Welcome to

DS3010:
DS-III: Computational Data Intelligence
Deep Learning / Multi-Layer Perceptron

Prof. Yanhua Li

Time: 11:00am – 12:50pm M & R
Location: HL 114
D-term 2022

Quiz 1 grading is done

Quiz 2 Week 5 (4/14 R)
Topics: To be announced

Project 1

- Two student presentations today.
- Grading will be available this afternoon at 5pm.

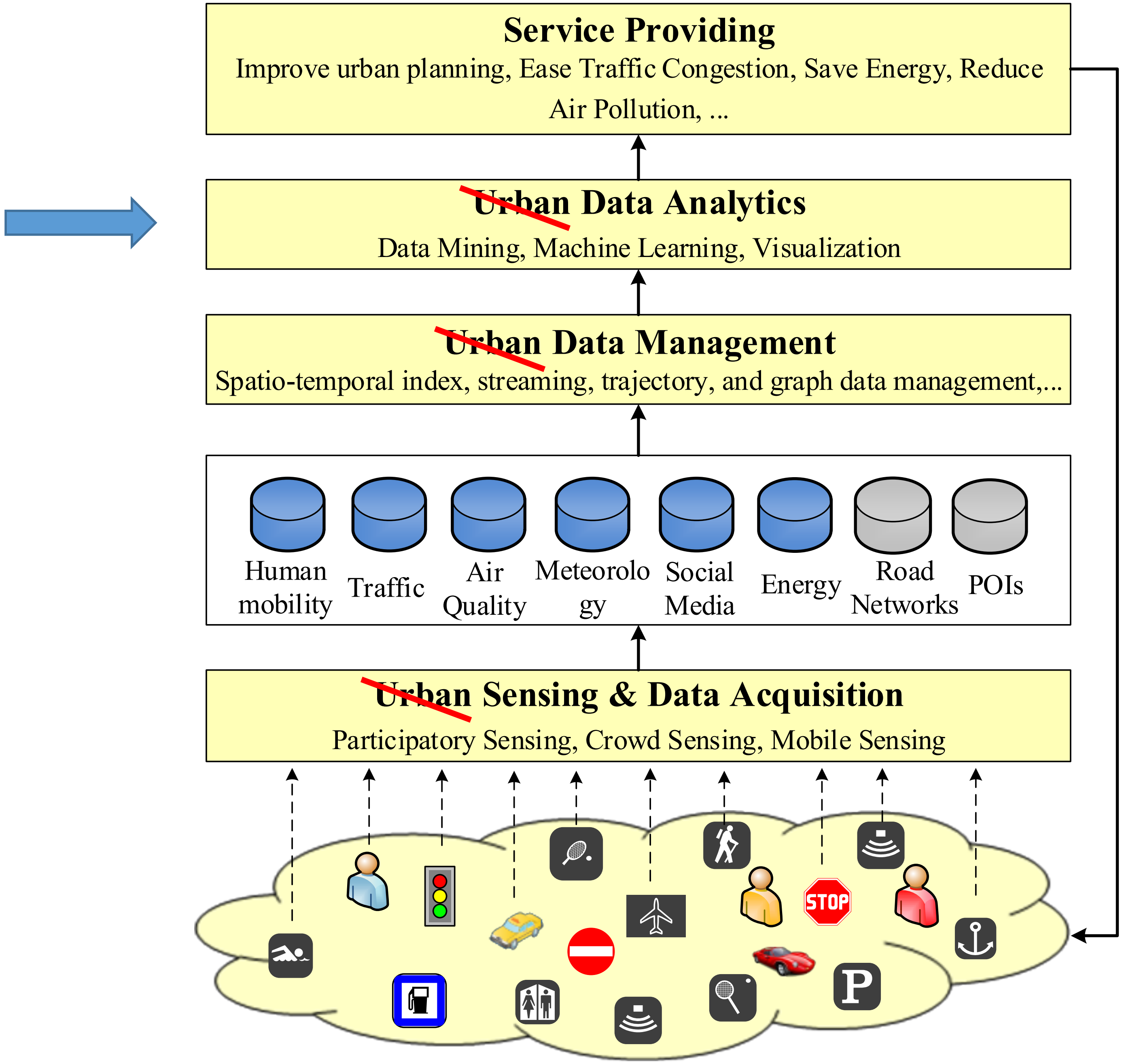
Project 2 due **Week 6 (4/18 M)**

- Logistic regression
- SVM
- Multi-layer perceptron/Deep learning
- https://github.com/ds3010s22/ds3010_projects/blob/main/Project2.ipynb
- Get started as early as possible, don’t wait for the last minute.

Project 3 due **Week 6 (4/18 M)**

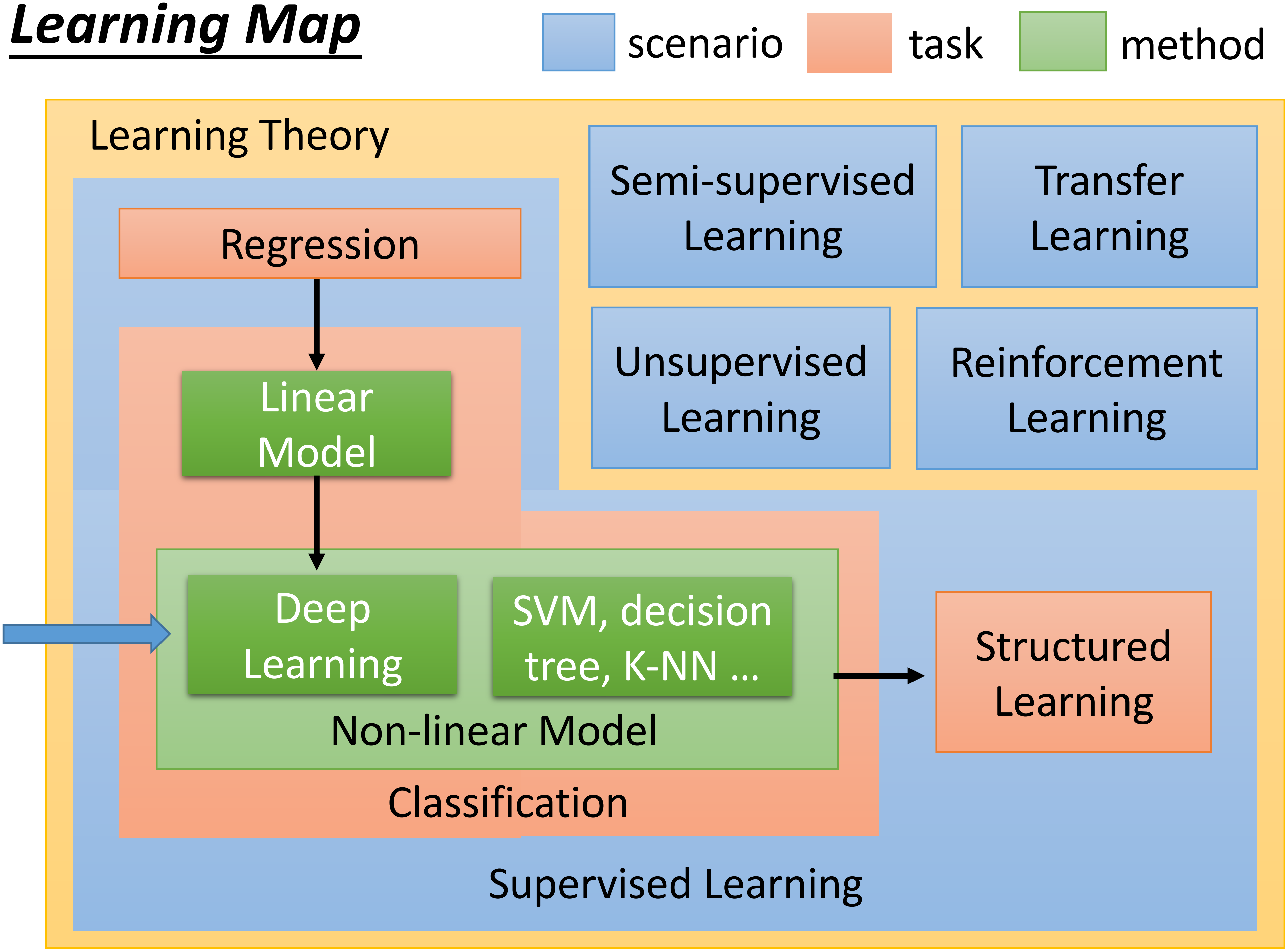
- Week 5 (4/11 M), Starting date
  Week 8 (5/2 M), Due.
  https://github.com/ds3010s22/ds3010_projects/blob/main/Project3_Business.ipynb
- Team work 2-4 student in a team.
- Start teaming up on Canvas

Data pipeline



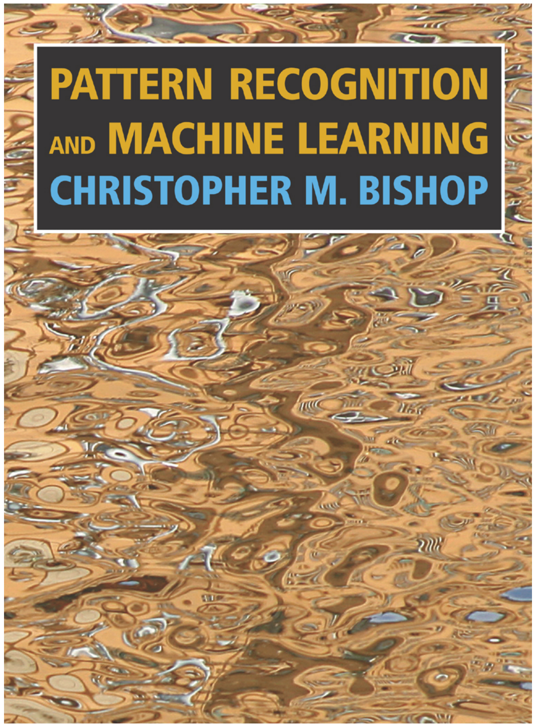
Urban Computing: concepts, methodologies, and applications.
Zheng, Y., et al. *ACM transactions on Intelligent Systems and Technology*.

Learning Map



References

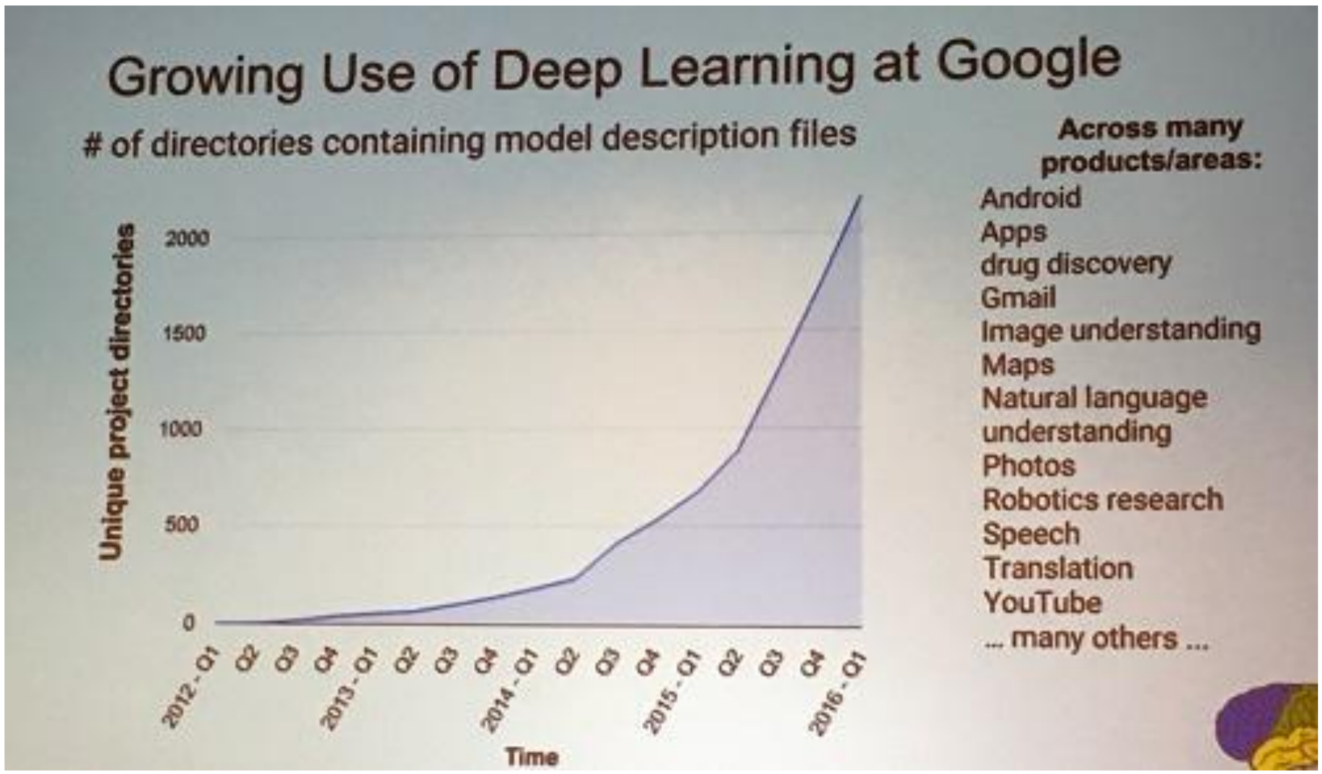
Classification
Multi-Layer
Perceptron / Deep
Learning



Bishop: Chapter 5.1

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

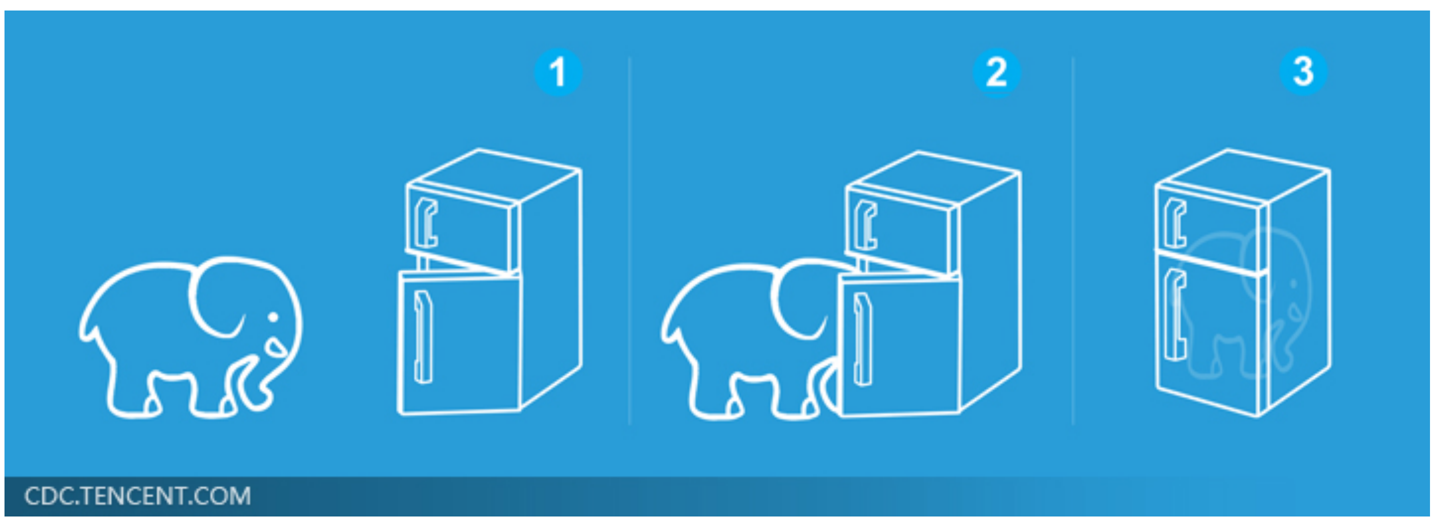
Ups and downs of Deep Learning

- 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

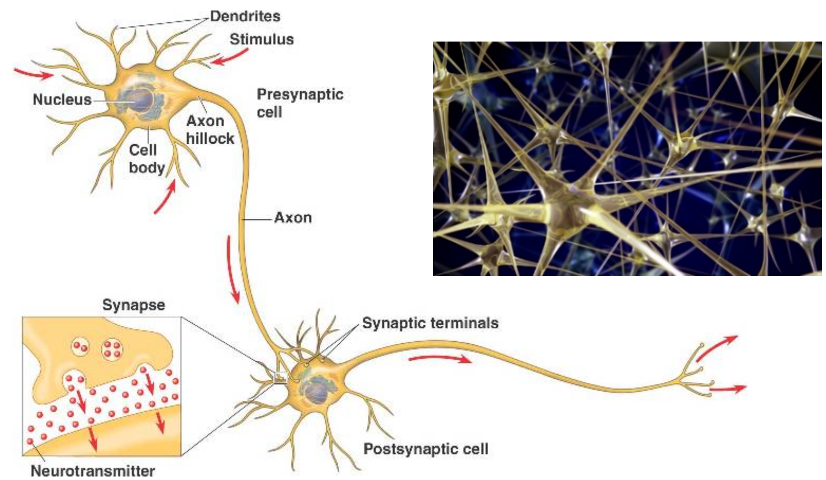
Three Steps for Deep Learning

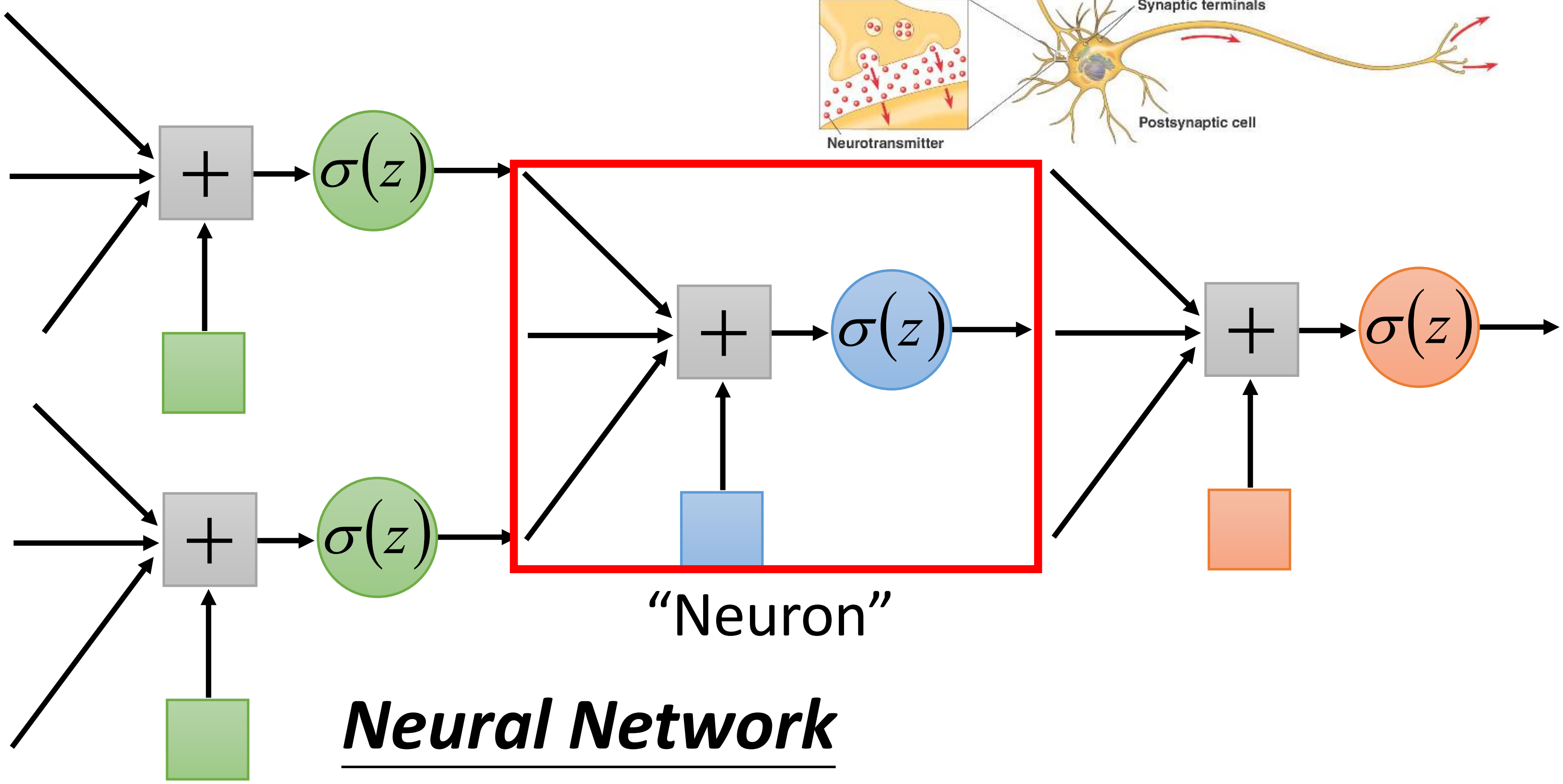
Step 1: Neural Network → Step 2: goodness of function → Step 3: pick the best function

Deep Learning is so simple ……



Neural Network



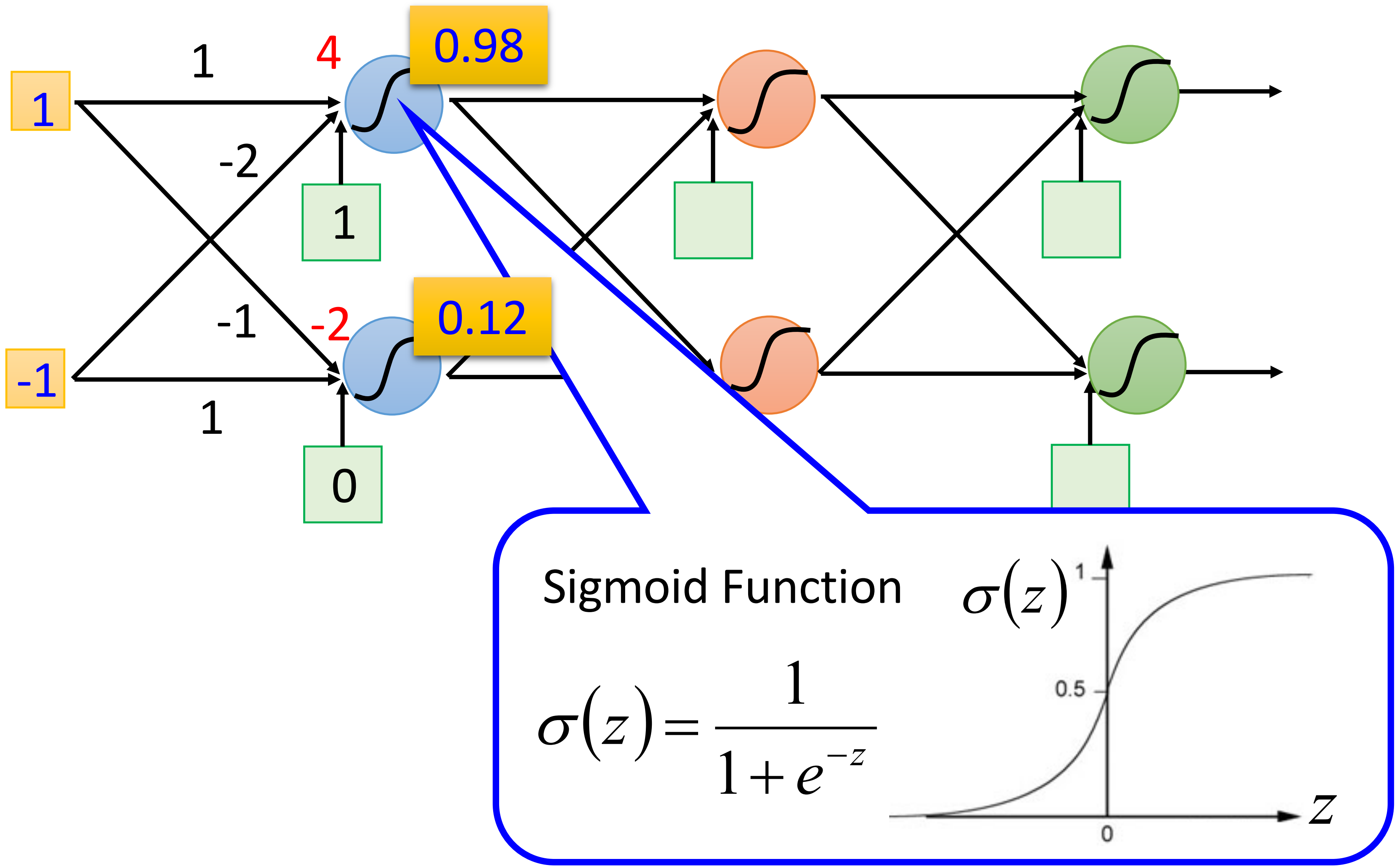


<u>Neural Network</u>

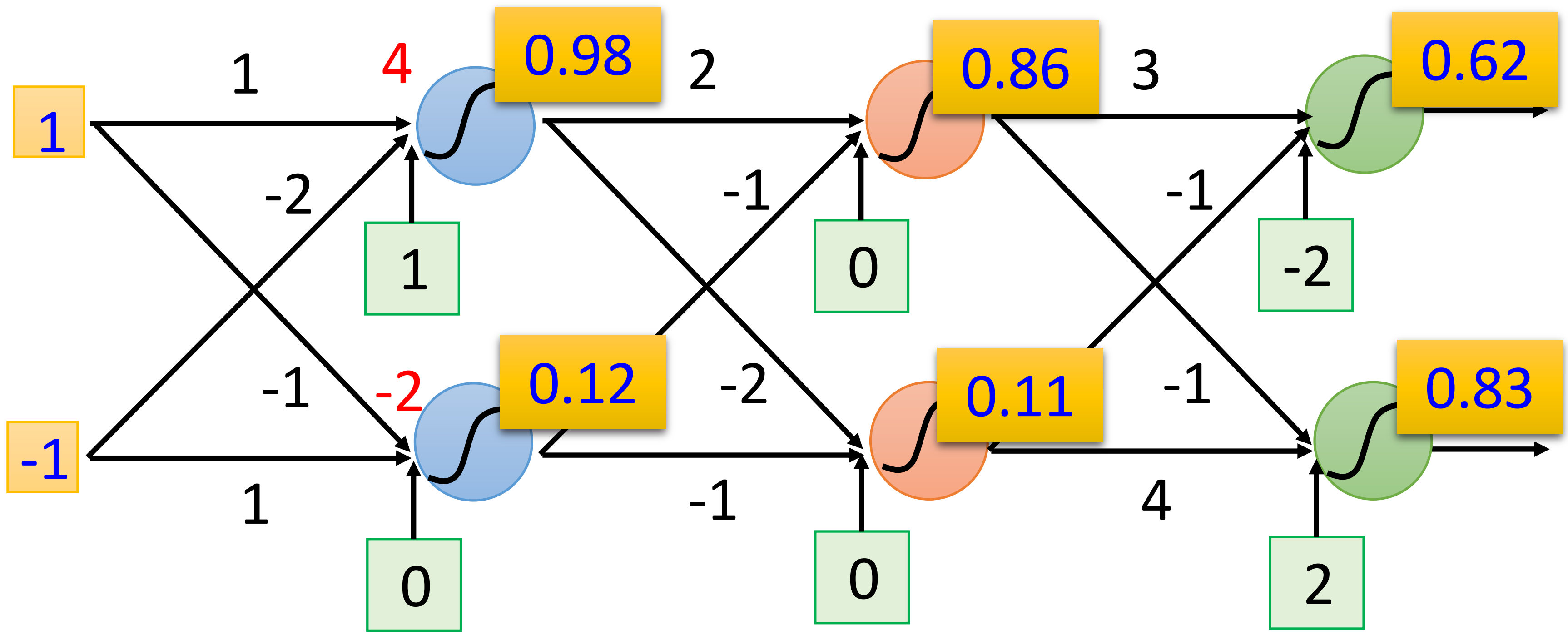
Different connection leads to different network structures

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

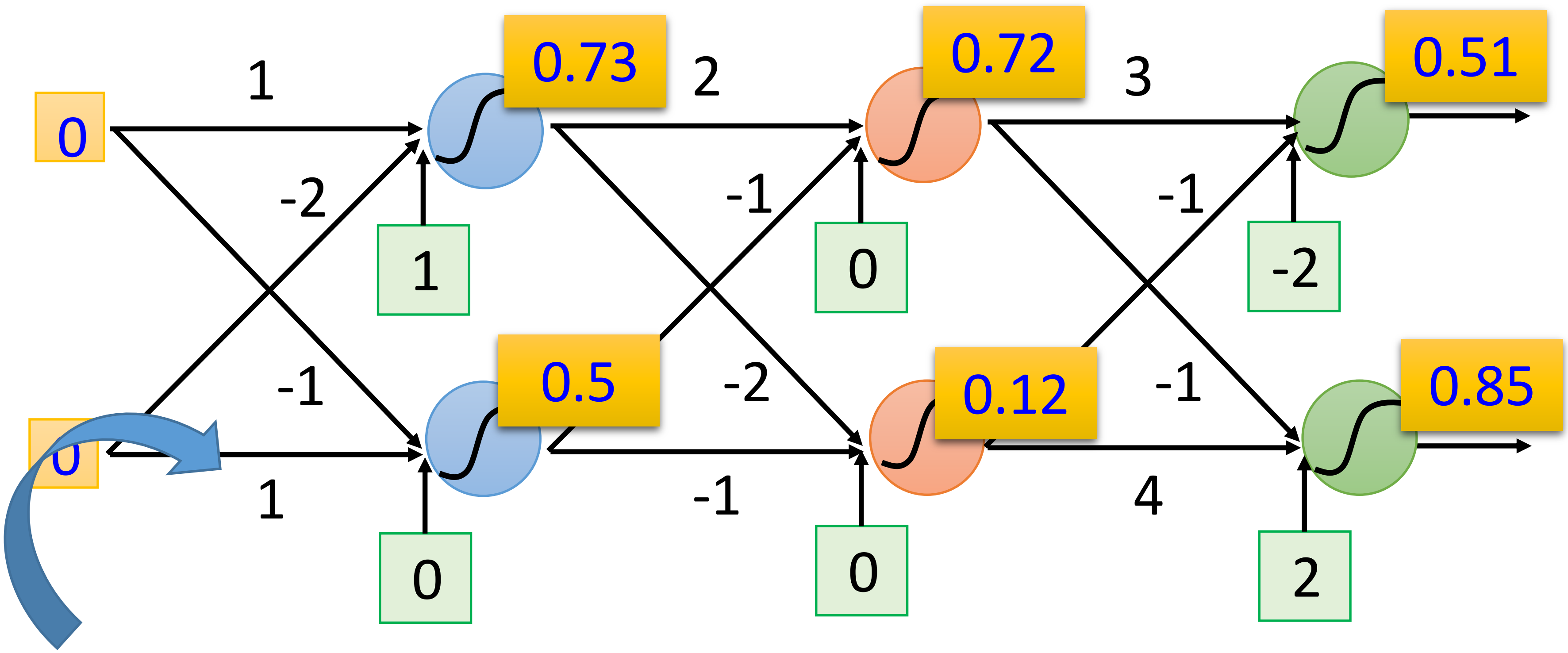
Fully Connect Feedforward Network



Fully Connect Feedforward Network



Fully Connect Feedforward Network



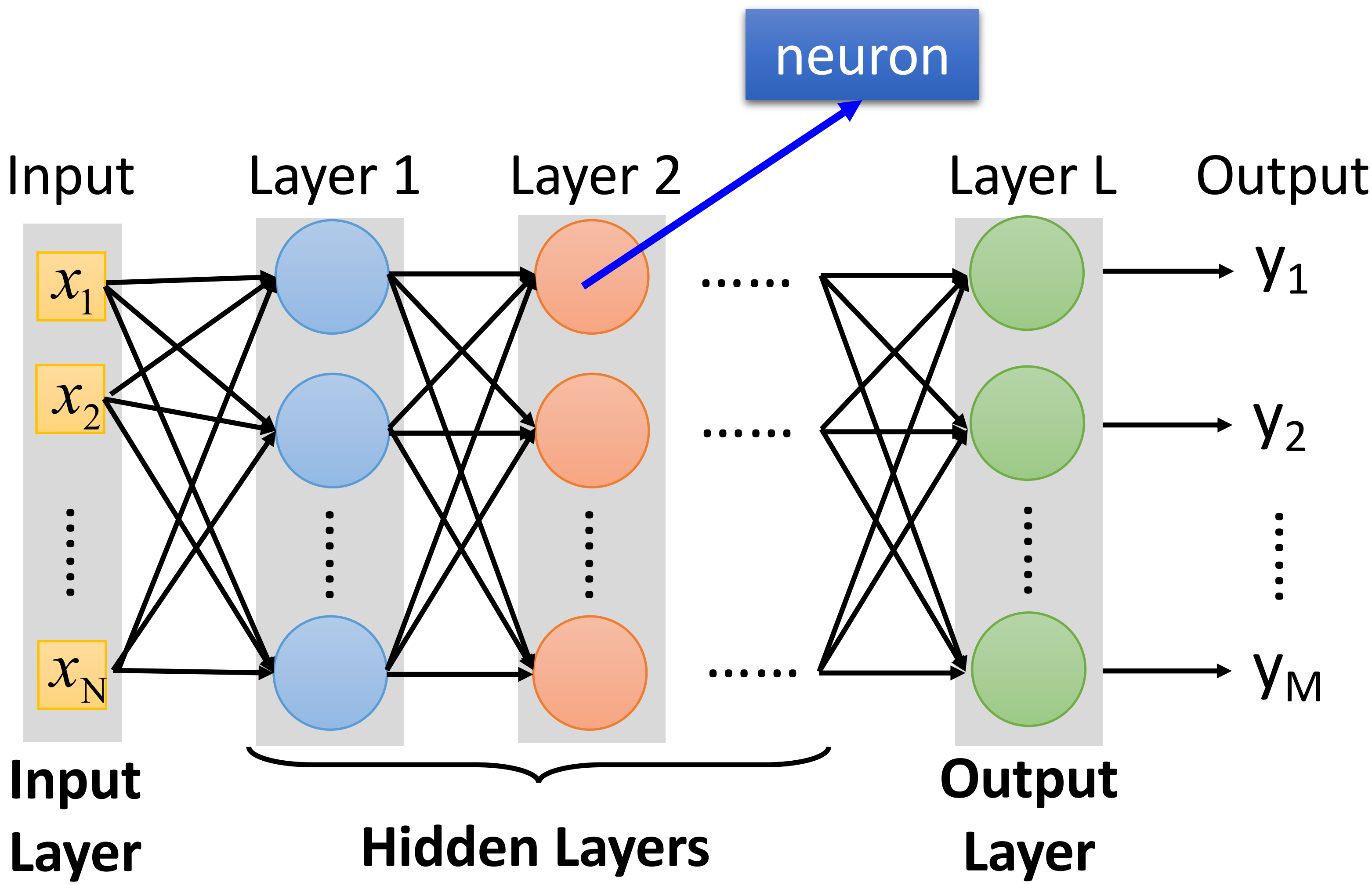
This is a function.

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

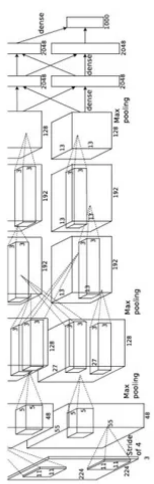


Deep = Many hidden layers

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

8 layers

16.4%



AlexNet (2012)

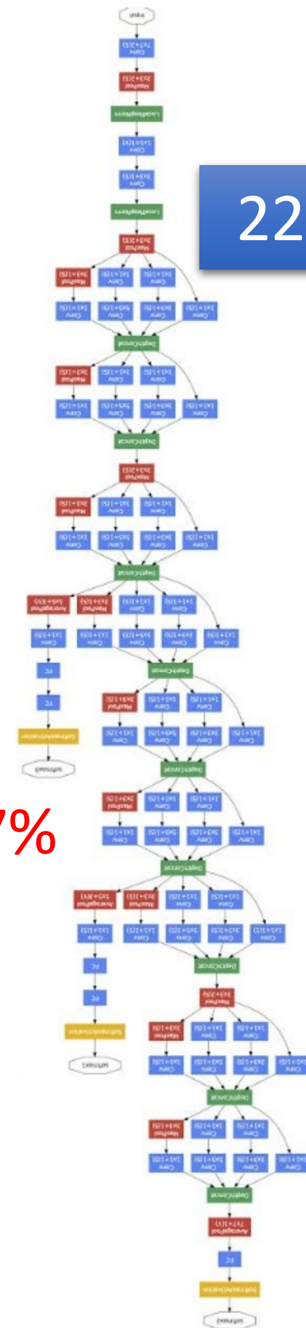
19 layers

7.3%

VGG (2014)

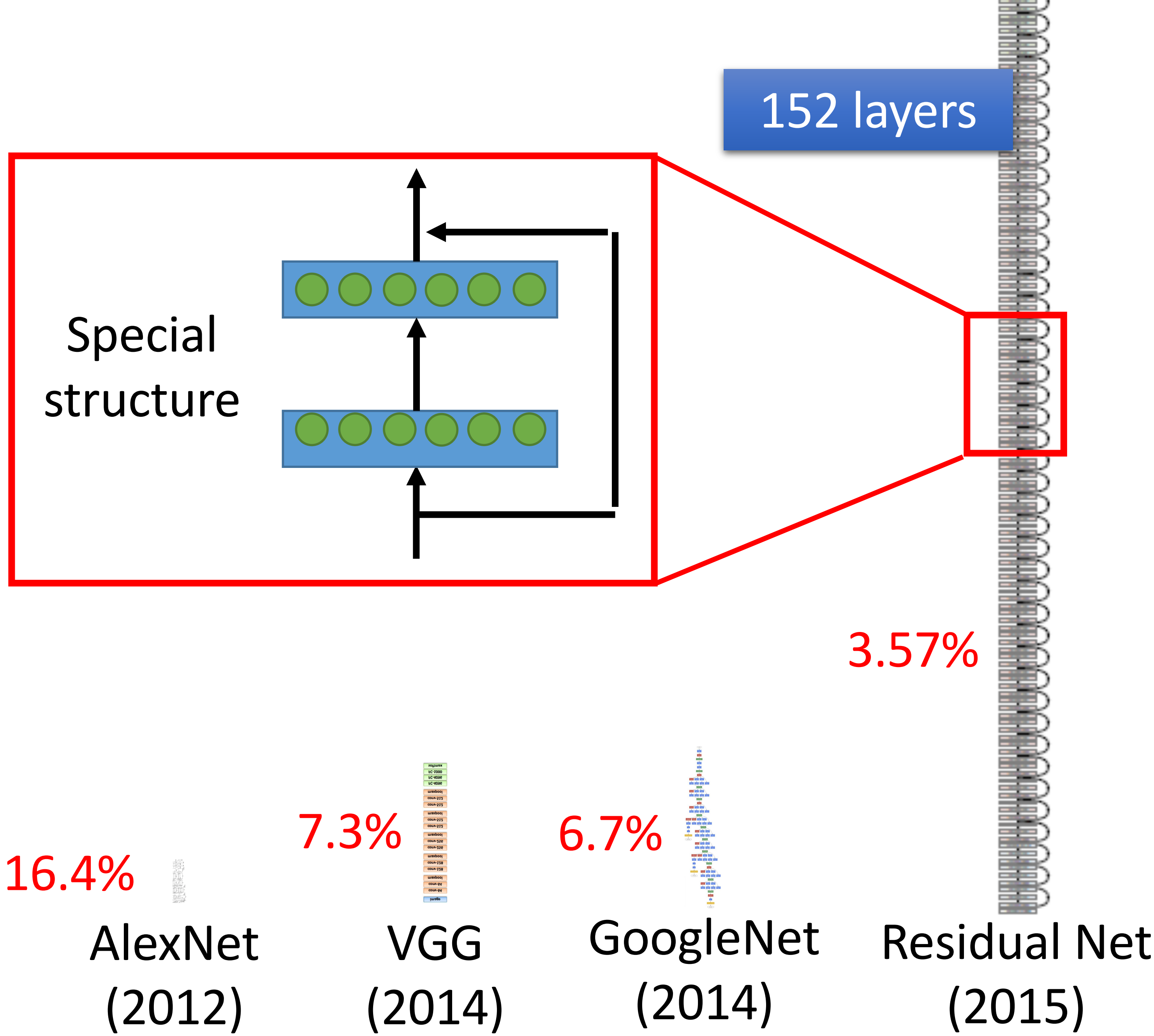
22 layers

6.7%

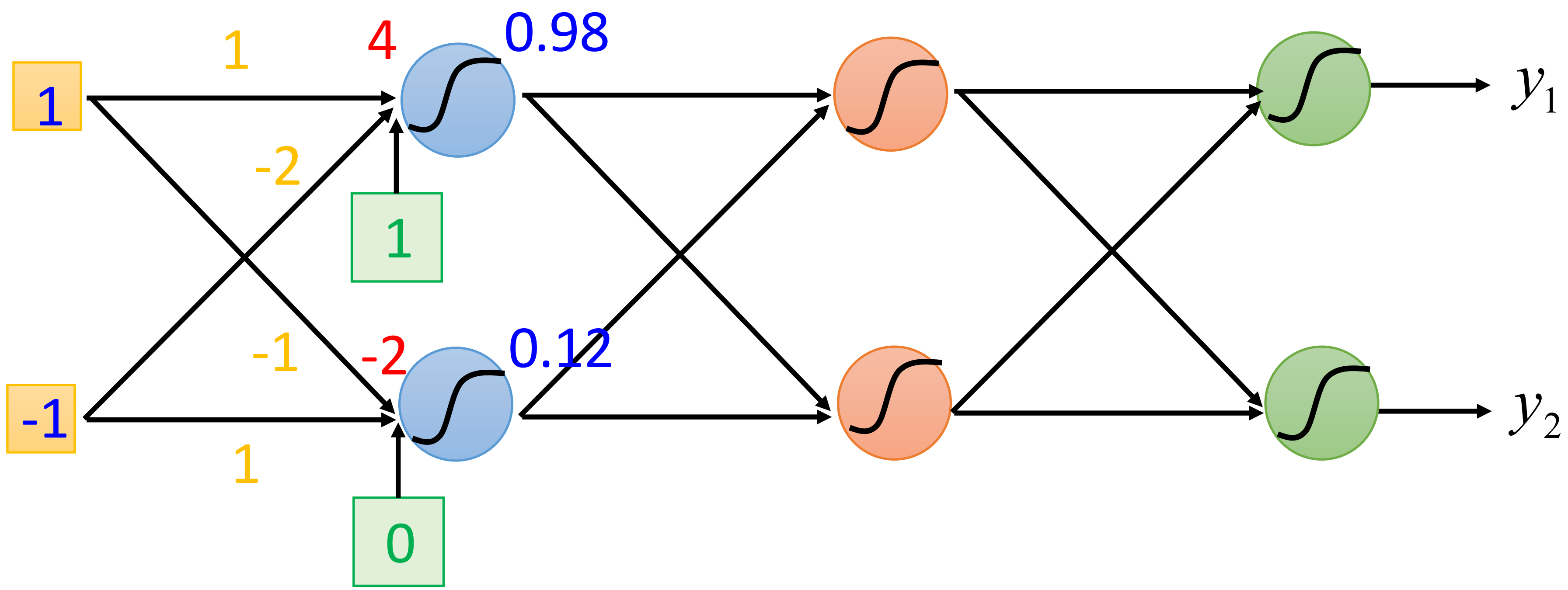


GoogleNet (2014)

Deep = Many hidden layers

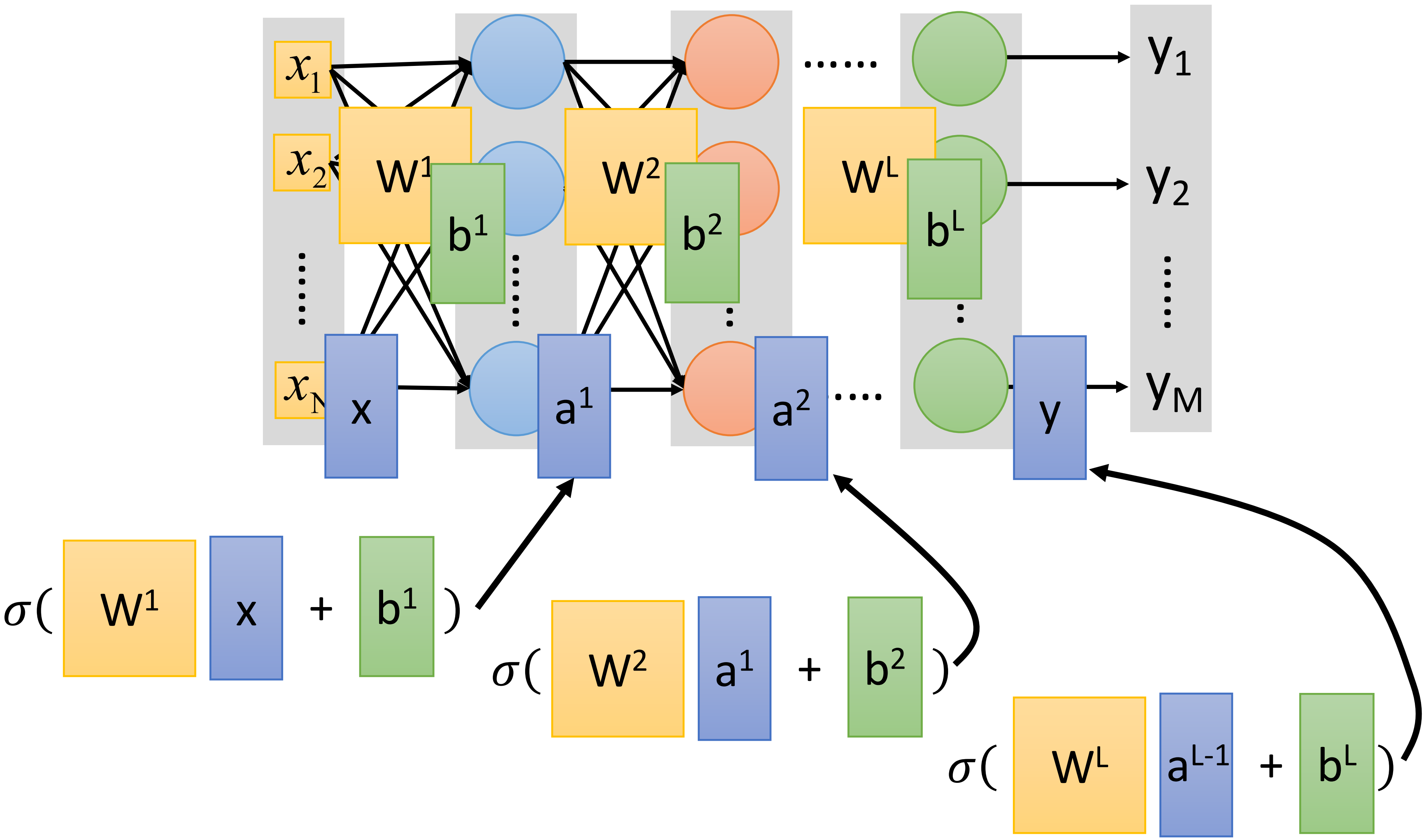


Matrix Operation

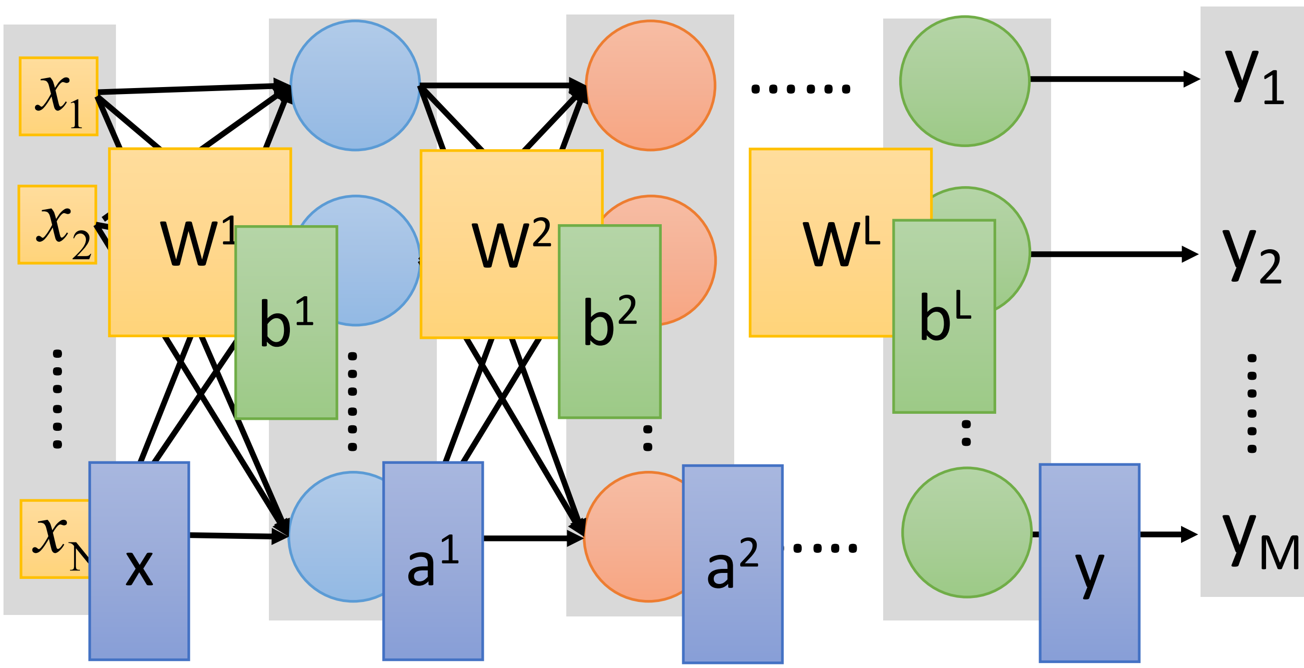


$$\sigma\Bigg(\underbrace{\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\begin{bmatrix} 4 \\ -2 \end{bmatrix}} \Bigg) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

Neural Network



Neural Network

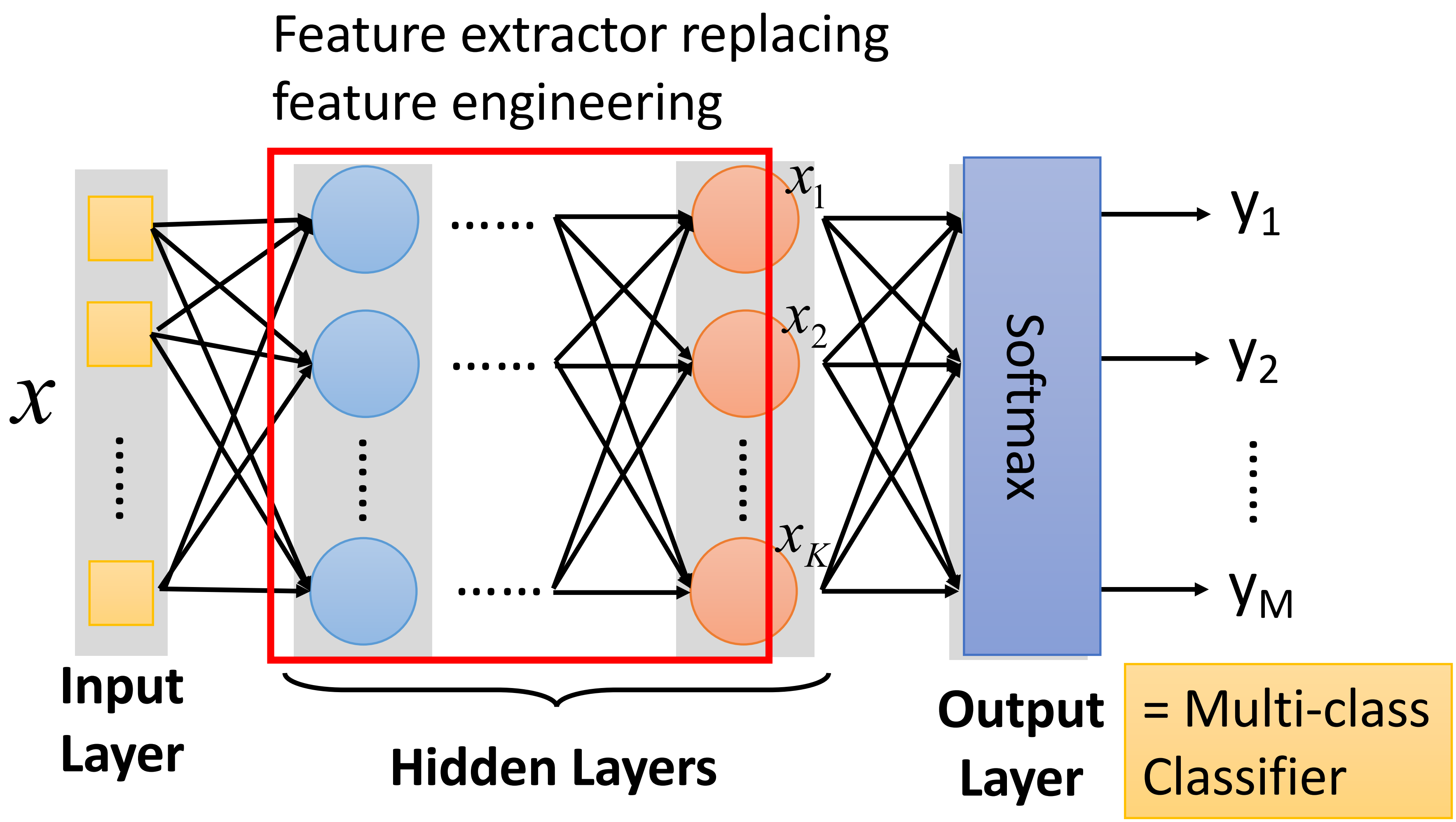


$$y = f(x)$$

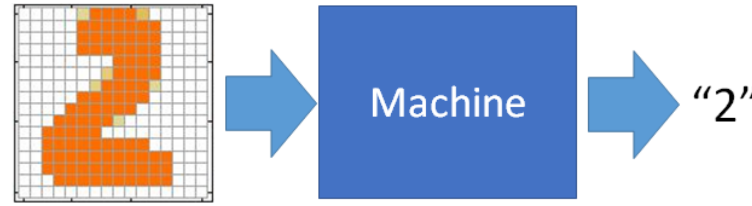
Using parallel computing techniques to speed up matrix operation

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

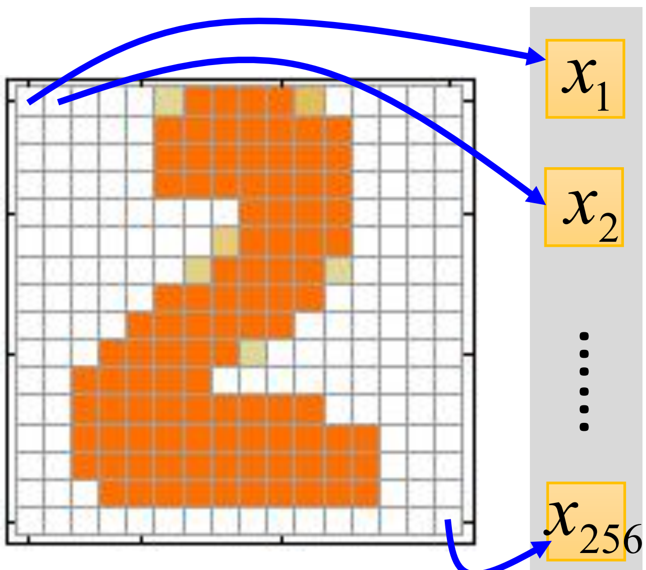
Output Layer
as Multi-Class Classifier



Example Application



Input

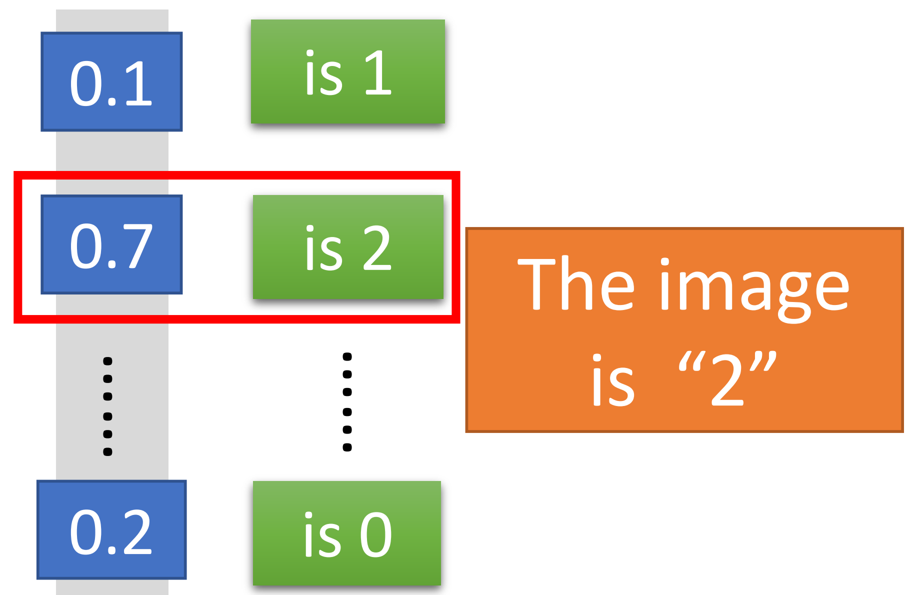


16 x 16 = 256

Ink → 1

No ink → 0

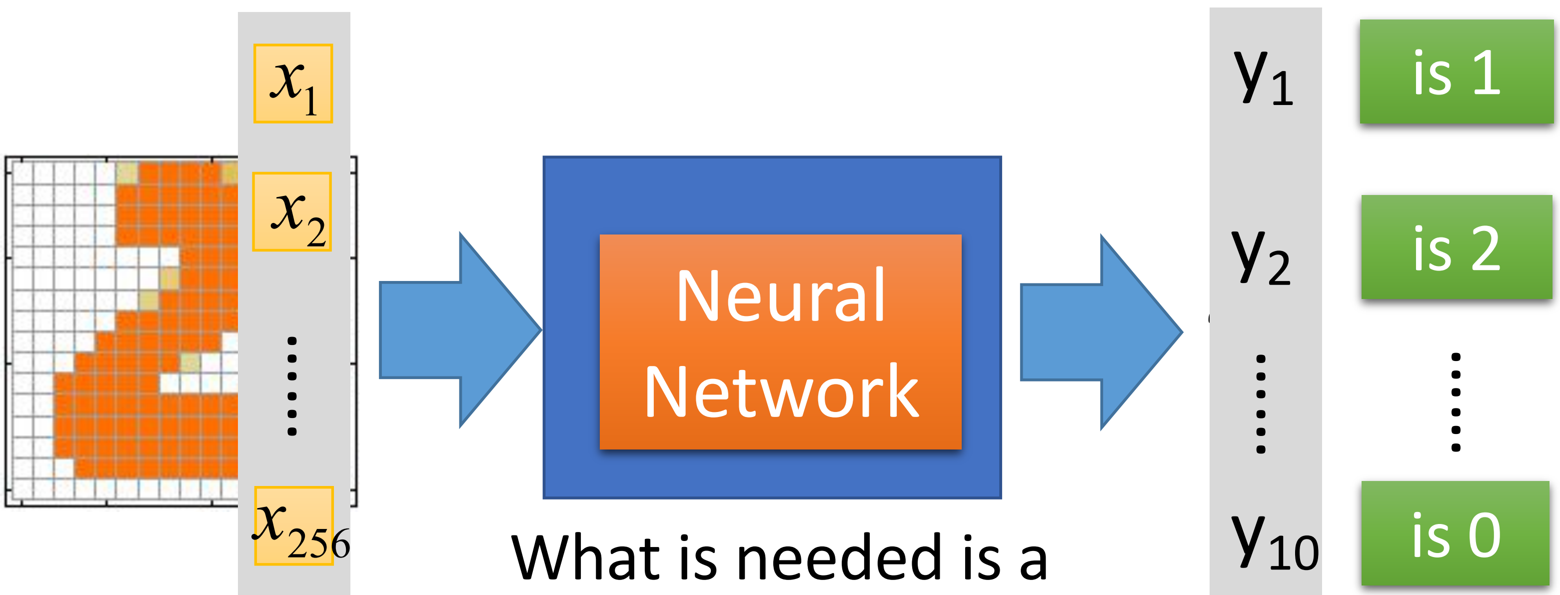
Output



Each dimension represents the confidence of a digit.

Example Application

- Handwriting Digit Recognition

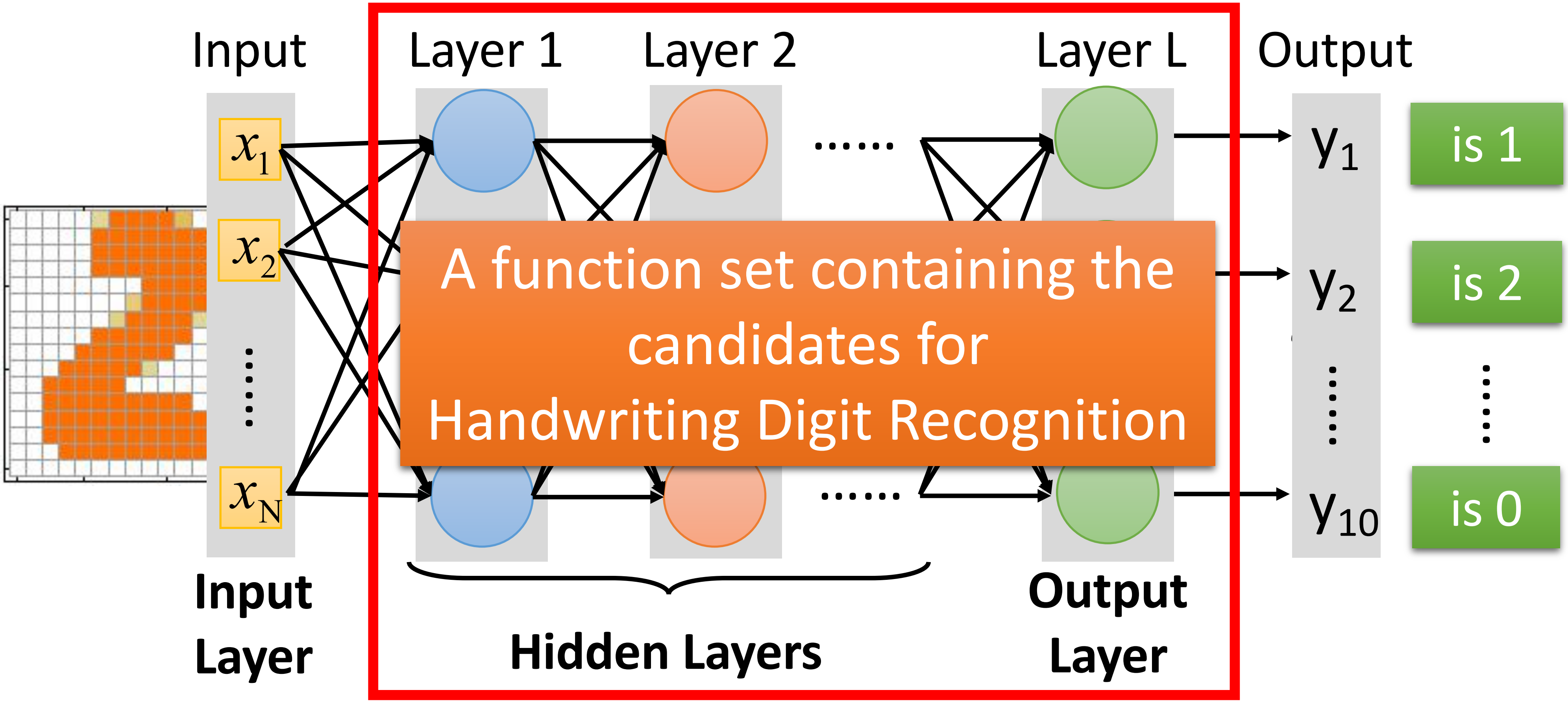


What is needed is a function ……

Input:
256-dim vector

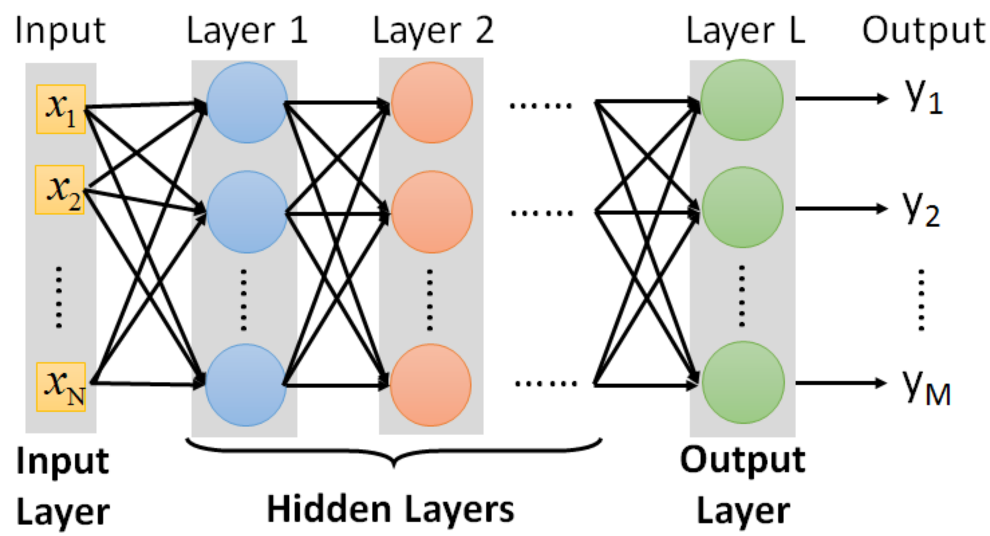
output:
10-dim vector

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

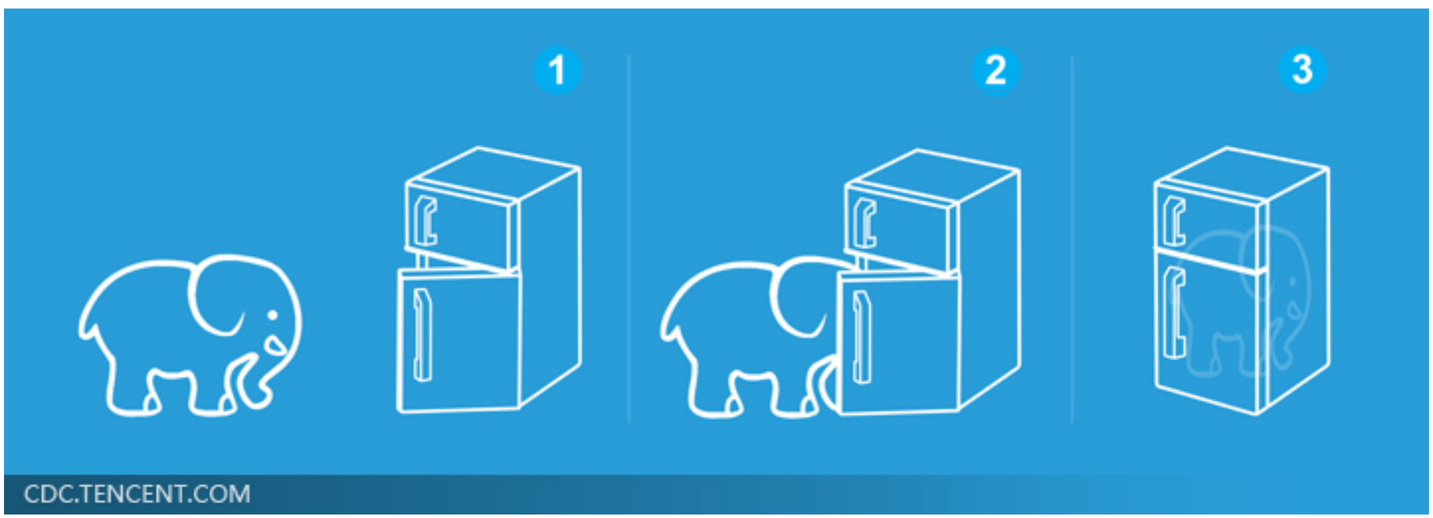
- Q: Can the structure be automatically determined?
  - E.g. Evolutionary Artificial Neural Networks
- Q: Can we design the network structure?

Convolutional Neural Network (CNN)

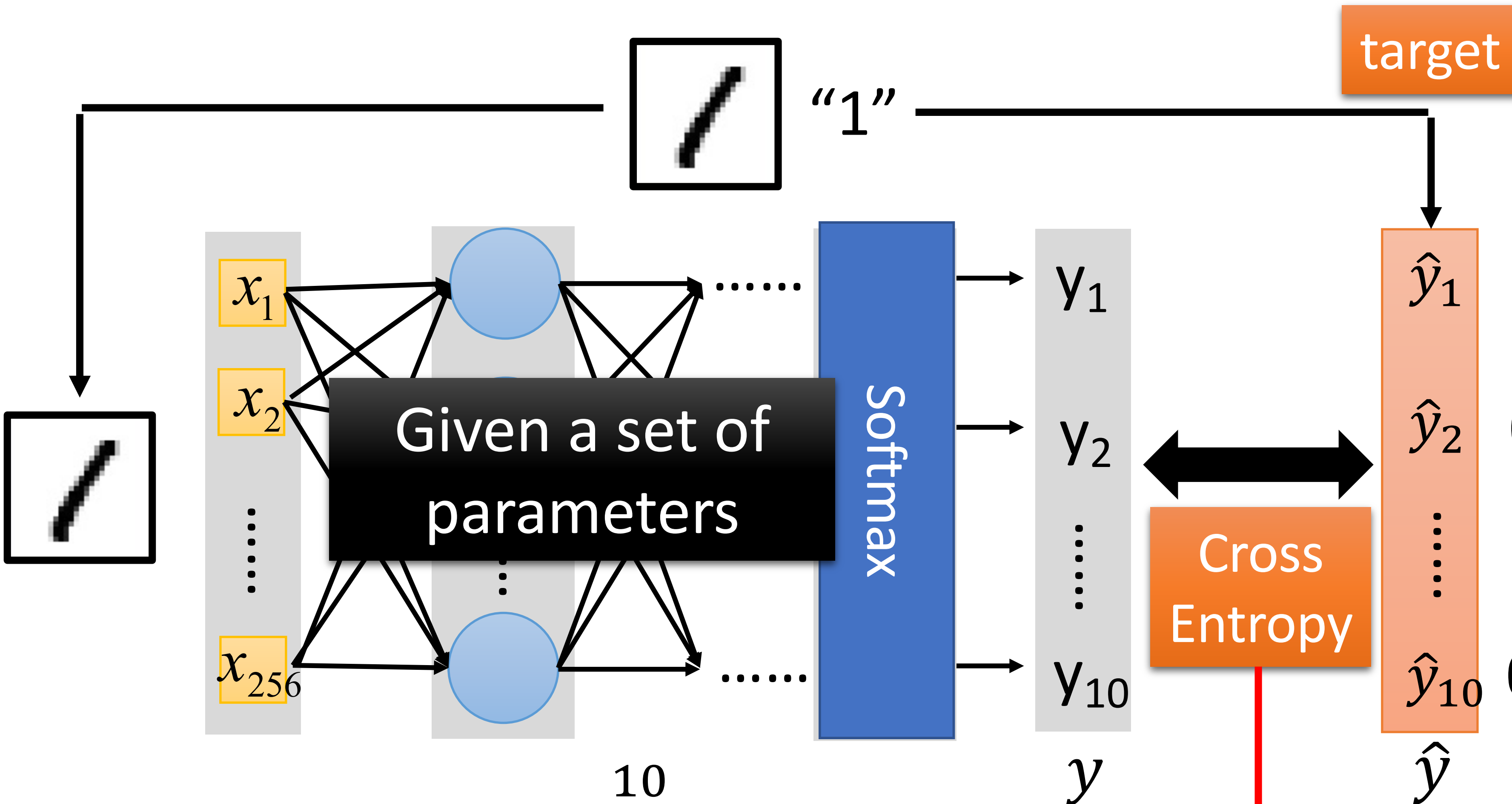
Three Steps for Deep Learning

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

Deep Learning is so simple ……



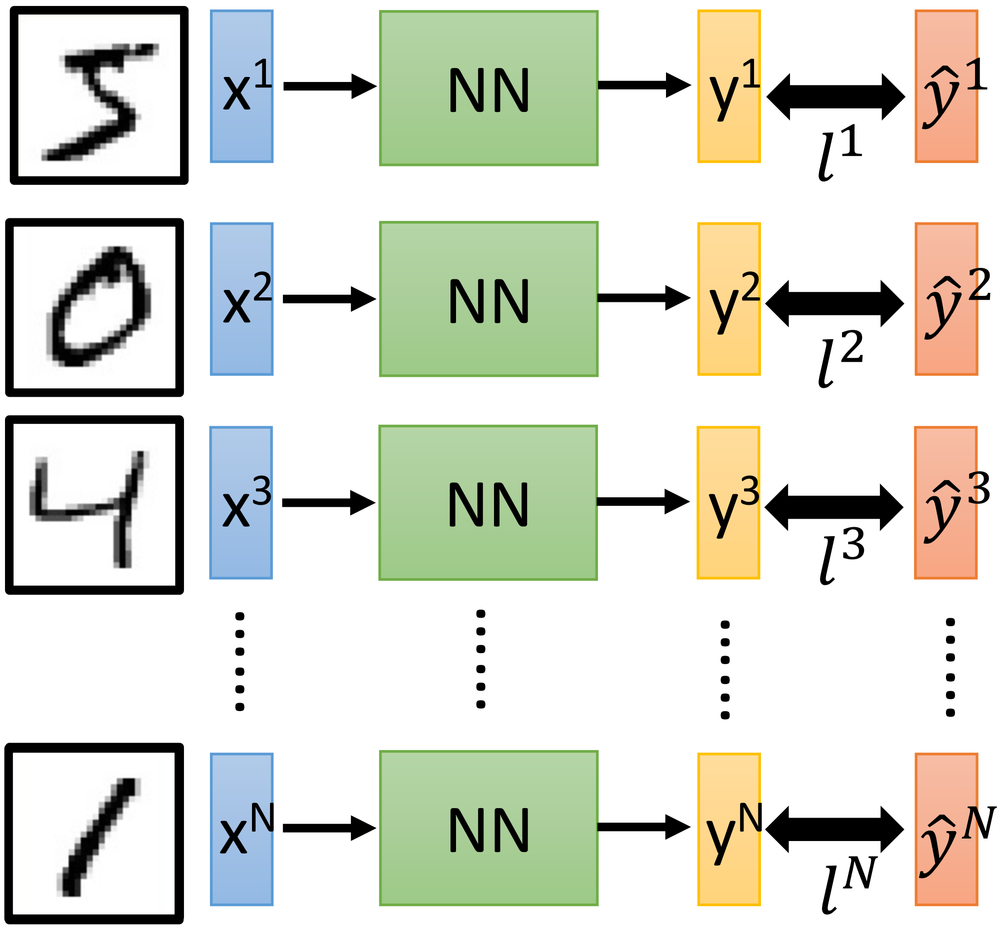
Loss for an Example



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

Total Loss

For all training data …



Total Loss:

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

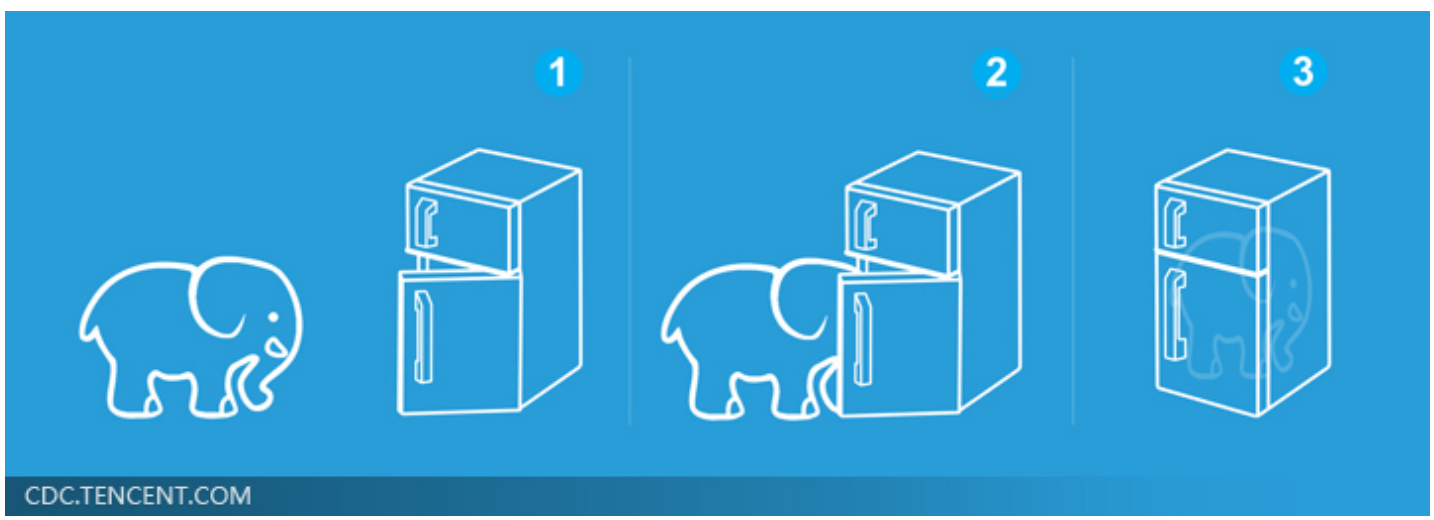
Find ***a function in function set*** that minimizes total loss L

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

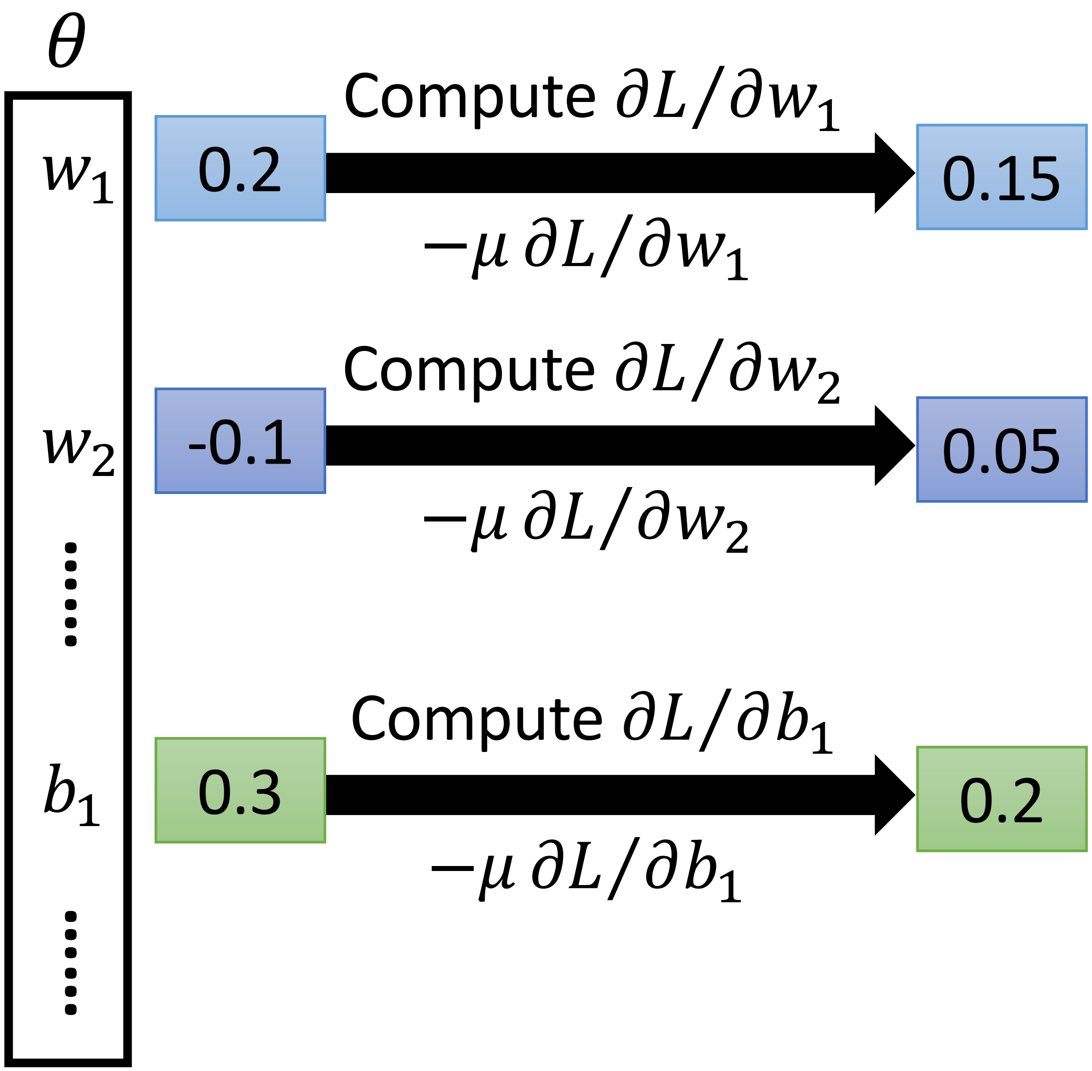
Three Steps for Deep Learning

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

Deep Learning is so simple ……



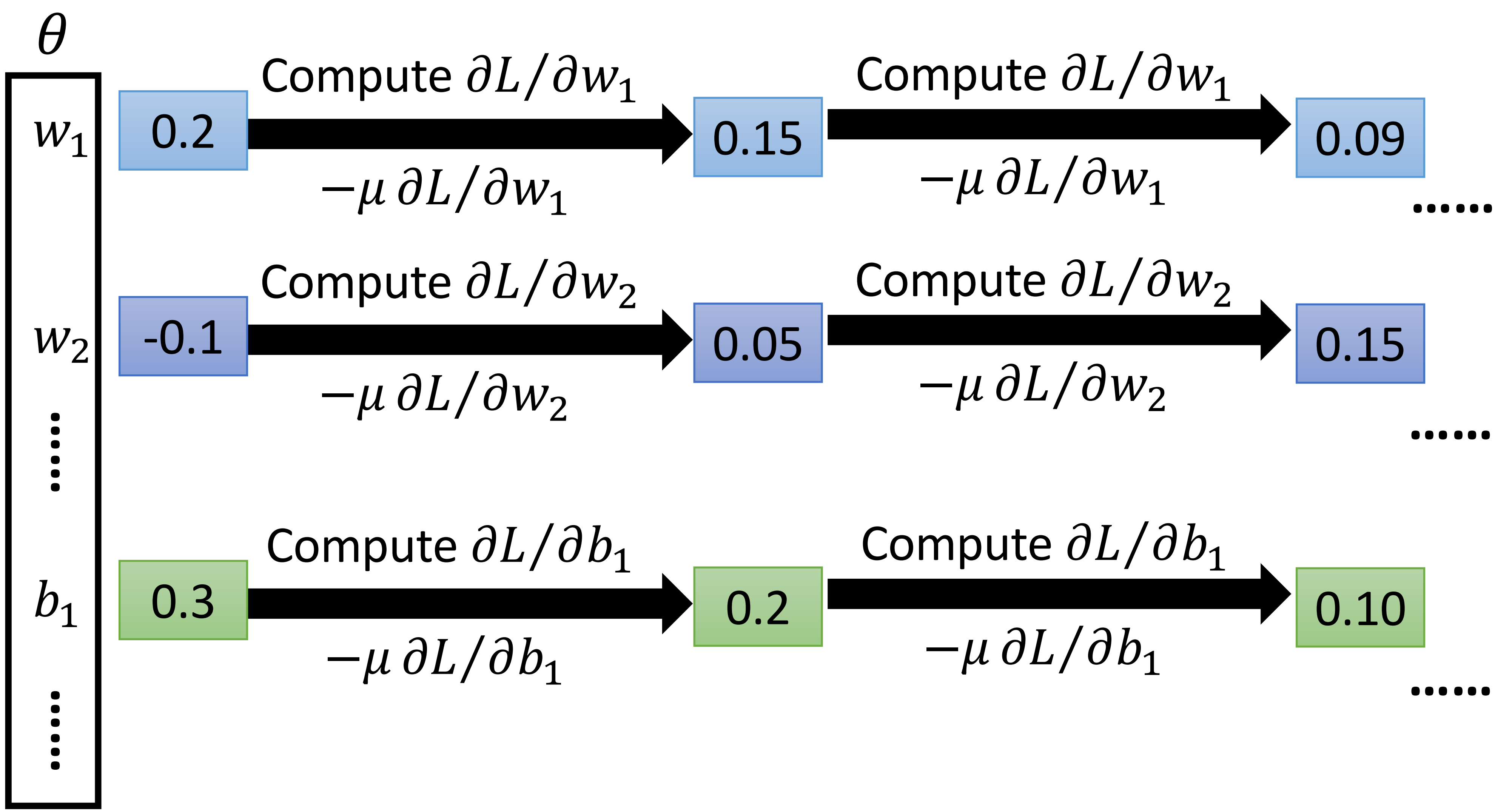
Gradient Descent



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

gradient

Gradient Descent



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

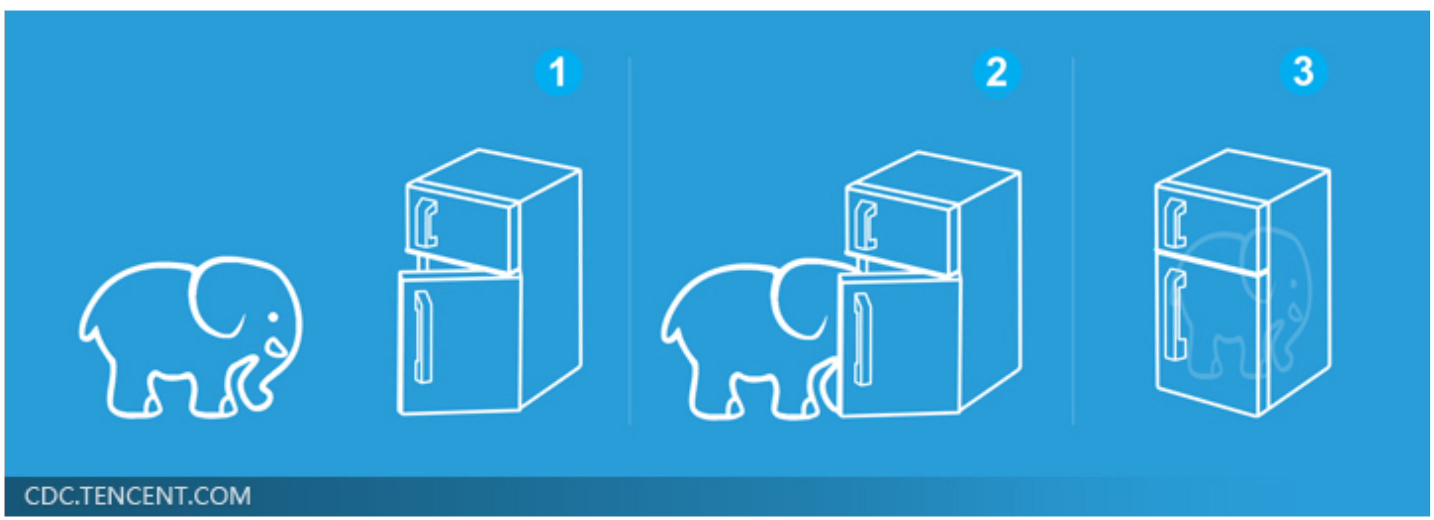
mxnet

libdnn

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 ……



Questions