Artificial Intelligence for Operation Research (AI4OR): Basics and Overview

Academy of Mathematics and Systems Sciences, CAS

May 4, 2023

Outline

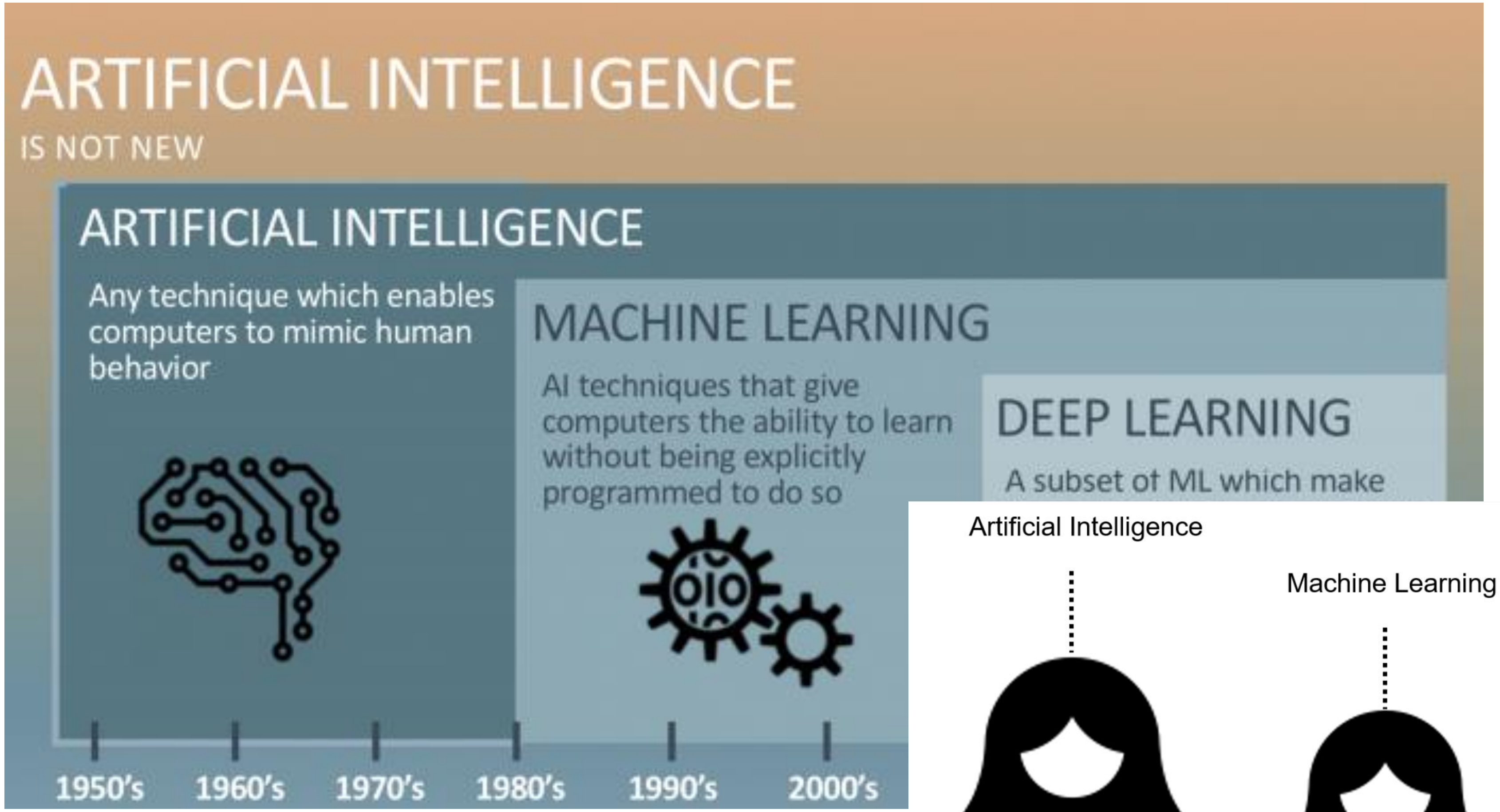
- Relationship of AI, ML, Deep Learning, and Neural Networks

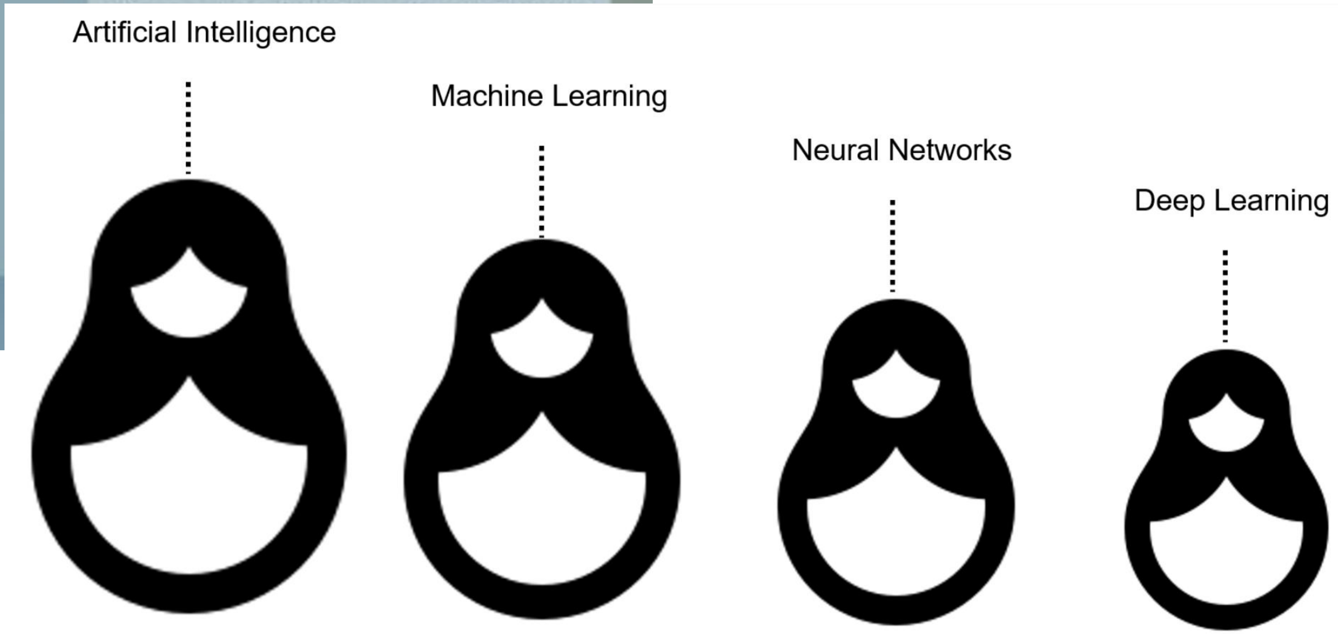
- Deep Learning
    - Neural Network (Architecture)
    - Total Cost (Loss)
    - Optimization
    - AI for Science

- Operation Research/Combinatorial Optimization
    - AI Meet Combinatorial Optimization
    - Learning Methods

- Agenda

AI is Not New





Machine Learning

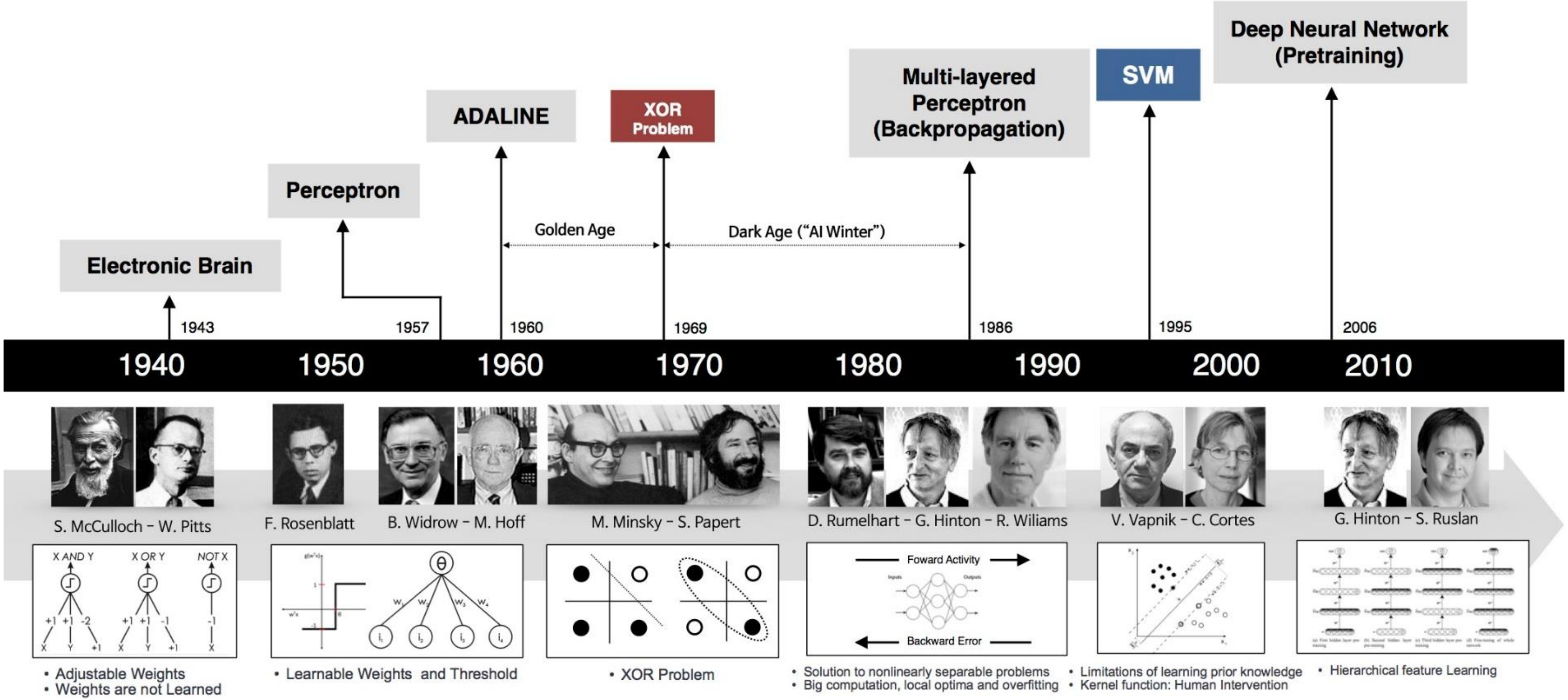
-Supervised Learning

-Unsupervised …

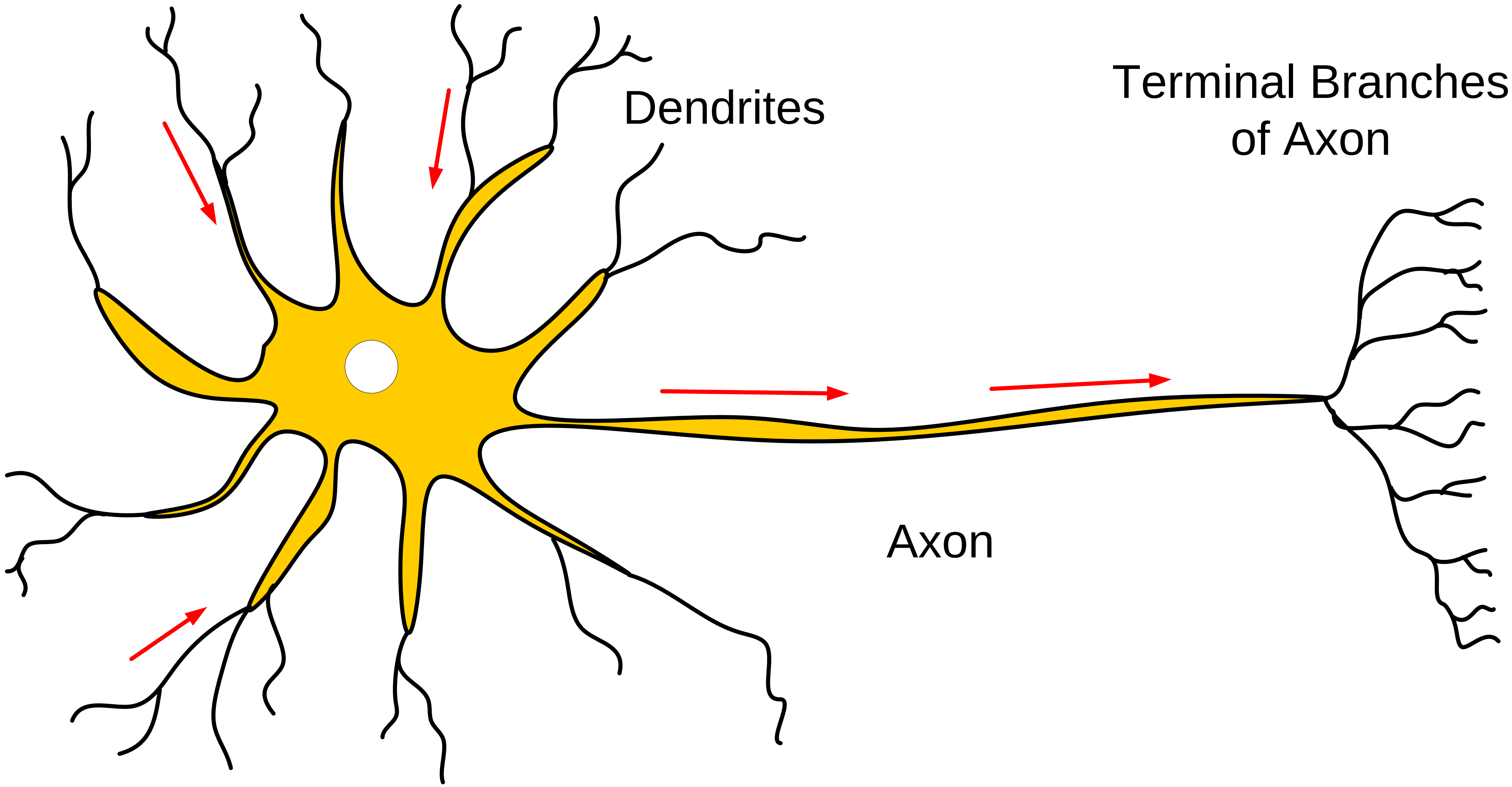
-Semi-Supervised …

-Reinforcement …

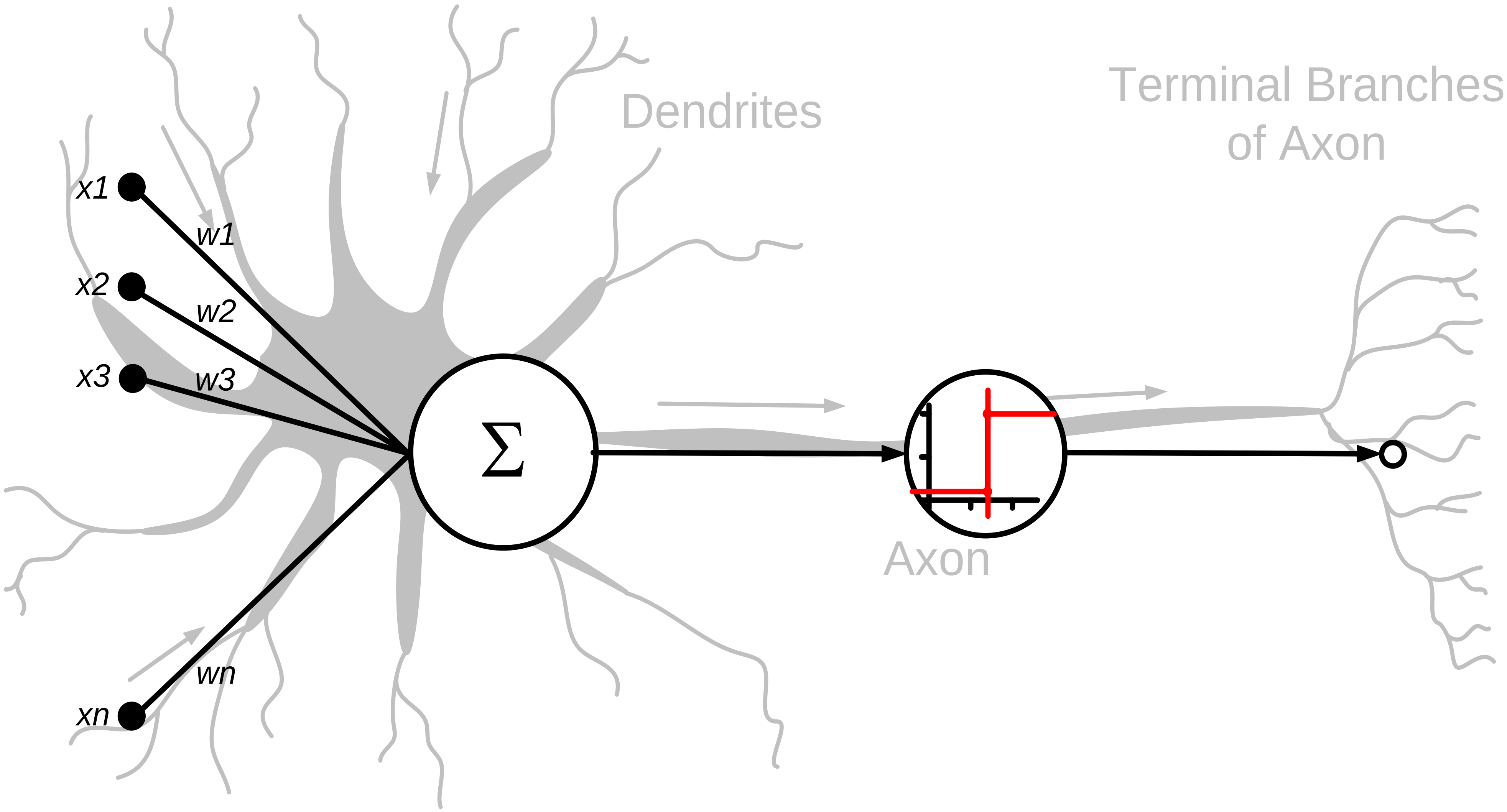
History of Neural Networks



What are Neural Networks? Biological Neurons

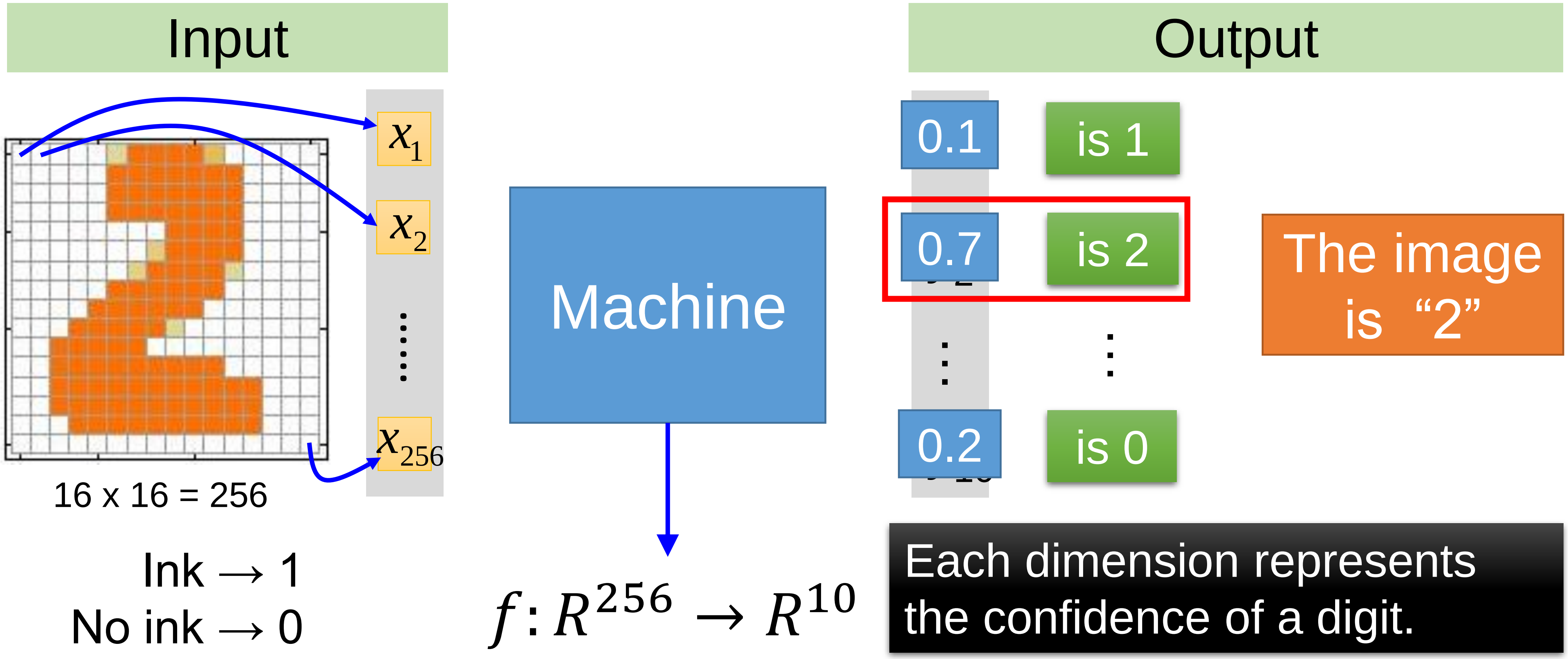


Artificial Neural Networks (ANN)



Slide credit : Andrew L. Nelson

Example: Handwriting Digit Recognition



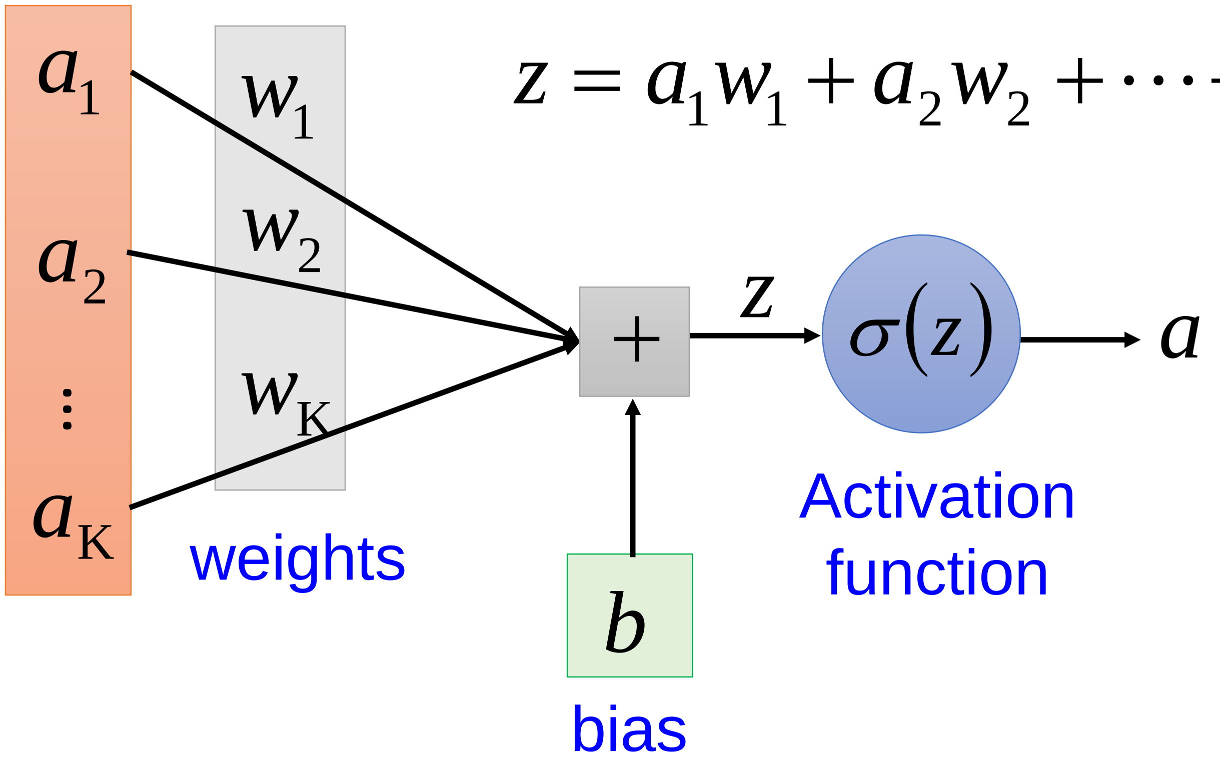
In deep learning, the function f is represented by neural network

The same for even more complex tasks.

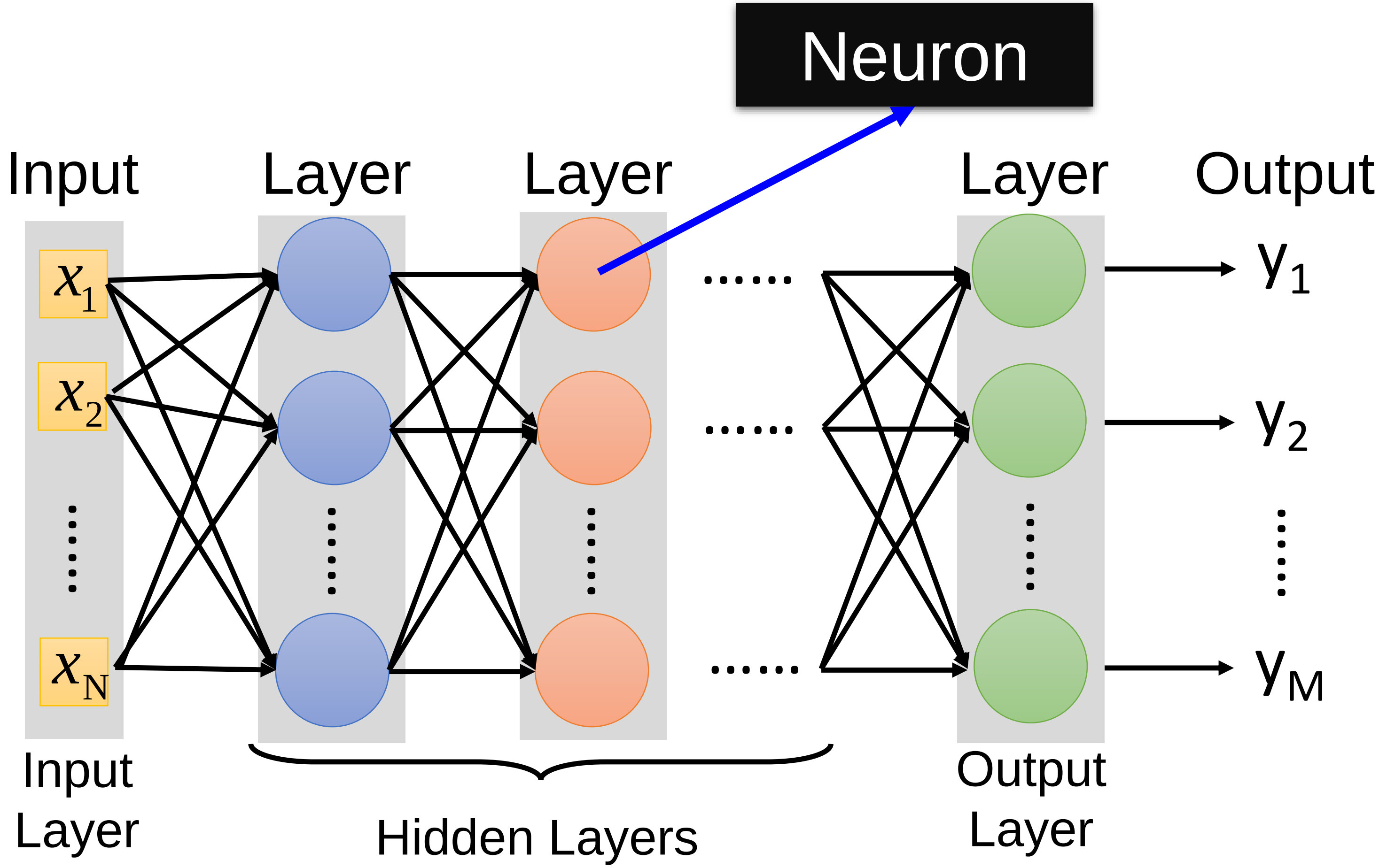
Element of Neural Network

<u>Neuron</u> $f: R^K \to R$

$$z = a_1 w_1 + a_2 w_2 + \cdots + a_K w_K + b$$



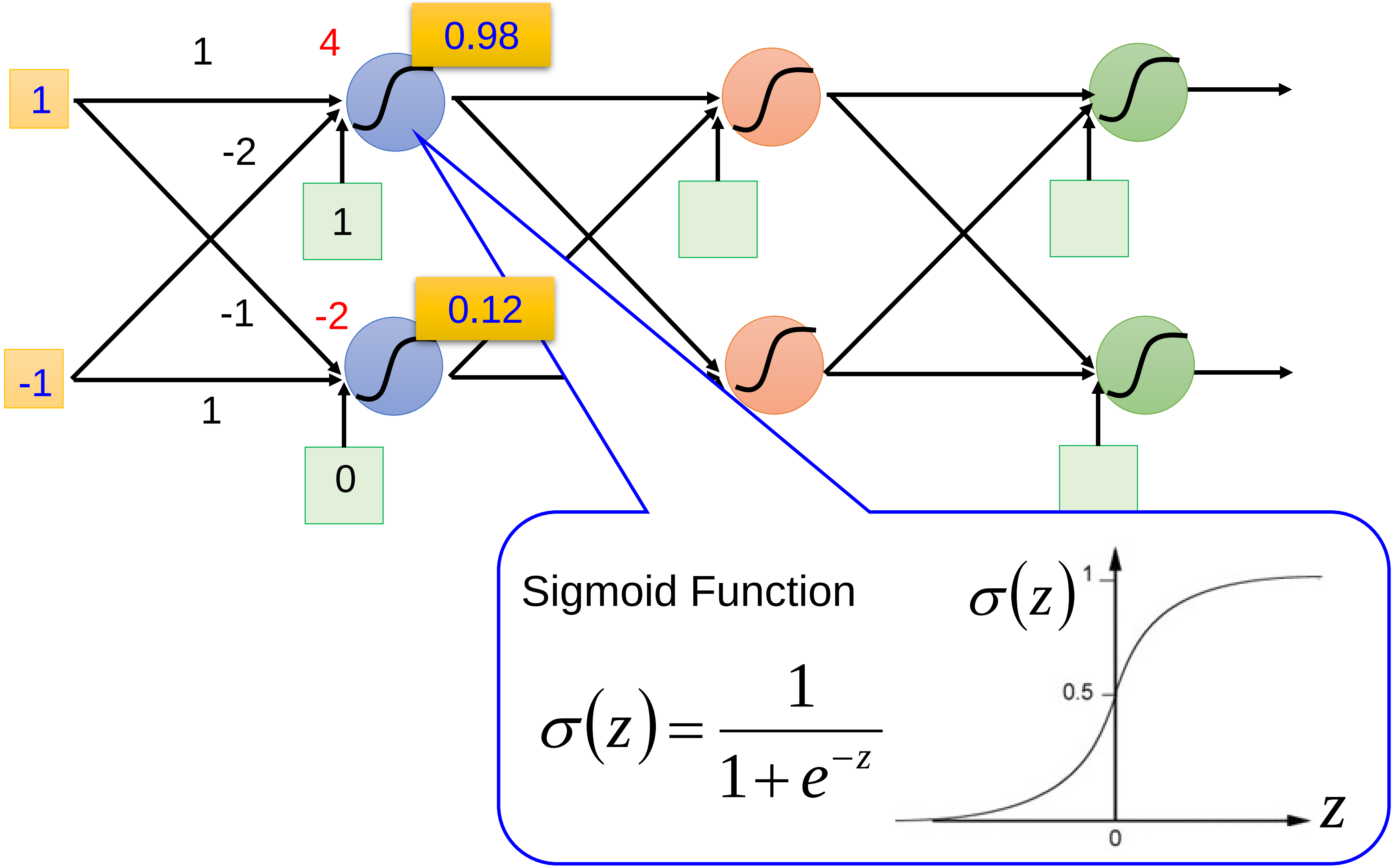
Neural Network



Deep means many hidden layers

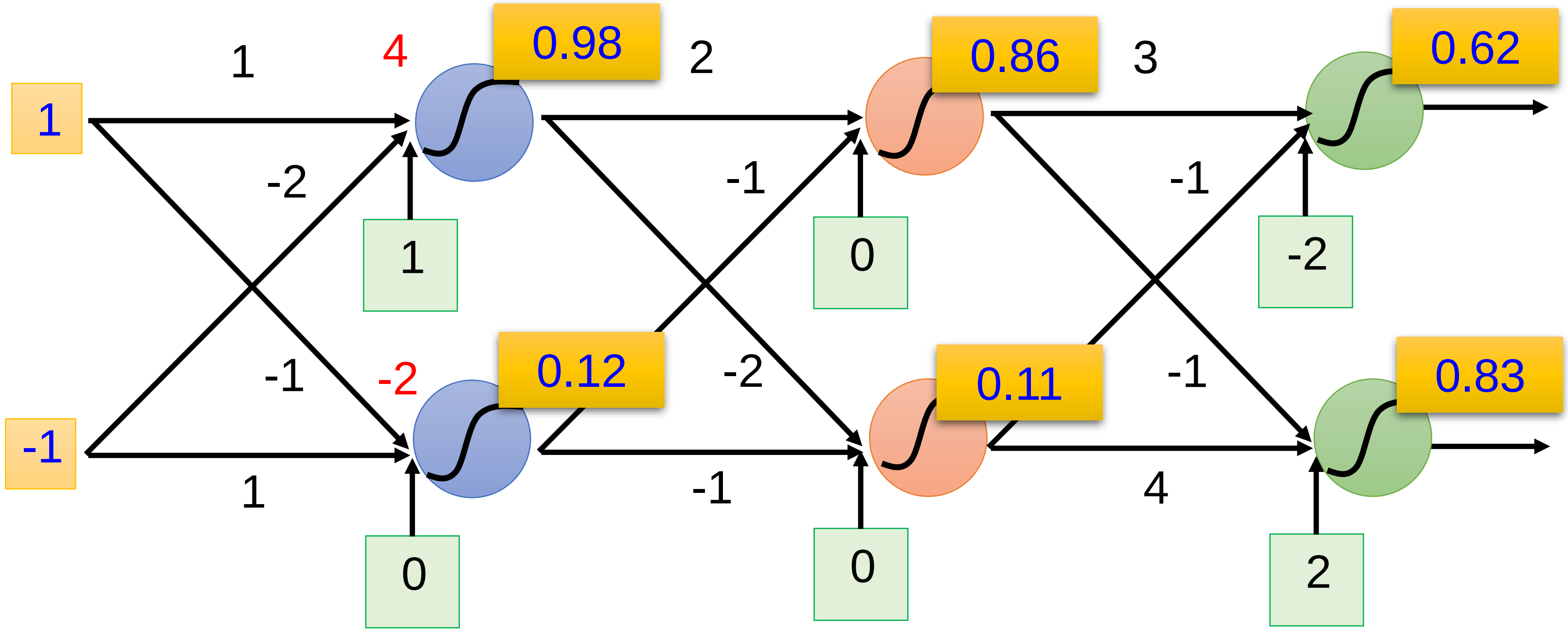
Fully Connected Feedforward Network

Example of Neural Network

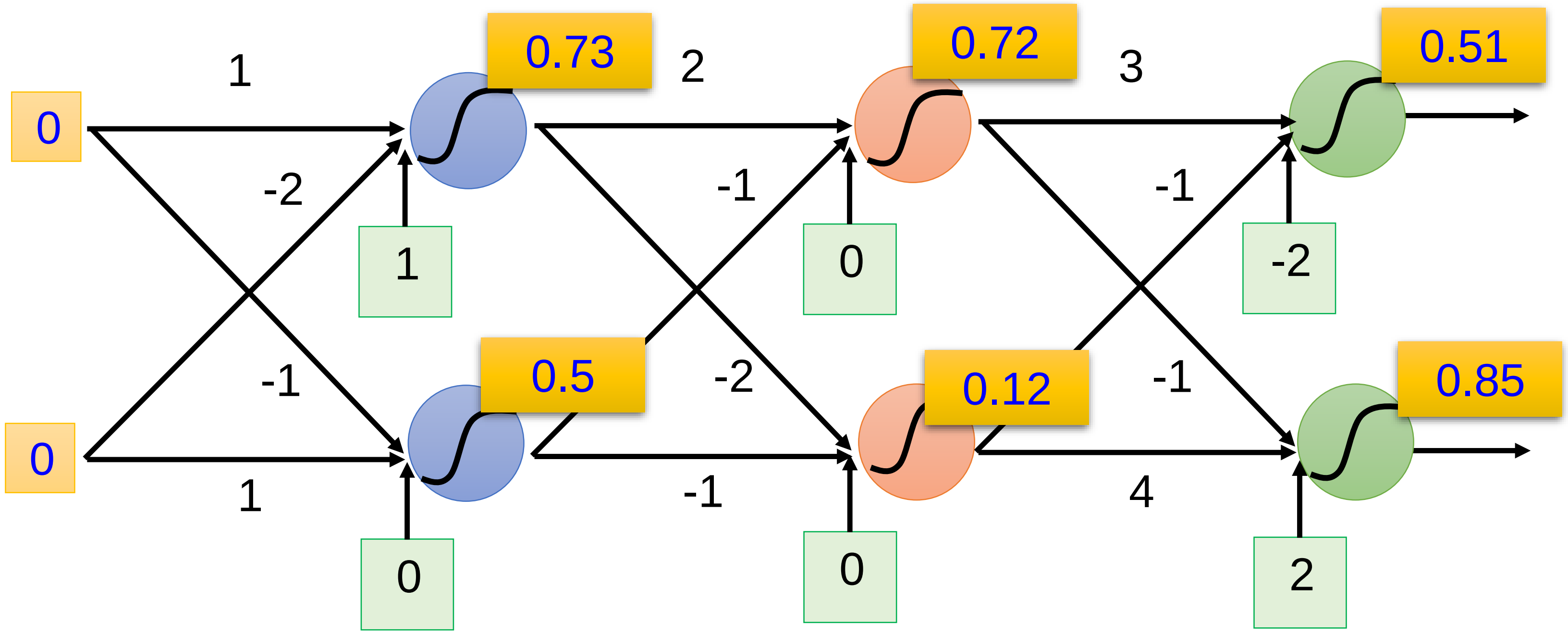


$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Example of Neural Network



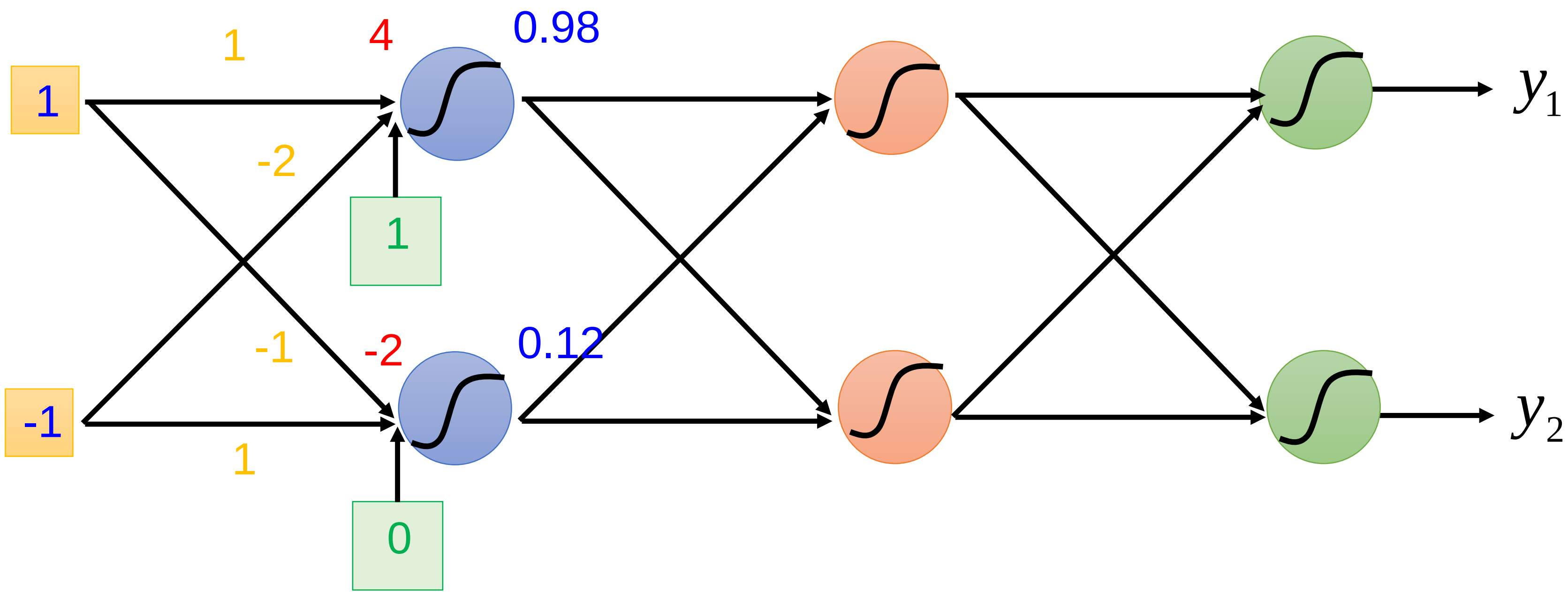
Example of Neural Network



$$f: R^2 \to R^2 \qquad f\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}\right) = \begin{bmatrix} 0.62 \\ 0.83 \end{bmatrix} \quad f\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0.51 \\ 0.85 \end{bmatrix}$$

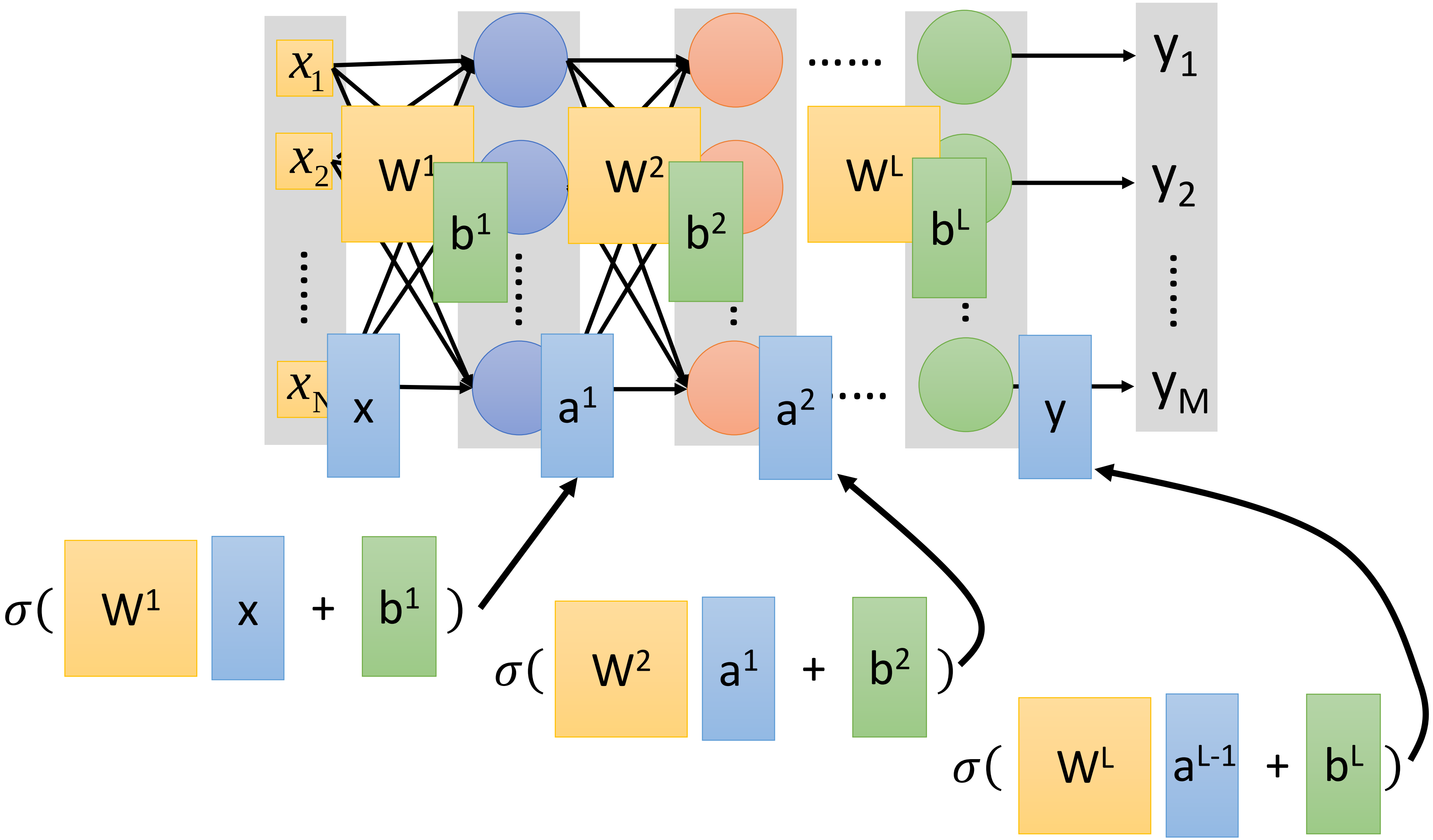
Different parameters define different function

Matrix Operation

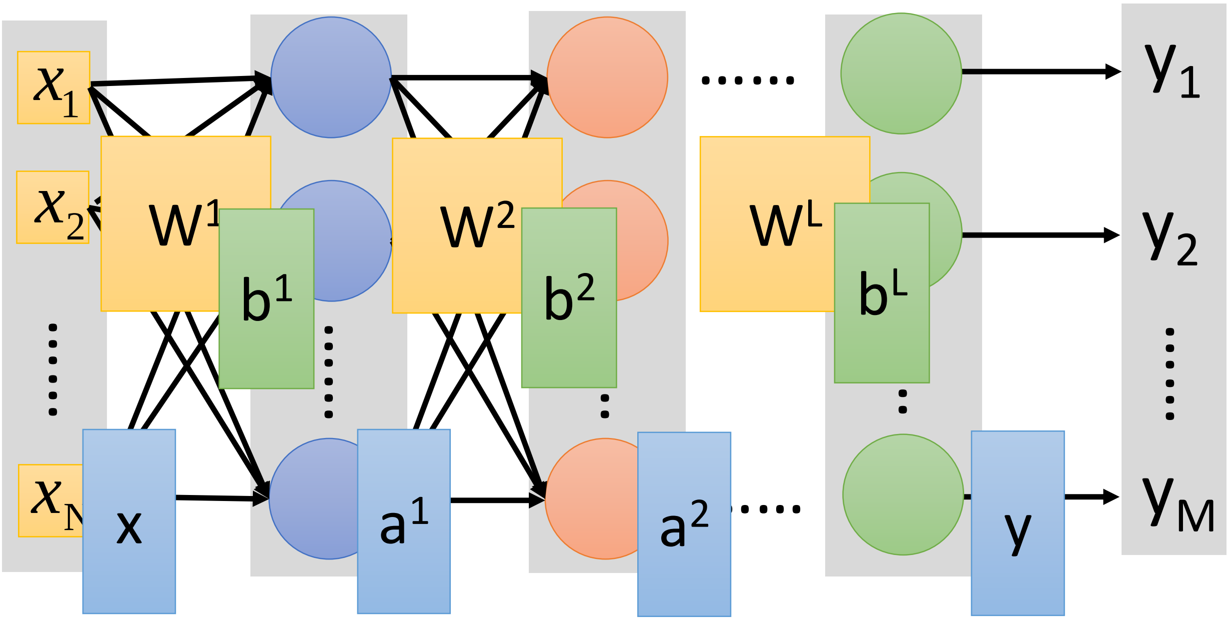


$$\sigma\left(\underbrace{\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\begin{bmatrix} 4 \\ -2 \end{bmatrix}} \right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

Neural Network



Neural Network



$$y = f(x)$$

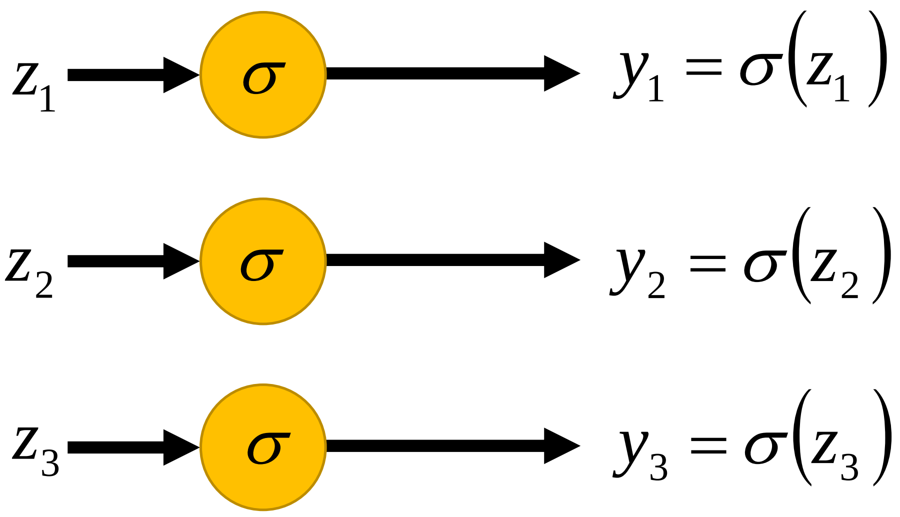
Using parallel computing techniques to speed up matrix operation

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

Softmax

- Softmax layer as the output layer

Ordinary Layer



In general, the output of network can be any value.

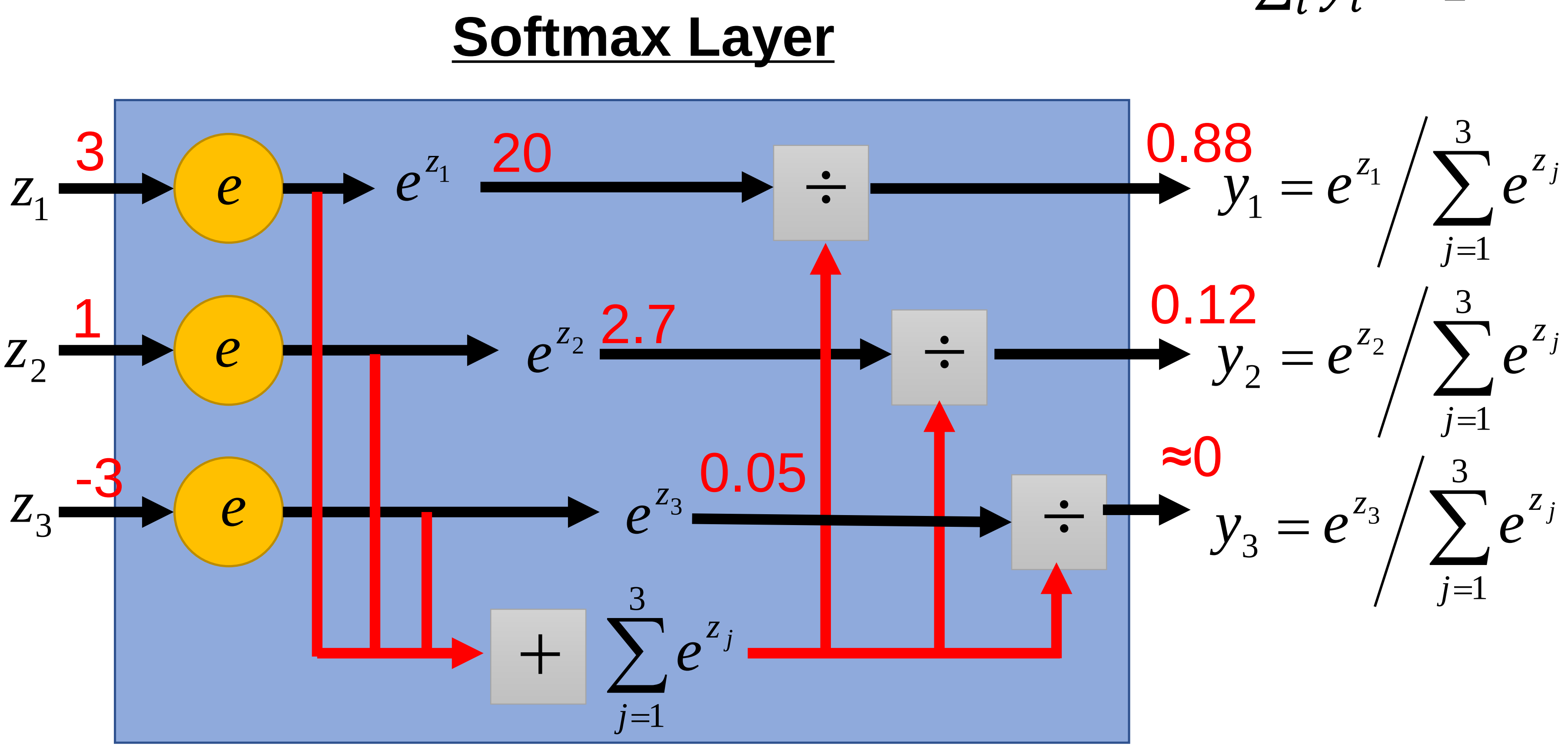
May not be easy to interpret!

Softmax

- Softmax layer as the output layer

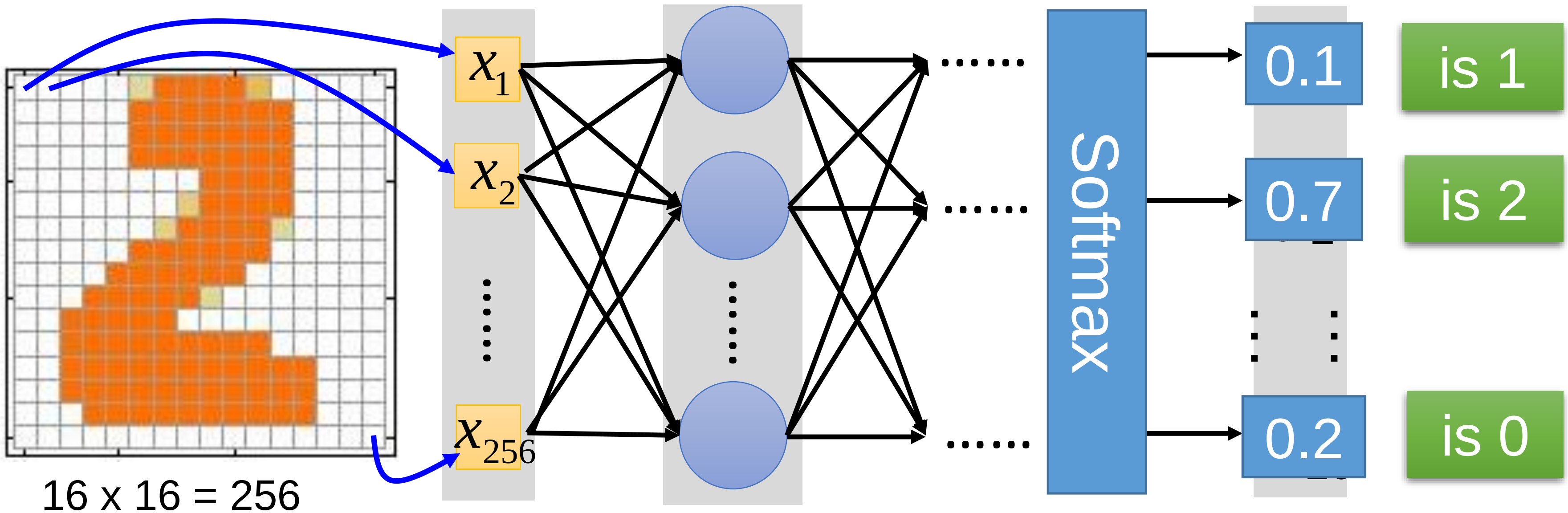
Probability:

- $1 > y_i > 0$
- $\sum_i y_i = 1$



How to set network parameters

$$\theta = \{W^1, b^1, W^2, b^2, \cdots W^L, b^L\}$$

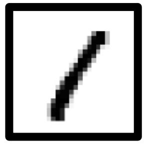


16 x 16 = 256

Ink → 1

No ink → 0

Set the network parameters θ such that ……

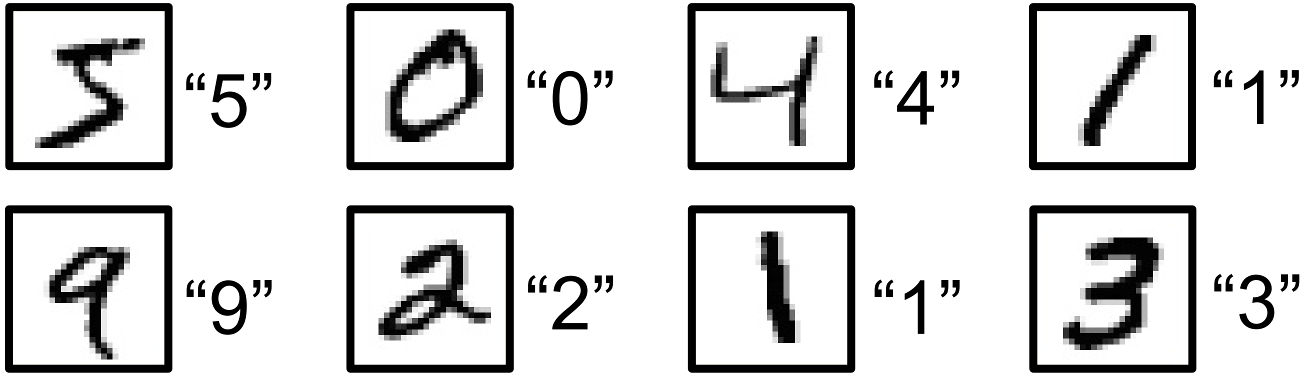
Input:  y_1 has the maximum value

Input:  y_2 has the maximum value

How to let the neural network achieve this

Training Data

- Preparing training data: images and their labels



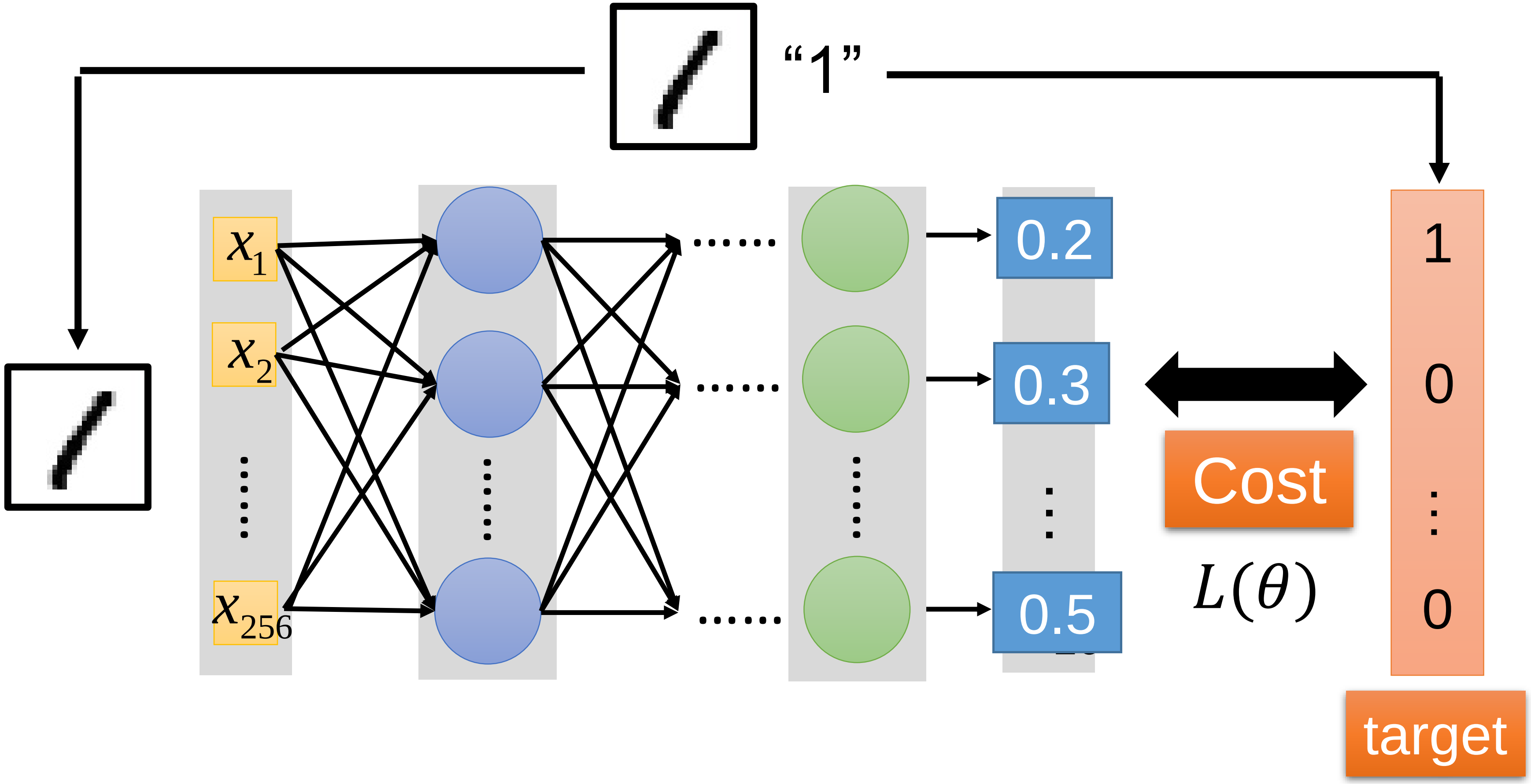
Using the training data to find the network parameters.

- **MNIST**: Training set of 60,000 examples vs Testing set of 10,000 examples.
- It is a subset of a larger set available from NIST.
- The digits have been size-normalized and centered in a fixed-size image.
- It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data.

http://yann.lecun.com/exdb/mnist/

Cost

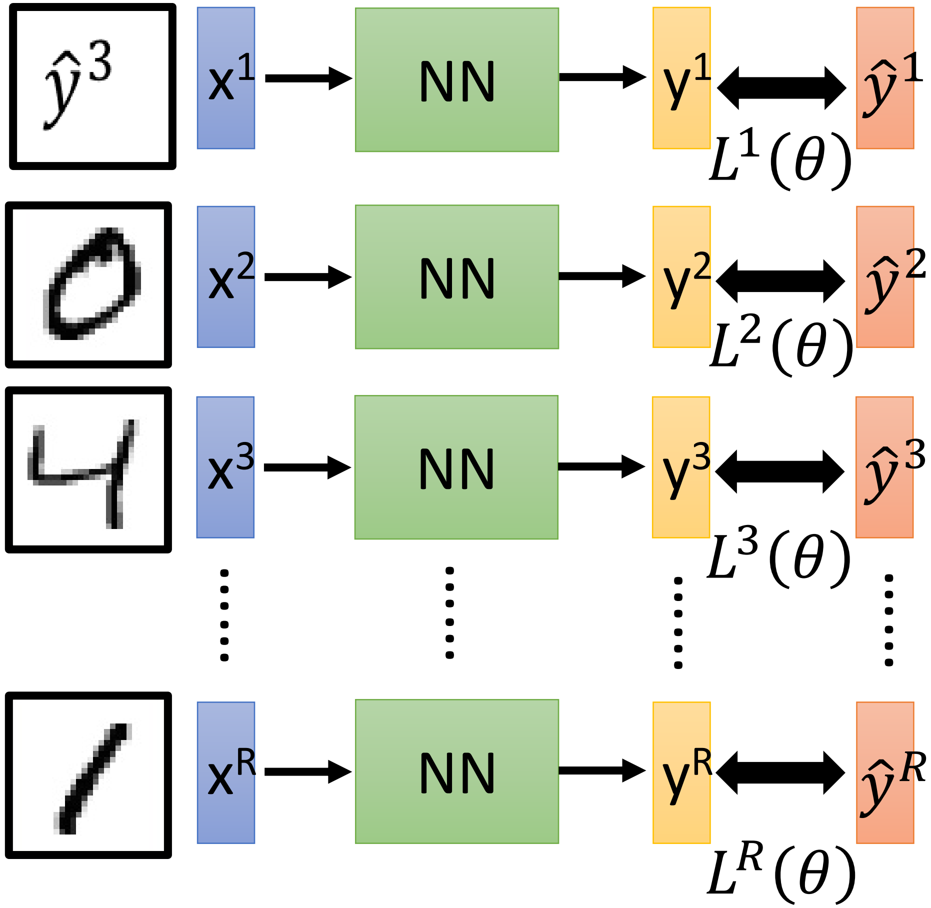
Given a set of network parameters θ, each example has a cost value.



Cost can be Euclidean distance or cross-entropy of the output and target

Total Cost

For all training data …



Total Cost:

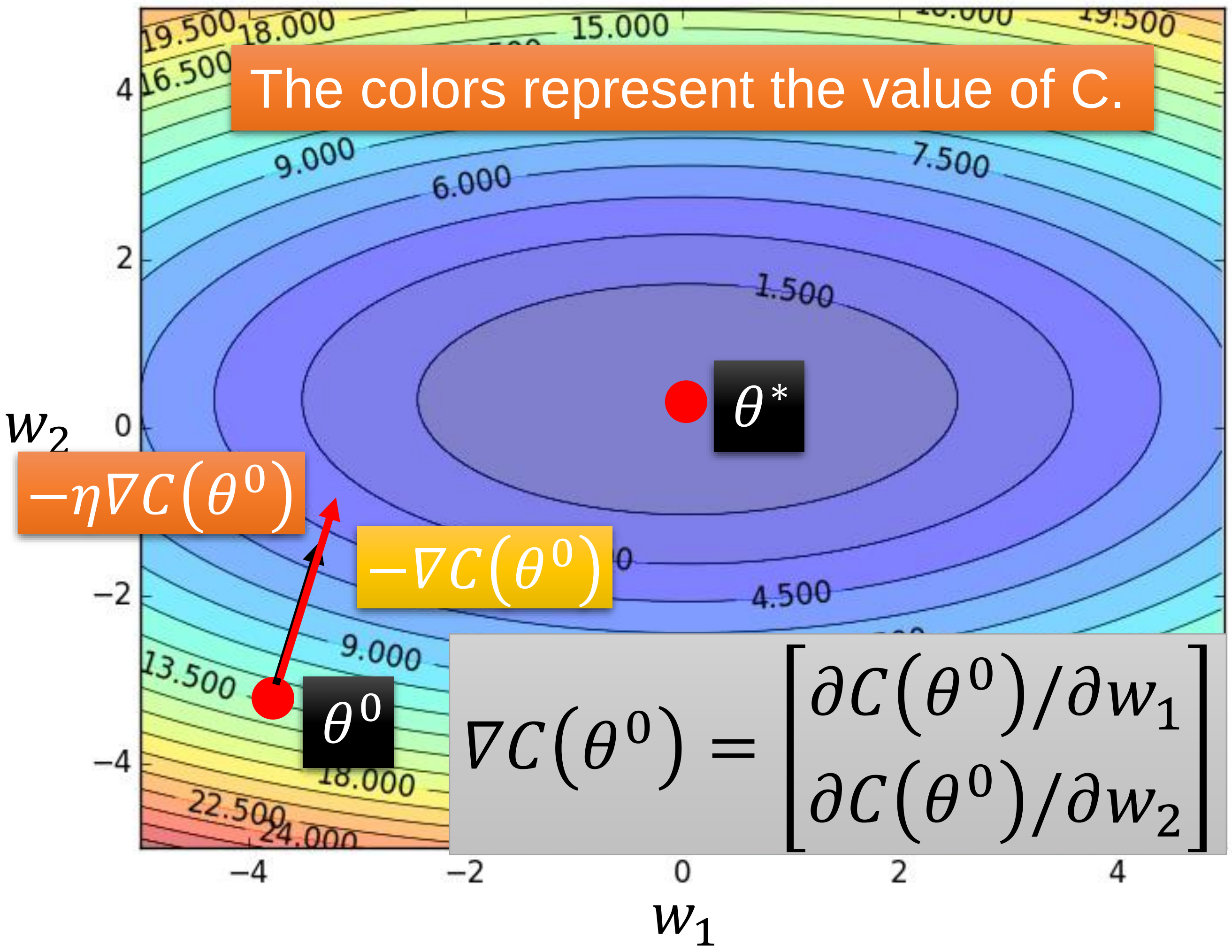
$$C(\theta) = \sum_{r=1}^{R} L^r(\theta)$$

How bad the network parameters θ is on this task?

Find the network parameters θ^* that minimize this value

Gradient Descent

Error Surface



$$\nabla C(\theta^0) = \begin{bmatrix} \partial C(\theta^0)/\partial w_1 \\ \partial C(\theta^0)/\partial w_2 \end{bmatrix}$$

Assume there are only two parameters w_1 and w_2 in a network.

$$\theta = \{w_1, w_2\}$$

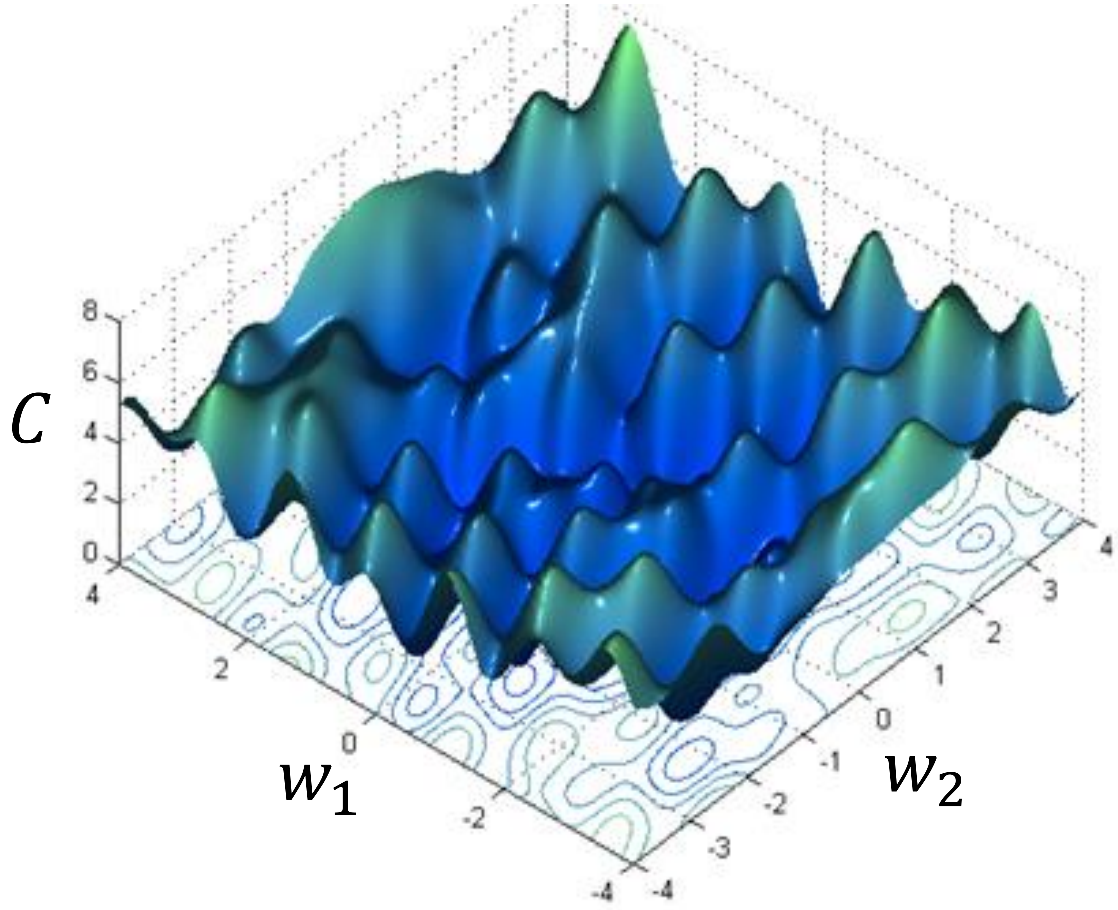
Randomly pick a starting point θ^0

Compute the negative gradient at θ^0 $\Rightarrow$ $-\nabla C(\theta^0)$

Times the learning rate η $\Rightarrow$ $-\eta \nabla C(\theta^0)$

Local Minima

- Gradient descent never guarantee global minima

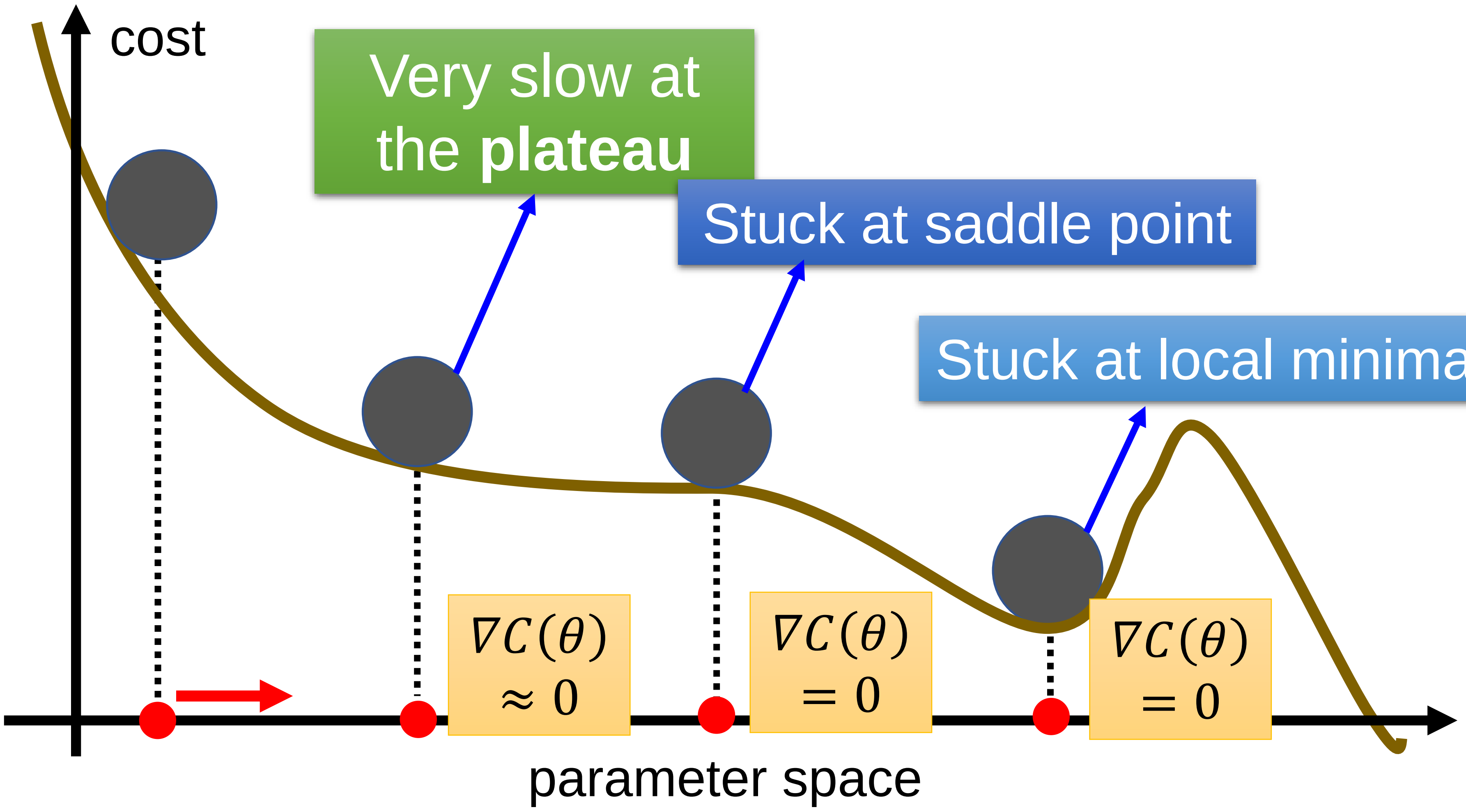


Different initial point θ^0

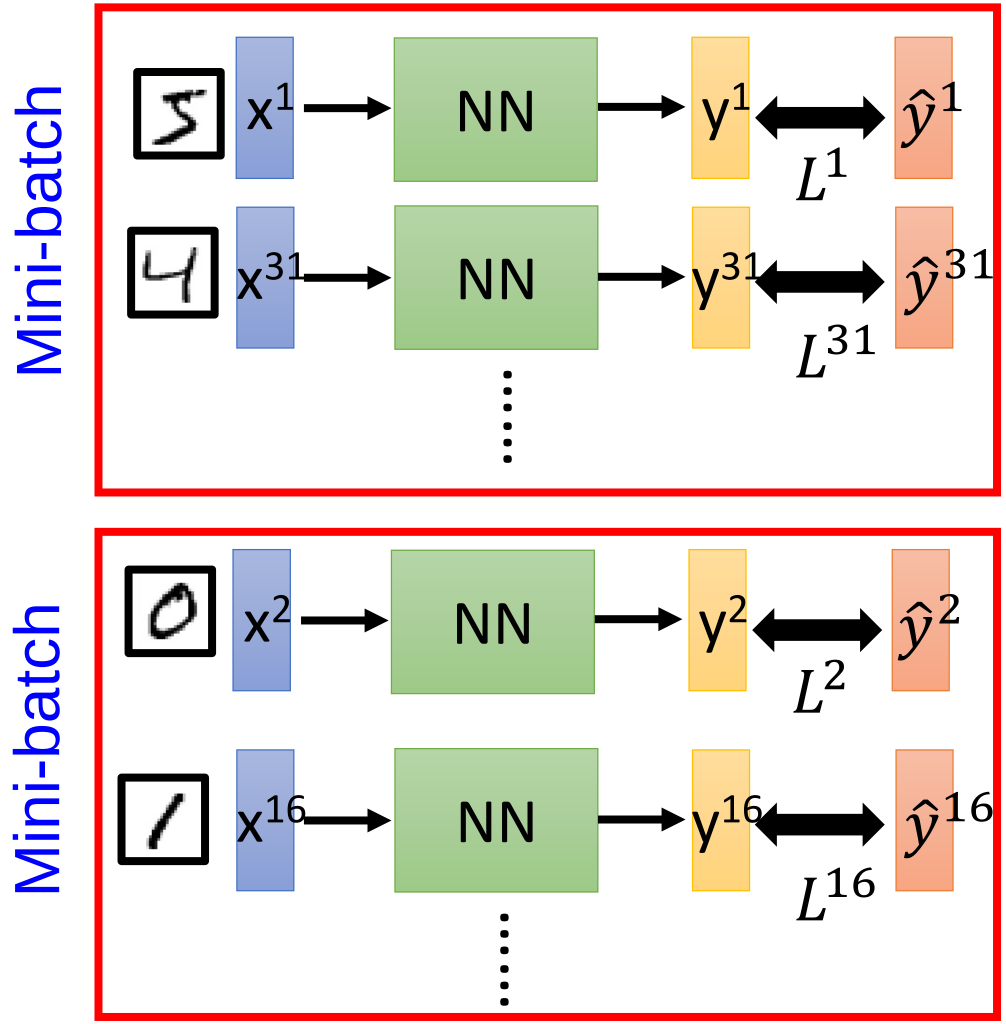
Reach different minima, so different results

Who is Afraid of Non-Convex Loss Functions?
http://videolectures.net/eml07_lecun_wia/

Besides local minima ……



Mini-batch



- Randomly initialize θ^0
- Pick the 1st batch

  $C = L^1 + L^{31} + \cdots$

  $\theta^1 \leftarrow \theta^0 - \eta \nabla C(\theta^0)$
- Pick the 2nd batch

  $C = L^2 + L^{16} + \cdots$

  $\theta^2 \leftarrow \theta^1 - \eta \nabla C(\theta^1)$

  ⋮
- Until all mini-batches have been picked

one epoch

Repeat this process

C is different each time when we update parameters!

Backpropagation

- A network can have millions of parameters.
  - Backpropagation is the way to compute the gradients efficiently (not today)
  - http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/DNN%20backprop.ecm.mp4/index.html

- Many toolkits can compute the gradients automatically

Summary

Three Steps for Deep Learning

- Define a set of functions (architecture)
- Goodness of function (loss or cost)
- Pick the best function (optimization)

When can we use deep learning to solve a problem?

- First, you want to find a function
- Second, you have lots of function input/output as training data

Diverse Architectures

- Convolutional Neural Network (CNN)
- Residual Neural Network (ResNet)
- Recurrent Neural Network (RNN)
- Transformer
- Graph Neural Network (GNN)

……

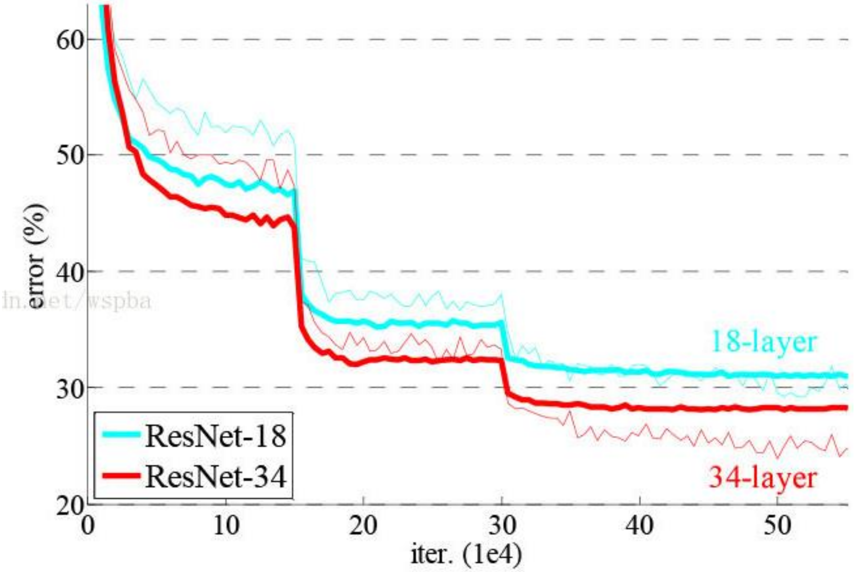
- Auto-encoder
- Generative Adversarial Network (GAN)
- Contrastive learning

……

Large Model Change Space

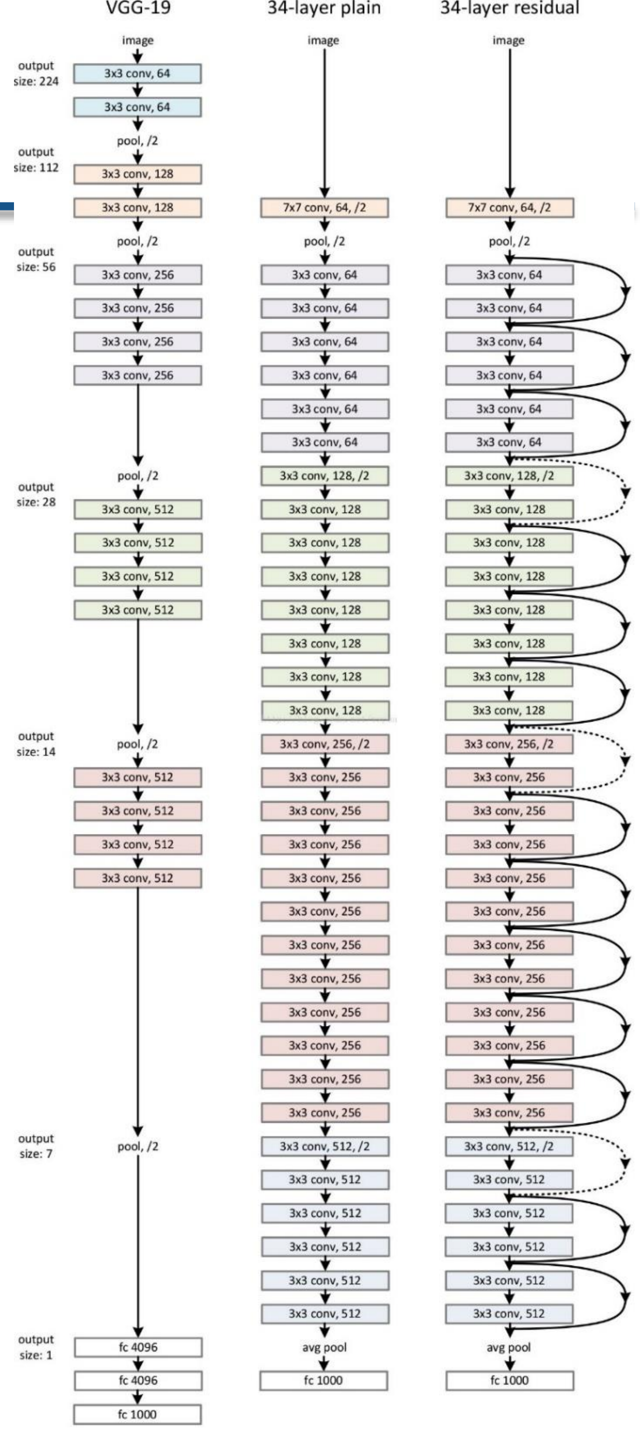
Deep Neural Networks

1998	2012	2014		2015
LeNet →	AlexNet →	VGGNet	InceptionNet →	ResNet



Deeper is Better?

Not surprised, more parameters, better performance

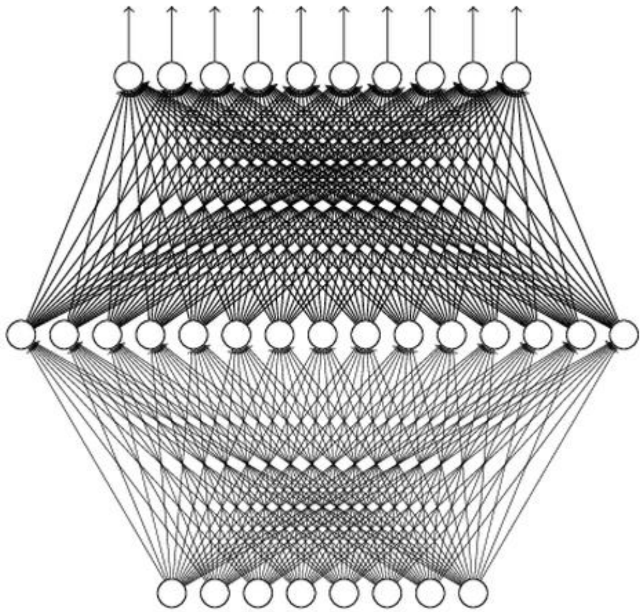


Universality Theorem

Any continuous function f

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

Can be realized by a network with one hidden layer (given **enough** hidden neurons)

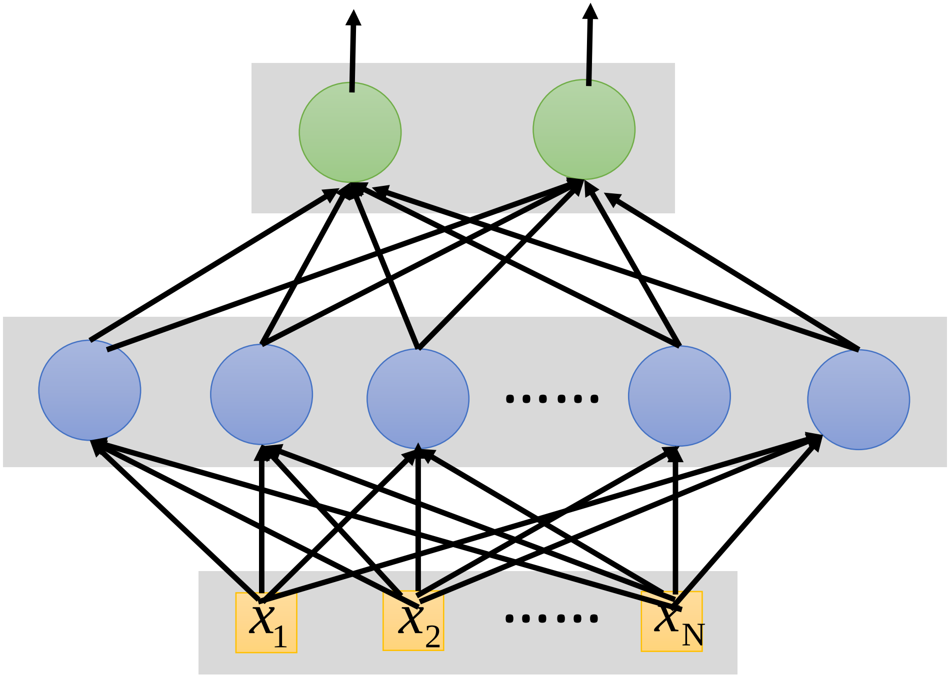


Reference for the reason:
http://neuralnetworksanddeeplearning.com/chap4.html

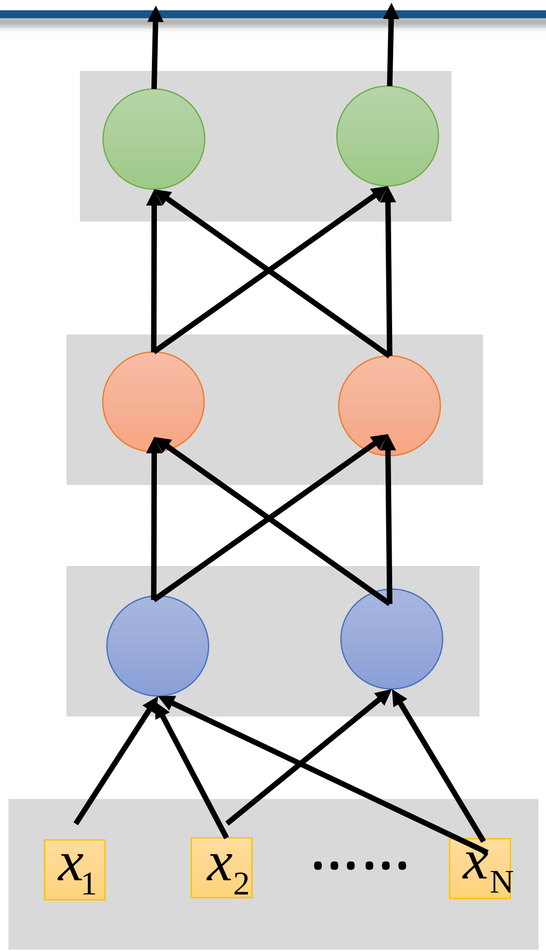
Why "Deep" neural network not "Fat" neural network?

Fat + Short v.s. Thin + Tall

The same number of parameters



Shallow

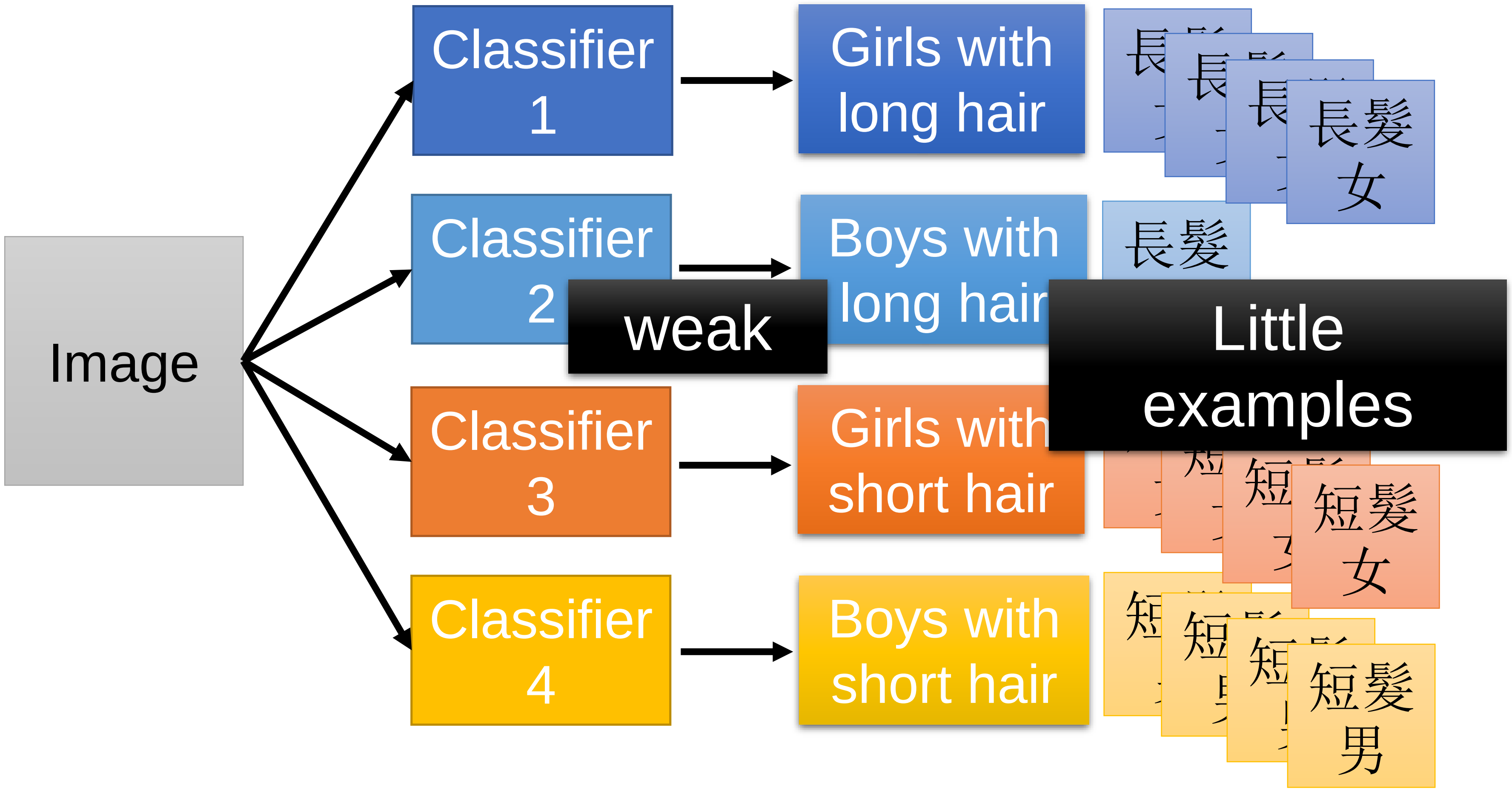


Deep

Which one is better?

Why Deep?

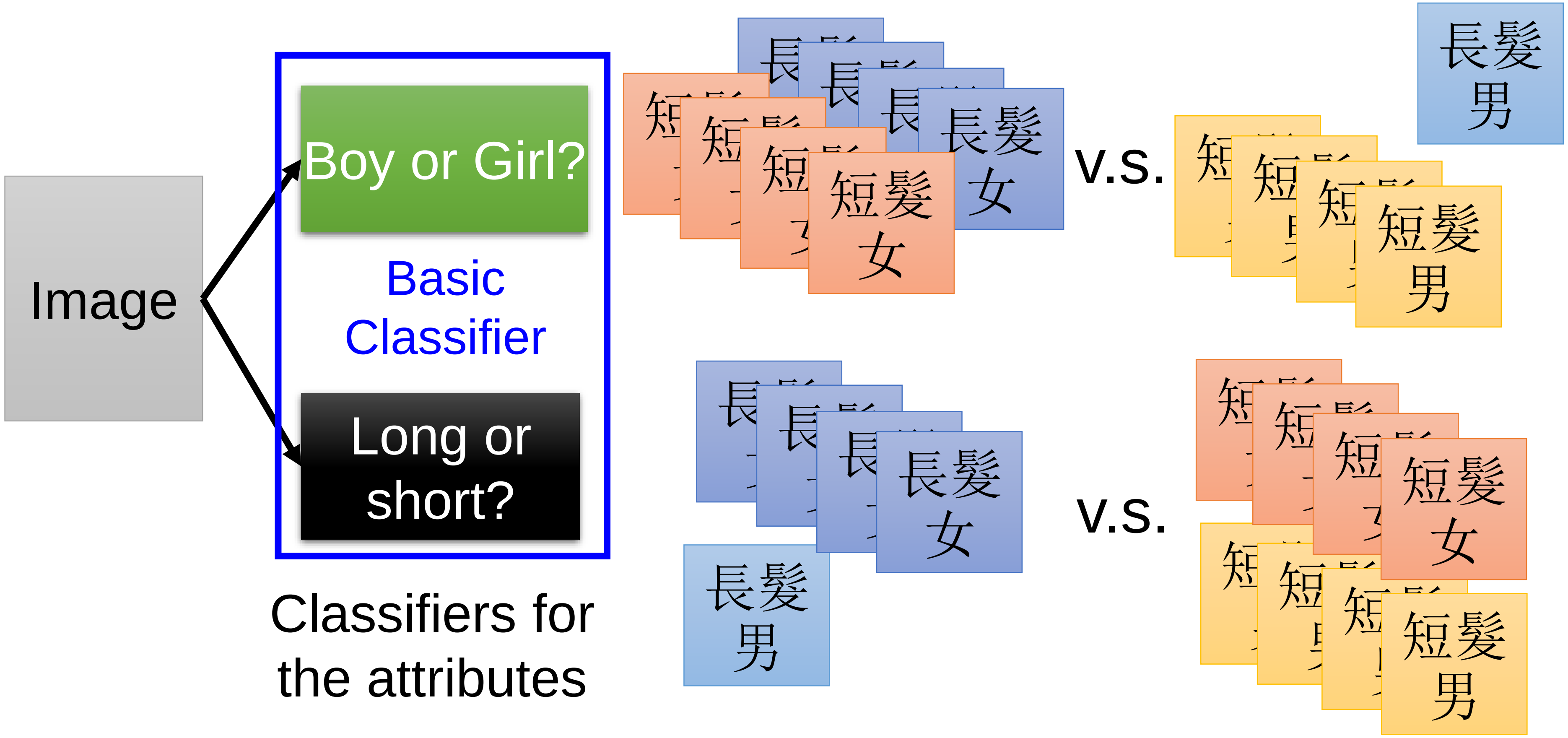
- Deep → Modularization



Why Deep?

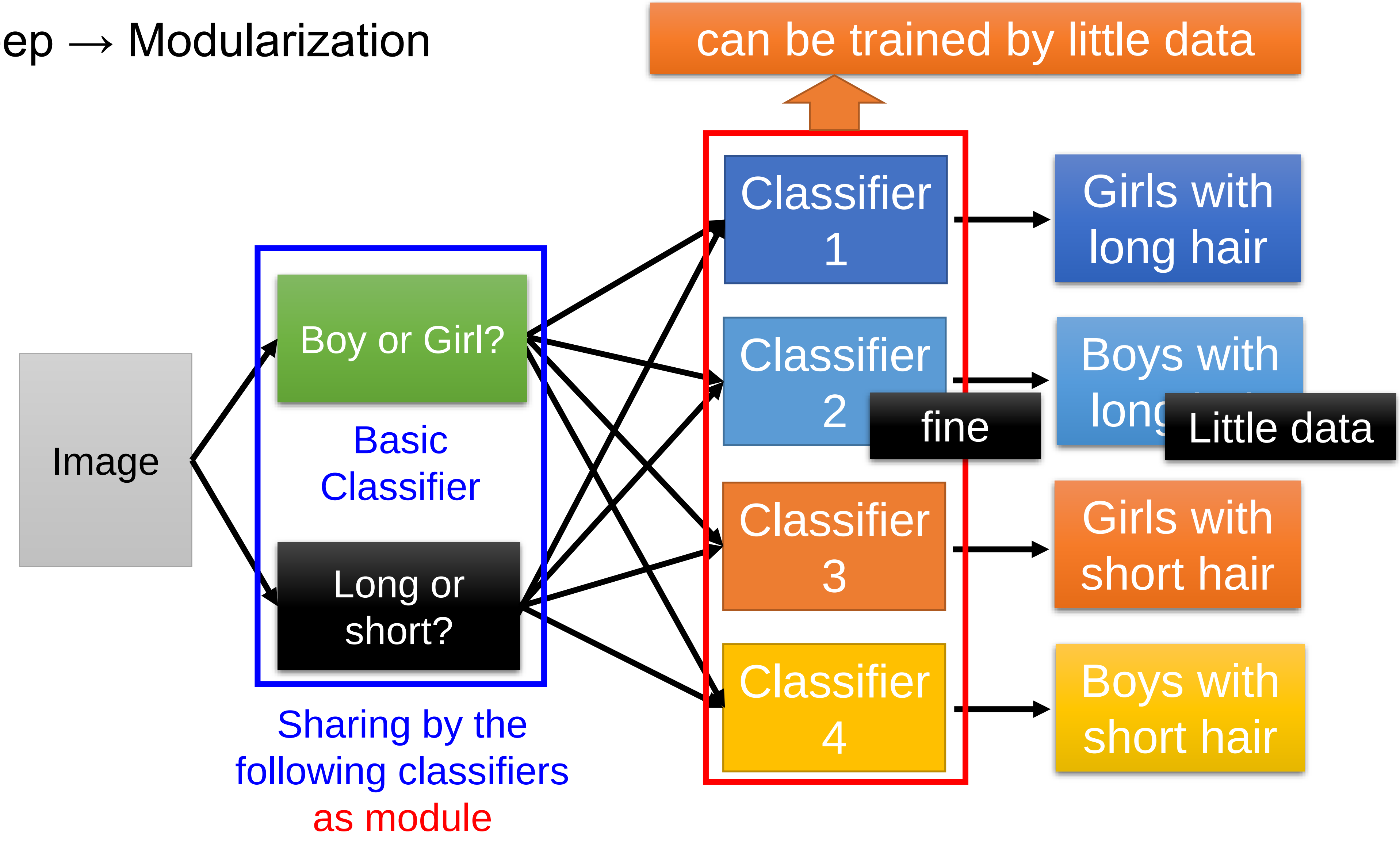
- Deep → Modularization

Each basic classifier can have sufficient training examples.



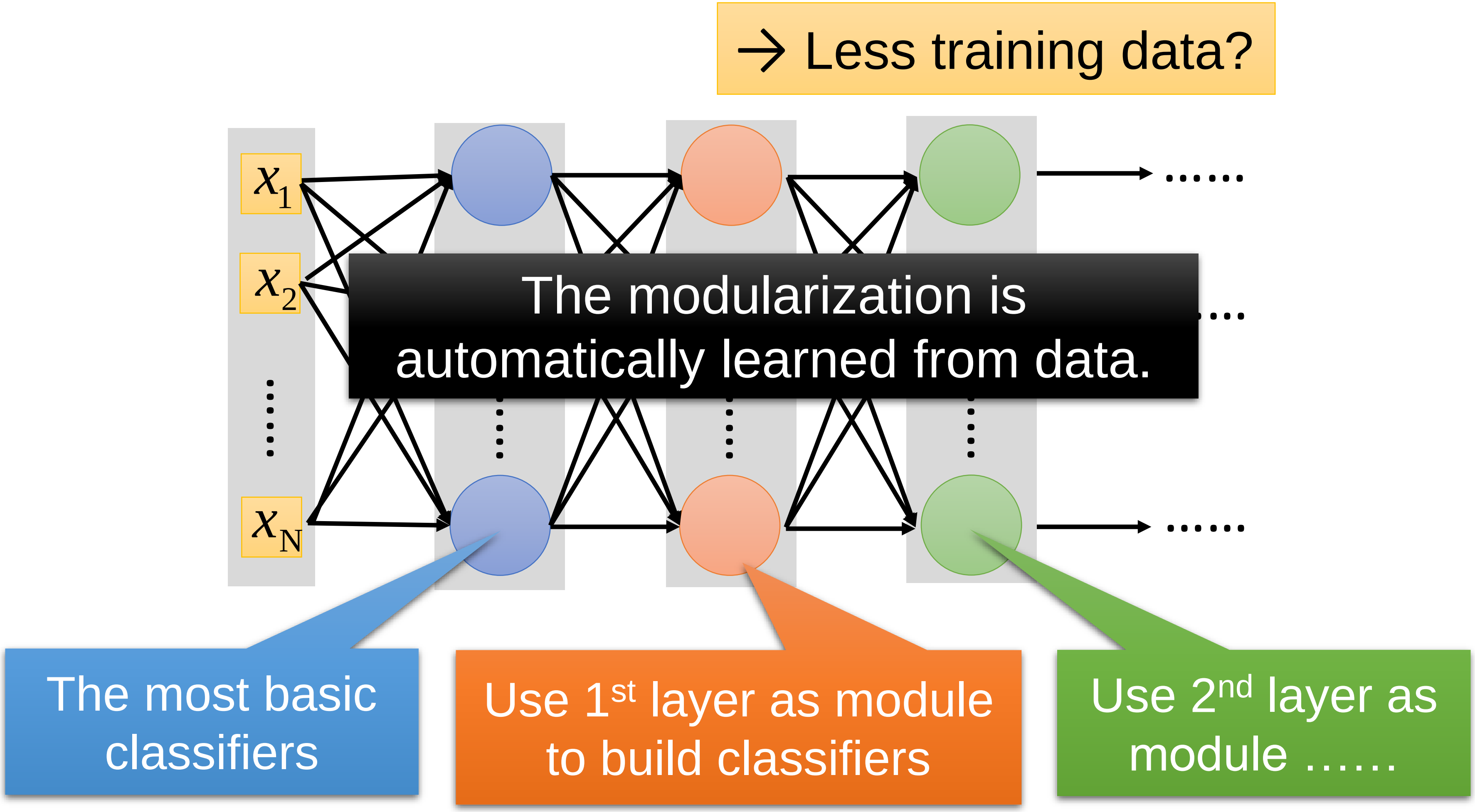
Why Deep?

- Deep → Modularization



Why Deep?

- Deep → Modularization

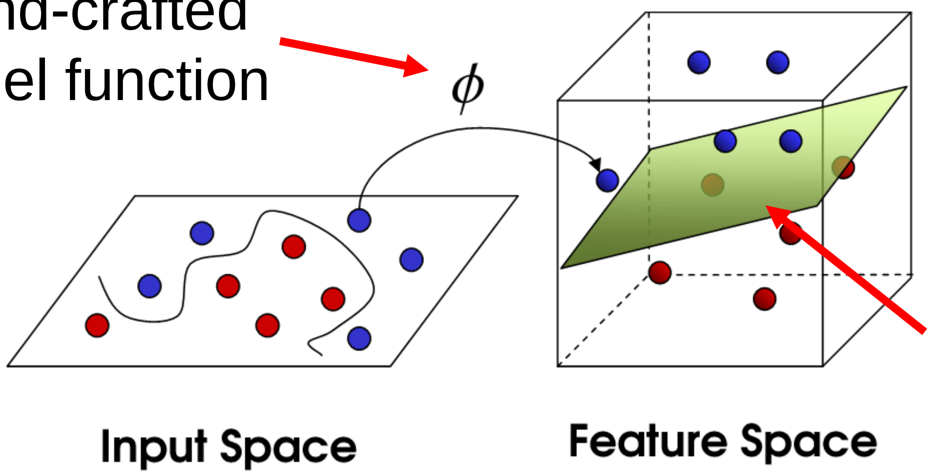


Comparison between Deep Learning and SVM

SVM

Hand-crafted kernel function

ϕ



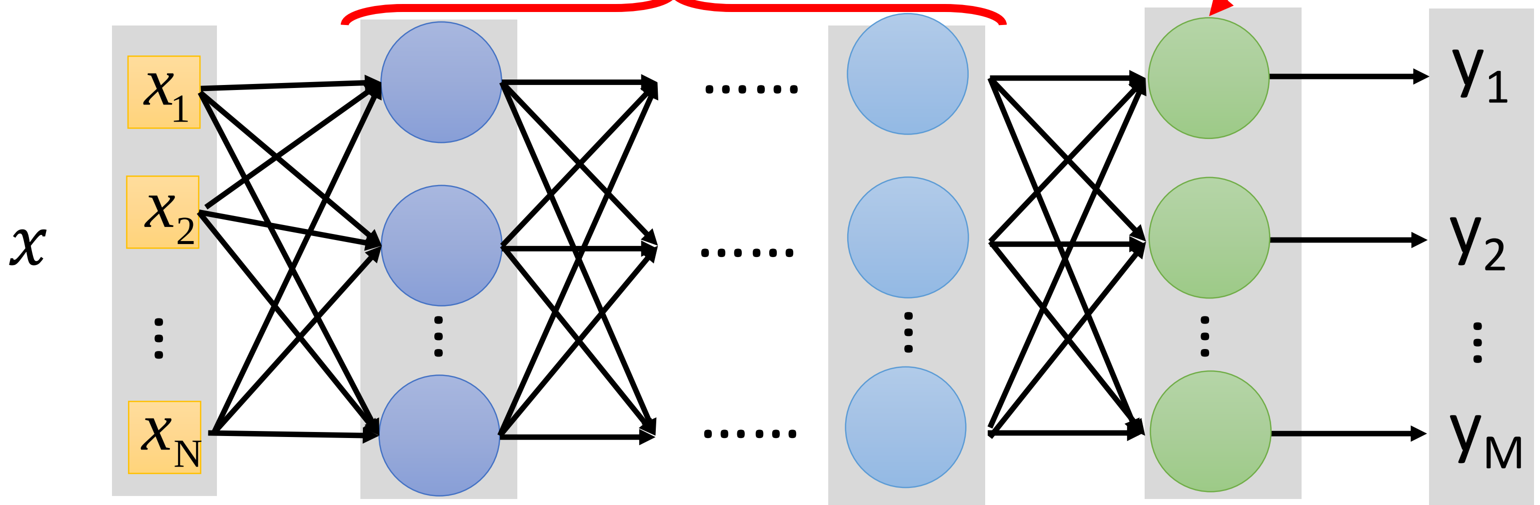
Apply simple classifier

Deep Learning

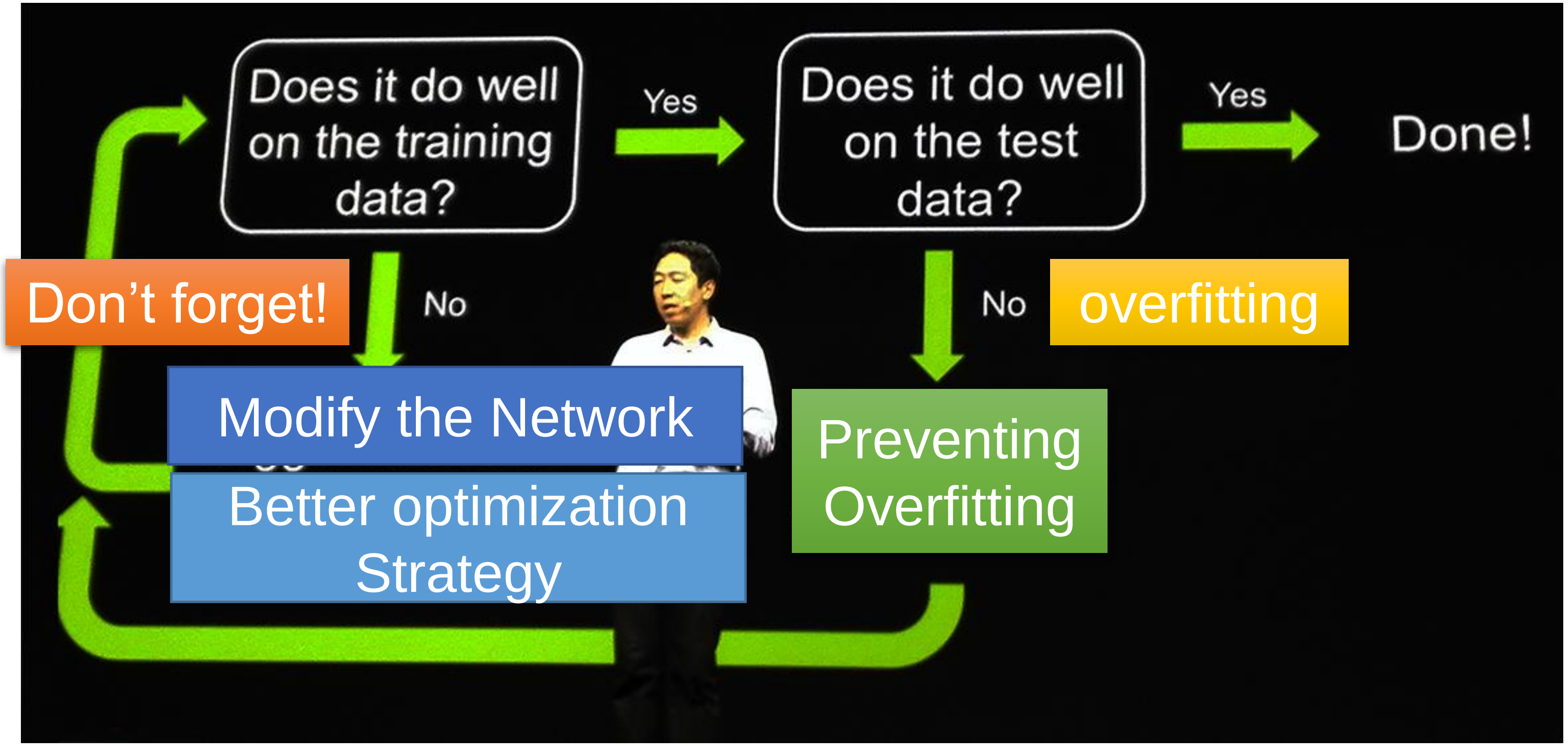
Learnable kernel

$\phi(x)$

simple classifier



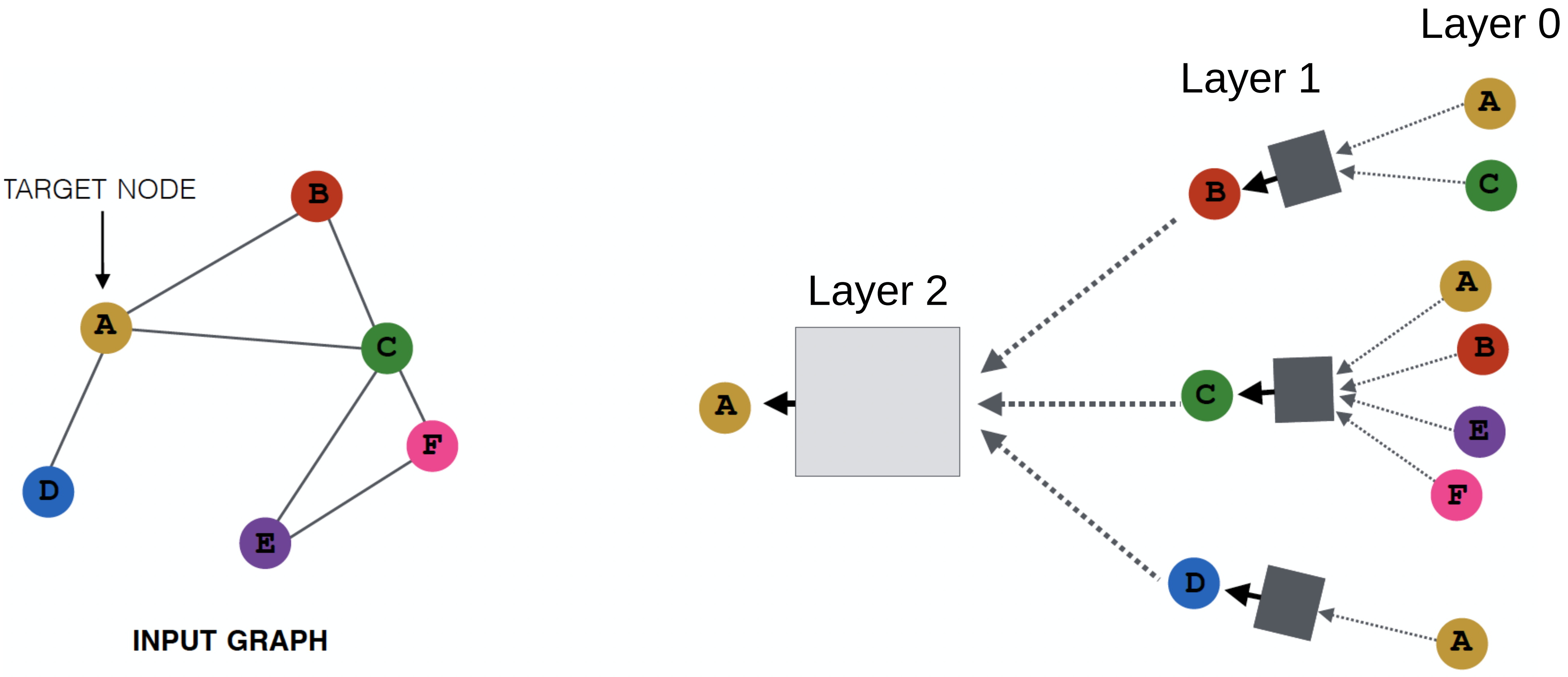
Recipe for Learning



http://www.gizmodo.com.au/2015/04/the-basic-recipe-for-machine-learning-explained-in-a-single-powerpoint-slide/

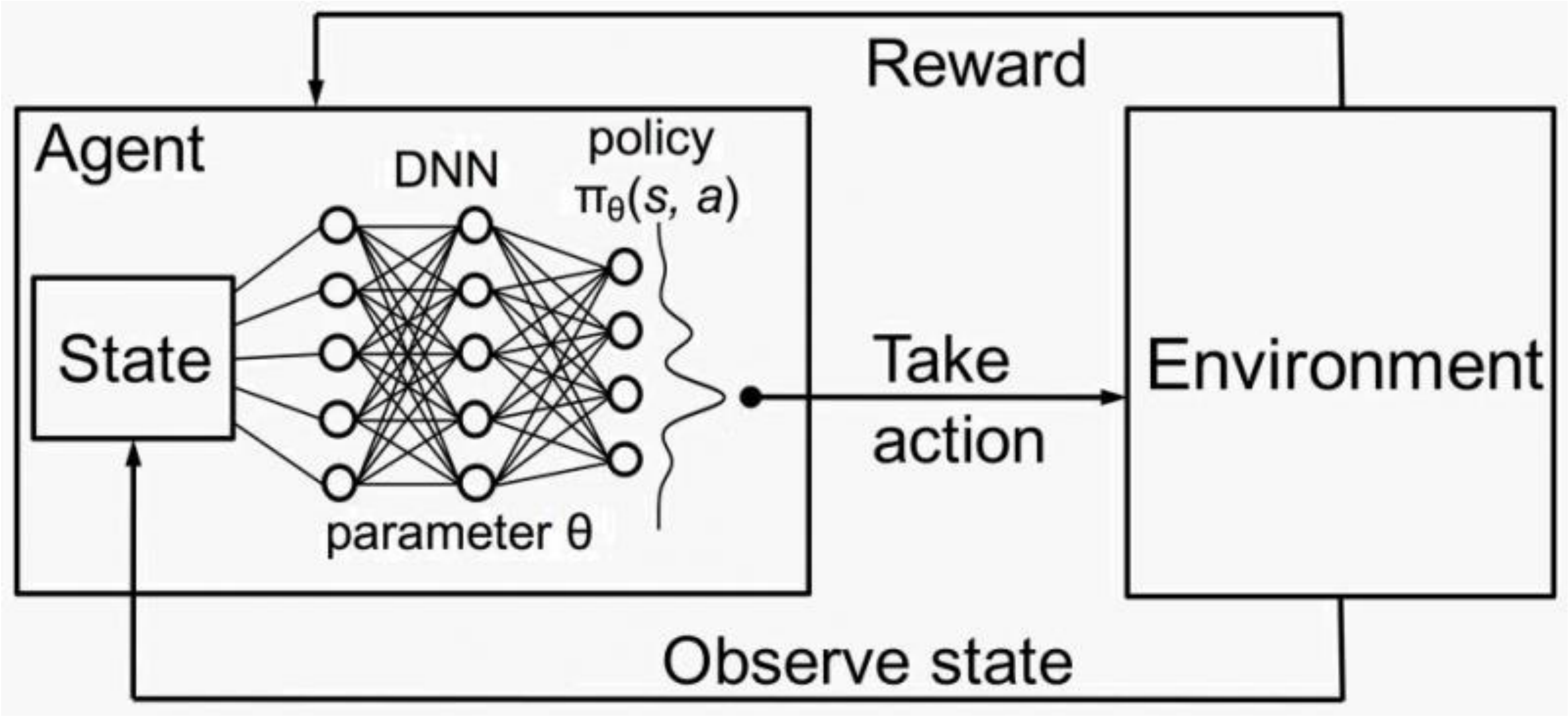
Graph Neural Networks

- Generate node embeddings by aggregating neighborhood information



GCN, GraphSAGE, GAT, …

Deep Reinforcement Learning Diagram

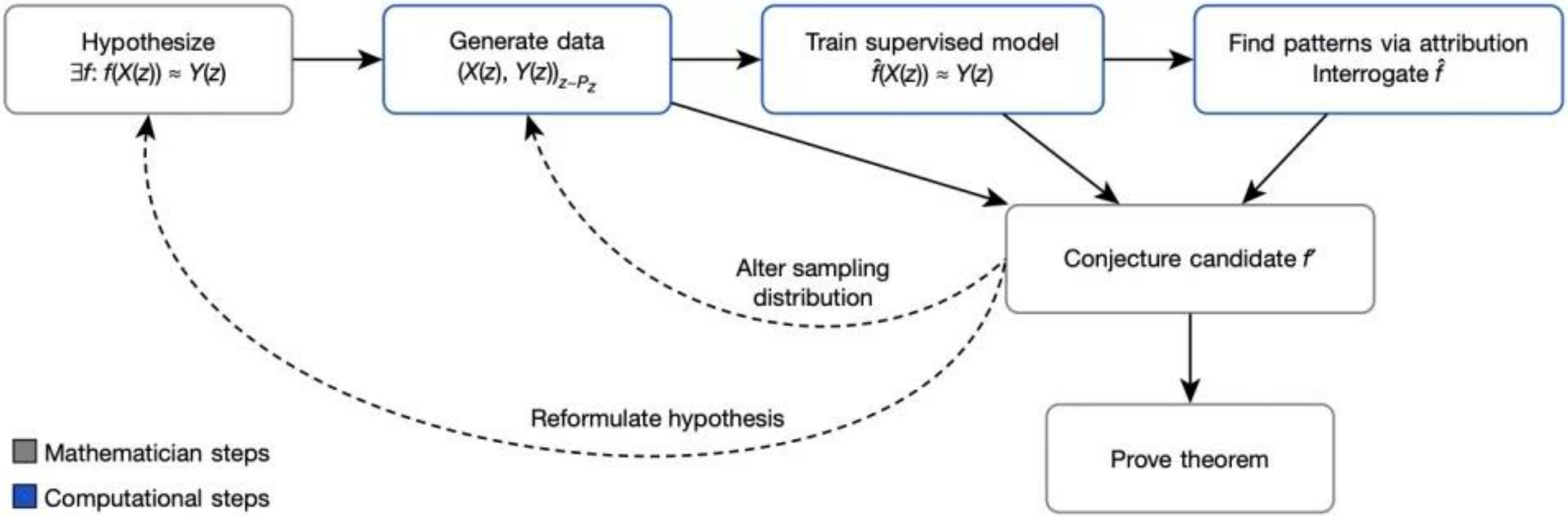


Deep Learning Revolution

Deep learning is everywhere
even for scientific discovery
AI for Science (AI4S)

Deep learning is everywhere
even for math

Advancing mathematics by guiding human intuition with AI



深度学习甚至被用来帮助证明或提出新的数学定理

Nature, 600, 70–74 (2021)

Outline

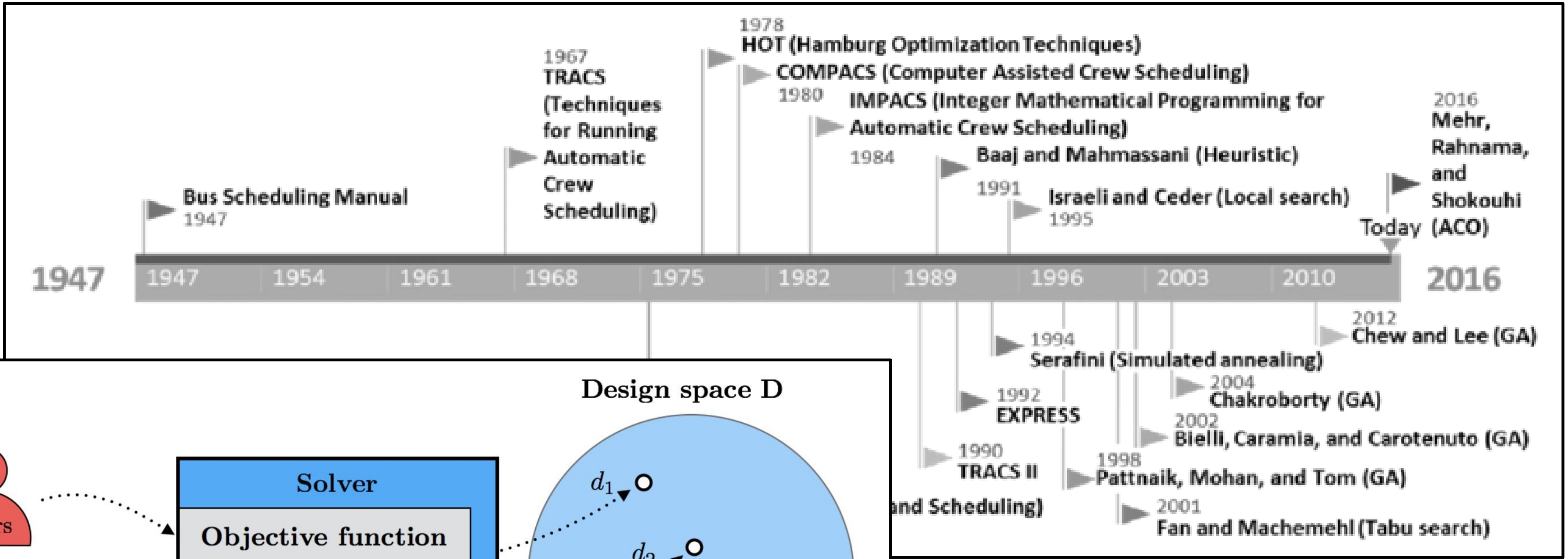
- Relationship of AI, ML, Deep Learning, and Neural Networks

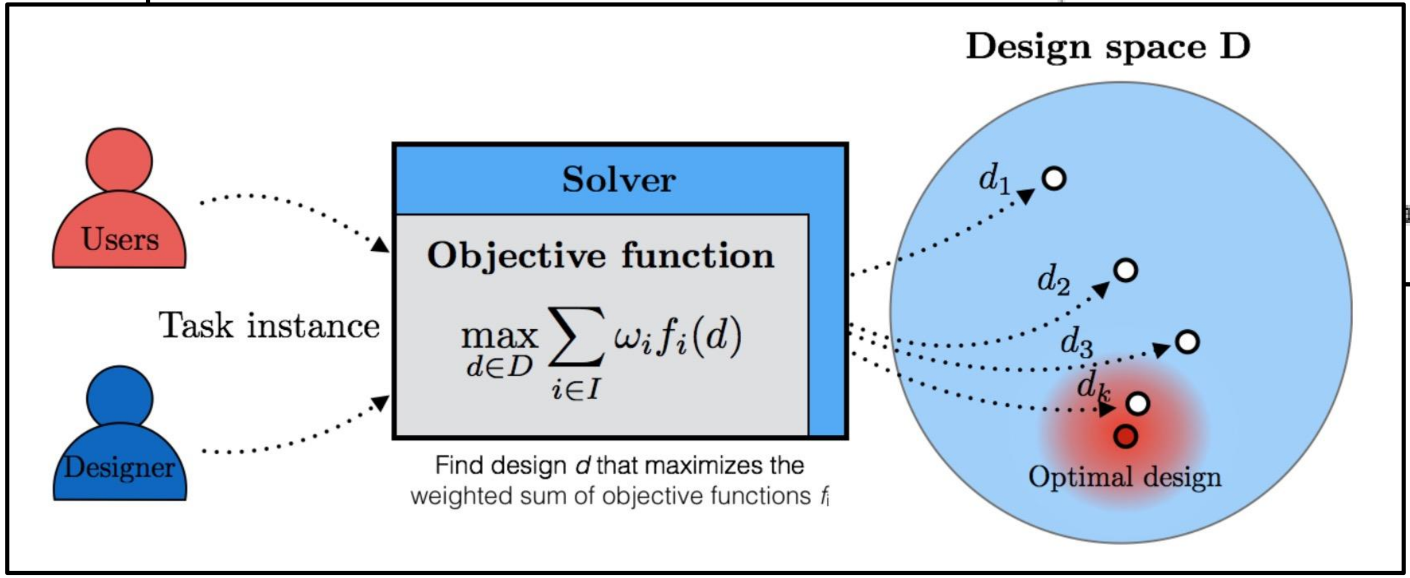
- Deep Learning
    - Neural Network (Architecture)
    - Total Cost (Loss)
    - Optimization
    - AI for Science

- Operation Research/Combinatorial Optimization
    - AI Meet Combinatorial Optimization
    - Learning Methods

- Agenda

Operation Research/Combinatorial Optimization





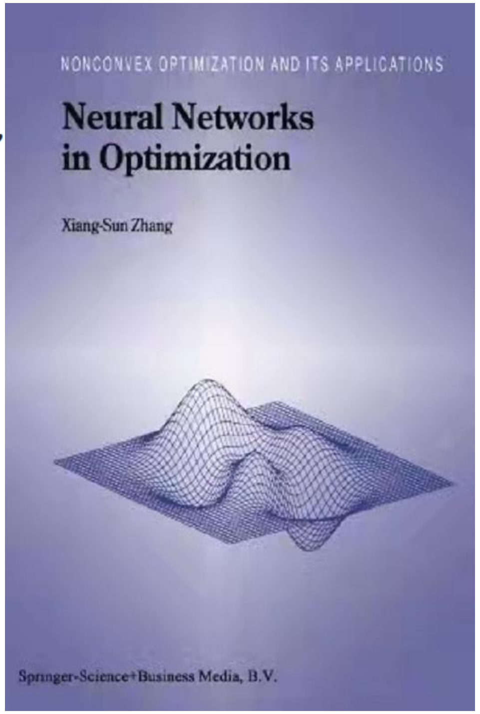
AI (Neural Networks) Meet Combinatorial Optimization

INFORMS Journal on Computing
Vol. 11, No. 1, Winter 1999

0899-1499/ 99 /1101-0015 $05.00

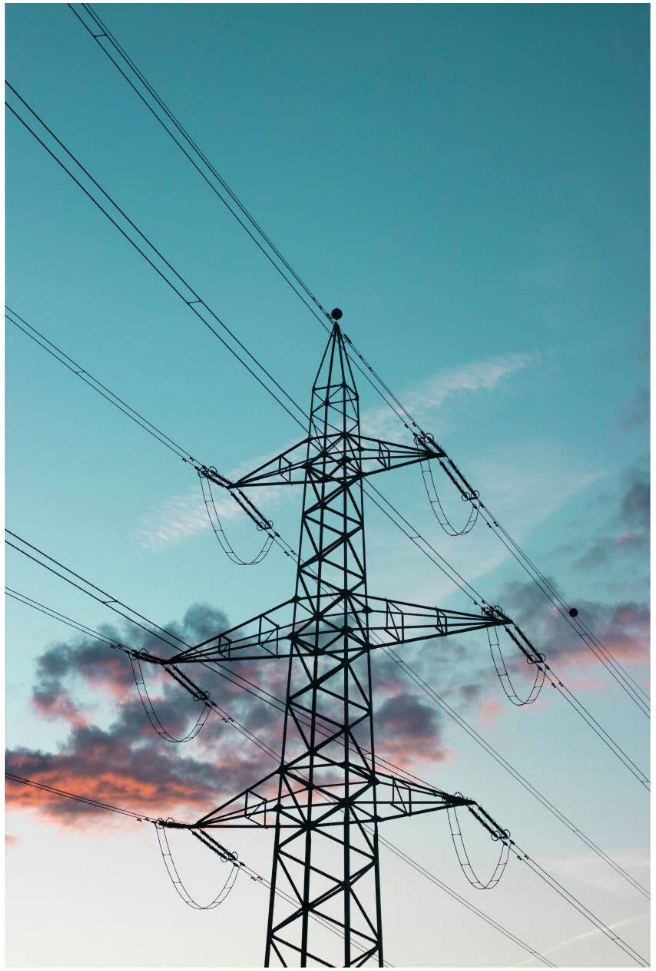
Neural Networks for Combinatorial Optimization: A Review of More Than a Decade of Research

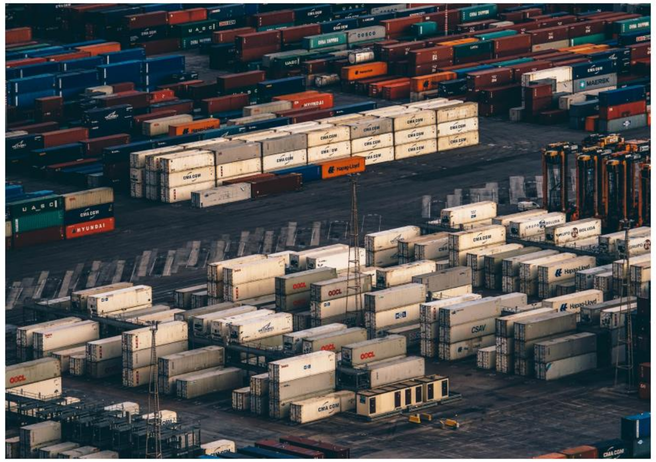
KATE A. SMITH / *School of Business Systems, Monash University, Clayton, Victoria, 3168,*
Email: ksmith@bs.monash.edu.au

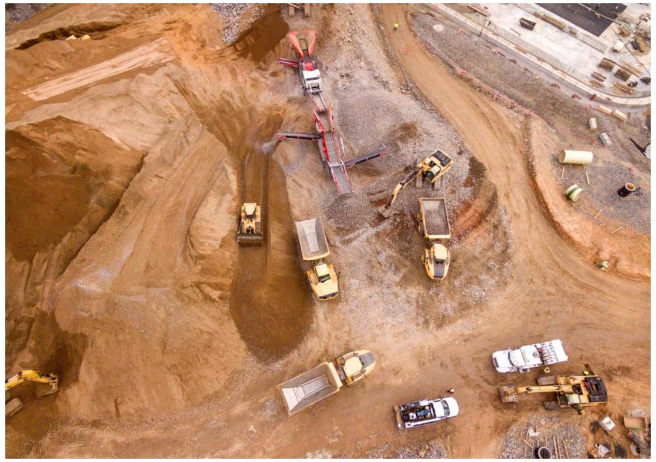


This is a long vision.

Operation Research/Combinatorial Optimization







Too long

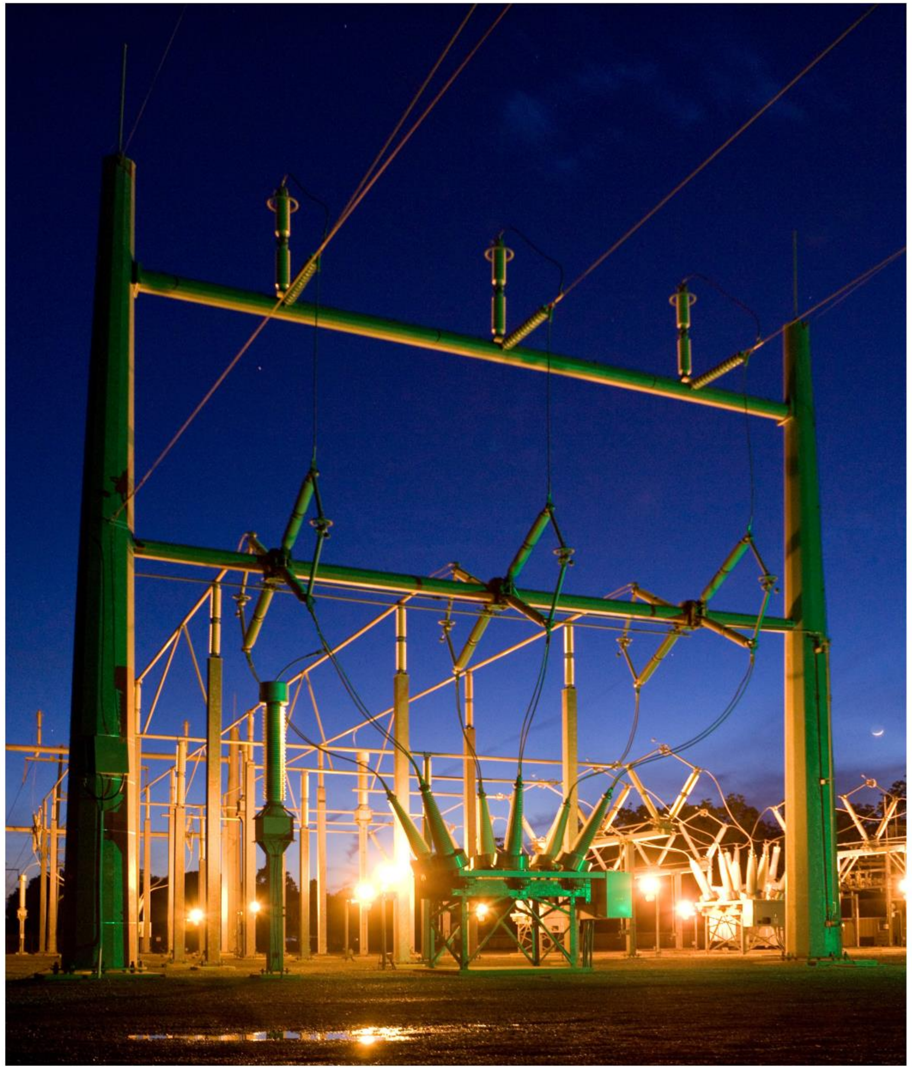
- Expert knowledge of how to make decisions
- Too expensive to compute
- Need for fast approximation

Too heuristic

- No idea which strategy will perform better
- Need a well-performing policy
- Need to discover policies

Aim: keep the **guarantees** provided by exact OR algorithms (feasibility or optimality)

Applications



- Many businesses care about solving similar problems repeatedly

- Solvers do not make any use of this aspect

- **Power systems and market**
  [Xavier et al. 2019]
  - Schedule 3.8 kWh ($400 billion) market annually in the US
  - Solved multiple times a day
  - 12x speed up combining ML and MILP

Structure Hypothesis

- Do not care about most instances that could exist;

- Instead, look at problem instances as data points from a specific, intractable, probability distribution;

- “Similar” instances show “similar” solving procedures.

Machine Learning or Imitation Learning

- Machine Learning is a collection of techniques for
  - learning patterns in or
  - understanding the structure of data
- often with the aim of performing data mining, i.e., recovering previously unknown, actionable information from the learned data.

- Typically, in ML (IL in particular) one has to “learn" from data (points in the so-called training set) a (nonlinear) function that predicts a certain score for new data points that are not in the training set.
- Each data point is represented by a set of features, which define its characteristics, and whose patterns should be learned.
- The techniques used in ML are diverse. Artificial (deep) neural networks are algorithmically boosted by first-order optimization methods.

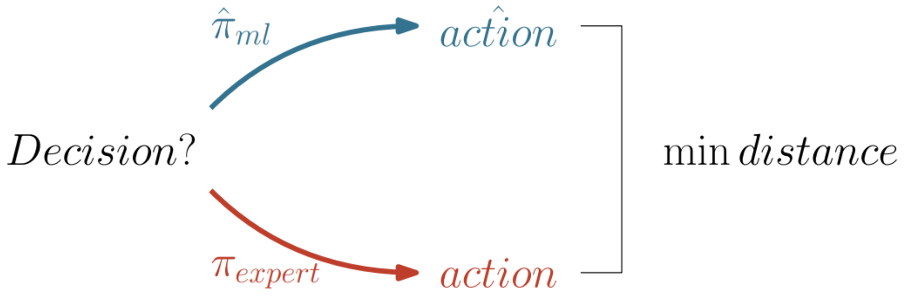
Learning Methods

Demonstration

- An expert/algorithm provides a policy
- Assumes theoretical/empirical knowledge about the decisions
- Decisions are too long to compute
- Supervised imitation learning

Experience

- Learn and discover new policies (better hopefully)
- Unsatisfactory knowledge (not mathematically well-defined)
- Decisions are too heuristic
- Reinforcement learning



Learning Methods (Not mutually exclusive)

Supervised

- Cannot beat the expert (an algorithm) → only interesting if the approximation is faster
- Can be unstable
- Cannot cope with equally good actions

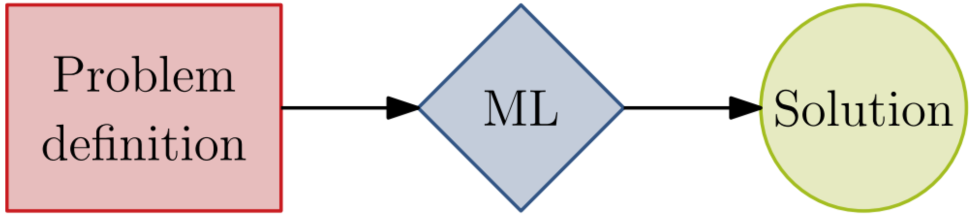
Reinforcement

- Reinforcement can potentially discover better policies
- Harder, with local maxima (exploration difficult)
- Need to define a reward

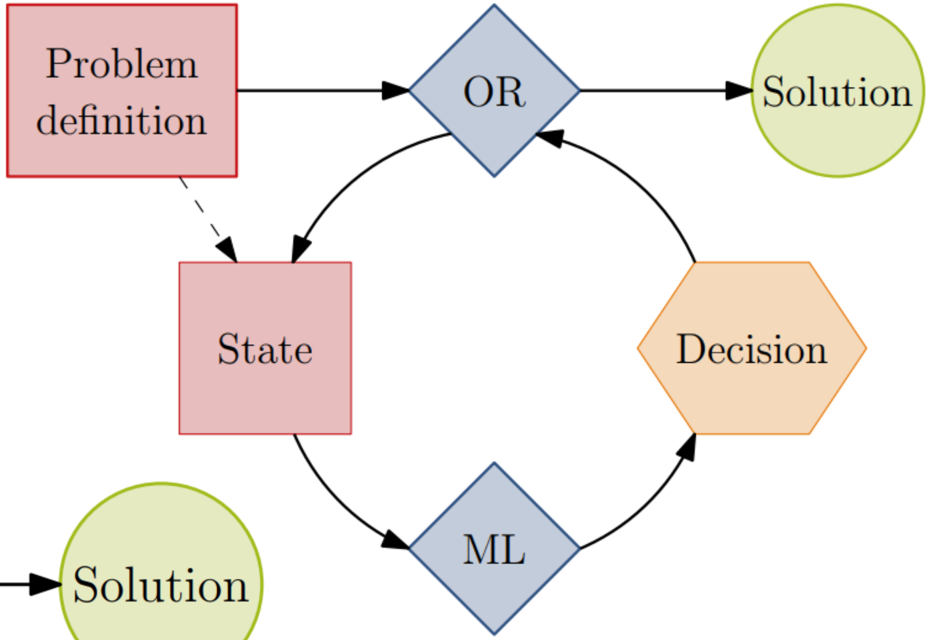
Algorithmic Structure

- How to build such algorithms? How to mix OR with ML?
- How to keep guarantees provided by OR algorithms (feasibility, optimality)?

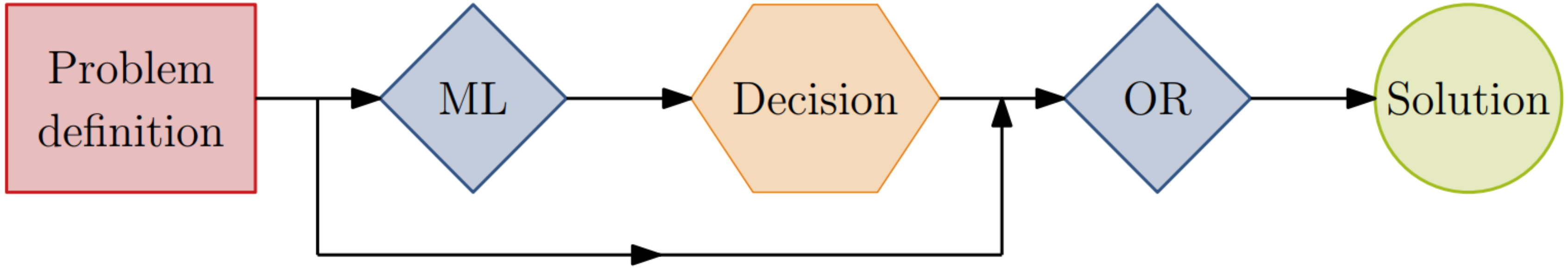
End to End Learning



Learning Repeated Decisions



Learning Properties

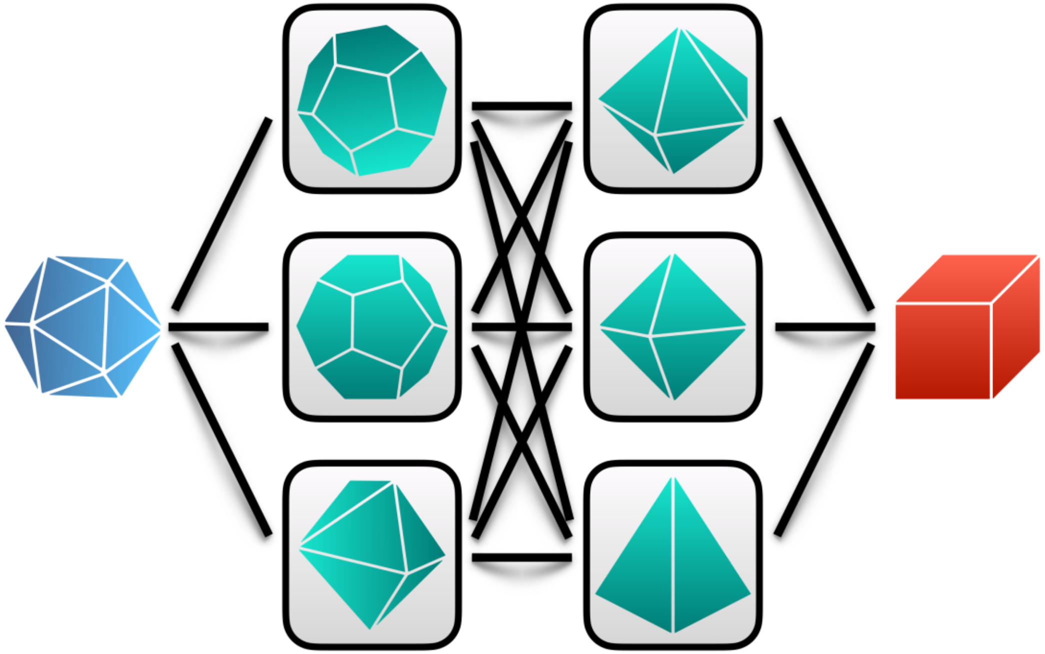


ML4CO NeurIPS 2021 Competition

- **Aim**: improving SOTA CO solvers by replacing key heuristic components with machine learning models.

- **Scientific question**: is machine learning a viable option for improving traditional CO solvers on specific problem distributions, when historical data is available?

Machine Learning for
Combinatorial Optimization
COMPETITION 2021



https://www.ecole.ai/2021/ml4co-competition/

Agenda

授课教师	节次	小节名称	学时
张世华（5.4）	1	智能驱动的运筹学概论	3
张世华（5.9）	2	旅行商问题	3
张世华（5.11）	3	稀疏线性规划	3
张世华（5.16）	4	混合整数规划	3
张世华（5.18）	5	几何学习	3
丁　超（5.23）	6	半定规划	3
丁　超（5.25）	7	非凸优化	3
闫桂英（5.30）	8	图组合优化与智能	3
闫桂英（6.1）	9	组合优化与图网络	3
闫桂英（6.6）	10	组合优化与博弈	3
张世华（6.8）	11	考核研讨会	2

1:30-3:10课堂讲授　　　3:20-4:10交流讨论

平时作业：任一主题综述或应用总结 (2篇，12页/篇)

考核研讨会：两轮演讲(分组研讨+课堂演讲讨论)

Acknowledgement

- The original contributors for all materials collected from the Internet (maybe without proper citation).

- All ML members at my lab.

AI4OR课程群

Thanks

zsh@amss.ac.cn

该二维码7天内(5月10日前)有效，重新进入将更新