投影片上的打字錯誤