

Introduction
Sep 18th & 21st, 2017

ADL x MLDS

YUN-NUNG (VIVIAN) CHEN

[HTTP://ADL.MIULAB.TW](http://ADL.MIULAB.TW)
[HTTP://MLDS.MIULAB.TW](http://MLDS.MIULAB.TW)



國立臺灣大學
National Taiwan University

Slides credited from Prof. Hung-Yi Lee

Announcement

Contact

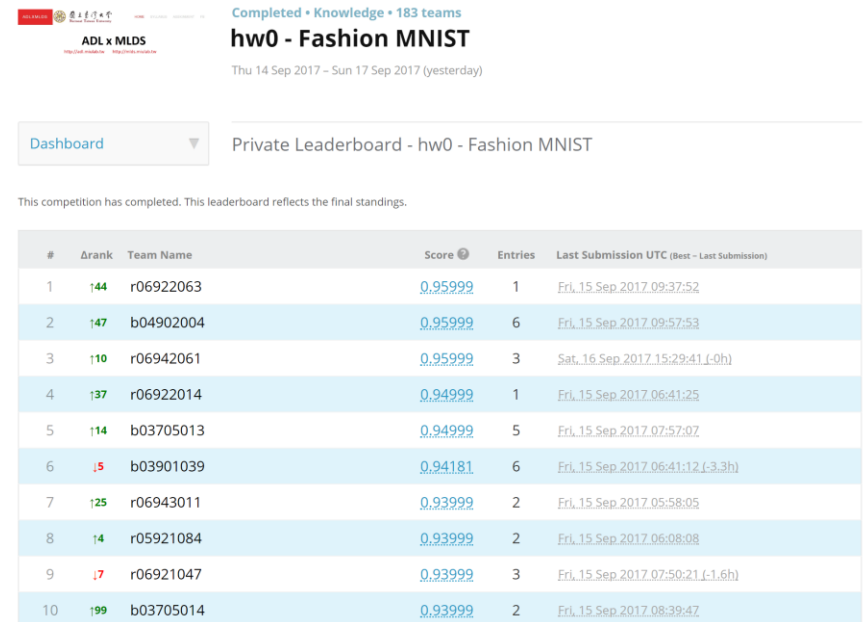
- TA email: adlxmlds@gmail.com
- Please include your 學號 and 姓名 in the email

Course Registration

- Registration codes for ADL were sent out
- Registration codes for MLDS will be sent out soon
- You are free to come to the class on Mon/Thur

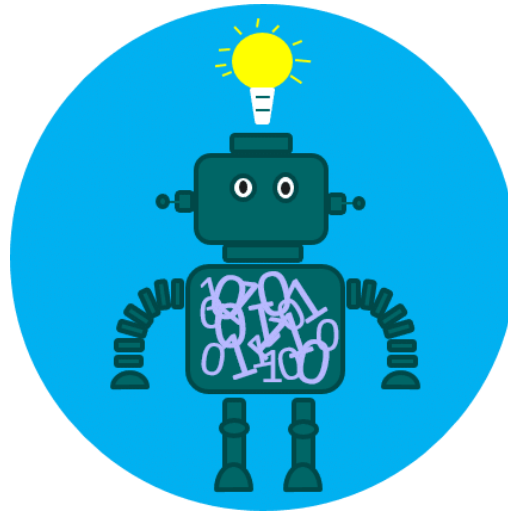
Others

- Additional bonus if you pass the HW0
- Please register the course ASAP
- The registration for GPU resources will be announced by the end of today's class



The screenshot shows the competition dashboard for 'ADL x MLDS' with the title 'hw0 - Fashion MNIST'. It indicates that the competition is completed and that 183 teams have participated. The competition ran from Thursday, September 14, 2017, to Sunday, September 17, 2017. The dashboard includes a 'Dashboard' dropdown menu and a 'Private Leaderboard - hw0 - Fashion MNIST' section. Below this, a message states: 'This competition has completed. This leaderboard reflects the final standings.' The leaderboard table is as follows:

#	Rank	Team Name	Score	Entries	Last Submission UTC (Best - Last Submission)
1	144	r06922063	0.95999	1	Fri, 15 Sep 2017 09:37:52
2	147	b04902004	0.95999	6	Fri, 15 Sep 2017 09:57:53
3	110	r06942061	0.95999	3	Sat, 16 Sep 2017 15:29:41 (-0h)
4	137	r06922014	0.94999	1	Fri, 15 Sep 2017 06:41:25
5	114	b03705013	0.94999	5	Fri, 15 Sep 2017 07:57:07
6	15	b03901039	0.94181	6	Fri, 15 Sep 2017 06:41:12 (-3.3h)
7	125	r06943011	0.93999	2	Fri, 15 Sep 2017 05:58:05
8	14	r05921084	0.93999	2	Fri, 15 Sep 2017 06:08:08
9	17	r06921047	0.93999	3	Fri, 15 Sep 2017 07:50:21 (-1.6h)
10	199	b03705014	0.93999	2	Fri, 15 Sep 2017 08:39:47



What is Machine Learning?

What Computers Can Do?



Programs can do the things you ask them to do

Program for Solving Tasks

Task: predicting positive or negative given a product review

“I love this product!” “It claims too much.” “It’s a little expensive.”

↓ program.py

+

if input contains “love”, “like”, etc.
output = positive

↓ program.py

-

if input contains “too much”, “bad”, etc.
output = negative

↓ program.py

?

“台灣第一波上市!”

↓ program.py

推

“規格好雞肋...”

↓ program.py

噓

“樓下買了我才考慮”

↓ program.py

?

Some tasks are complex, and we don’t know how to write a program to solve them.

Learning \approx Looking for a Function

Task: predicting positive or negative given a product review

“I love this product!” “It claims too much.” “It’s a little expensive.”

↓ f

+

↓ f

-

↓ f

?

“台灣第一波上市!”

↓ f

推

“規格好雞肋...”

↓ f

噓

“樓下買了我才考慮”

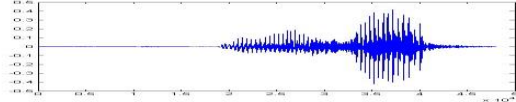
↓ f

?

Given a large amount of data, the machine learns what the function f should be.

Learning \approx Looking for a Function



Speech Recognition

$$f(\text{  }) = \text{“你好”}$$

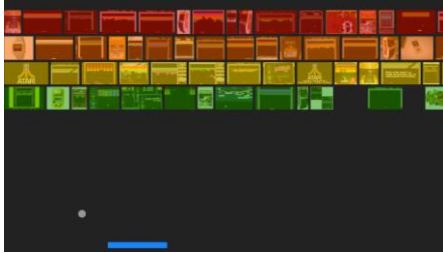
Handwritten Recognition

$$f(\text{  }) = \text{“2”}$$

Weather forecast

$$f(\text{  Thursday }) = \text{“  Saturday”}$$

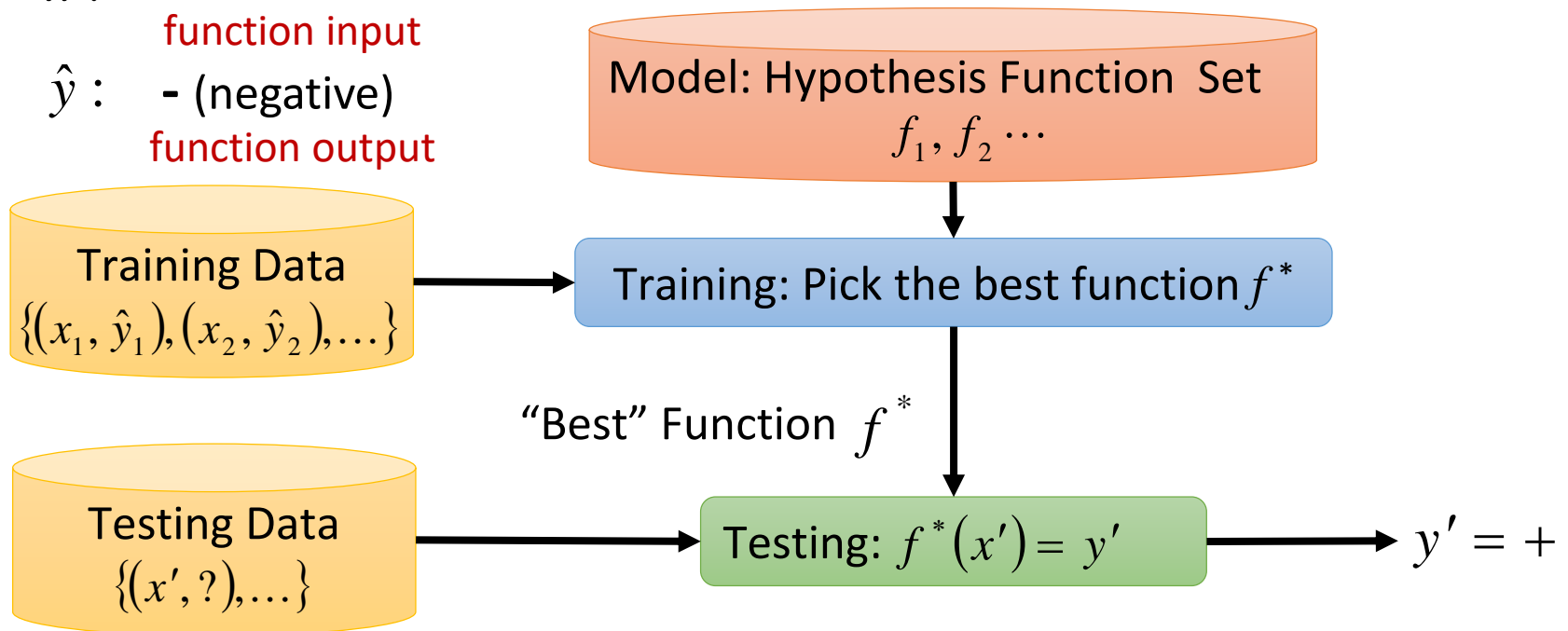
Play video games

$$f(\text{  }) = \text{“move left”}$$

Machine Learning Framework

x : “It claims too much.”
function input

\hat{y} : - (negative)
function output



Training is to pick the best function given the observed data
Testing is to predict the label using the learned function

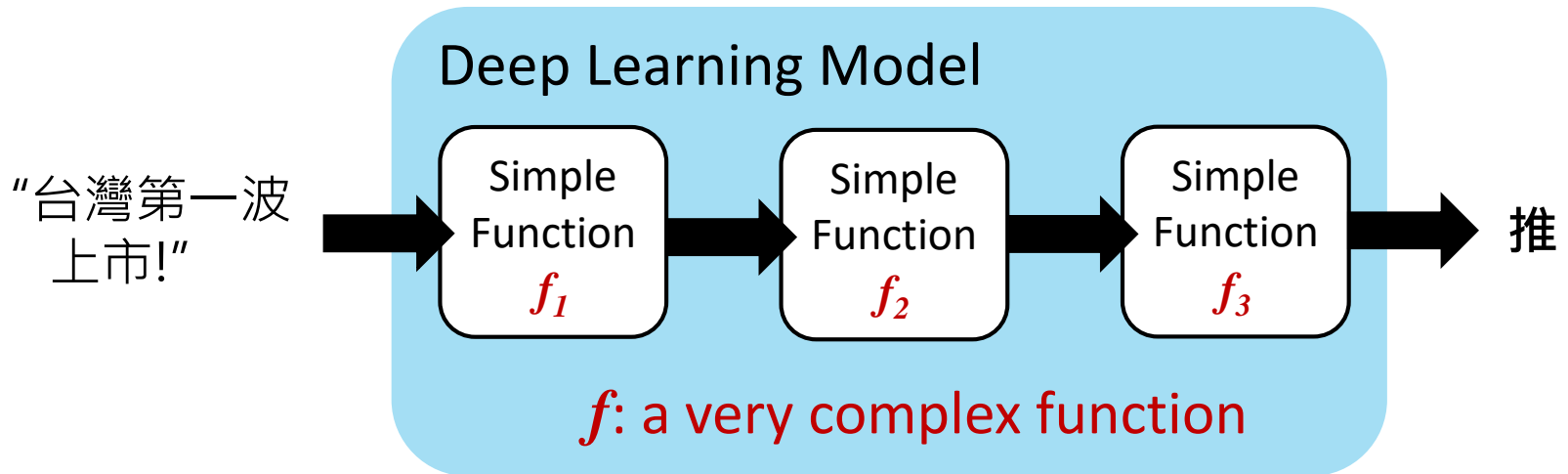


What is Deep Learning?

A subfield of machine learning

Stacked Functions Learned by Machine

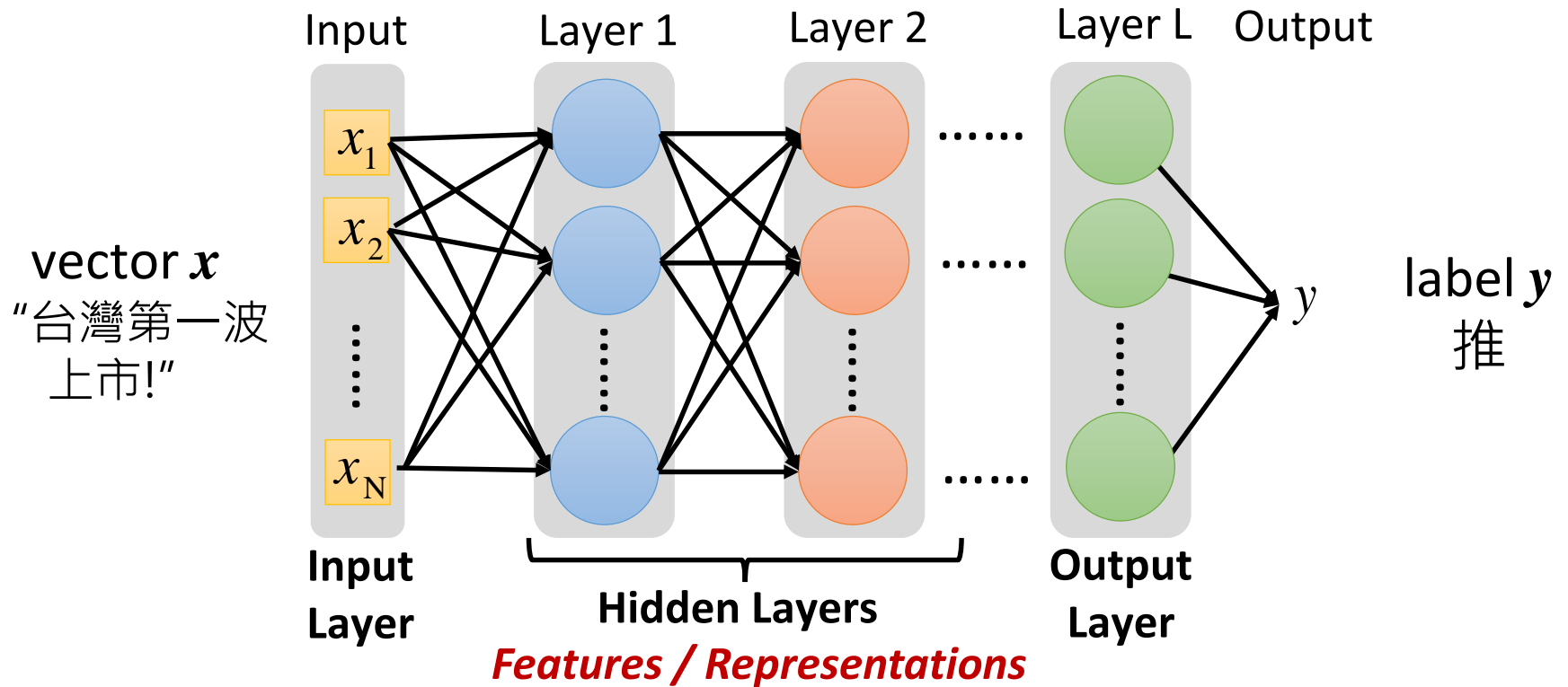
Production line (生產線)



End-to-end training: what each function should do is learned automatically

Deep learning usually refers to *neural network* based model

Stacked Functions Learned by Machine

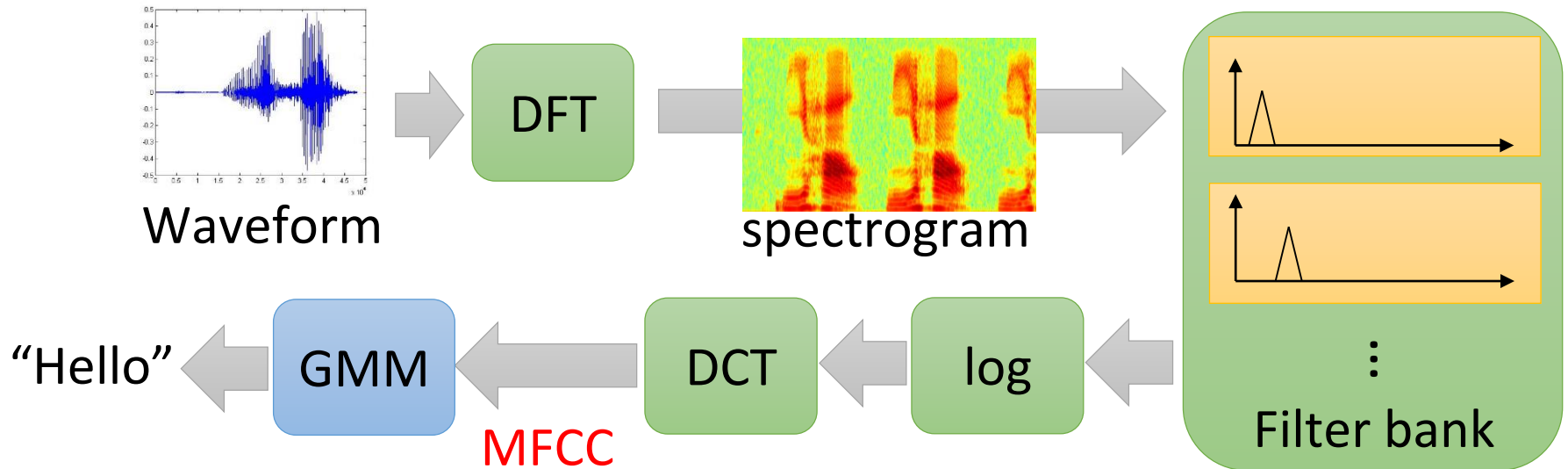


Representation Learning attempts to learn good features/representations


Deep Learning attempts to learn (multiple levels of) representations and an output

Deep v.s. Shallow – Speech Recognition

Shallow Model

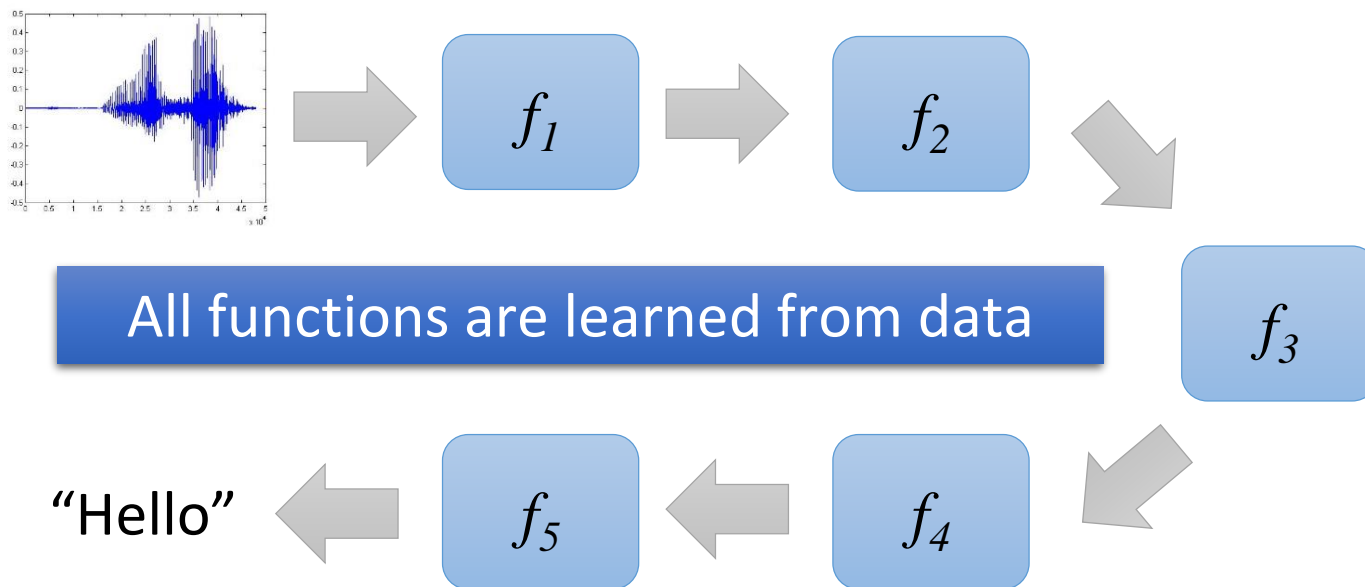


Each box is a simple function in the production line:

 :hand-crafted  :learned from data

Deep v.s. Shallow – Speech Recognition

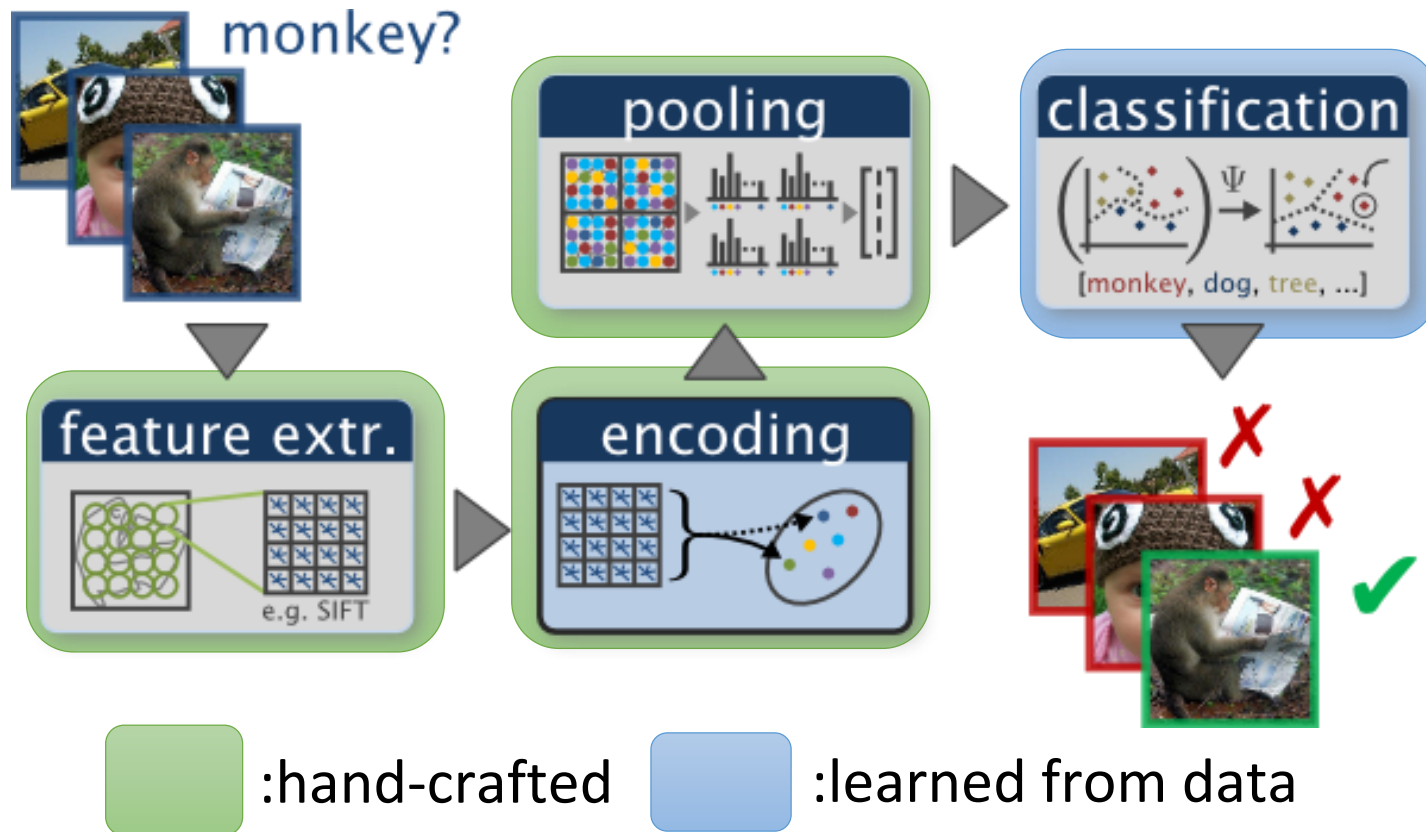
Deep Model



Less engineering labor, but machine learns more

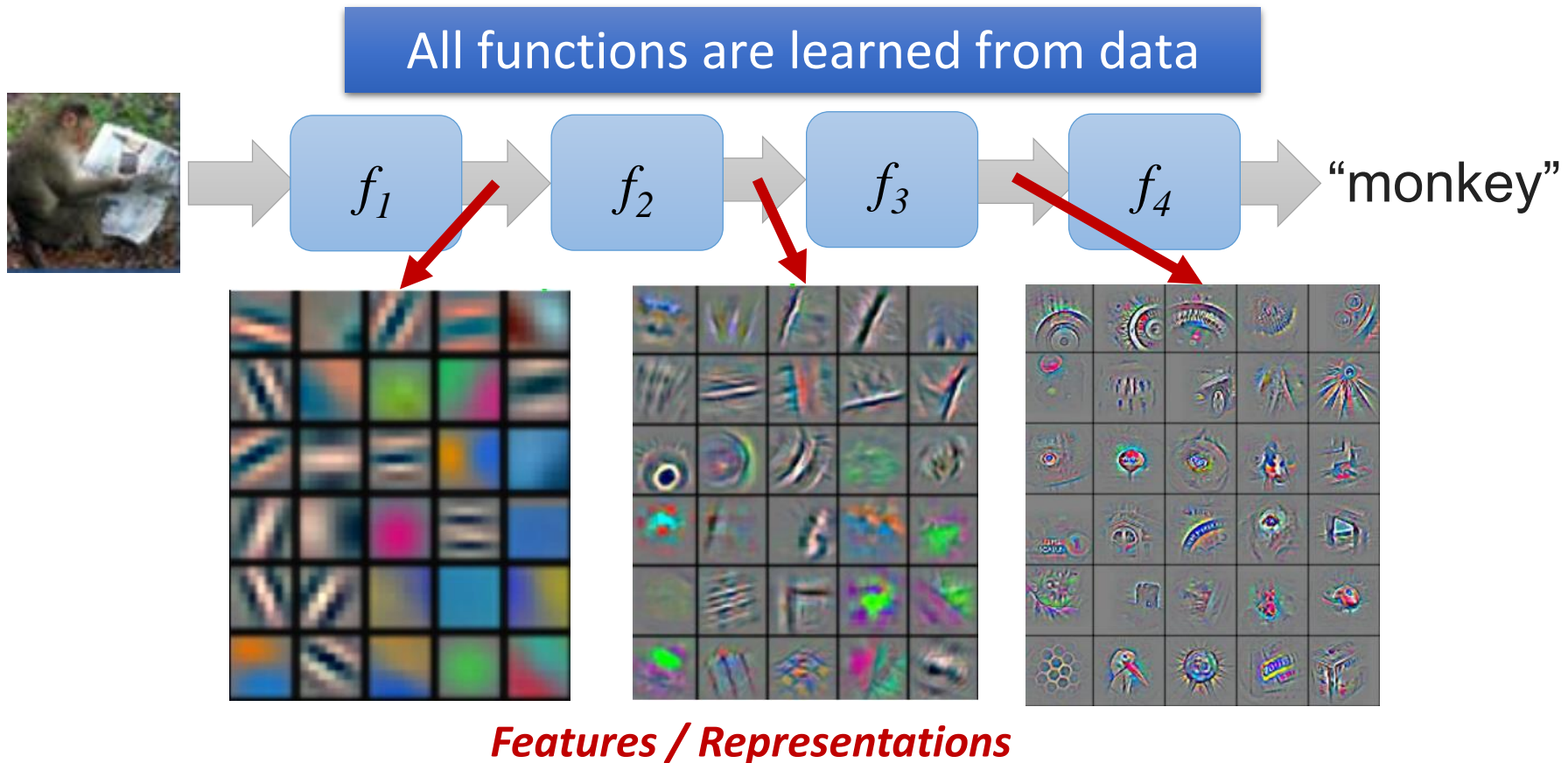
Deep v.s. Shallow – Image Recognition

Shallow Model

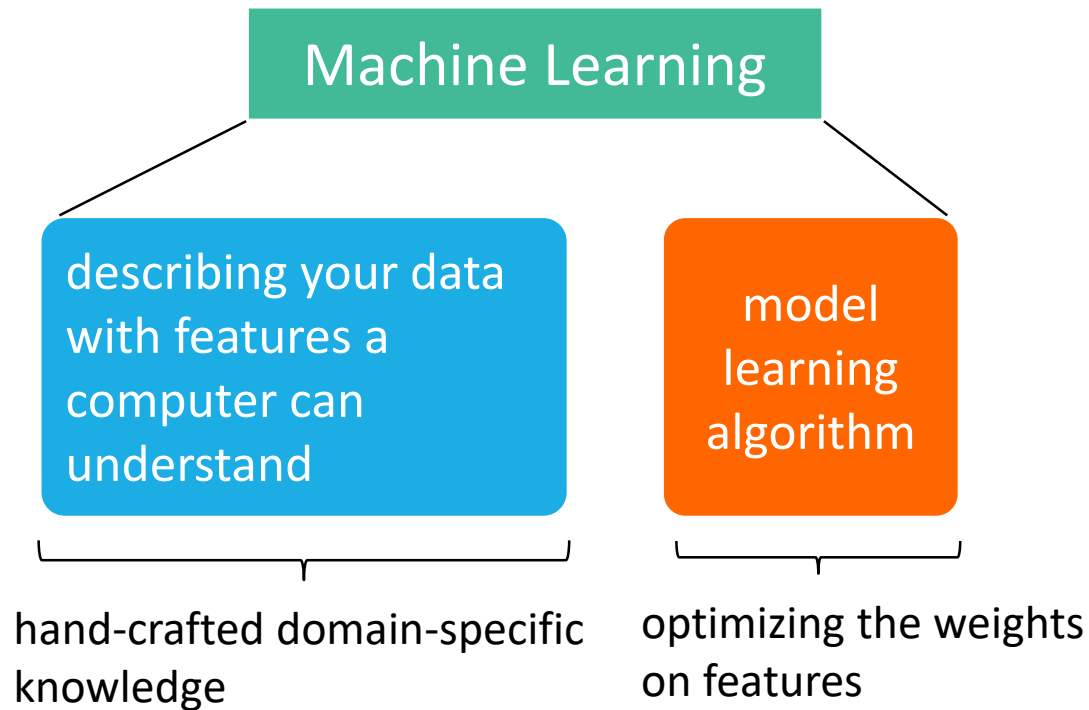


Deep v.s. Shallow – Image Recognition

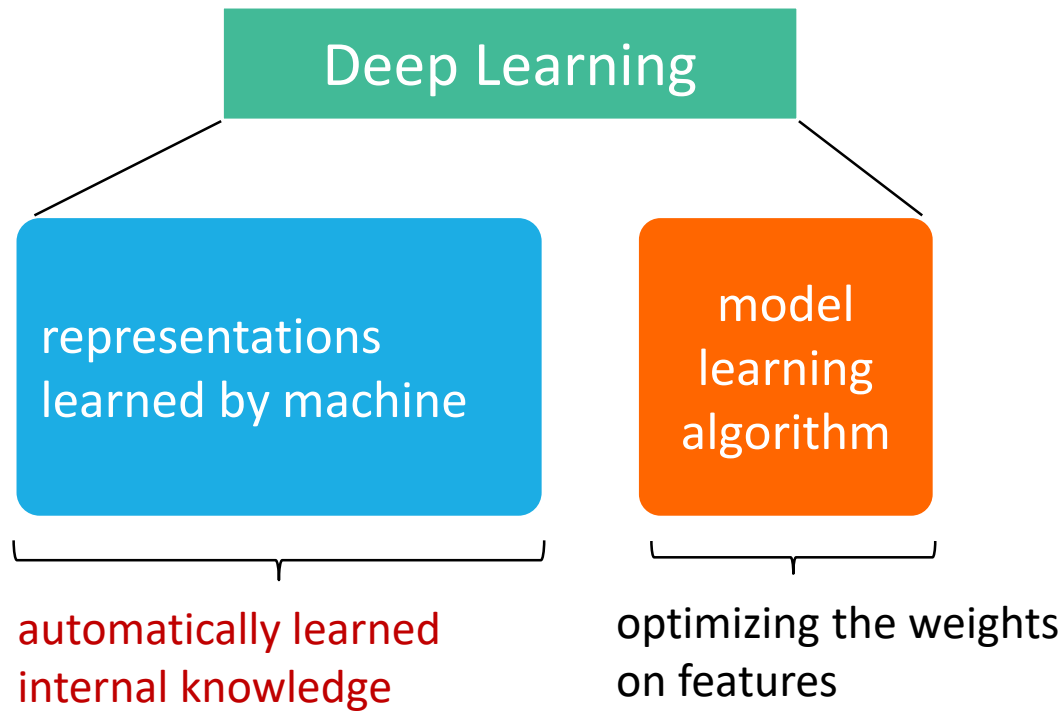
Deep Model



Machine Learning v.s. Deep Learning

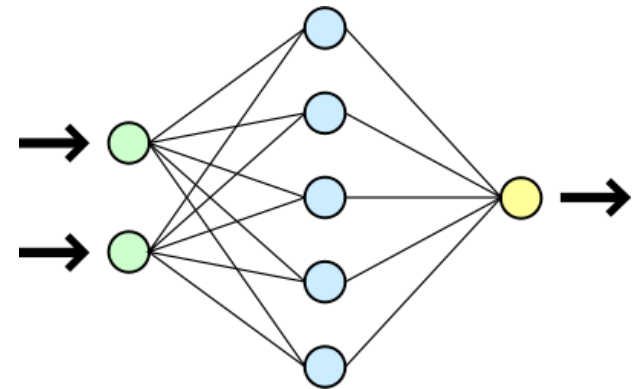
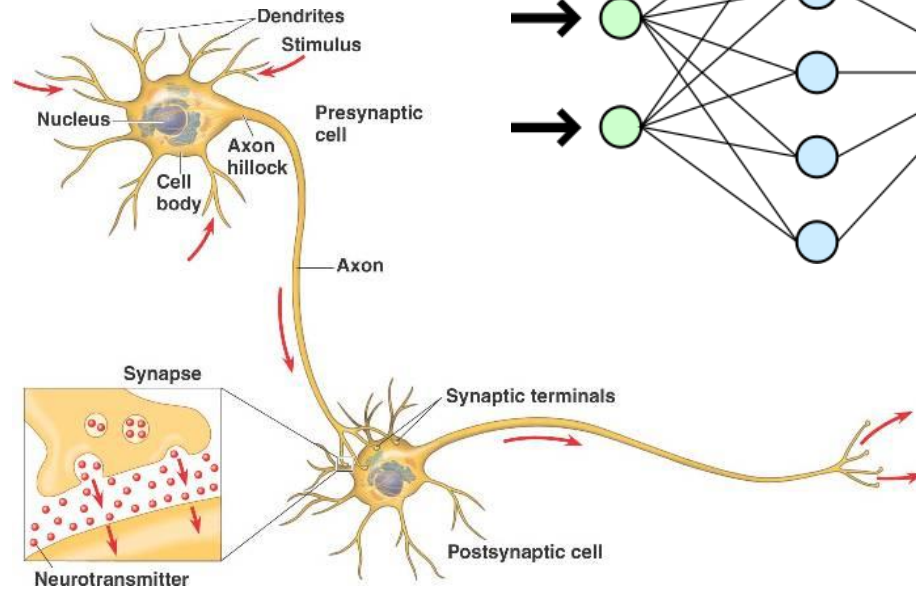


Machine Learning v.s. Deep Learning

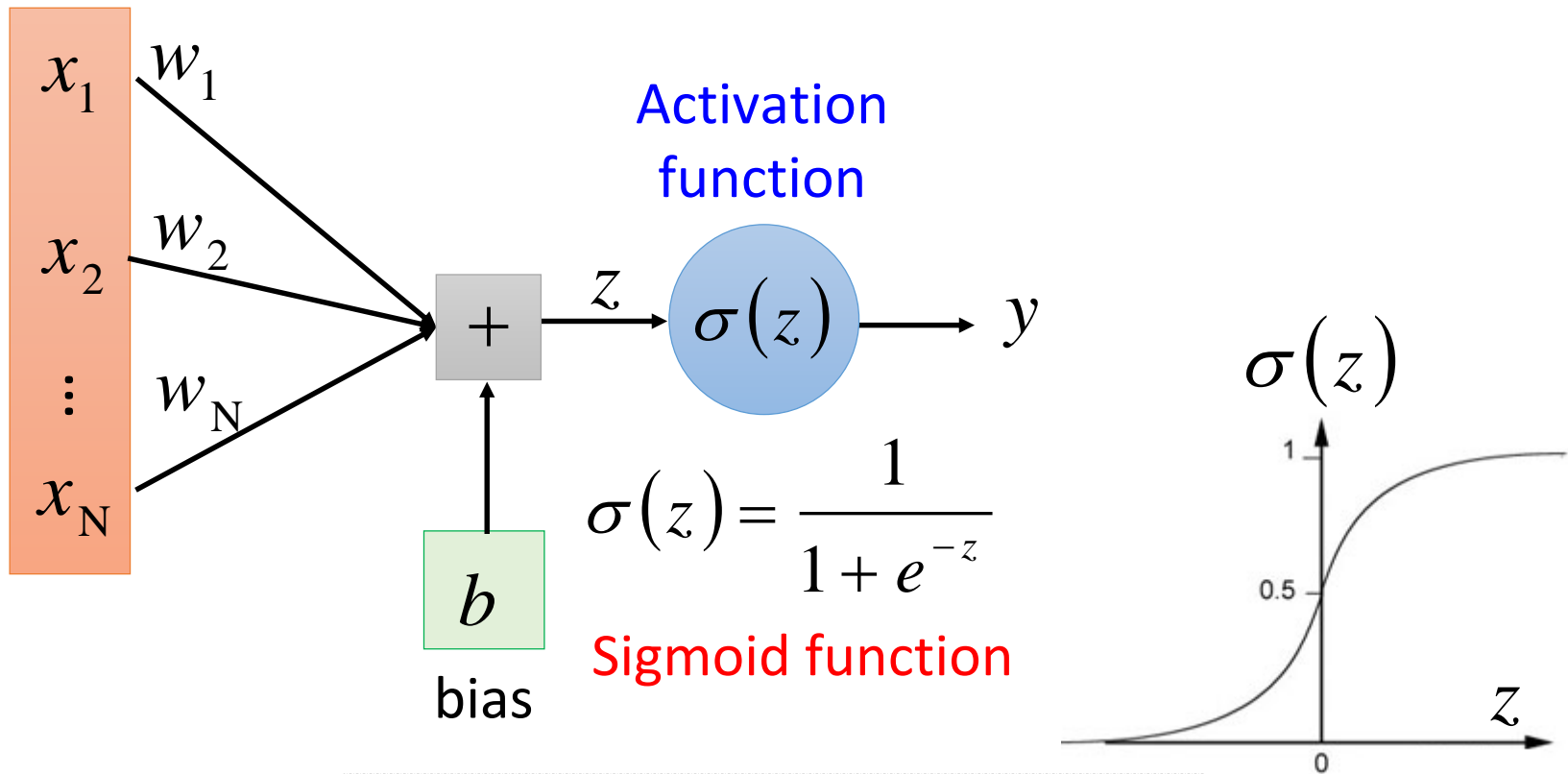


Deep learning usually refers to *neural network* based model

Inspired by Human Brain



A Single Neuron



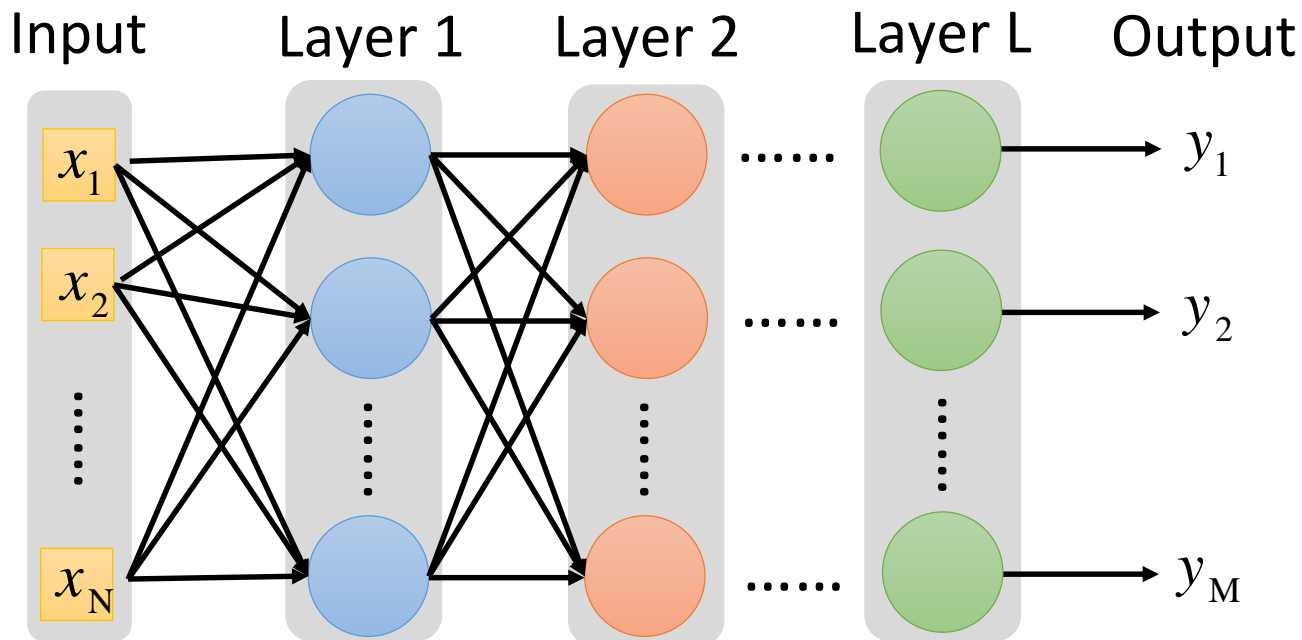
Each neuron is a very simple function

Deep Neural Network

A neural network is a complex function:

$$f : R^N \rightarrow R^M$$

Cascading the neurons to form a neural network



Each layer is a simple function in the production line

History of Deep Learning

1960s: Perceptron (single layer neural network)

1969: Perceptron has limitation

1980s: Multi-layer perceptron

1986: Backpropagation

1989: 1 hidden layer is “good enough”, why deep?

2006: RBM initialization (**breakthrough**)

2009: GPU

2010: **breakthrough in Speech Recognition** (Dahl et al., 2010)

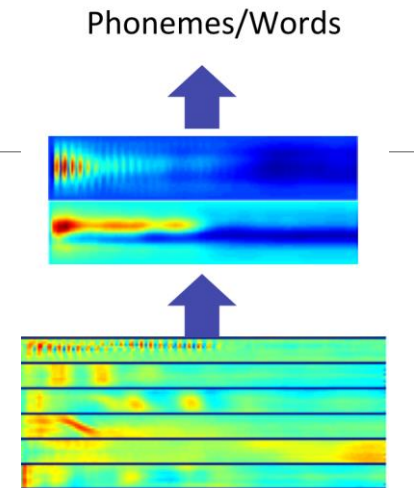
2012: **breakthrough in ImageNet** (Krizhevsky et al. 2012)

2015: **“superhuman”** results in Image and Speech Recognition

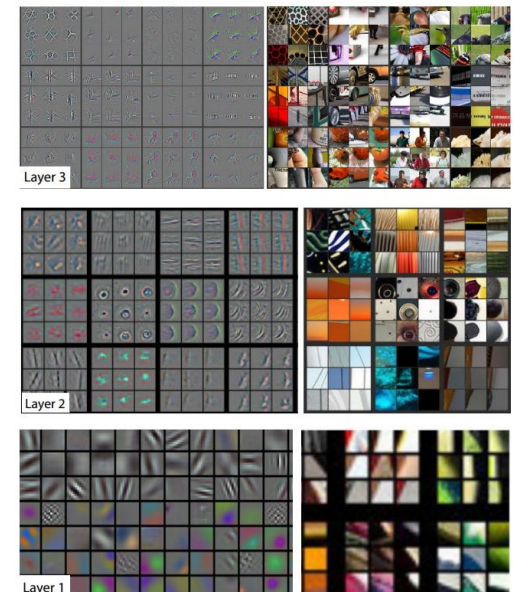
Deep Learning Breakthrough

First: Speech Recognition

Acoustic Model	WER on RT03S FSH	WER on Hub5 SWB
Traditional Features	27.4%	23.6%
Deep Learning	18.5% (-33%)	16.1% (-32%)



Second: Computer Vision



History of Deep Learning

1960s: Perceptron (single layer neural network)

1969: Perceptron has limitation

1980s: Multi-layer perceptron

1986: Backpropagation

1989: 1 hidden layer is “good enough”, why deep?

2006: RBM initialization (**breakthrough**)

2009: GPU

2010: **breakthrough in Speech Recognition** (Dahl et al., 2010)

2012: **breakthrough in ImageNet** (Krizhevsky et al. 2012)

2015: “**superhuman**” results in Image and Speech Recognition

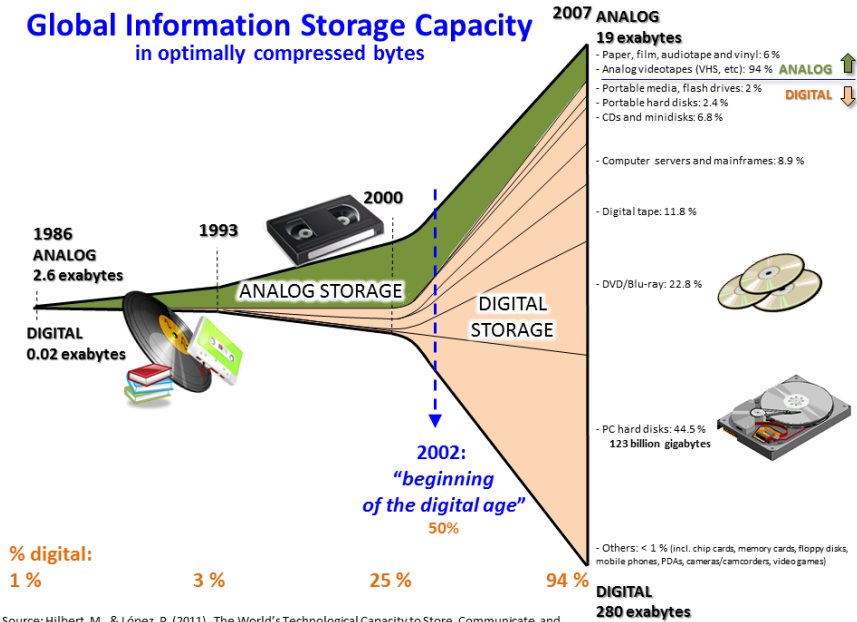
Why does deep learning show breakthrough in applications after 2010?

Reasons why Deep Learning works

Big Data

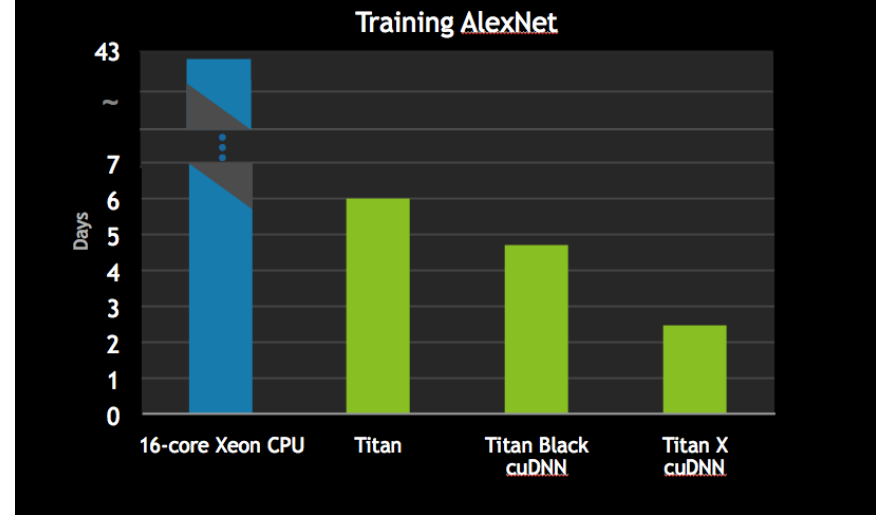
GPU

Global Information Storage Capacity
in optimally compressed bytes



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60–65. <http://www.martinhilbert.net/WorldInfoCapacity.html>

TITAN X FOR DEEP LEARNING



Why to Adopt GPU for Deep Learning?

GPU is like a brain

Human brains create *graphical imagination* for mental thinking

台灣好吃牛肉麵



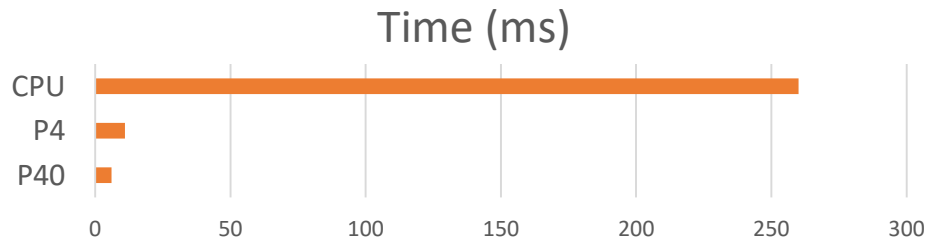
Why Speed Matters?

Training time

- Big data increases the training time
- Too long training time is not practical

Inference time

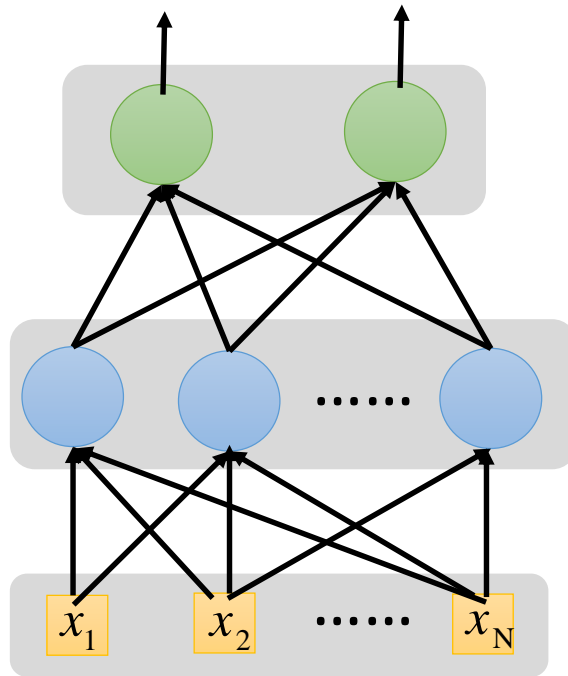
- Users are not patient to wait for the responses



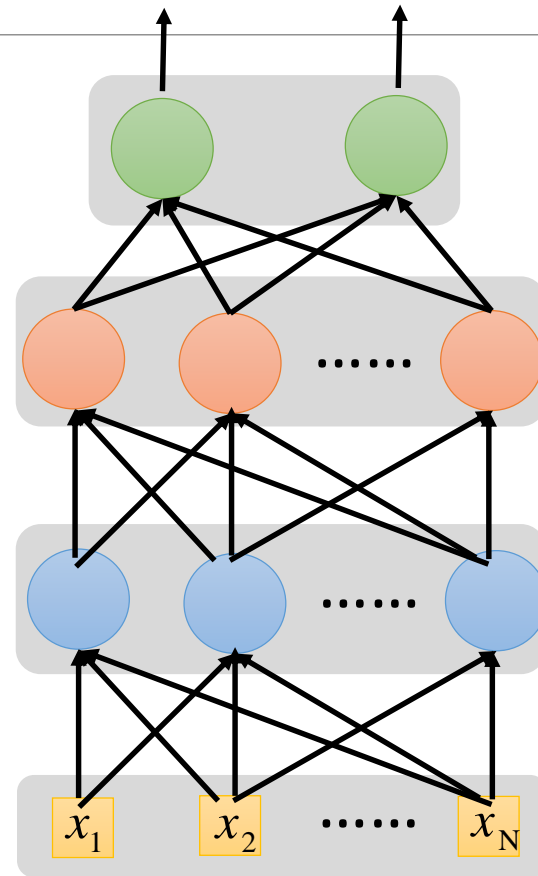
GPU enables the real-world applications using the computational power

Why Deeper is Better?

Deeper \rightarrow More parameters



Shallow



Deep

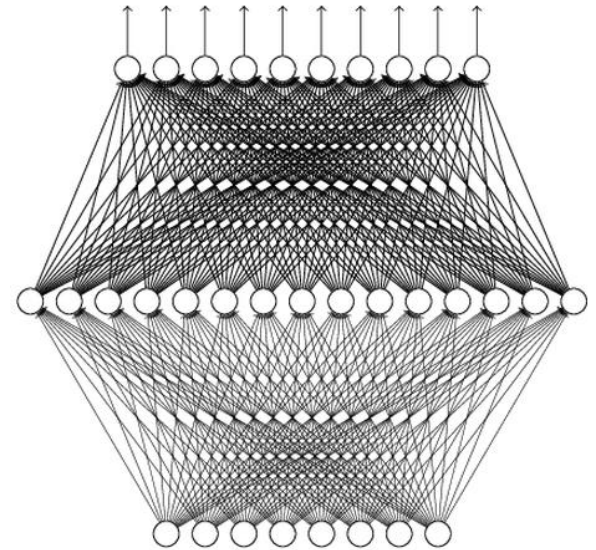
Universality Theorem

Any continuous function f

$$f : \mathbb{R}^N \rightarrow \mathbb{R}^M$$

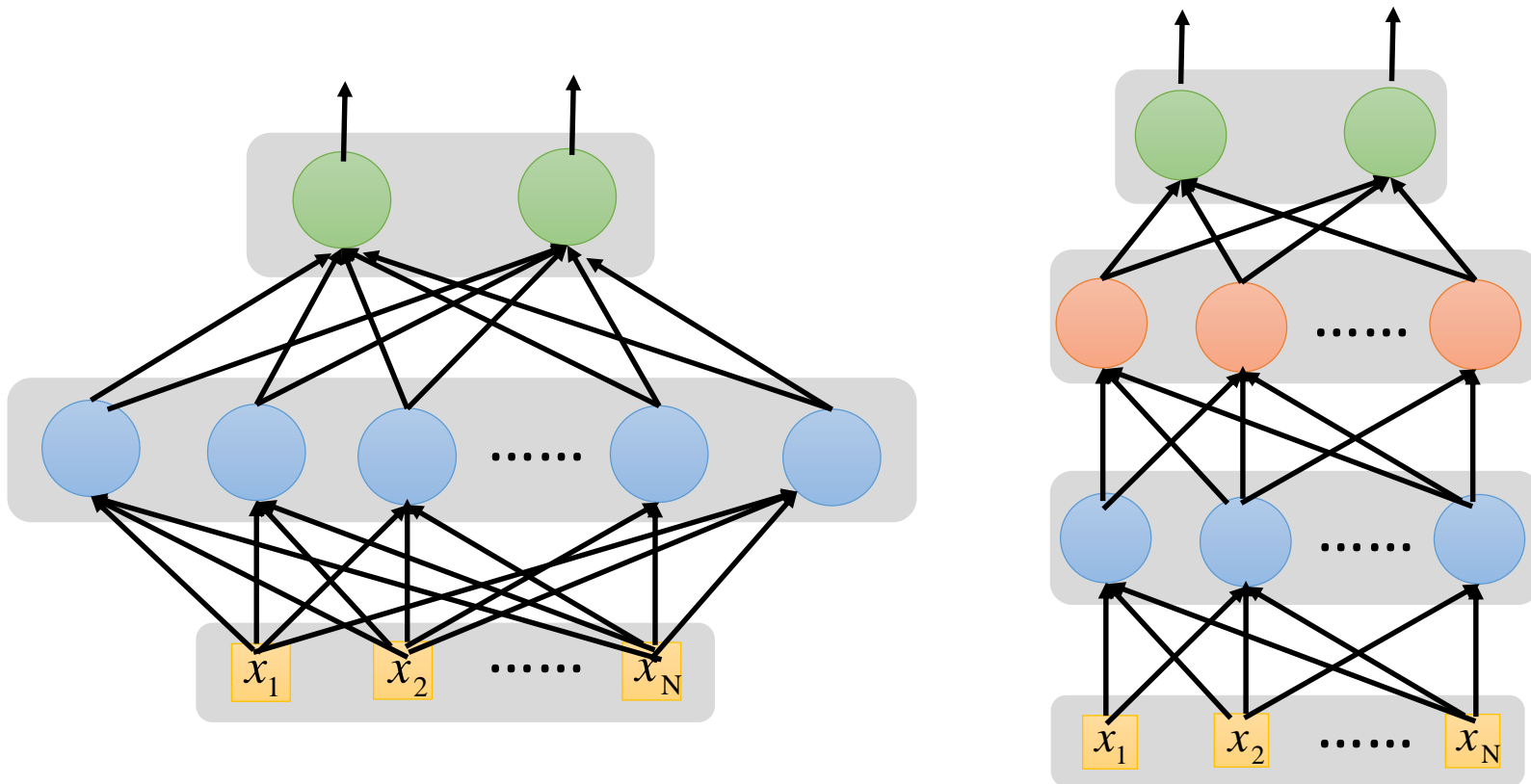
can be realized by a network with only hidden layer

Why “deep” not “fat”?

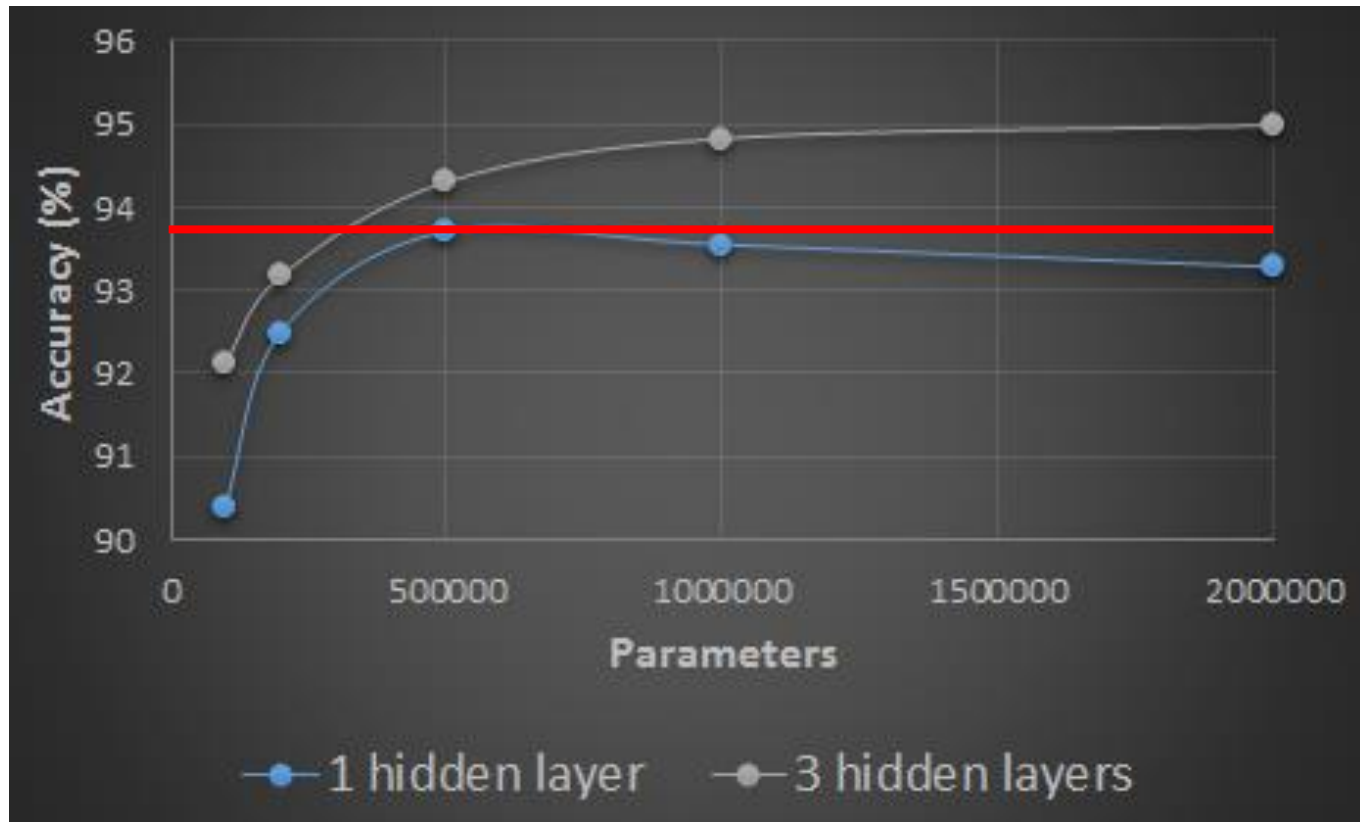


Fat + Shallow v.s. Thin + Deep

Two networks with the same number of parameters



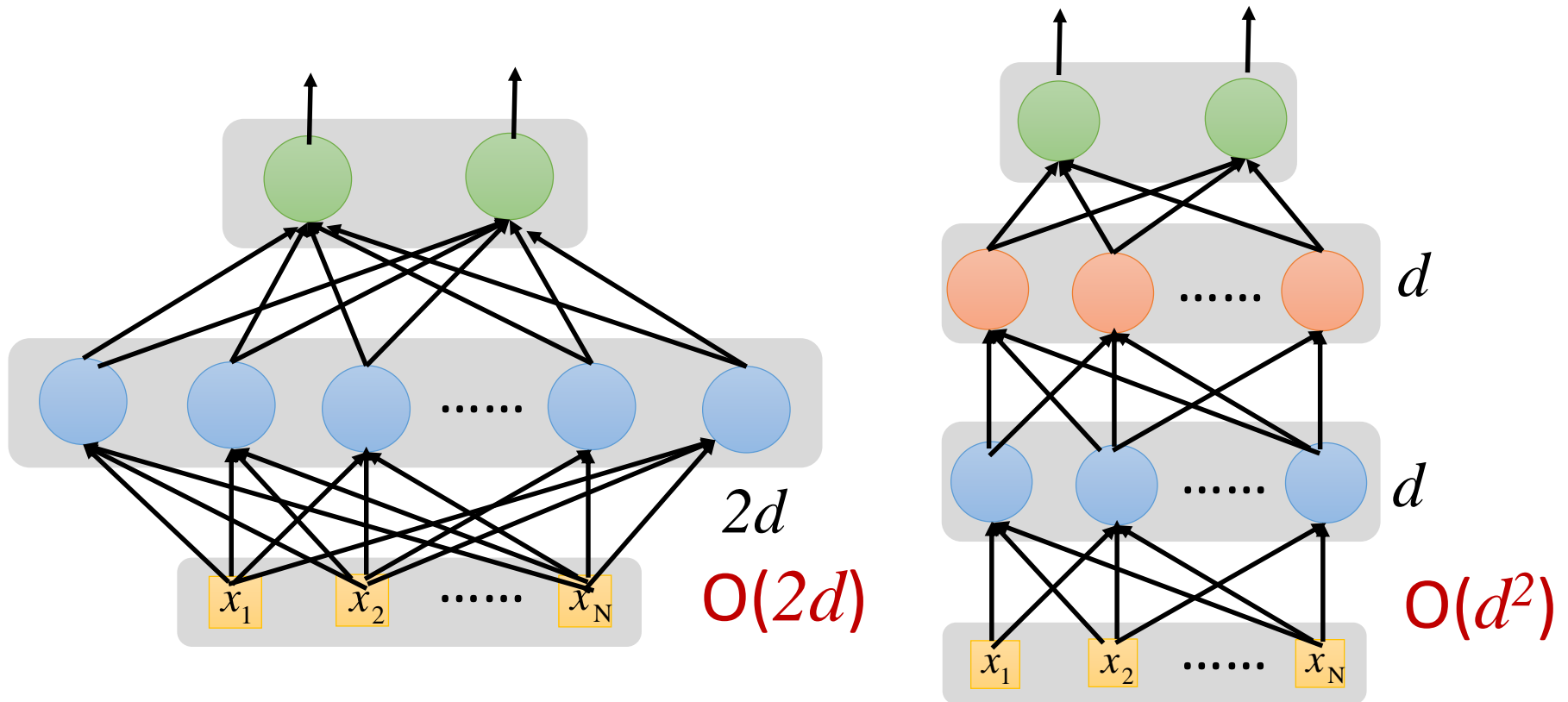
Fat + Shallow v.s. Thin + Deep Hand-Written Digit Classification



The deeper model uses less parameters to achieve the same performance

Fat + Shallow v.s. Thin + Deep

Two networks with the same number of parameters





How to Apply?

How to Frame the Learning Problem?

The learning algorithm f is to map the input domain X into the output domain Y

$$f : X \rightarrow Y$$

Input domain: word, word sequence, audio signal, click logs

Output domain: single label, sequence tags, tree structure, probability distribution

Output Domain – Classification

Sentiment Analysis

“這規格有誠意!” → +

“太爛了吧~” → -

Speech Phoneme Recognition



→ /h/

Handwritten Recognition



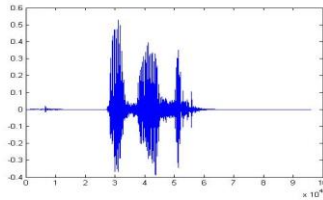
→ 2

Output Domain – Sequence Prediction

POS Tagging

“推薦我台大後門的餐廳” → 推薦/VV 我/PN 台大/NR 後門/NN
的/DEG 餐廳/NN

Speech Recognition



→ “大家好”

Machine Translation

“How are you doing today?” → “你好嗎?”

Learning tasks are decided by the output domains

Input Domain – How to Aggregate Information

Input: word sequence, image pixels, audio signal, click logs

Property: continuity, temporal, importance distribution

Example

- CNN (convolutional neural network): local connections, shared weights, pooling
 - AlexNet, VGGNet, etc.
- RNN (recurrent neural network): temporal information

Network architectures should consider the input domain properties

How to Frame the Learning Problem?

The learning algorithm f is to map the input domain X into the output domain Y

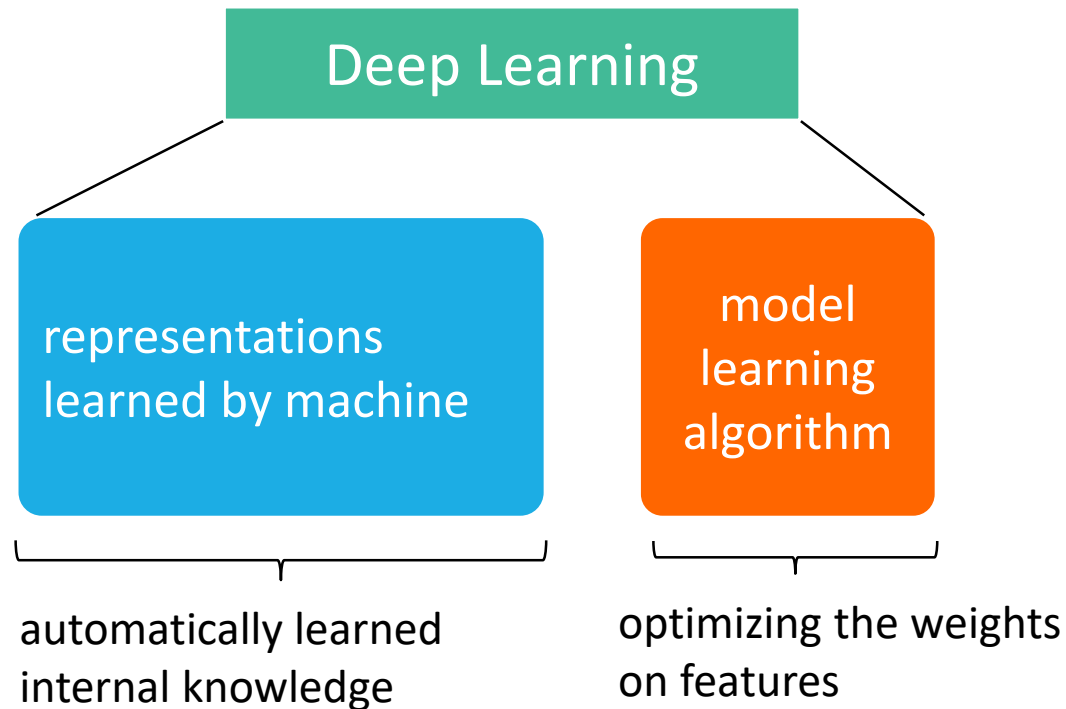
$$f : X \rightarrow Y$$

Input domain: word, word sequence, audio signal, click logs

Output domain: single label, sequence tags, tree structure, probability distribution

Network design should leverage input and output domain properties

“Applied” Deep Learning



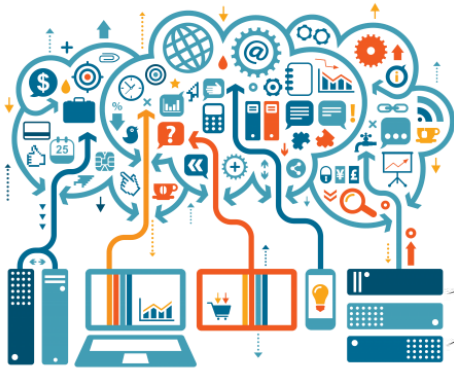
How to frame a task into a learning problem and design the corresponding model

Core Factors for Applied Deep Learning

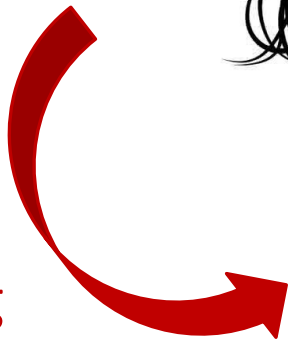
1. Data: big data
2. Hardware: GPU computing
3. **Talent**: design algorithms to allow networks to work for the specific problems



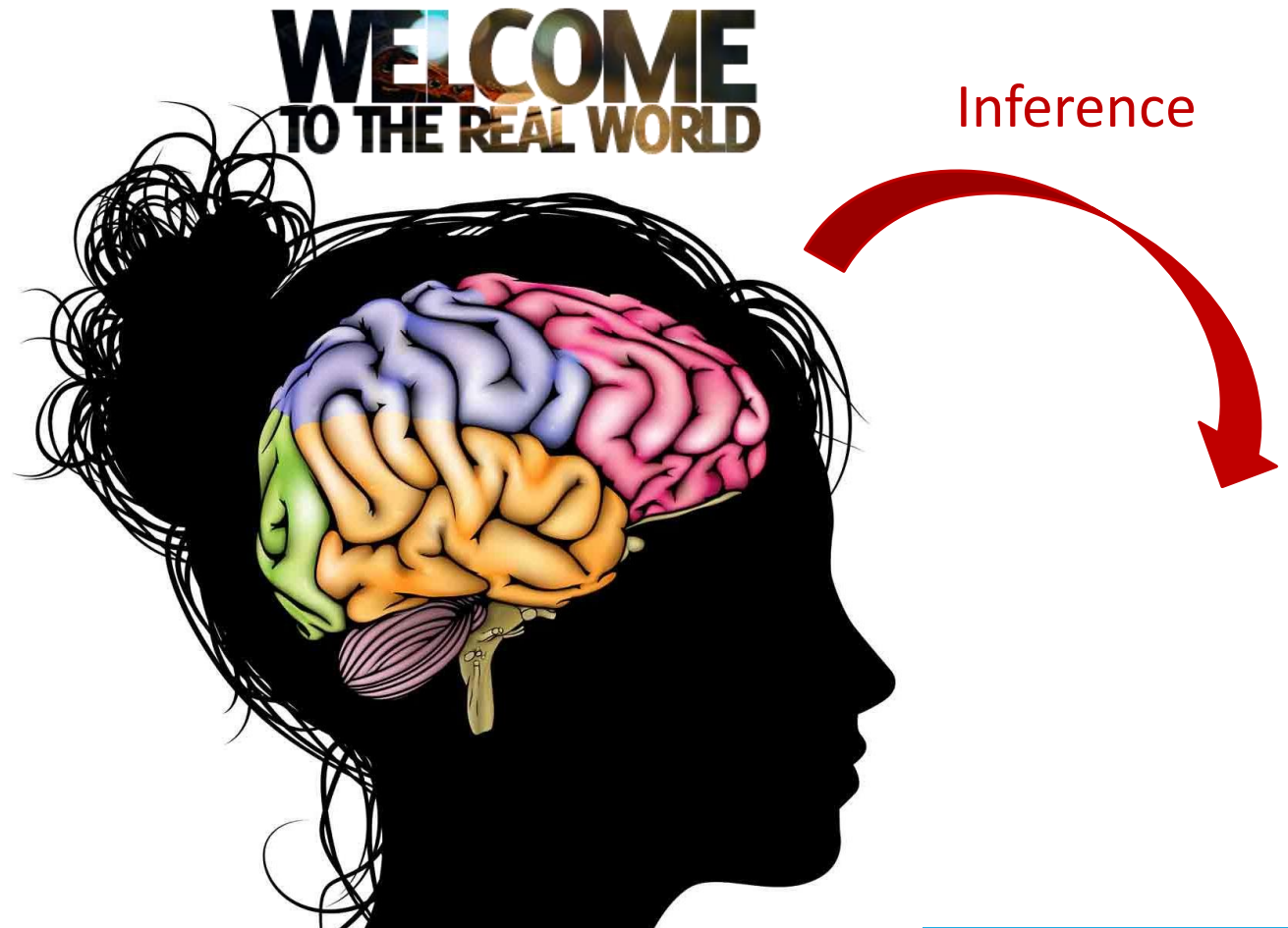
Concluding Remarks



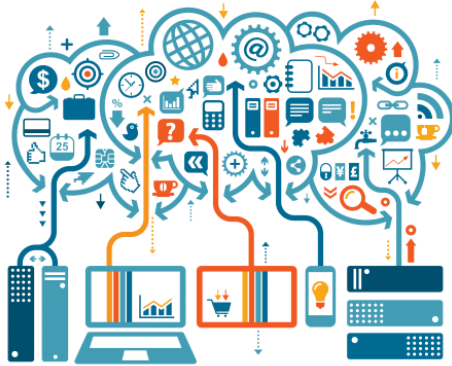
Training



Concluding Remarks



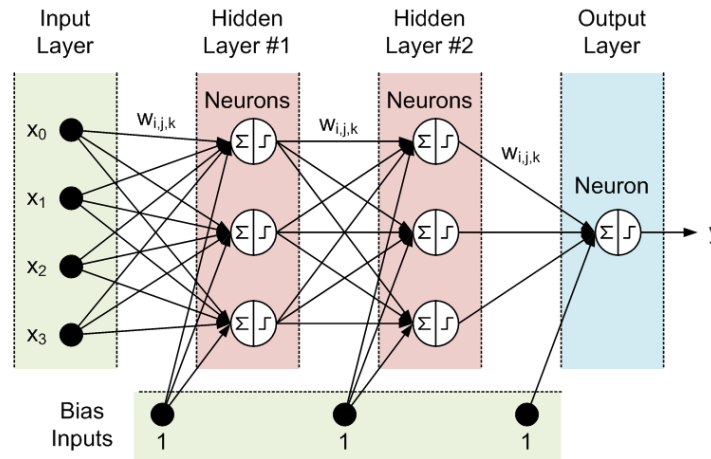
Concluding Remarks



**WELCOME
TO THE REAL WORLD**

Inference

Training



Main focus: how to apply deep learning to the real-world problems

Reference

Reading Materials

- Academic papers will be put in the website

Deep Learning

- Goodfellow, Bengio, and Courville, “Deep Learning,” 2016.
<http://www.deeplearningbook.org>
- Michael Nielsen, “Neural Networks and Deep Learning”
<http://neuralnetworksanddeeplearning.com>