

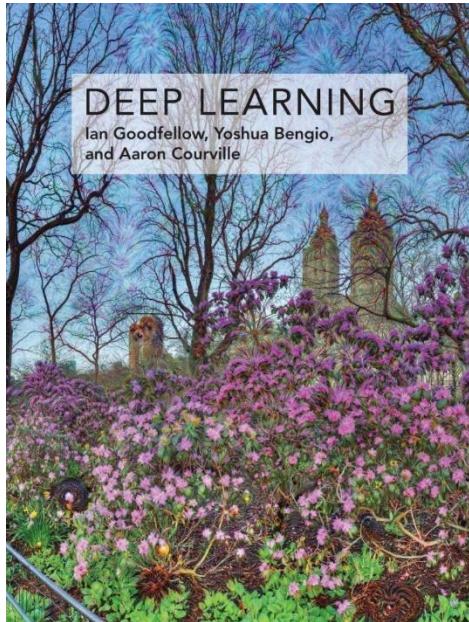
# Introduction to Deep Learning

中国科学院数学与系统科学研究院

张世华

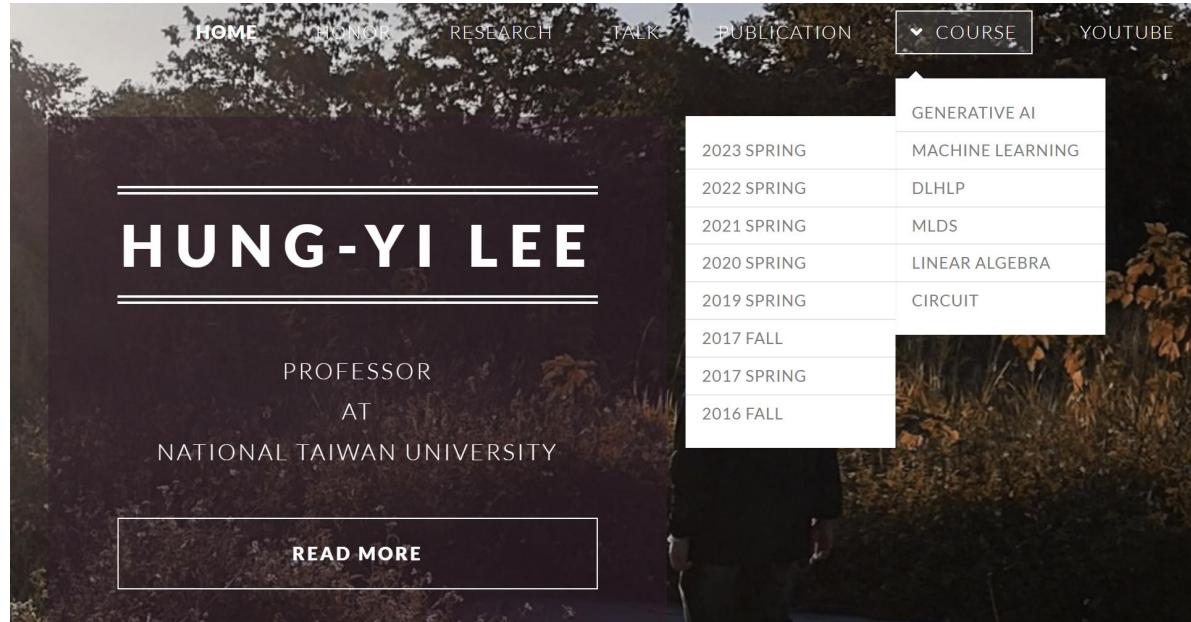
[zsh@amss.ac.cn](mailto:zsh@amss.ac.cn)

# Basic References



Book

<http://www.deeplearningbook.org/>



Deep Learning Tutorial  
Hung-yi Lee (李宏毅)

<https://speech.ee.ntu.edu.tw/~hylee/ml/2019-spring.php>

# Overview of Deep Learning

# What's Deep Learning?

大众眼中的我们



工程师眼中的我们



数学家眼中的我们



我们眼中的自己



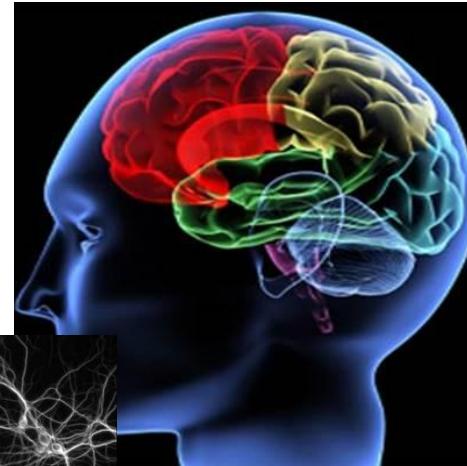
实际中的我们



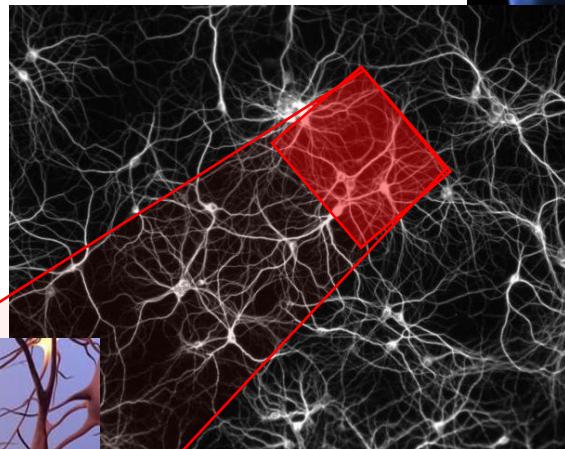
Deep learning is a **subfield** of machine learning, and neural networks make up the **backbone** of deep learning algorithms.

我们了解了深度学习的形式，但是是否真正理解了深度学习？

# What's Neural Networks?

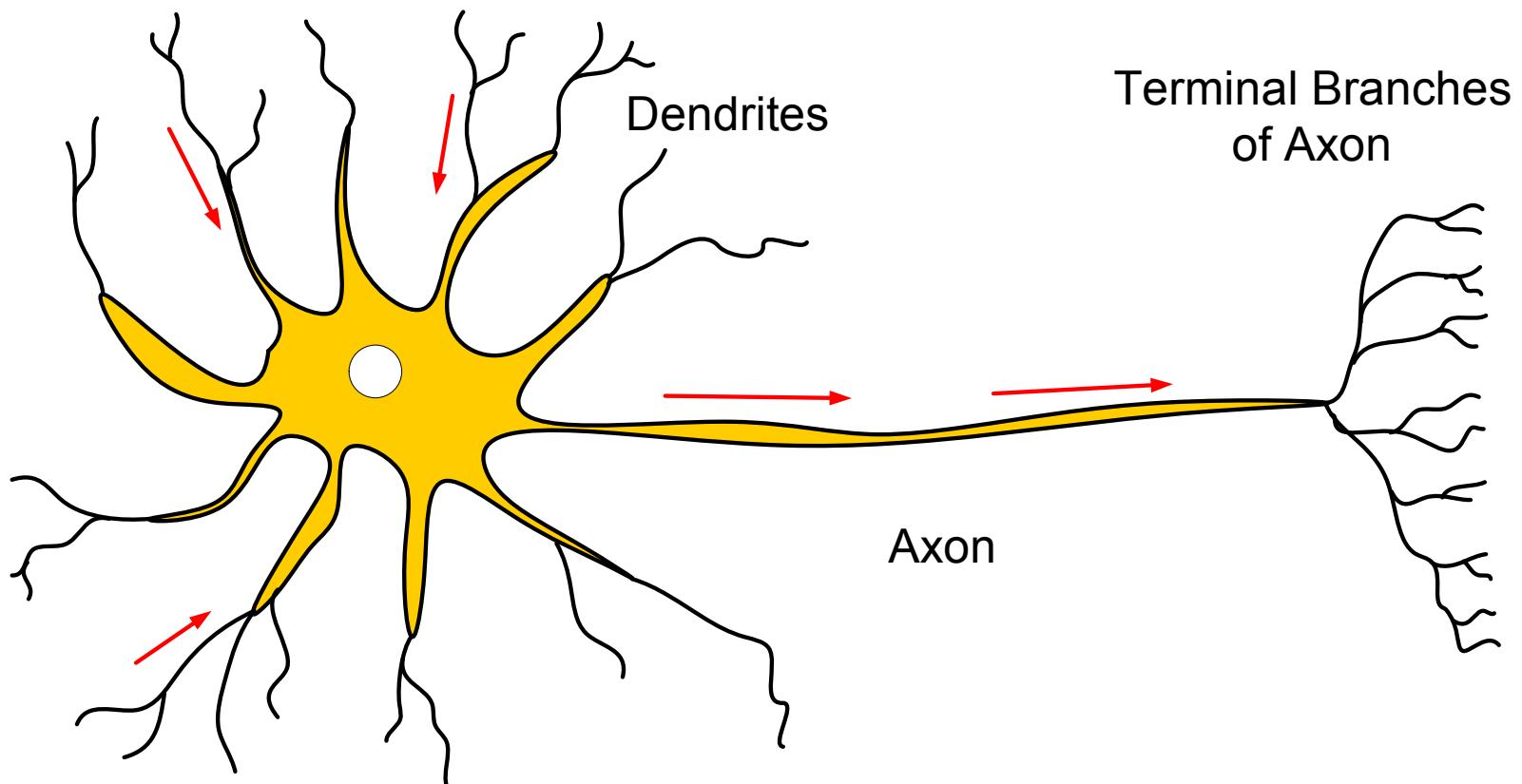


Biological  
Neurons



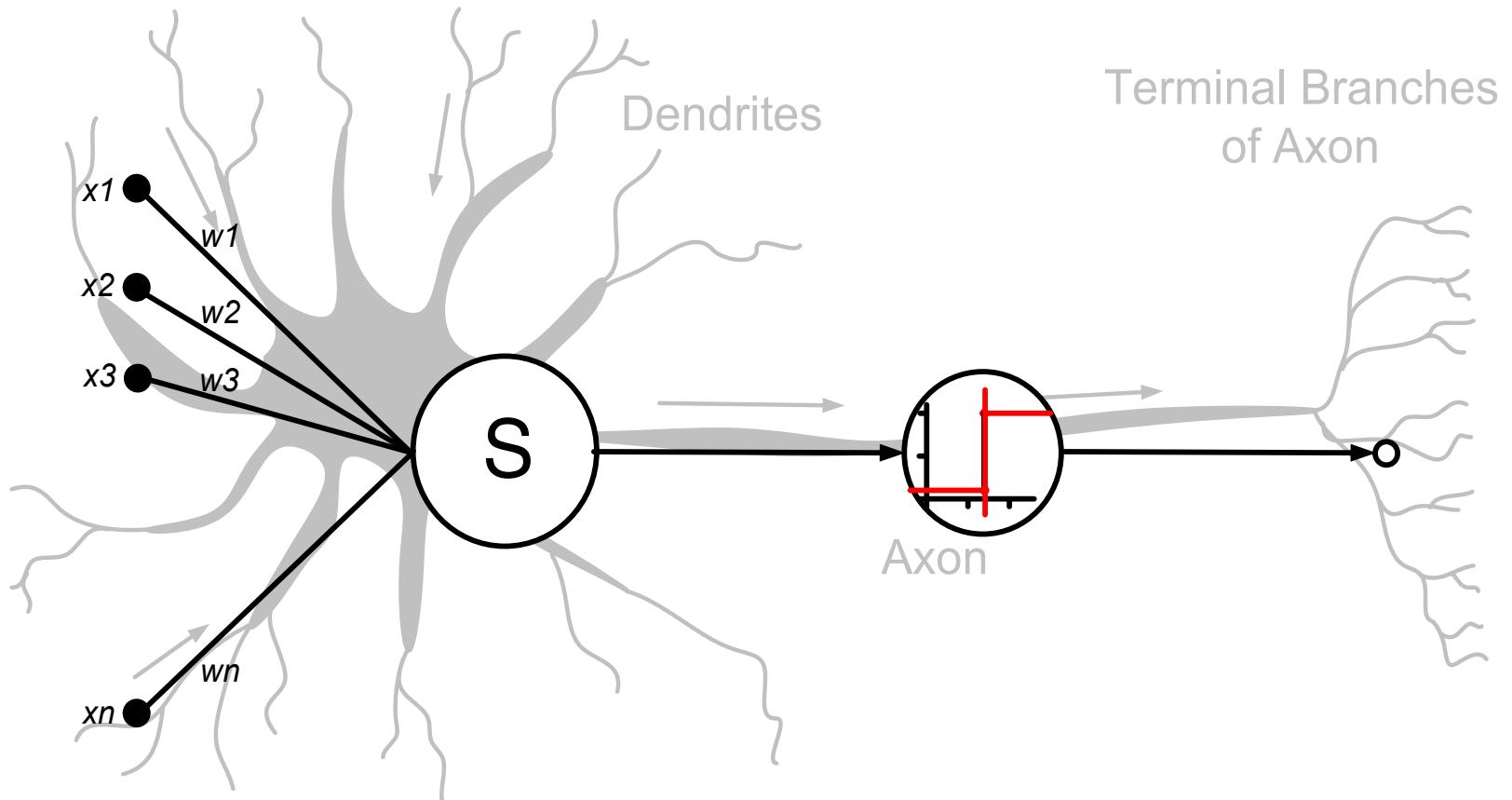
- 人类大脑约包含500-1000亿个 ( $10^{11}$ ) 神经元
- 细胞信号通过1,000,000,000,000,000 ( $10^{15}$ ) 突触连接传递到对方

# What's Neural Networks? Biological Neurons



树突(dendrite)和轴突(axon)

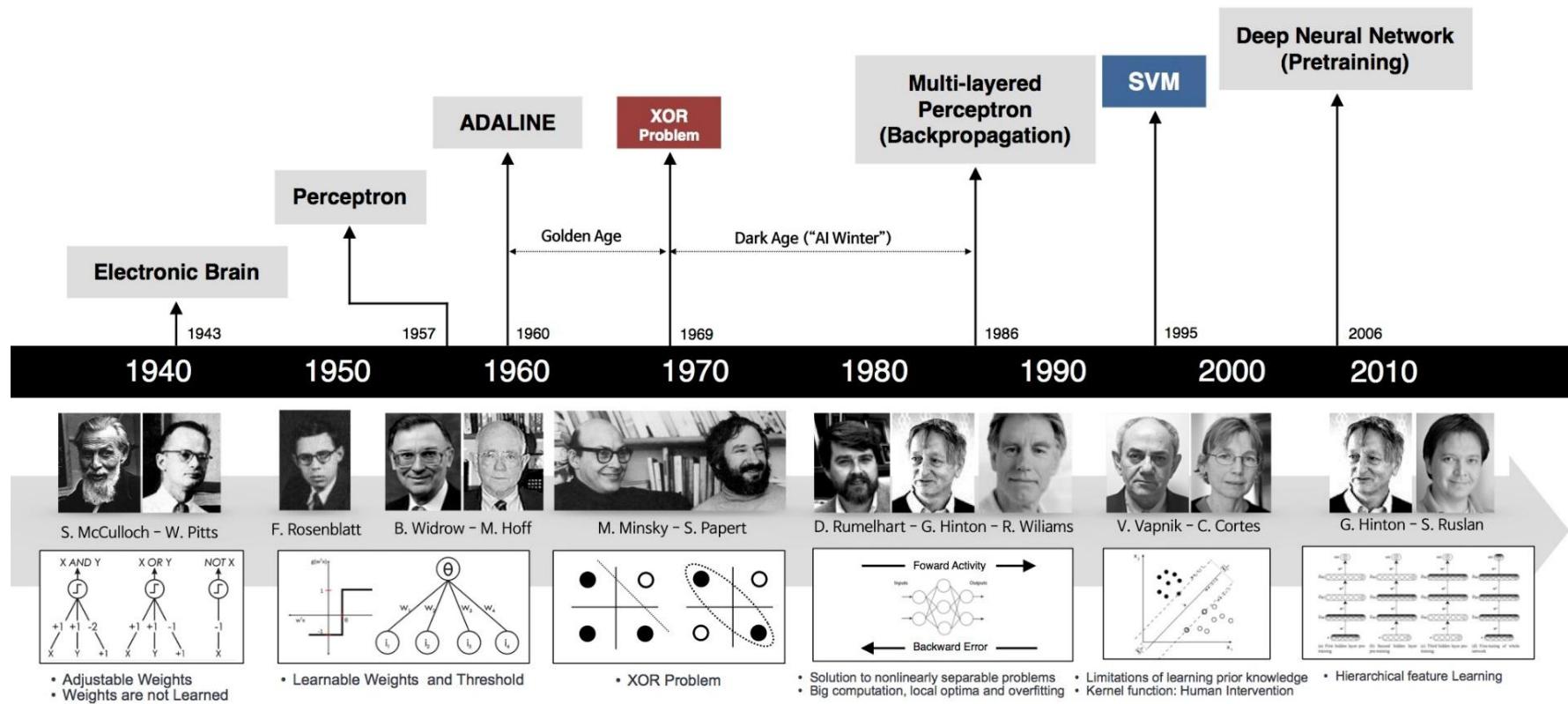
# Artificial Neural Networks (ANN)



Slide credit : Andrew L. Nelson

ANN在干什么?怎么得到w?是否可以应用于新来的数据?

# History of Neural Networks



By Wei Di , Jianing Wei , Anurag Bhardwaj

# What's Deep Learning?

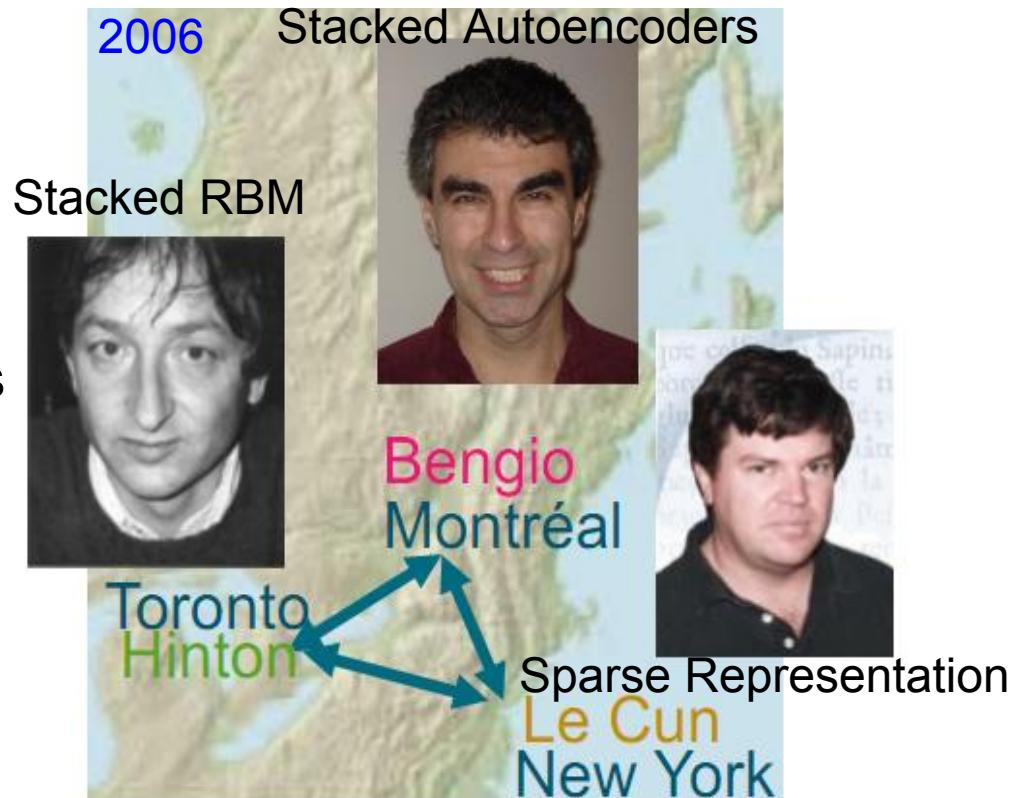
- A particular class of **learning algorithms**.
- Rebranded **neural networks**: with multiple layers.
- Inspired by the **neuronal** architecture of the Brain.
- Renewed interest in the area due to a few recent **breakthroughs**.
- Learn **parameters** from data.
- **Nonlinear** classification or clustering.

Deep architecture is an **ensemble** of shallow networks

# Deep Learning Revolution

**Before 2006**

**Fail** to train deep architectures



**Fathers of the Deep Learning Revolution Receive ACM A.M. Turing Award (2018)**

<https://awards.acm.org/about/2018-turing>

# Old Wine in a New Bottle!



Big Data  
(Digitalization)



Computation  
(Moore's Law, GPUs)



Algorithmic  
Progress

“2006”

- Faster machines (GPU's!)
- More data
- New methods for unsupervised pre-training

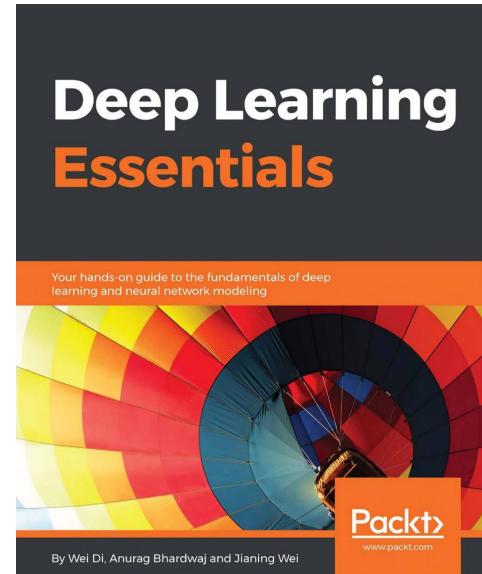
# Deep Learning Revolution

- It seems that most theoretical breakthroughs had already been made by the **1980s-1990s**, so what else has changed in the past decade?
- A not-too-controversial theory is that the success of deep learning is largely a success of **engineering**.

If you treat the **theoretical development of deep learning** as the **engine**, fast **computer**, the development of graphics processing units (**GPU**) and the occurrence of massive **labeled datasets** are the **fuels**.

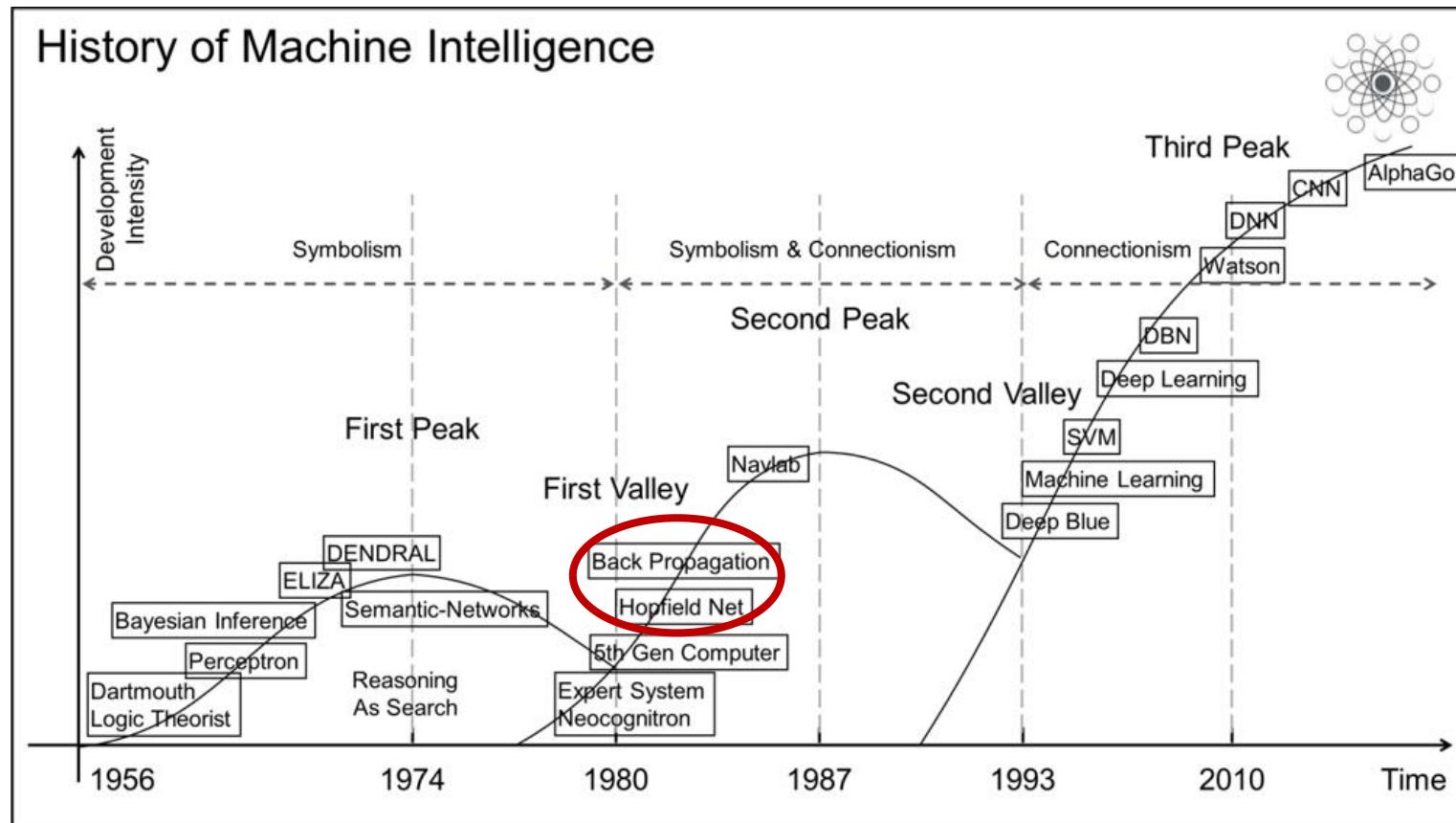
Ng

Andrew

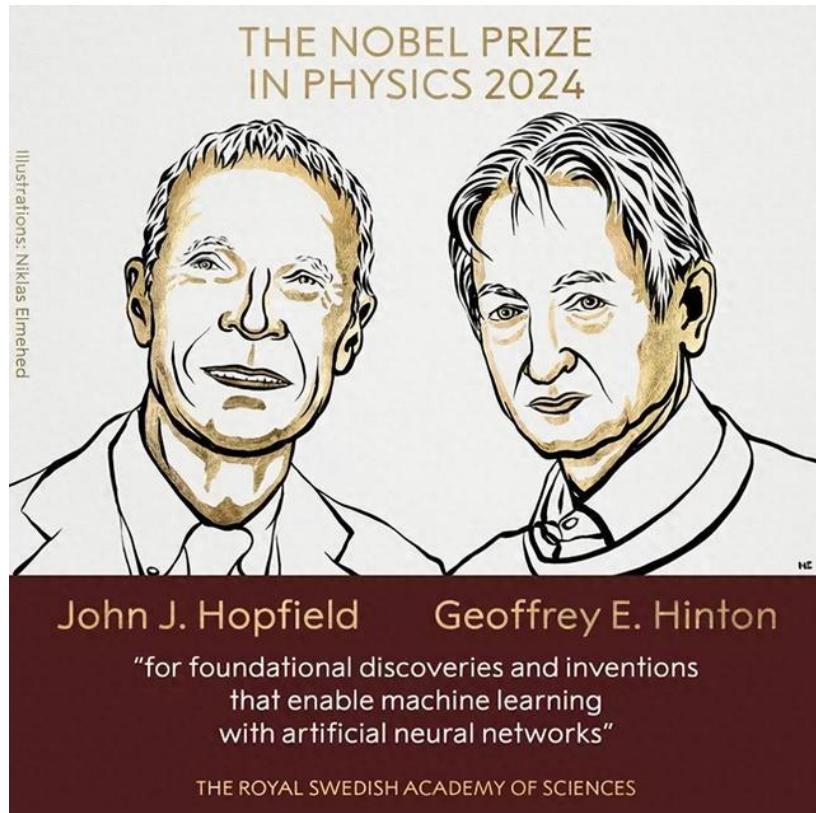


# History of Machine Intelligence

## Deep learning and Artificial (Machine) Intelligence



# Nobel Prize for AI

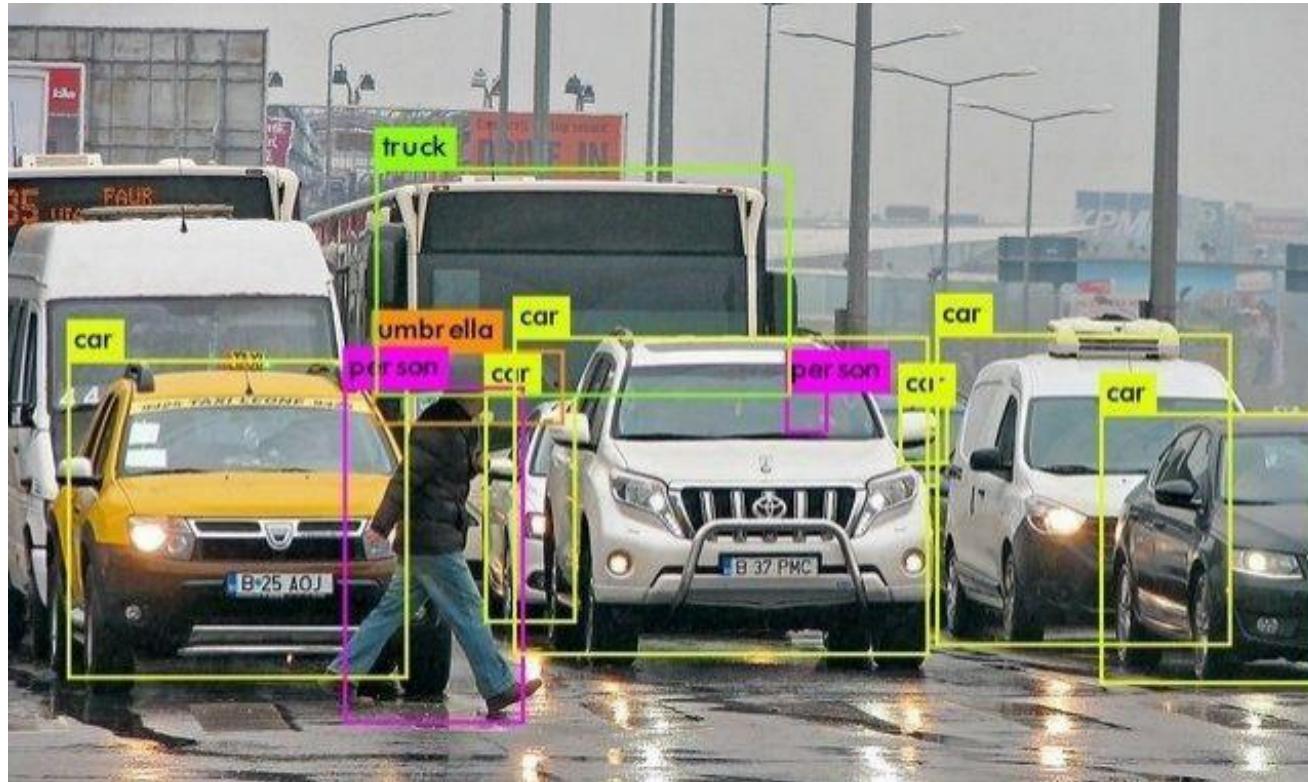


John J. Hopfield and Geoffrey E. Hinton 获  
2024年诺贝尔物理学奖

“基于人工神经网络实现机器学习  
的基础性发现和发明”

Deep learning is everywhere  
even for **scientific discovery**

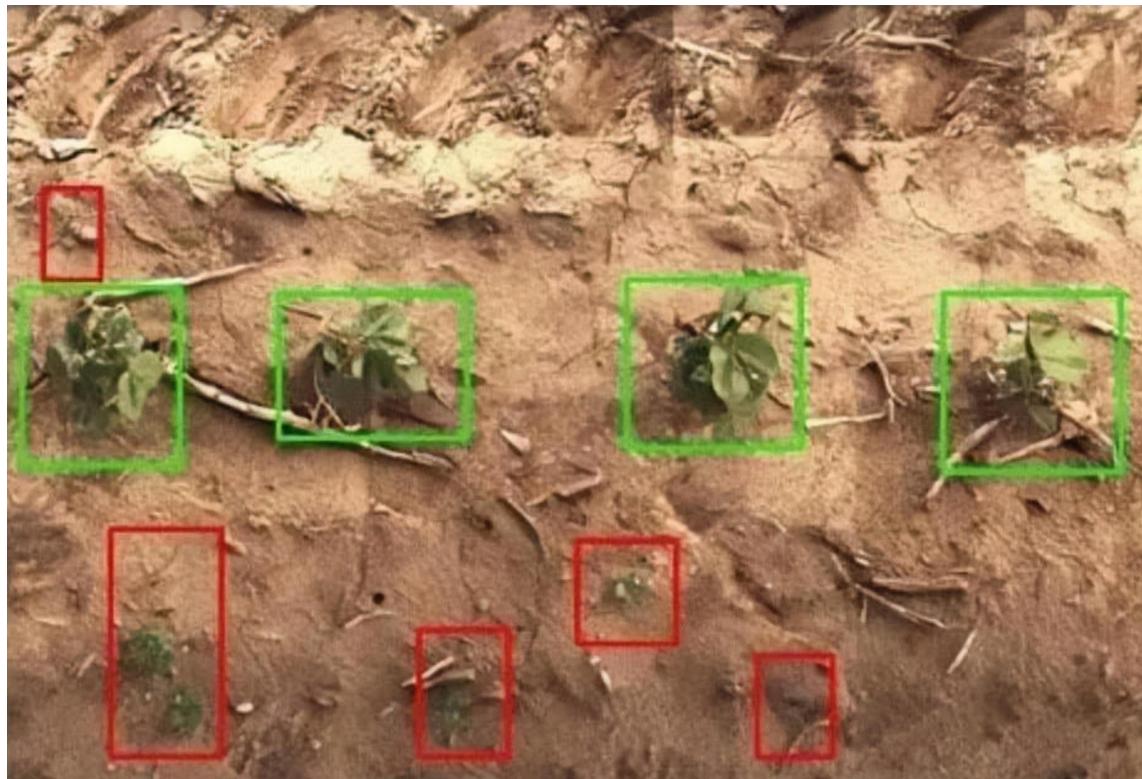
# Automatic Drive



# Deep learning helps cars identify objects on the road

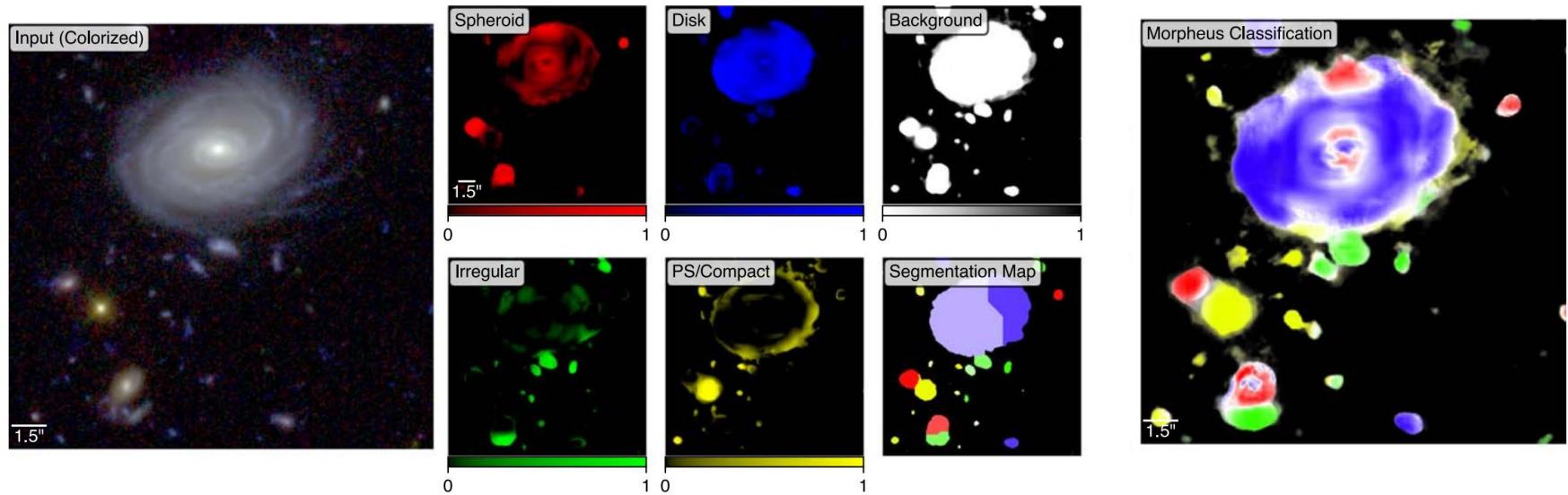
Preprint arXiv:1811.10399, 2018

# Agriculture



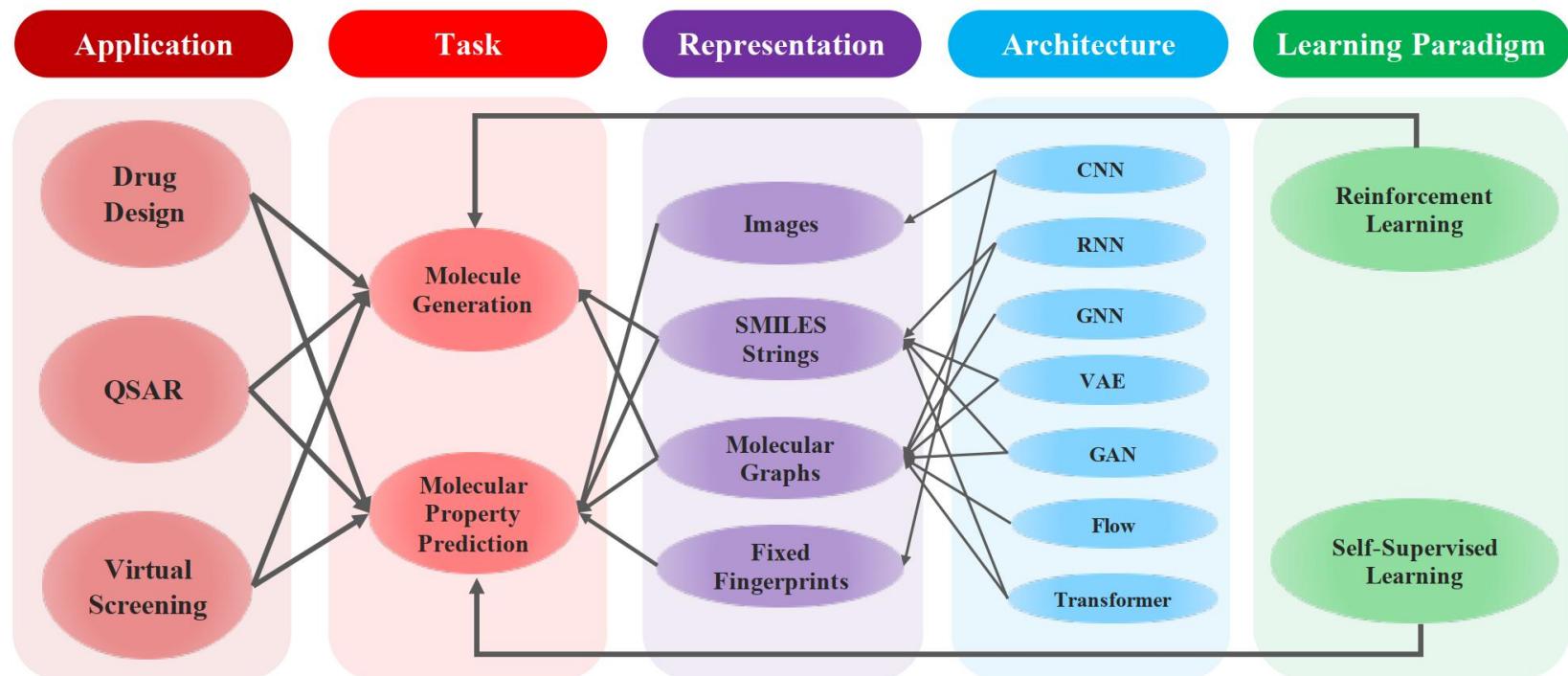
AI recognizes crops (green) and weeds (red), and instructs sprayer nozzles to avoid crops and spray only weeds

# Astronomy

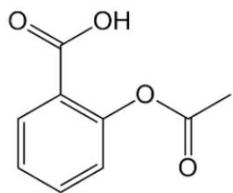


Morpheus helps human astronomers to complete the task of classifying space objects

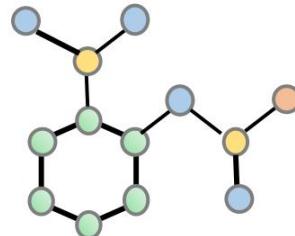
# AI-driven Drug Discovery



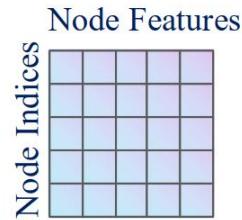
# AI-driven Drug Discovery



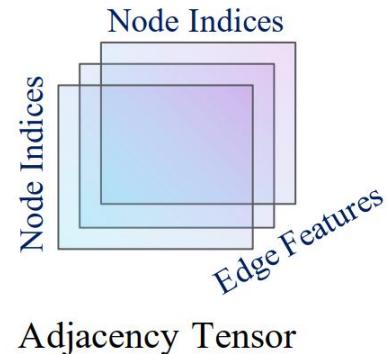
**A. Kekulé Diagram**



**C. Molecular Graph**

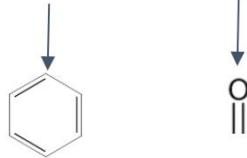


Node Feature Matrix



Adjacency Tensor

$[0 \ 0 \ 0 \ 1 \ \dots \ 0 \ 0 \ 0 \ 0 \ 1]$



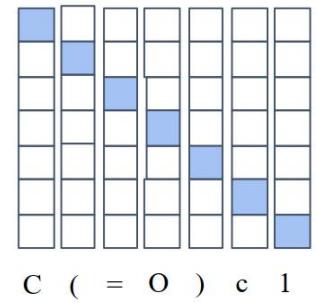
**B. Fingerprints**

CC(=O)Oc1ccccc1C(=O)O

**D. SMILES String**

Tokenization

One-Hot Encoding



Small Molecule Representations

# AlphaFold: a solution to a 50-year-old grand challenge in biology

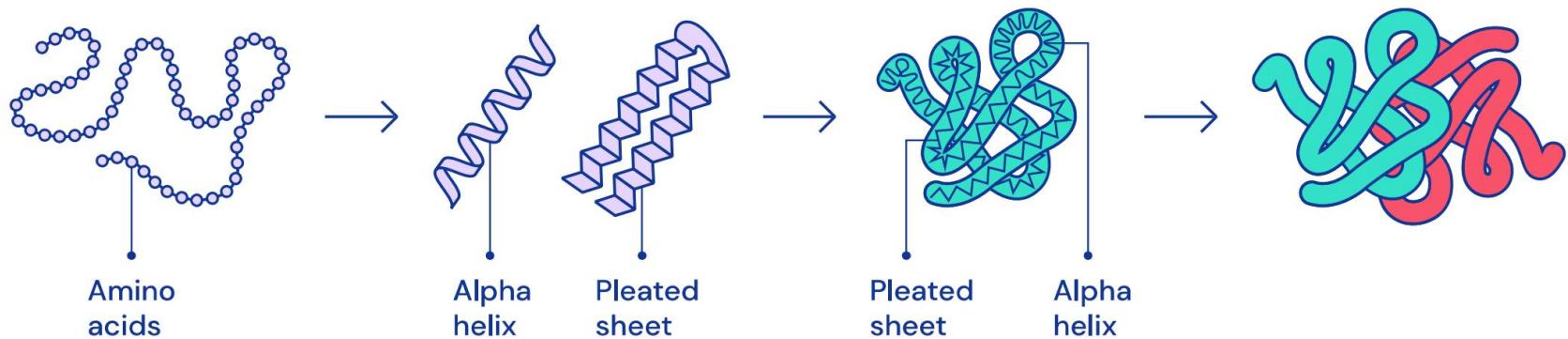
## What is the protein folding problem?

Every protein is made up of a sequence of amino acids bonded together

These amino acids interact locally to form shapes like helices and sheets

These shapes fold up on larger scales to form the full three-dimensional protein structure

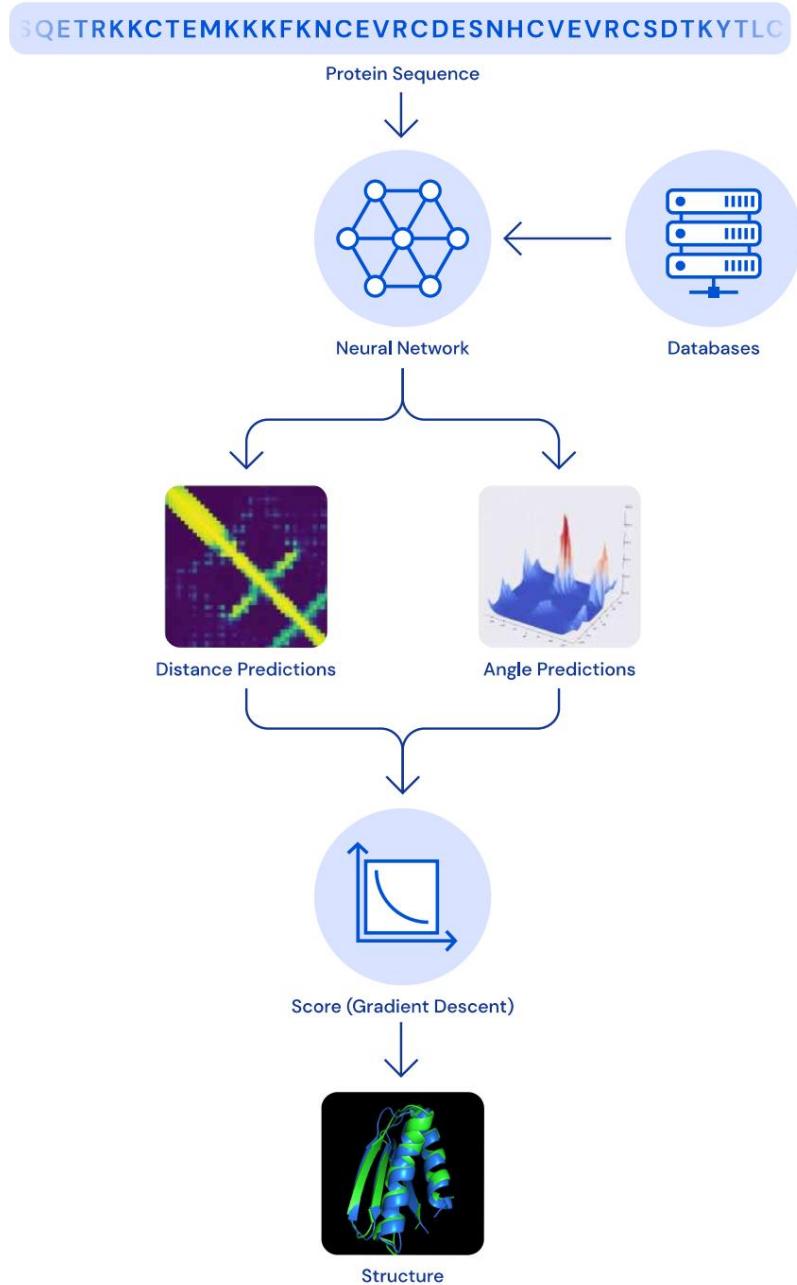
Proteins can interact with other proteins, performing functions such as signalling and transcribing DNA



# AlphaFold

The architecture of the AlphaFold system:

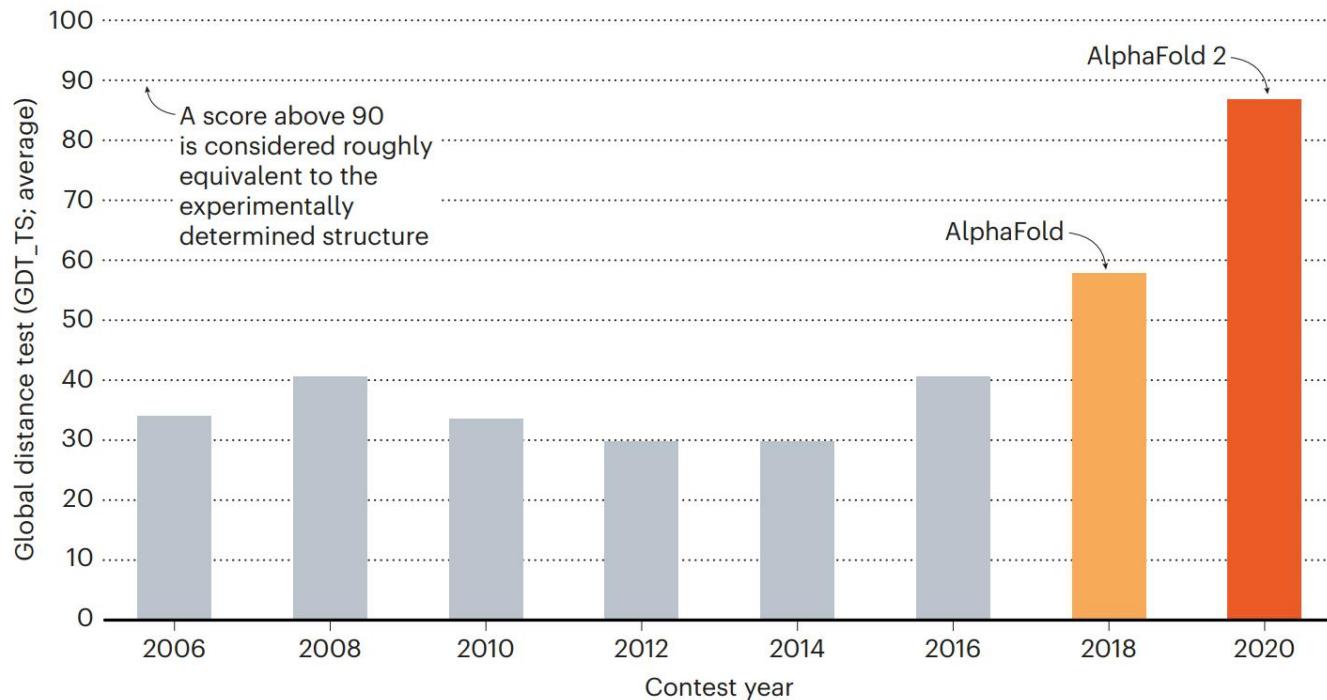
Predicting structure from protein sequence



# AlphaFold: a solution to a 50-year-old grand challenge in biology

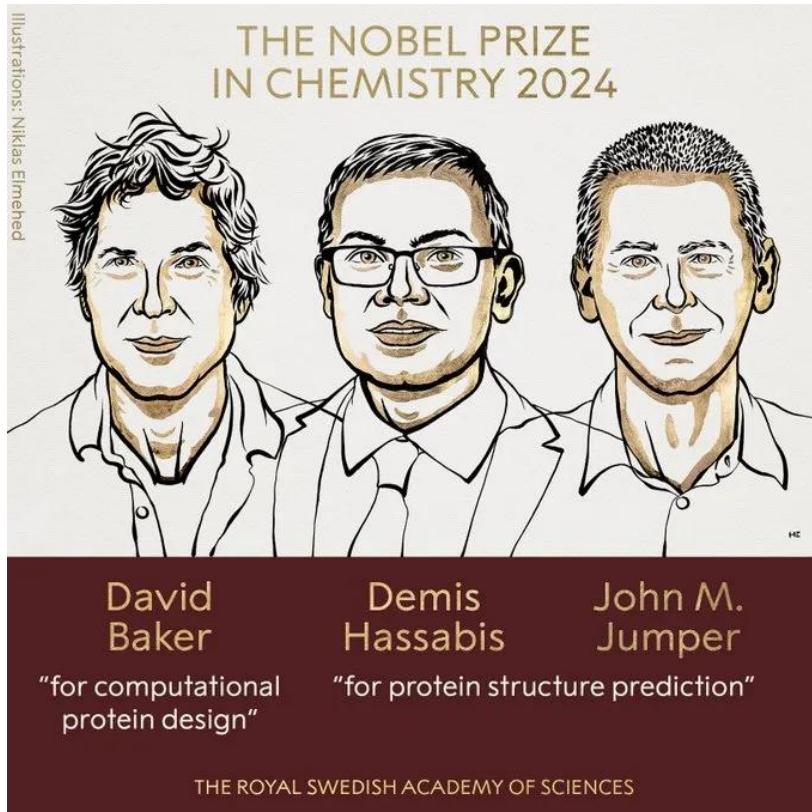
## STRUCTURE SOLVER

DeepMind's AlphaFold 2 algorithm significantly outperformed other teams at the CASP14 protein-folding contest — and its previous version's performance at the last CASP.



DeepMind's AlphaFold2 algorithm **significantly outperformed** other teams at the CASP14 protein-folding contest.

# Nobel Prize for AI4S



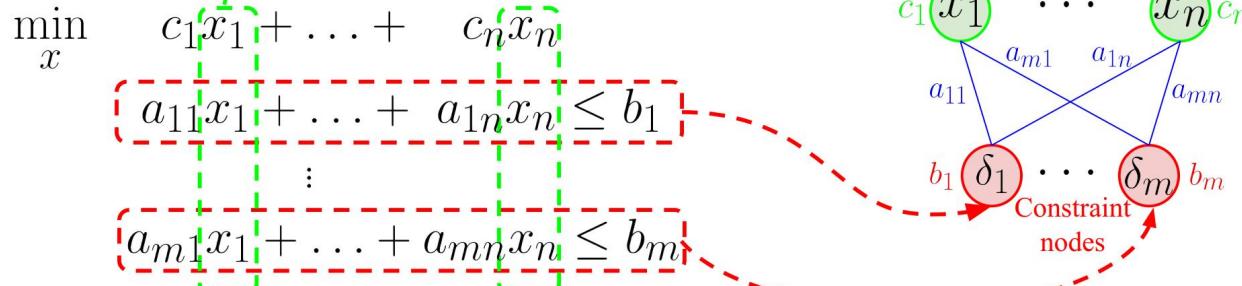
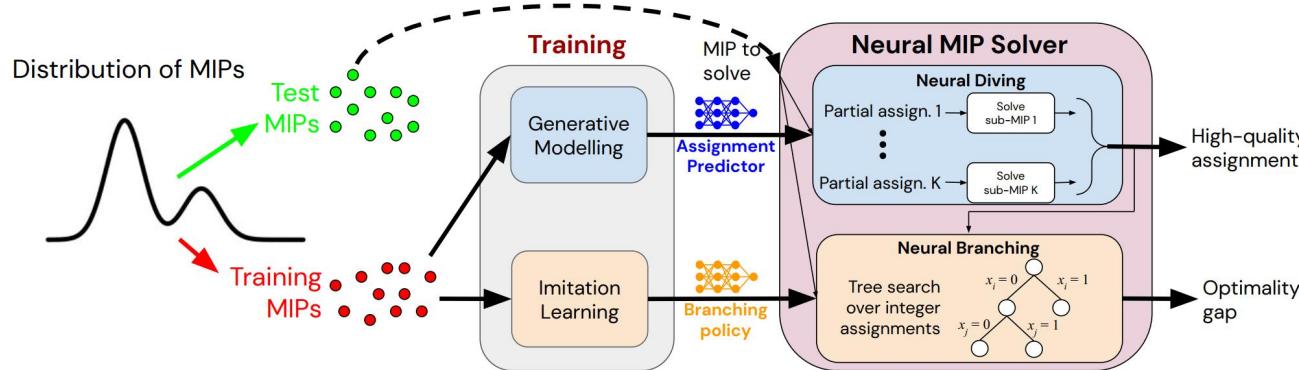
2024年诺贝尔化学奖

David Baker  
“计算蛋白质设计”

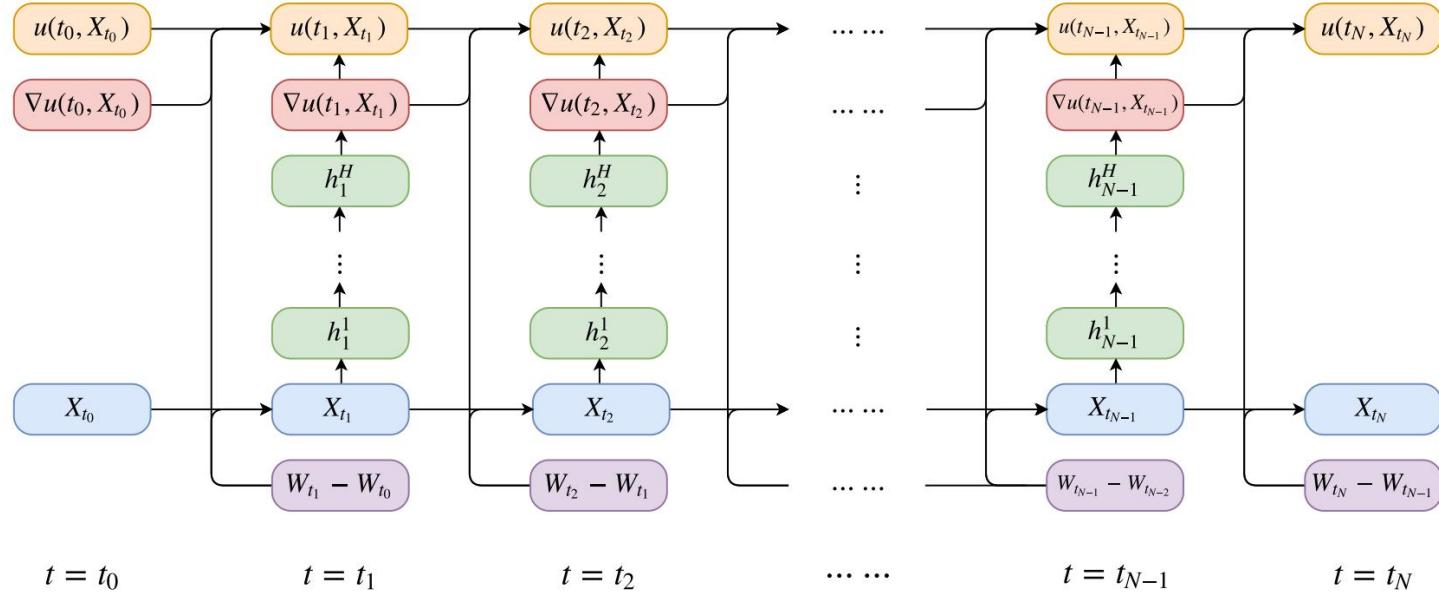
Demis Hassabis 和 John M. Jumper  
“蛋白质结构预测”

# Deep Learning and Combinatorial Optimization

## Solving Mixed Integer Programs Using Neural Networks



# Deep Learning and Numerical Methods

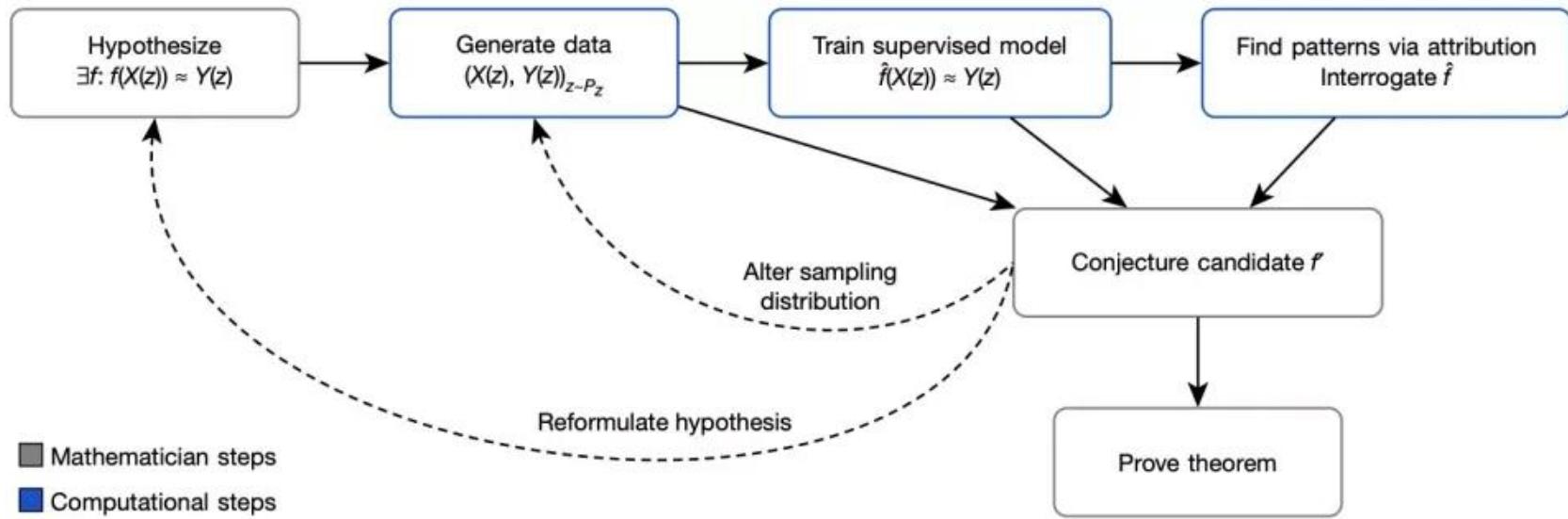


Rough sketch of the architecture of the deep BSDE solver

Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations

Commun. Math. Stat. 5, 349-380 (2017)

# 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

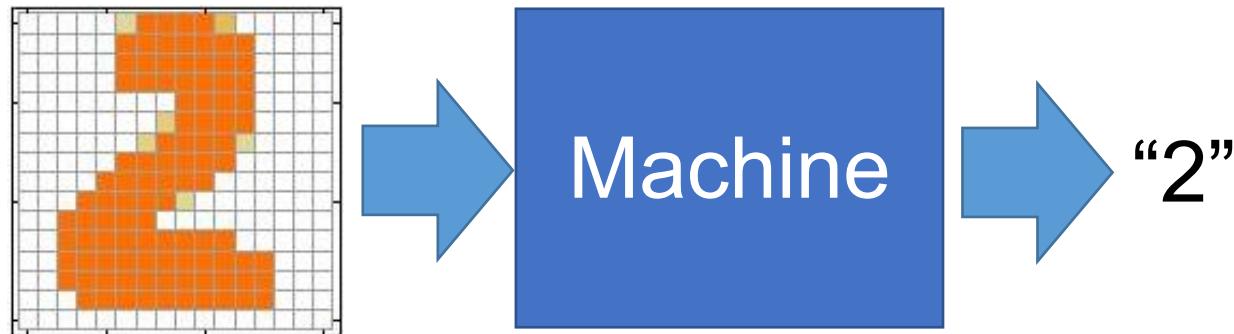
Nature, 600, 70–74 (2021)

# Essentials of Deep Learning

What people already knew in 1980s

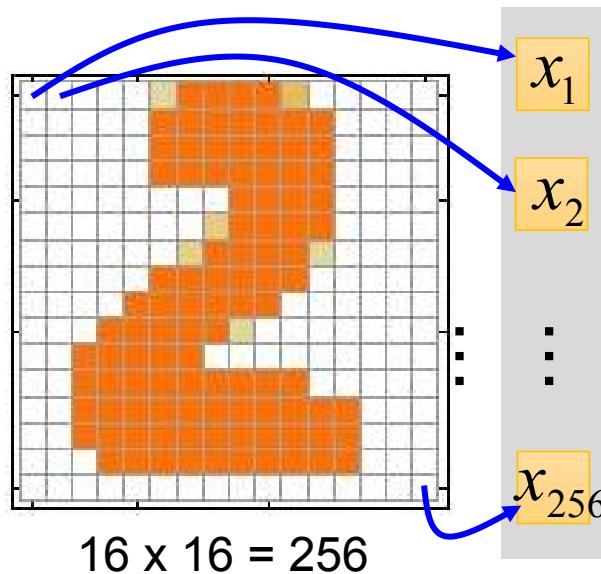
# Example Application

- Handwriting Digit Recognition



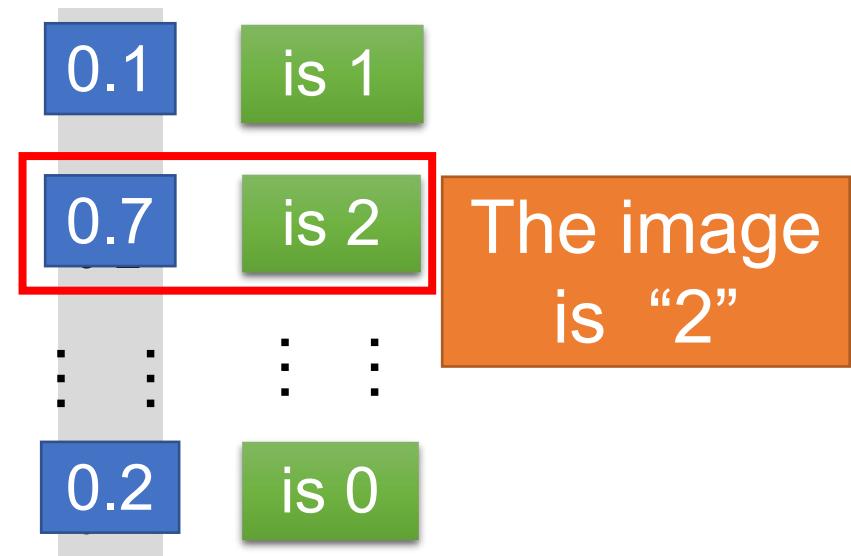
# Handwriting Digit Recognition

Input



Ink  $\rightarrow 1$   
No ink  $\rightarrow 0$

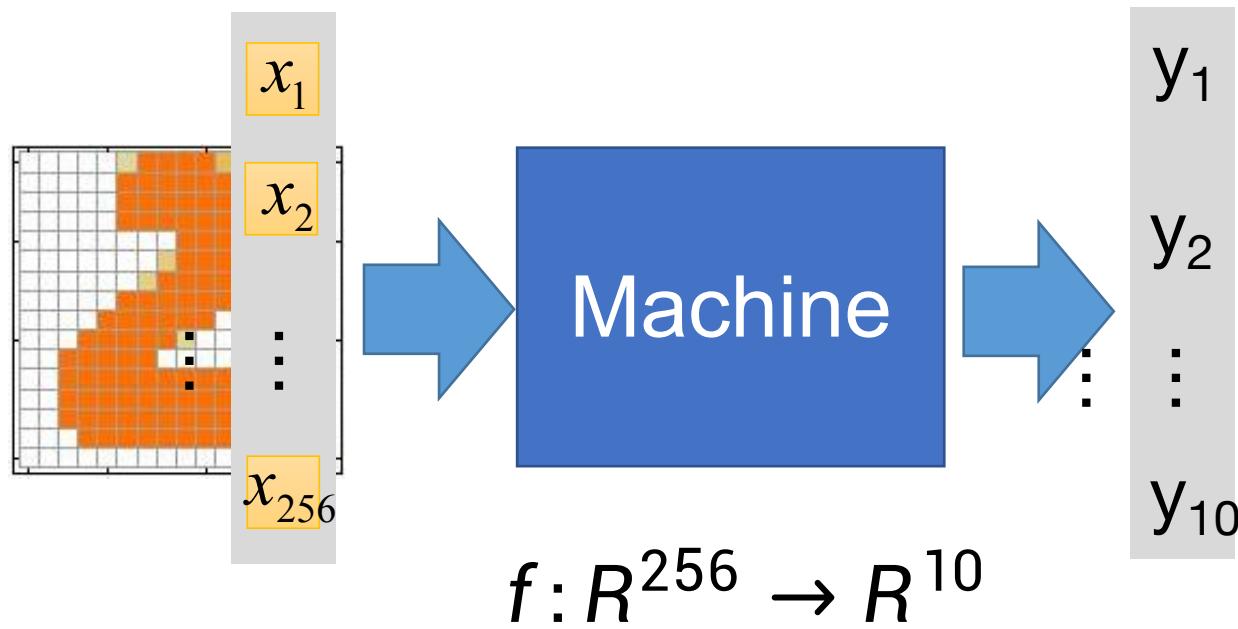
Output



Each dimension represents the confidence of a digit.

# Example Application

- Handwriting Digit Recognition

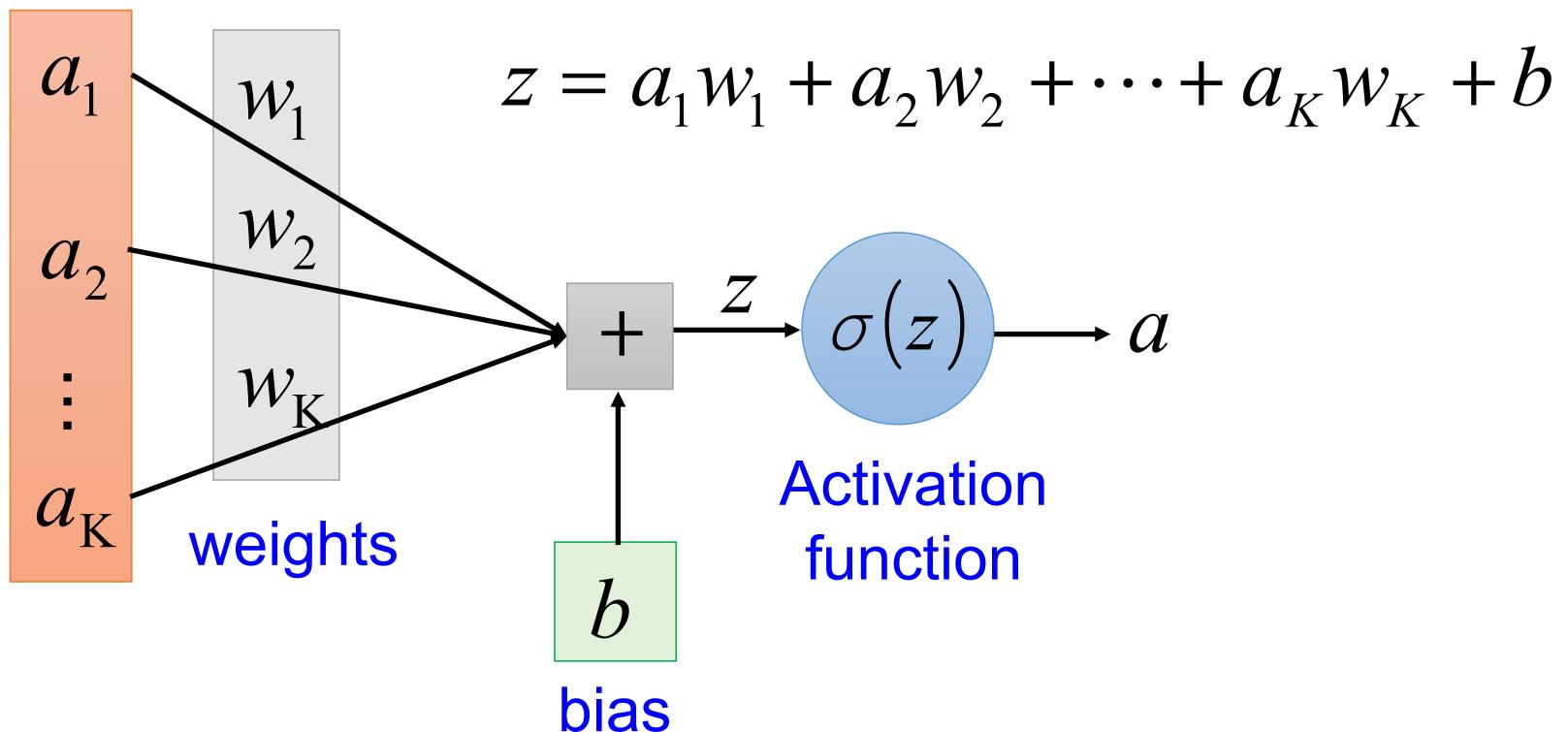


The task of Machine learning is to look for a function  $f$ .

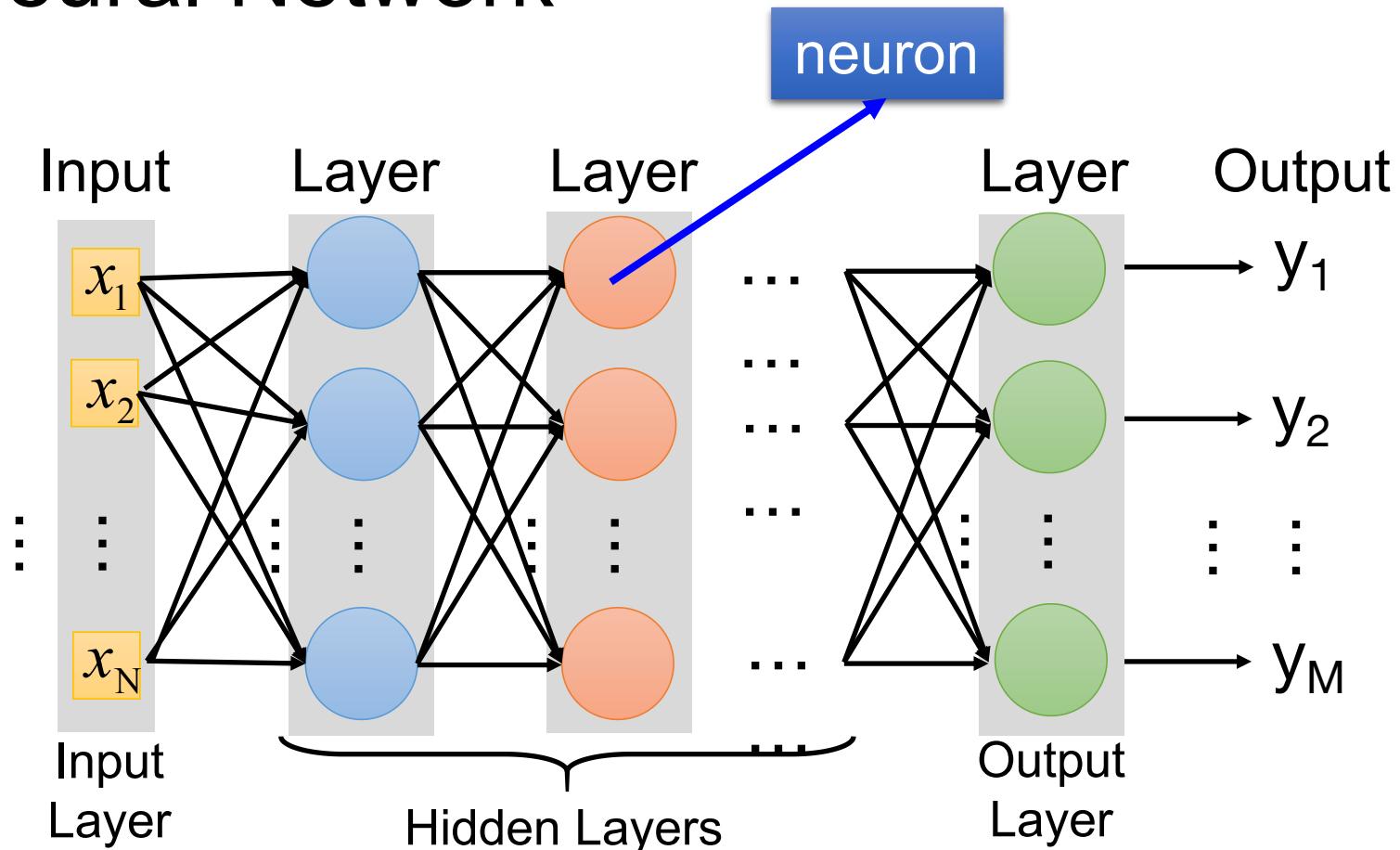
In deep learning, the function  $f$  is represented by neural network

# Element of Neural Network

Neuron  $f: R^K \rightarrow R$



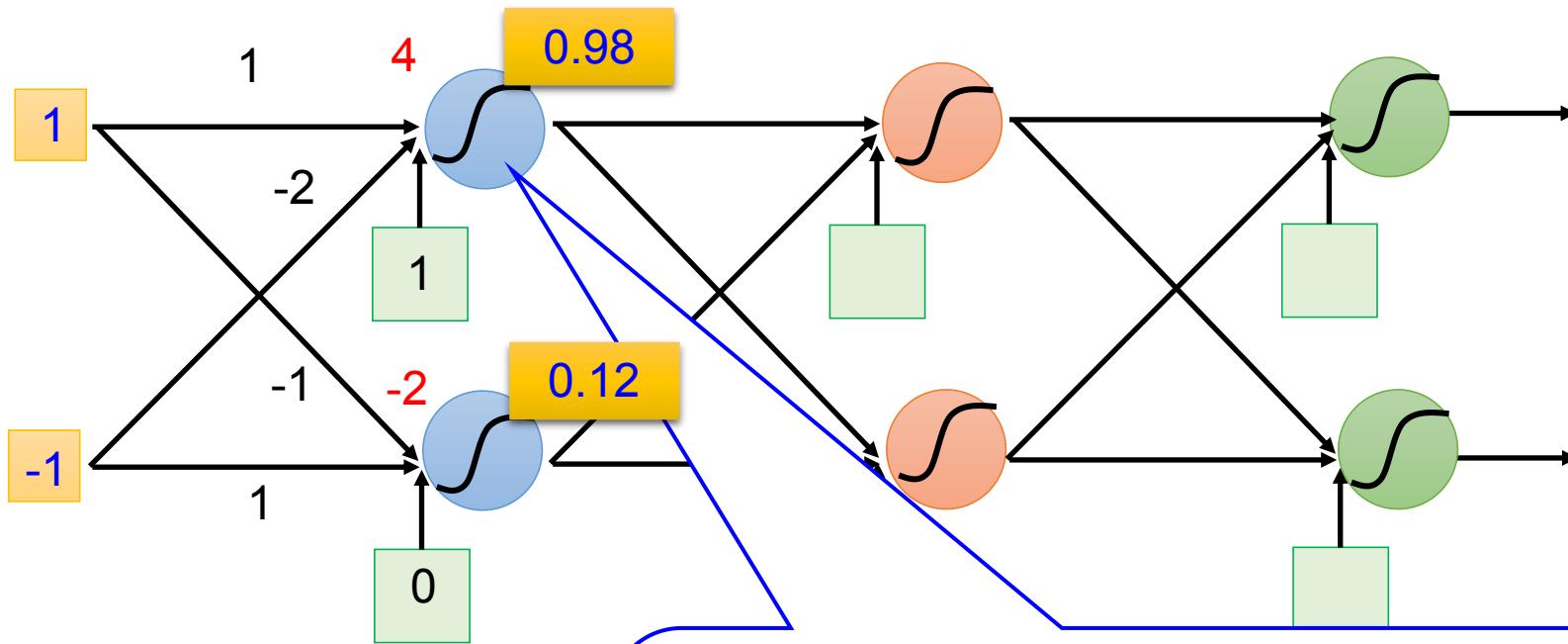
# Neural Network



Deep means many hidden layers

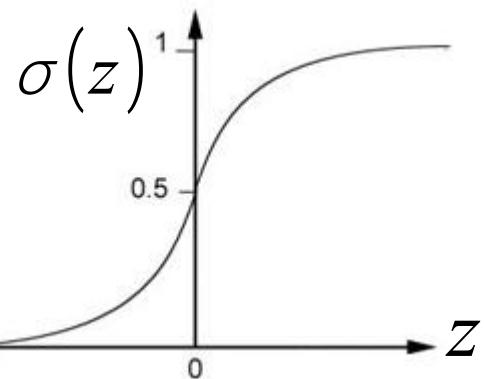
Fully Connected Feedforward Network

# Example of Neural Network

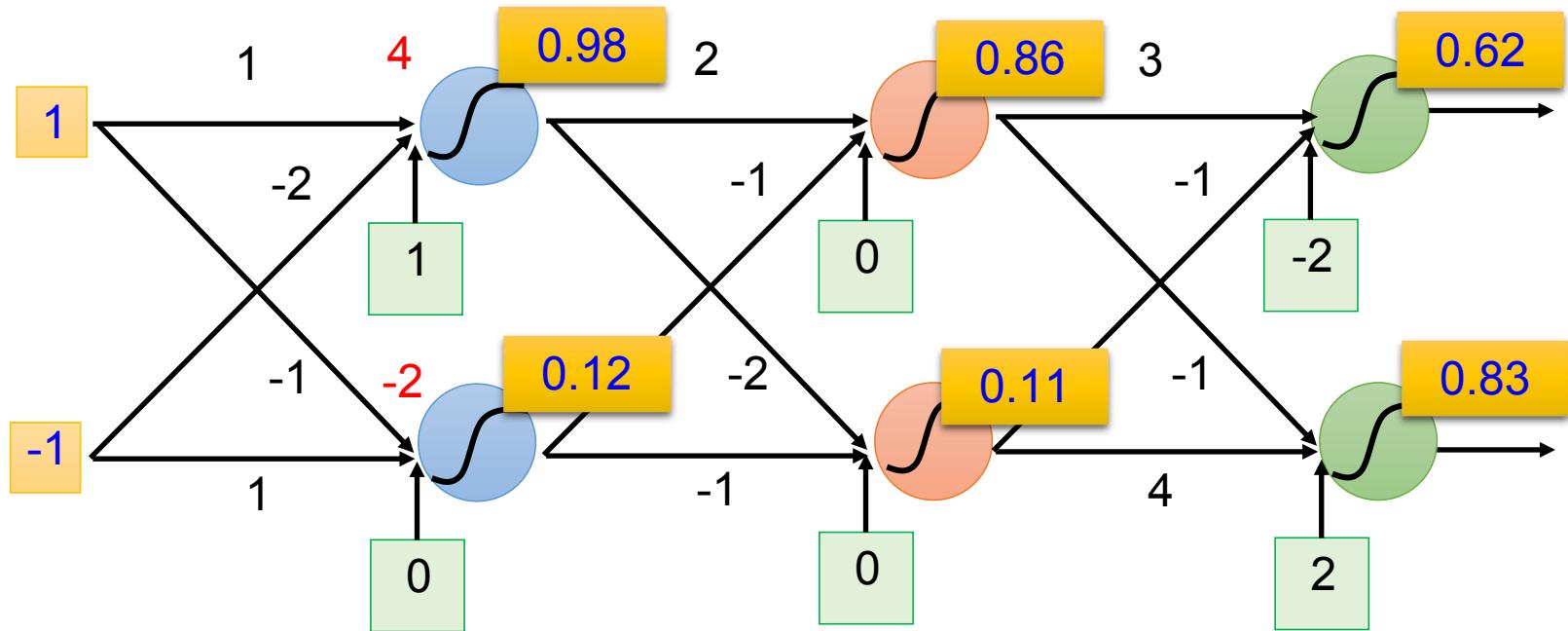


Sigmoid Function

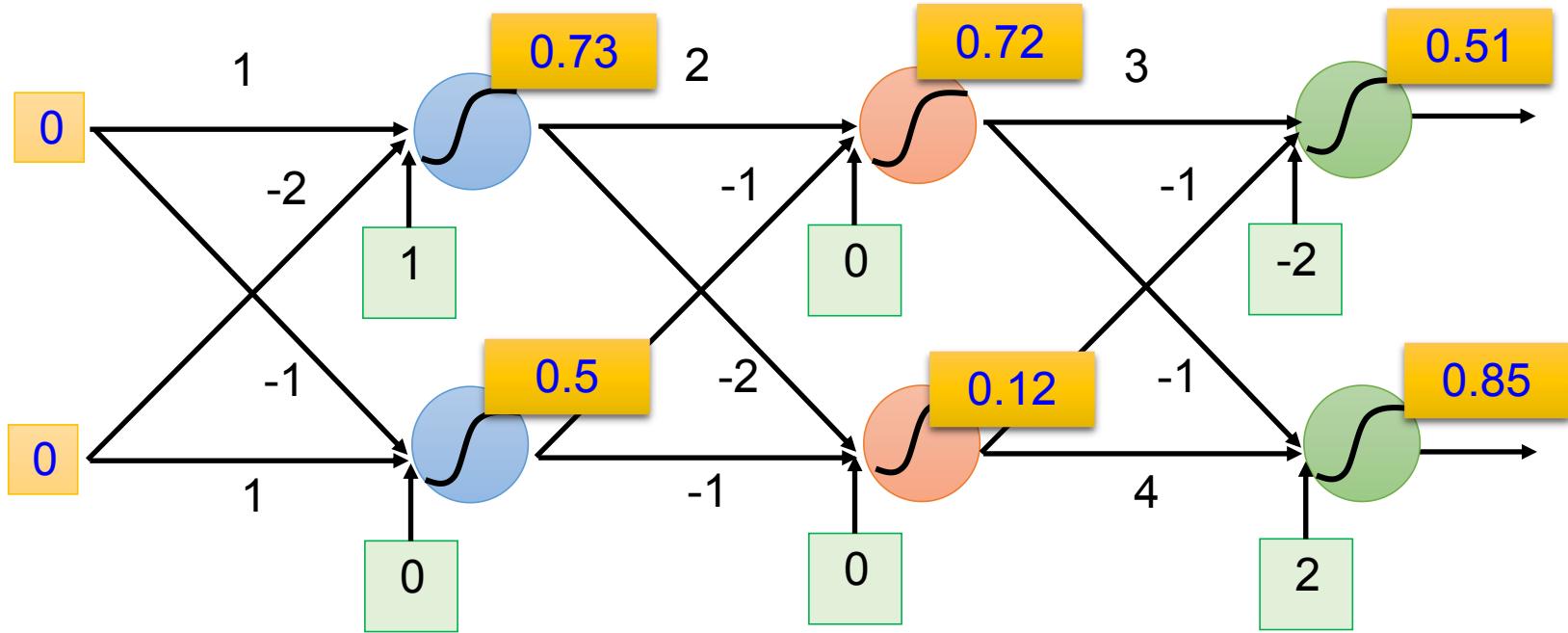
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$



# Example of Neural Network



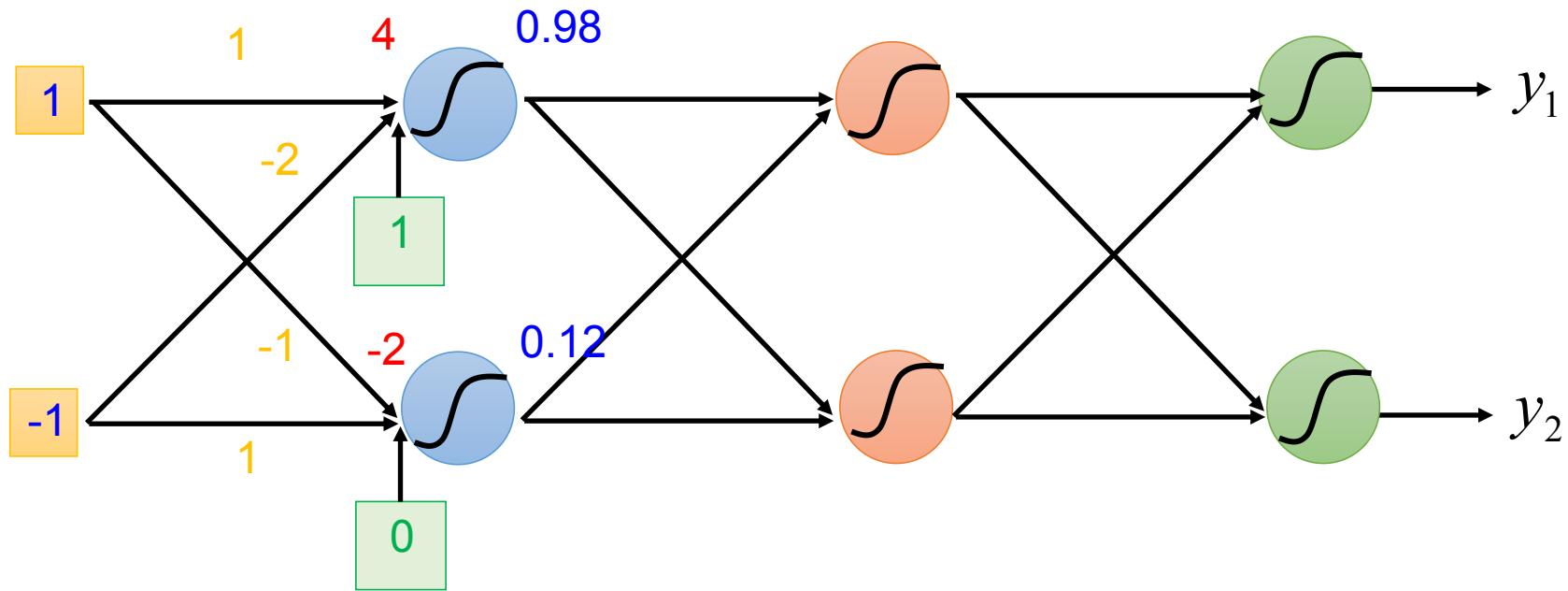
# Example of Neural Network



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

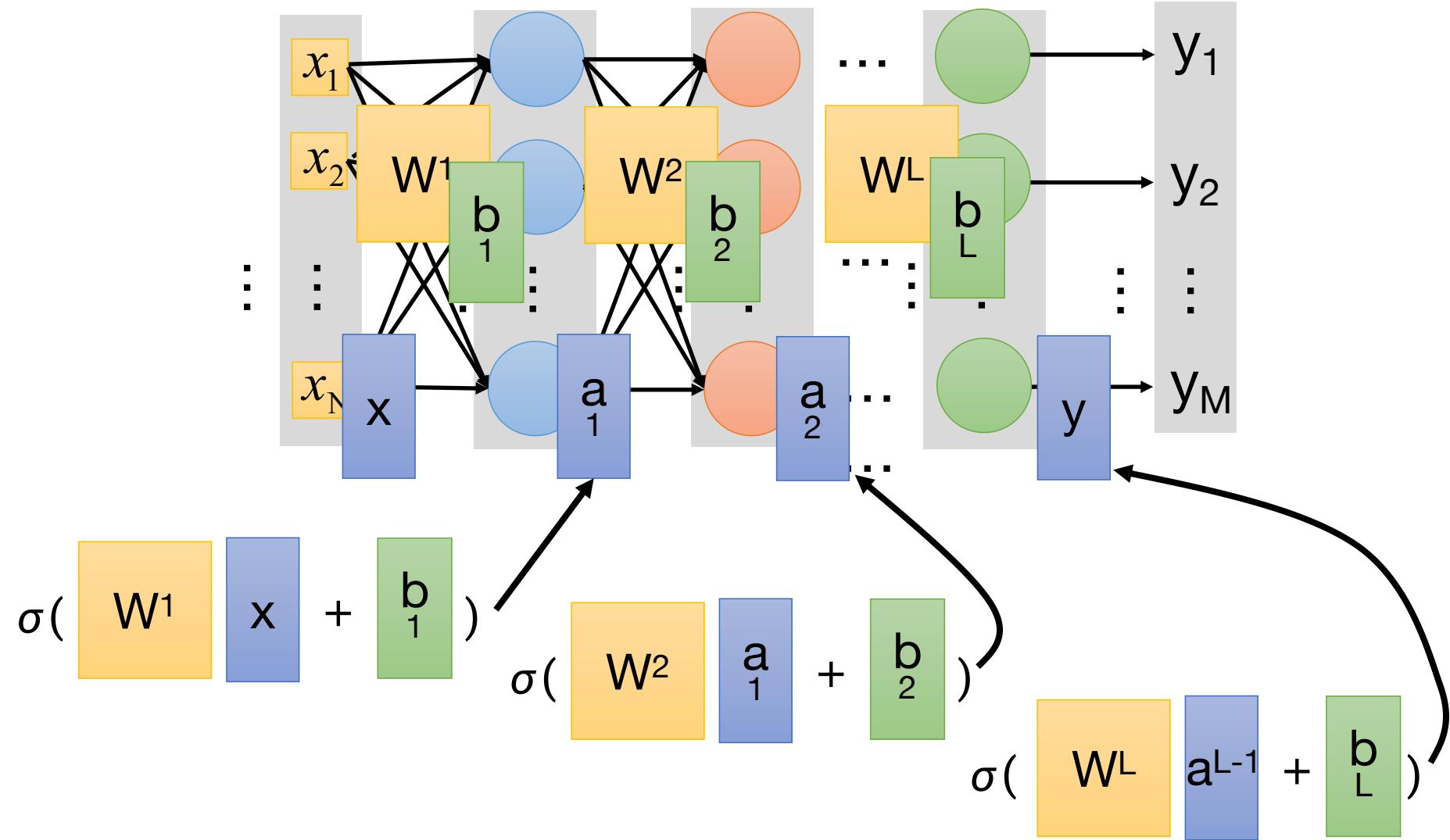
Different parameters define different function

# Matrix Operation

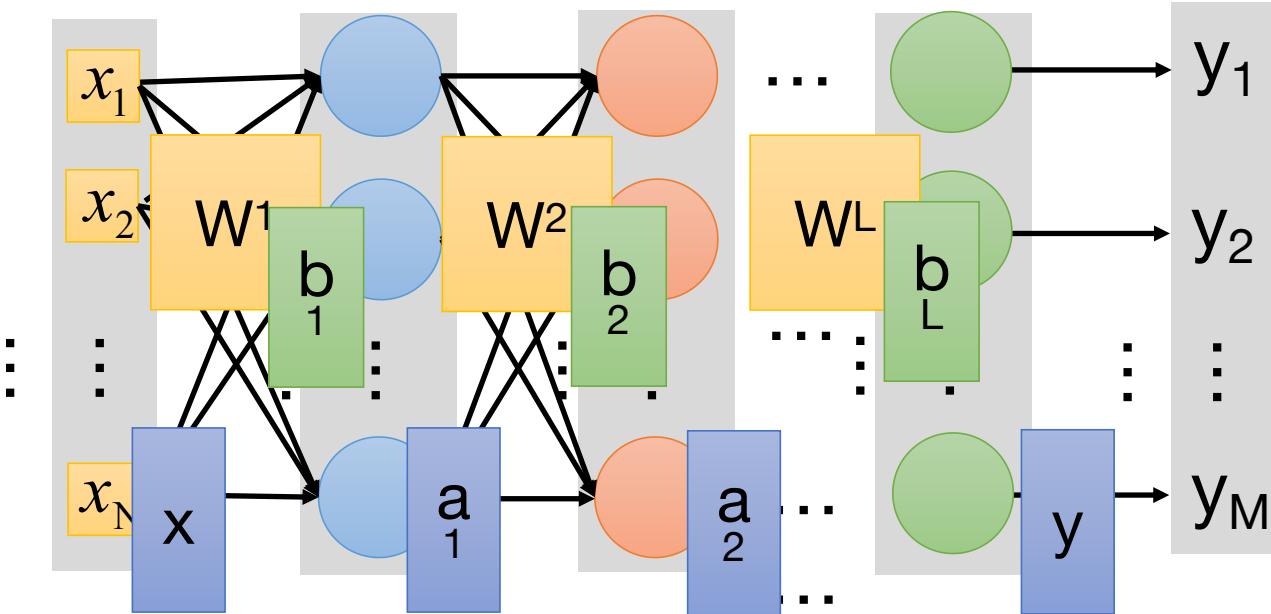


$$\sigma \left( \underbrace{\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\begin{bmatrix} 4 \\ -2 \end{bmatrix}} \right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$

# 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

$$z_1 \rightarrow \sigma \rightarrow y_1 = \sigma(z_1)$$

$$z_2 \rightarrow \sigma \rightarrow y_2 = \sigma(z_2)$$

$$z_3 \rightarrow \sigma \rightarrow y_3 = \sigma(z_3)$$

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

May not be easy to interpret!

# Softmax

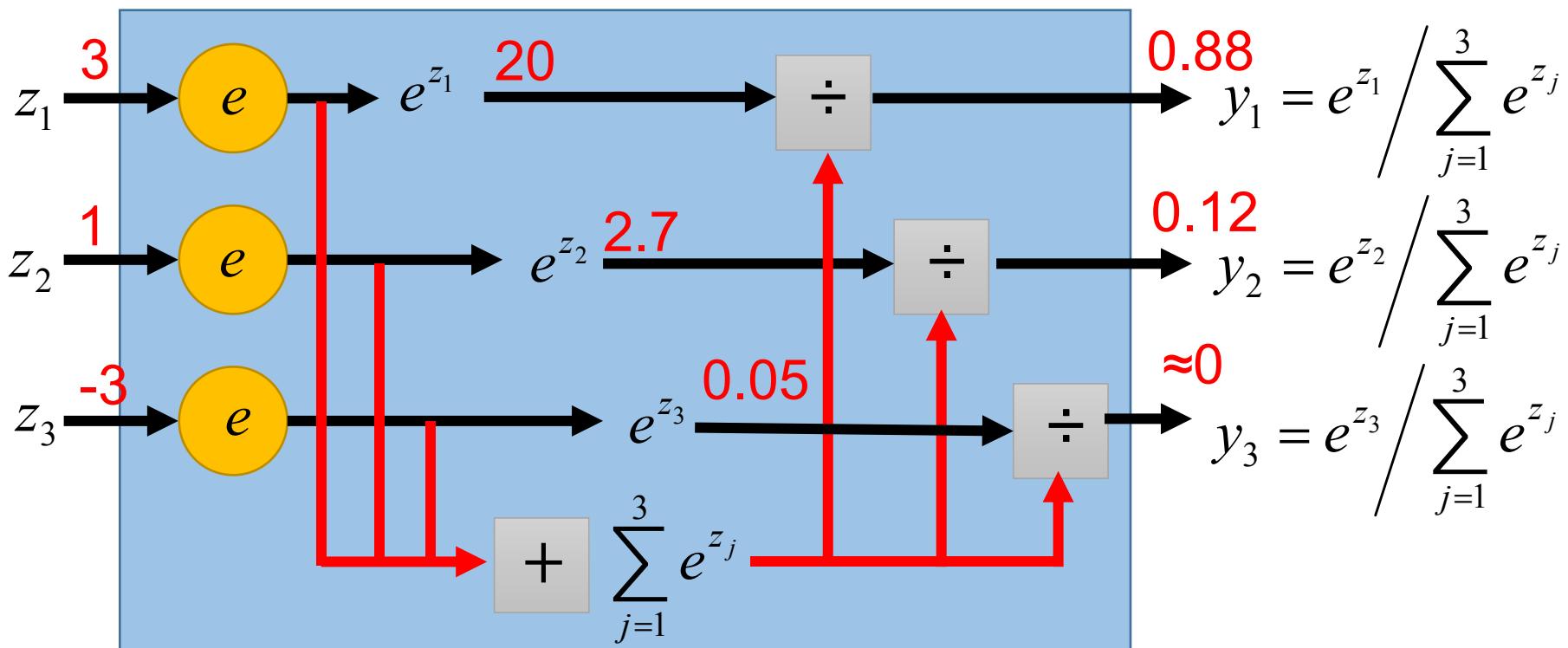
- Softmax layer as the output layer

**Probability:**

■  $1 > y_i > 0$

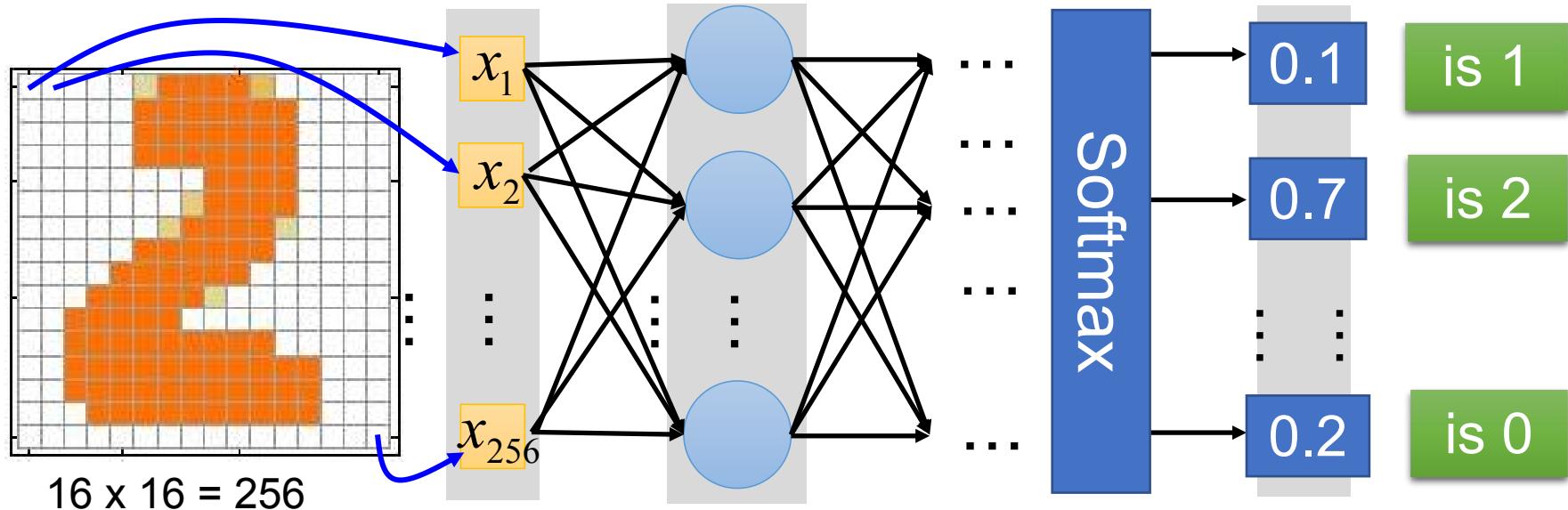
■  $\sum_i y_i = 1$

## Softmax Layer



# How to Set Network Parameters

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



$$16 \times 16 = 256$$

Ink  $\rightarrow 1$

No ink  $\rightarrow$

$$0$$

Set the network parameters  $\theta$  such that .....

Input:   $\rightarrow$   $y_1$  has the maximum value

Input:   $\rightarrow$   $y_2$  has the maximum value

How to let the neural network achieve this

# Training Data

- Preparing training data: images and their labels



“5”



“0”



“4”



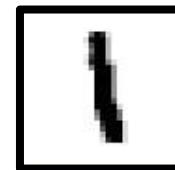
“1”



“9”



“2”



“1”



“3”

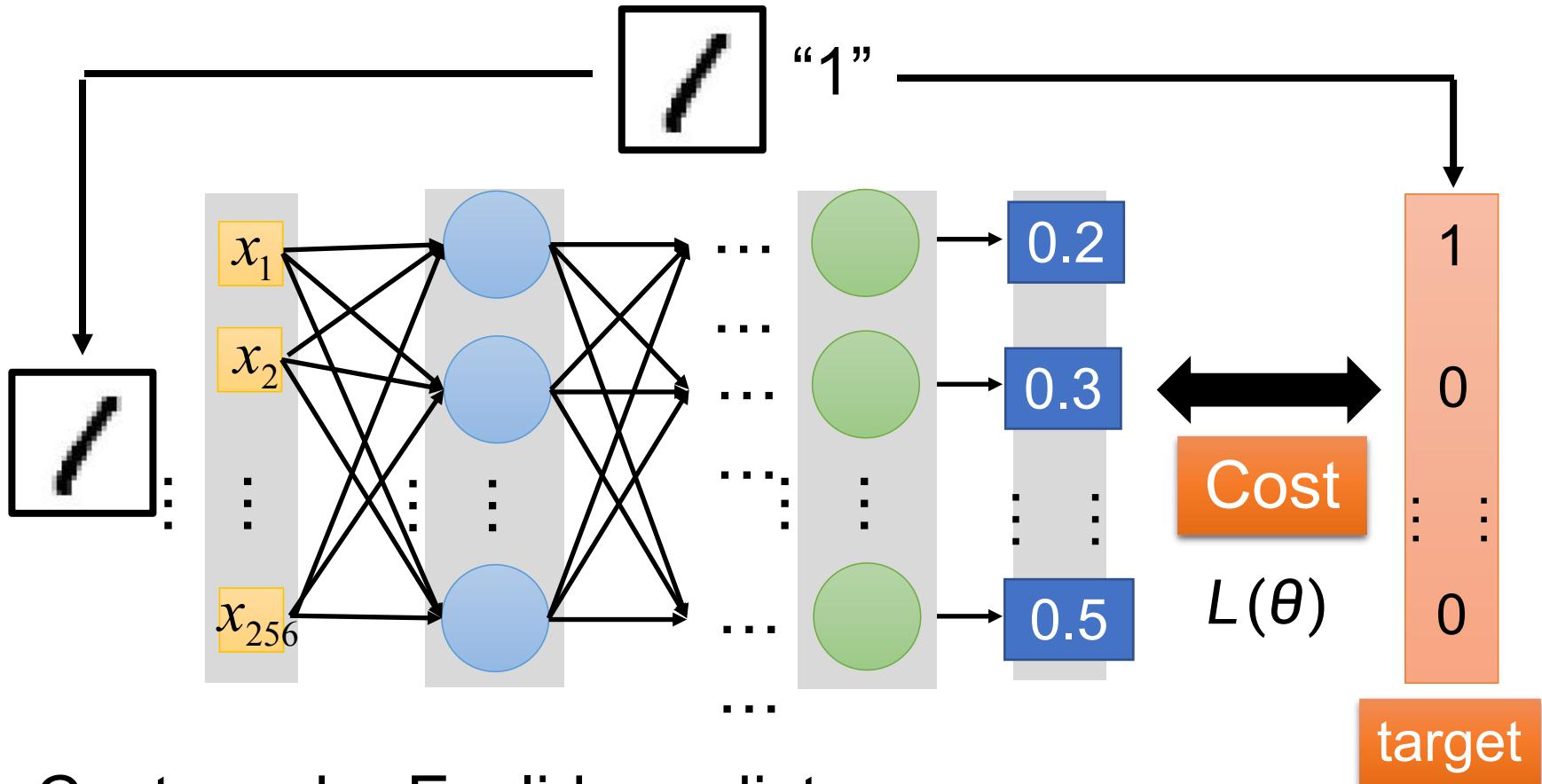
Using the training data to find the network parameters.

# The **MNIST** Database of handwritten digits

- 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.

# Cost

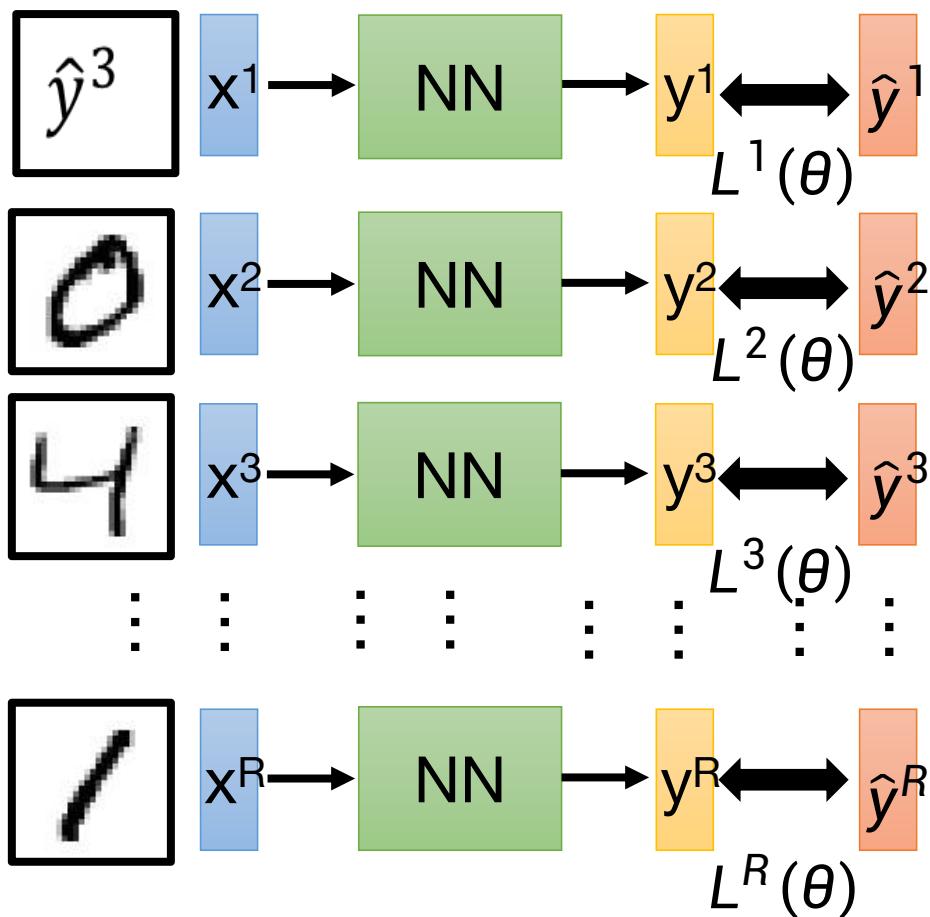
Given a set of network parameters  $\theta$ , each example has a cost value.



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

# Total Cost

For all training data ...



Total Cost:

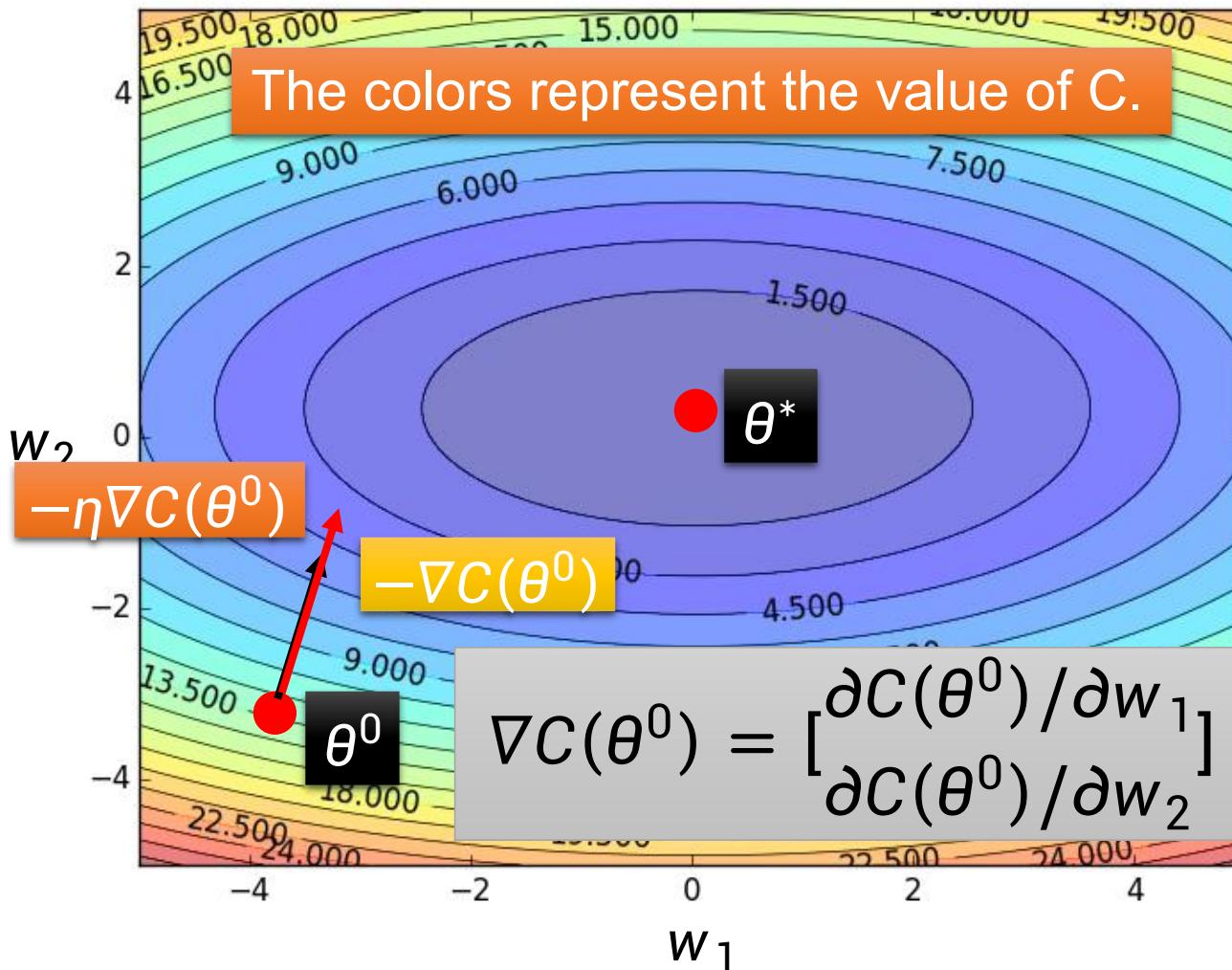
$$C(\theta) = \sum_{r=1}^R L^r(\theta)$$

How bad the network parameters  $\theta$  is on this task?

Find the network parameters  $\theta^*$  that minimize this value

# Gradient Descent

## Error Surface



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

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

Randomly pick a starting point  $\theta^0$

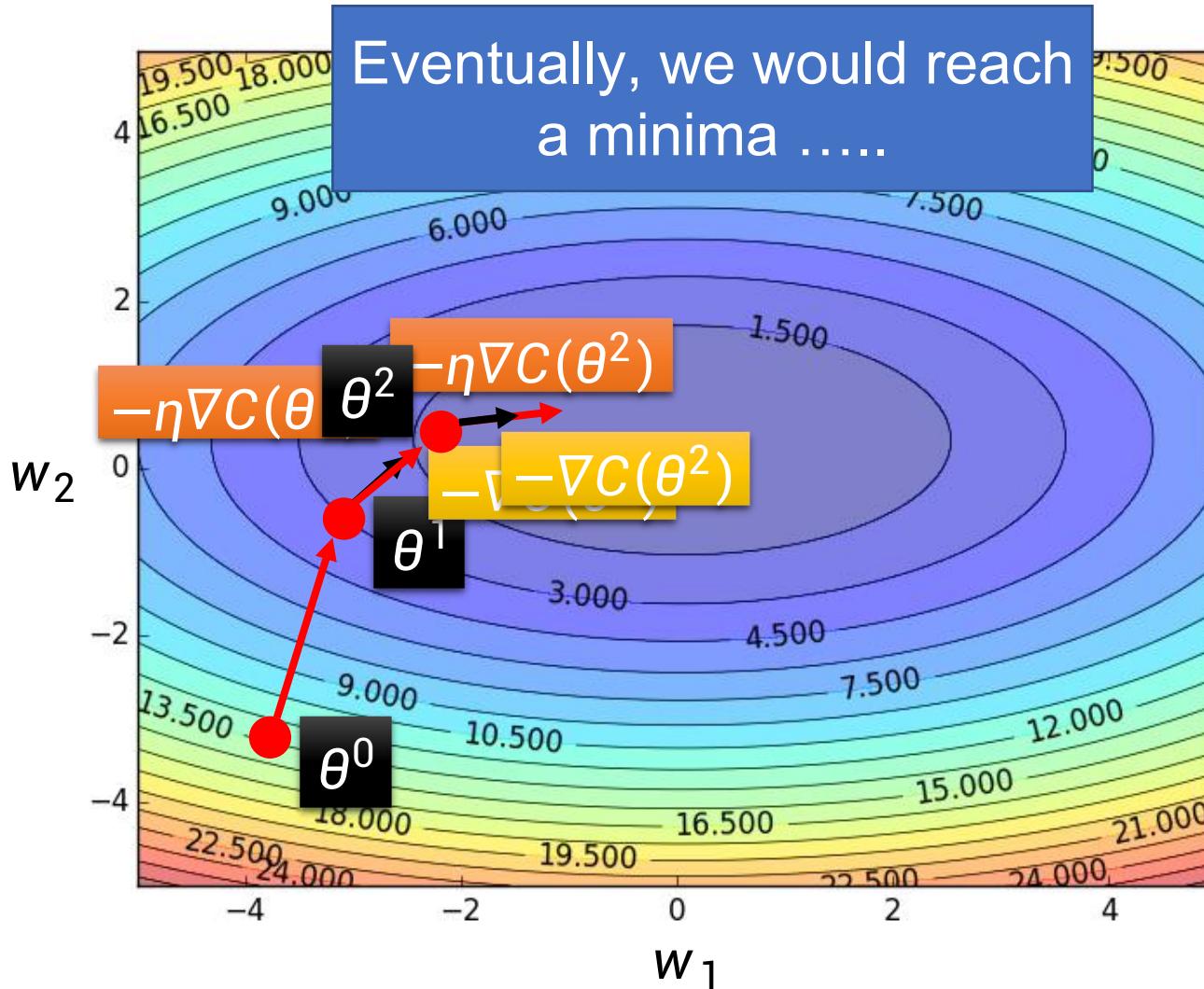
Compute the negative gradient at  $\theta^0$

$$\rightarrow -\nabla C(\theta^0)$$

Times the learning rate  $\eta$

$$\rightarrow -\eta \nabla C(\theta^0)$$

# Gradient Descent



Randomly pick a starting point  $\theta^0$

Compute the negative gradient at  $\theta^0$

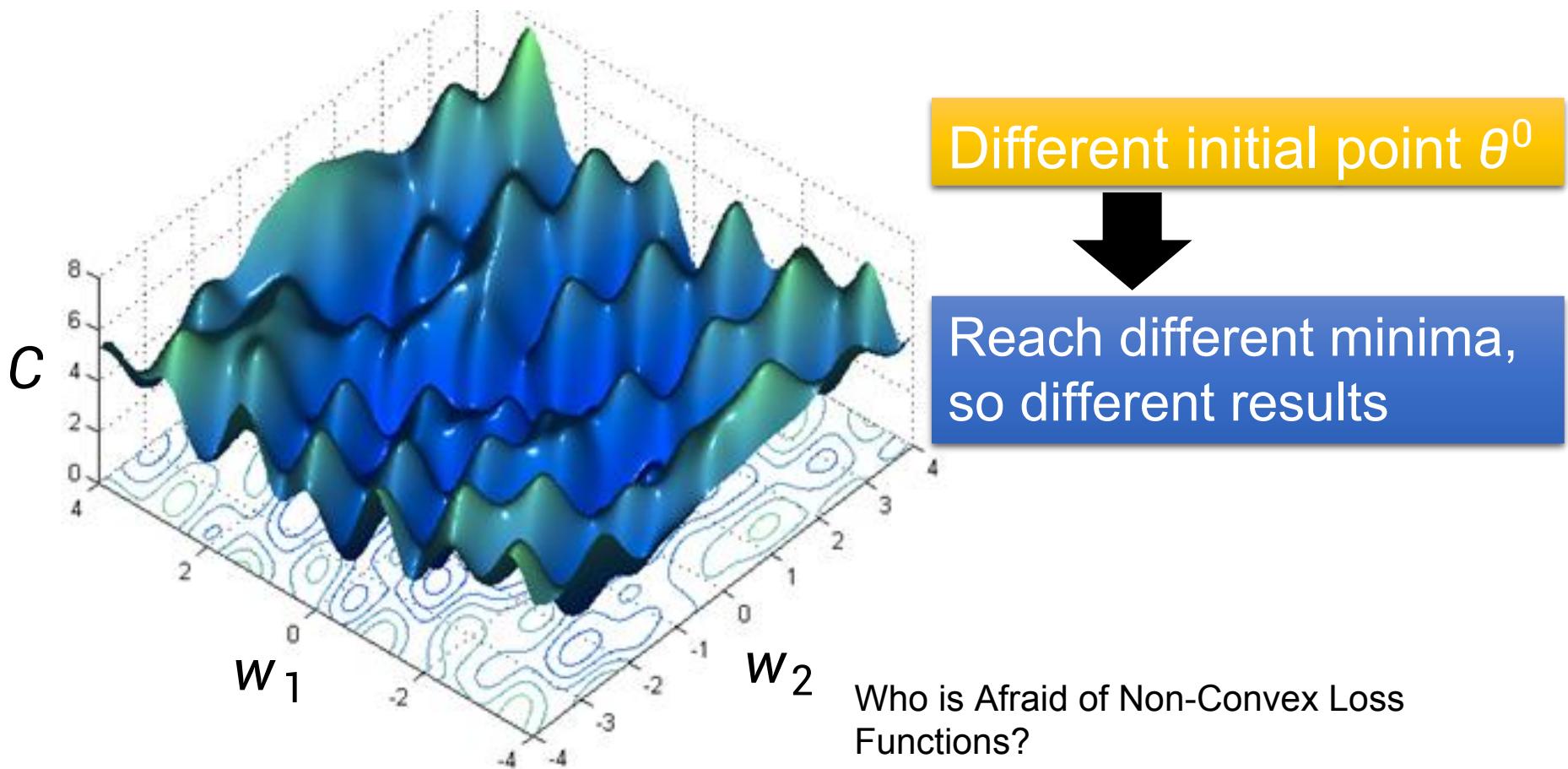
$\rightarrow -\nabla C(\theta^0)$

Times the learning rate  $\eta$

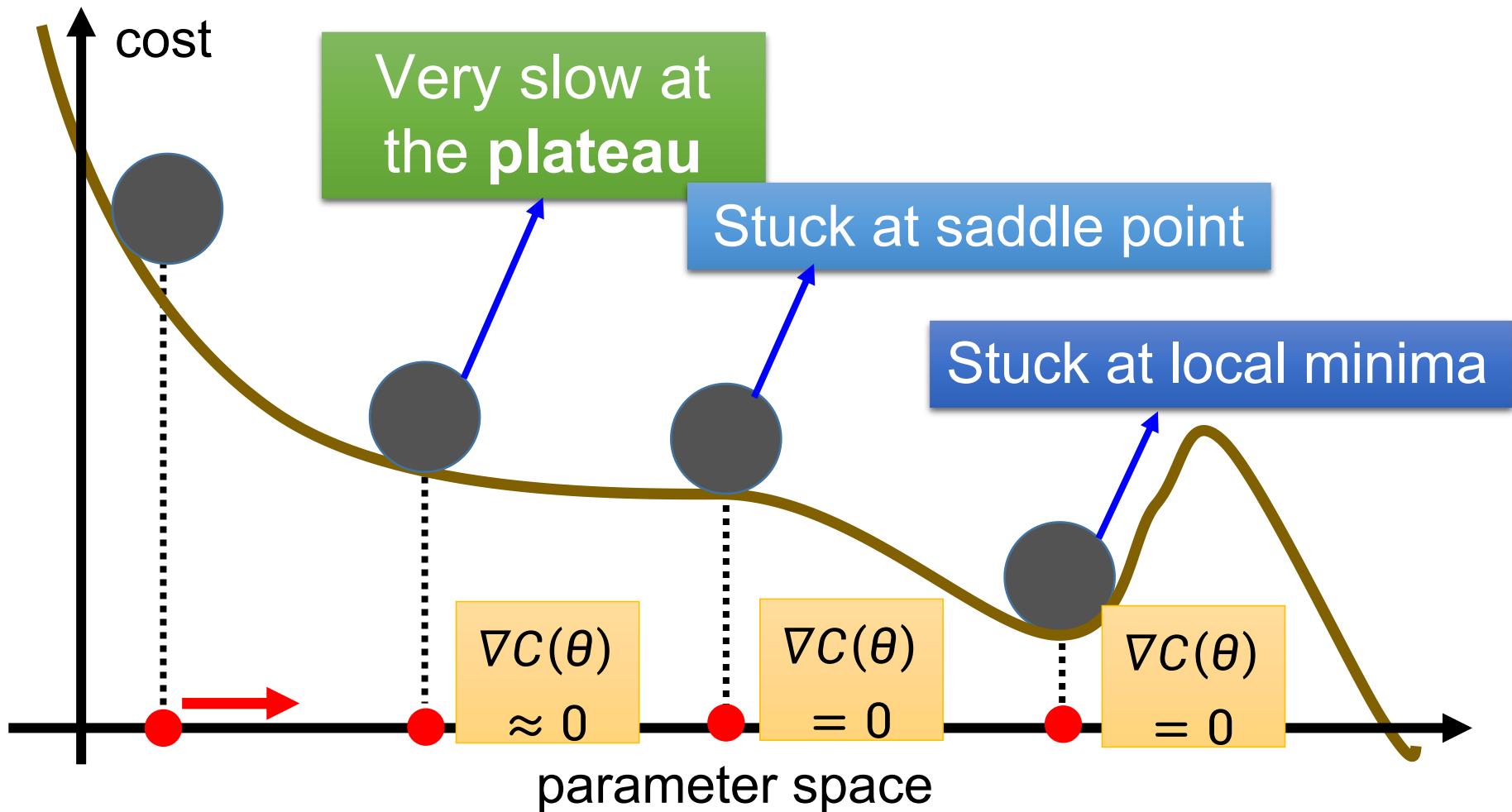
$\rightarrow -\eta \nabla C(\theta^0)$

# Local Minima

- Gradient descent never guarantee global minima

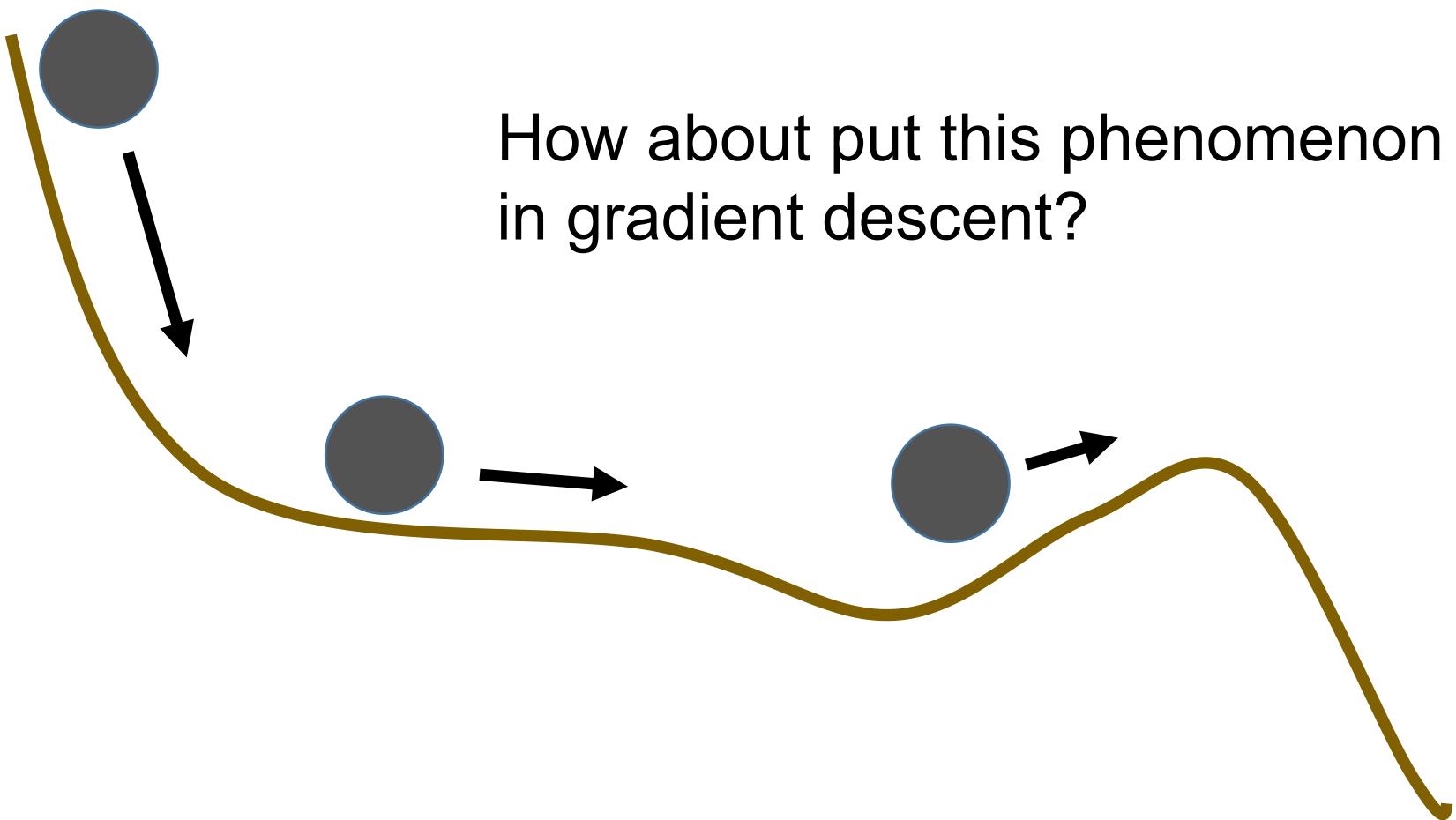


# Besides Local Minima .....



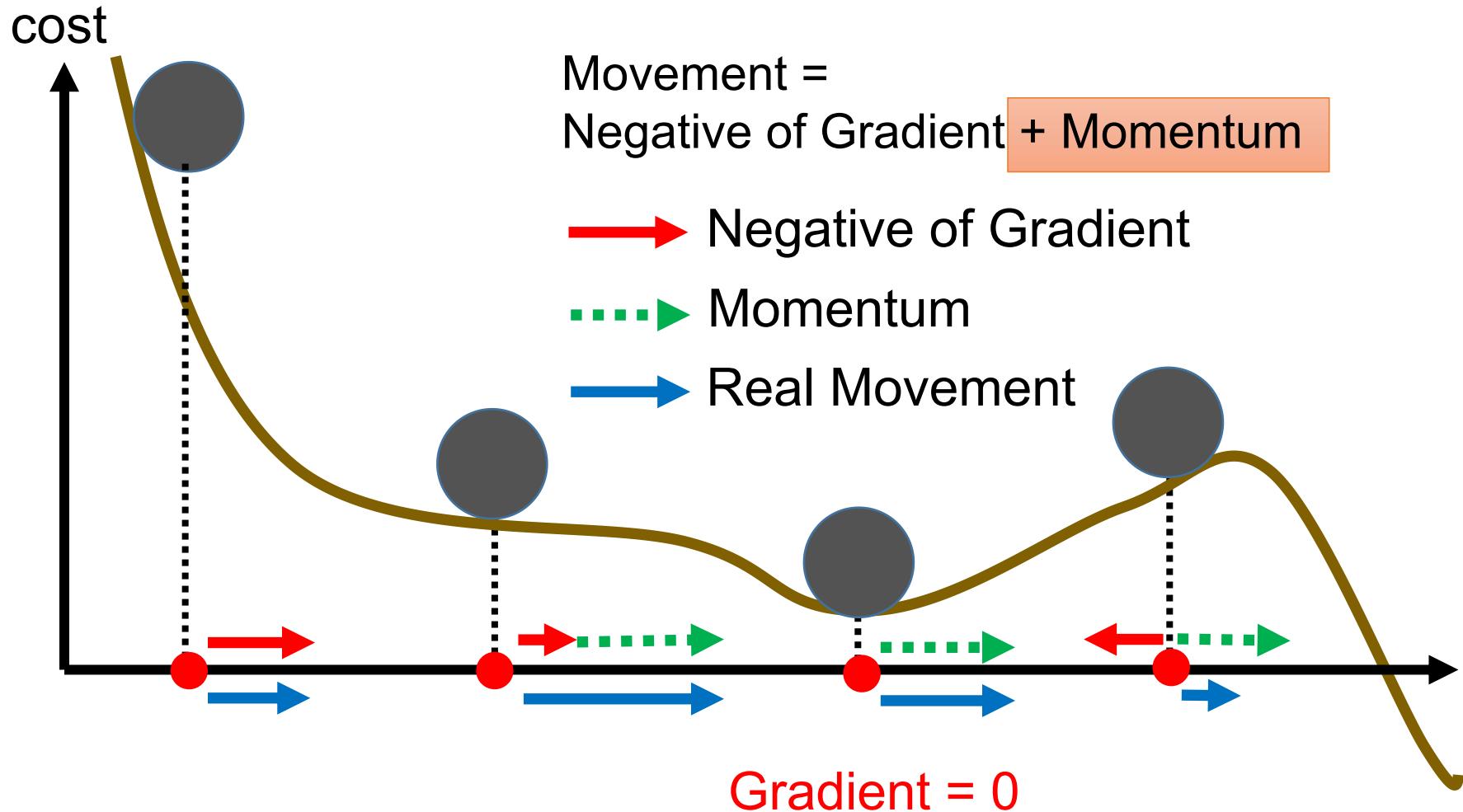
# In Physical World .....

- Momentum



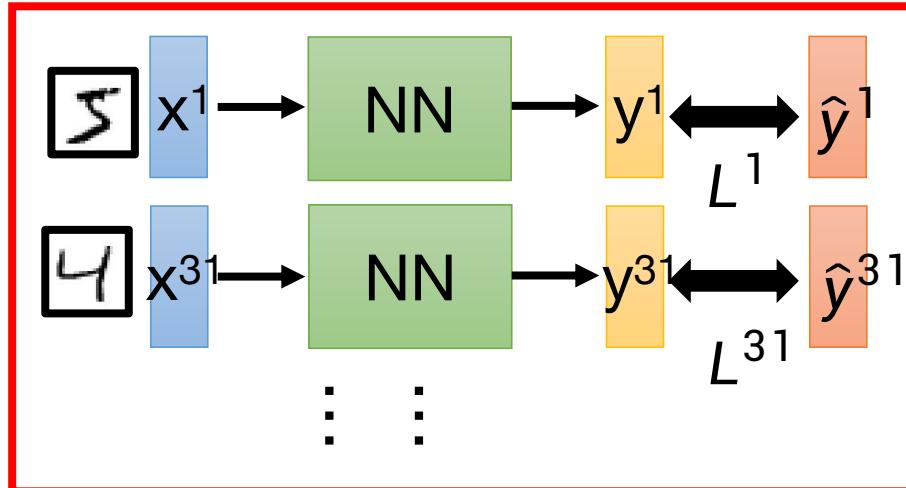
# Momentum

Still not guarantee reaching global minima, but give some hope .....

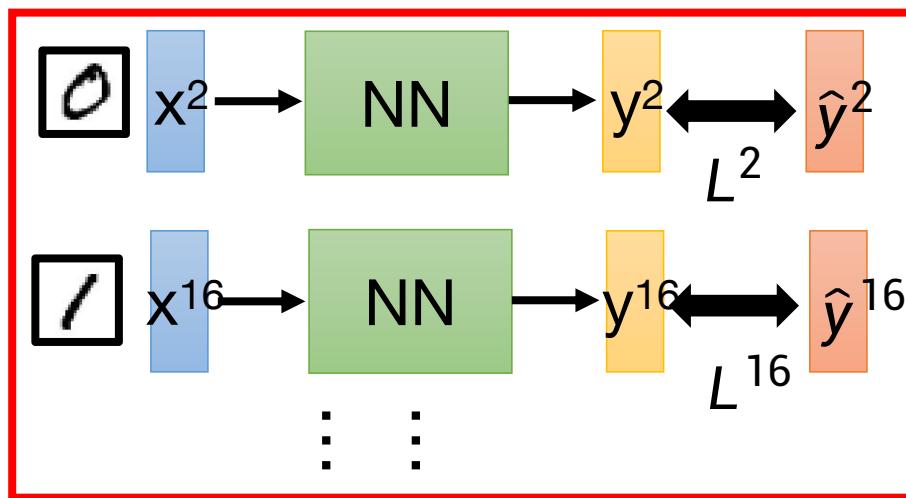


# Mini-batch

Mini-batch



Mini-batch

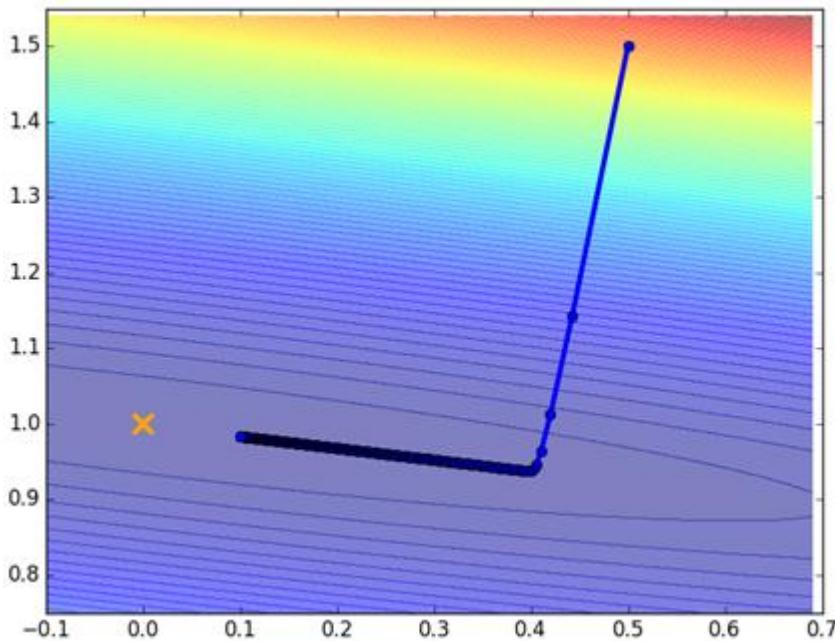


- Randomly initialize  $\theta^0$
- Pick the 1<sup>st</sup> batch
- $$C = L^1 + L^{31} + \dots$$
- $$\theta^1 \leftarrow \theta^0 - \eta \nabla C(\theta^0)$$
- Pick the 2<sup>nd</sup> batch
- $$C = L^2 + L^{16} + \dots$$
- $$\theta^2 \leftarrow \theta^1 - \eta \nabla C(\theta^1)$$
  
⋮

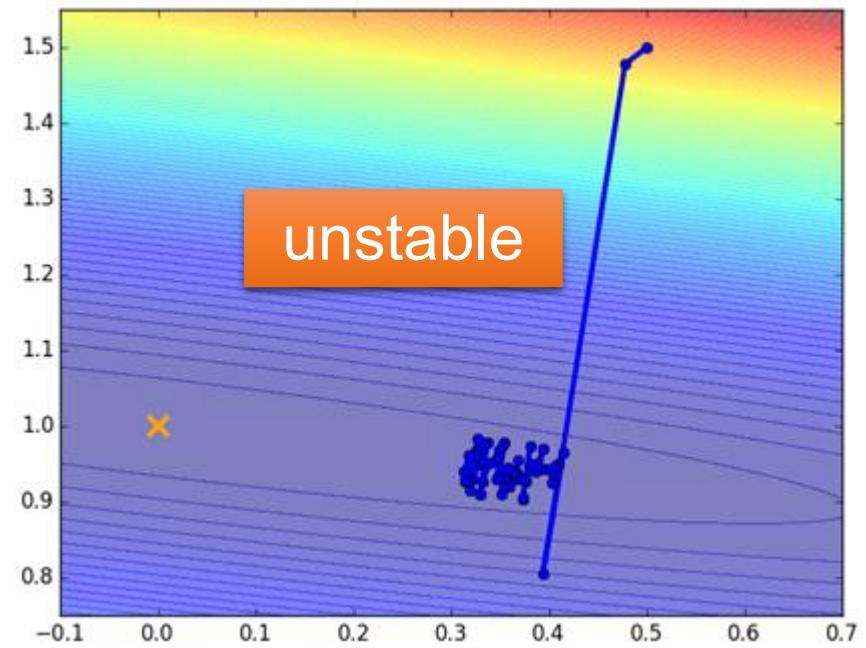
C is different each time when we update parameters!

# Mini-batch

Original Gradient Descent



With Mini-batch



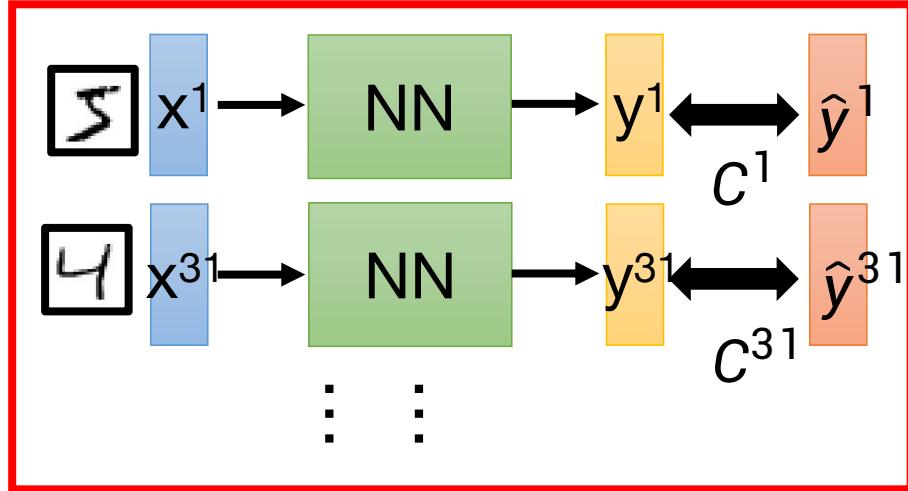
The colors represent the total C on all training data.

# Mini-batch

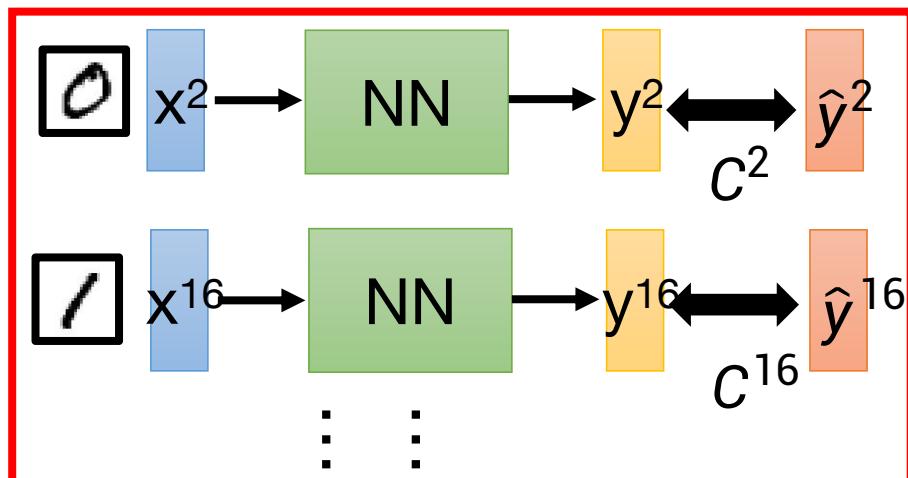
Faster

Better!

Mini-batch



Mini-batch



➤ Randomly initialize  $\theta^0$

➤ Pick the 1<sup>st</sup> batch

$$C = C^1 + C^{31} + \dots$$

$$\theta^1 \leftarrow \theta^0 - \eta \nabla C(\theta^0)$$

➤ Pick the 2<sup>nd</sup> batch

$$C = C^2 + C^{16} + \dots$$

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

⋮

➤ Until all mini-batches have been picked

one epoch

Repeat the above process

# 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](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

theano



[http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS\\_2015\\_2/Lecture/Theano%20DNN.ecm.mp4/index.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/Theano%20DNN.ecm.mp4/index.html)

# Conclusion

## Three Steps for Deep Learning

- Define a set of function (**architecture**)
- Goodness of function (**loss**)
- Pick the best function (**optimization**)

## When can we use deep learning to **solve a problem?**

- First, you want **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)

.....

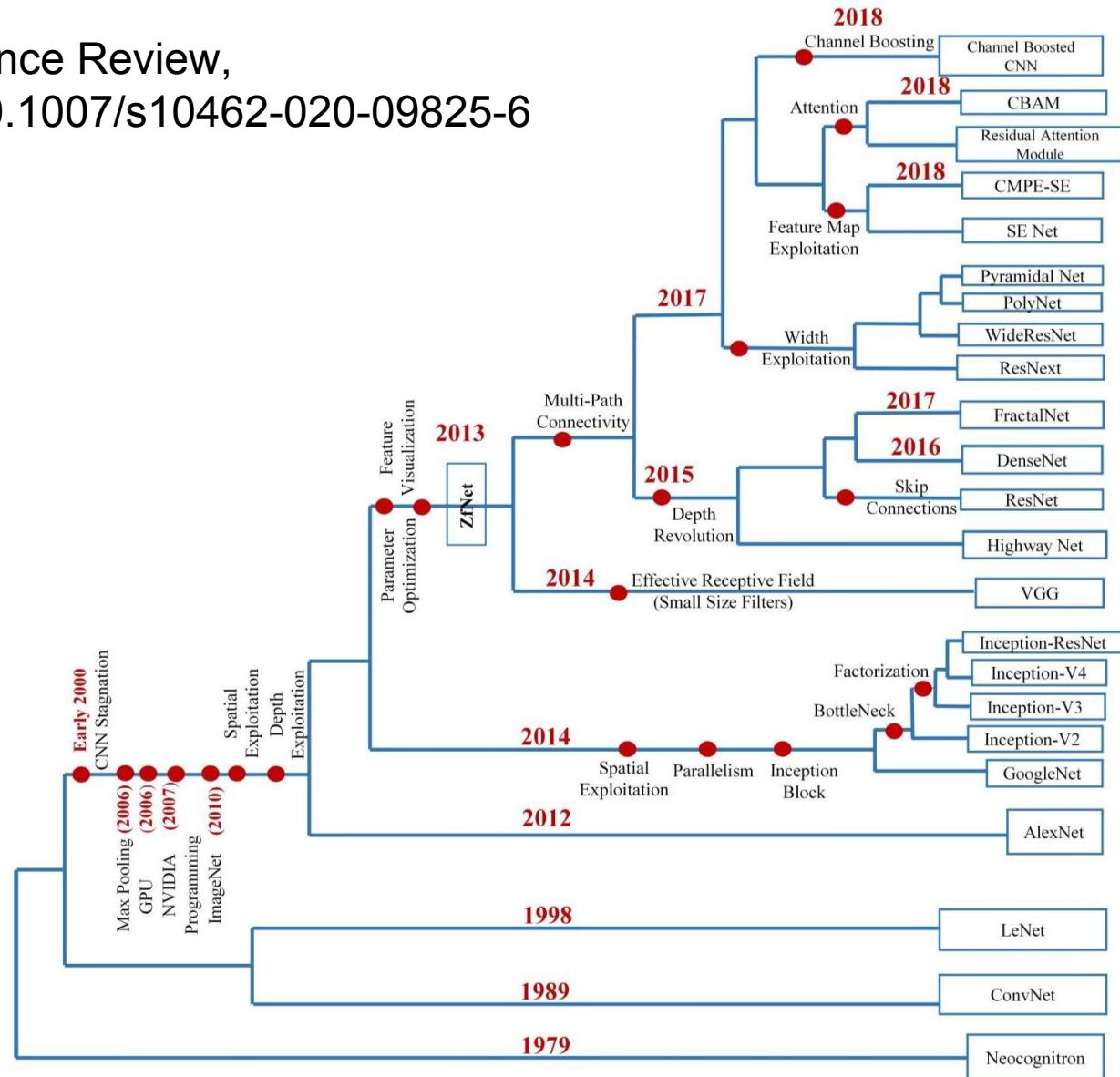
- Auto-encoder
- Generative Adversarial Network (GAN)
- Contrastive learning

.....

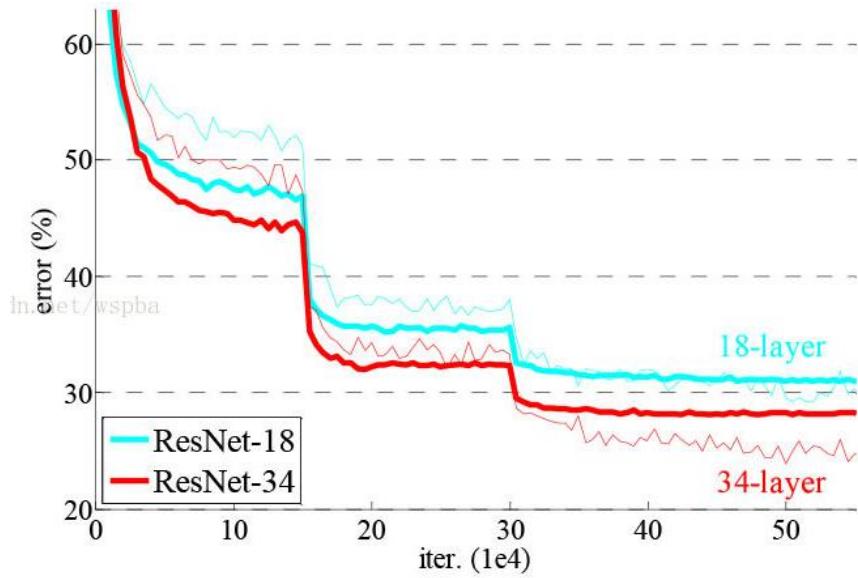
**Large Model Change Space**

# Evolutionary history of deep CNNs

Artificial Intelligence Review,  
<https://doi.org/10.1007/s10462-020-09825-6>

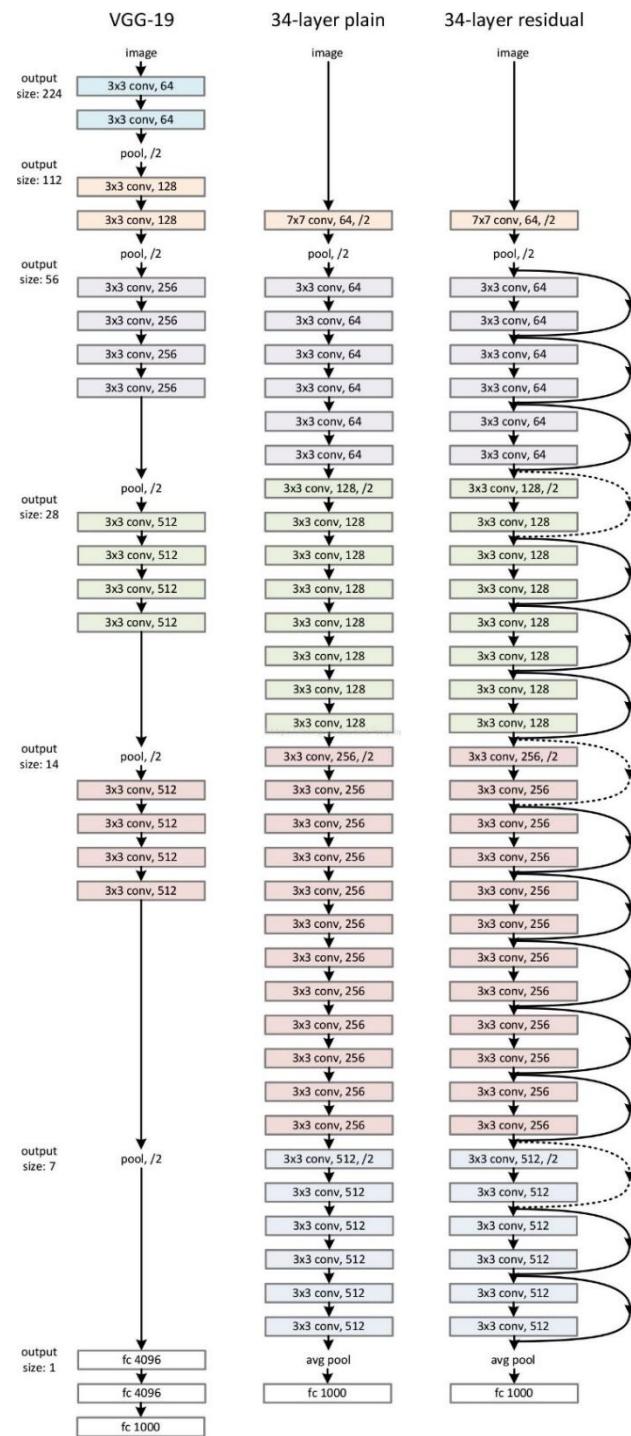


# Deep Neural Networks



Deeper is Better?

Not surprised, more parameters,  
better performance



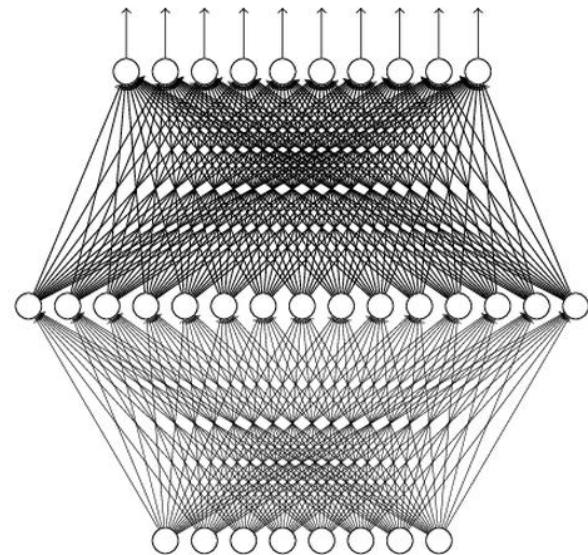
# Universality Approximation Theorem

Any continuous function  $f$

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

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

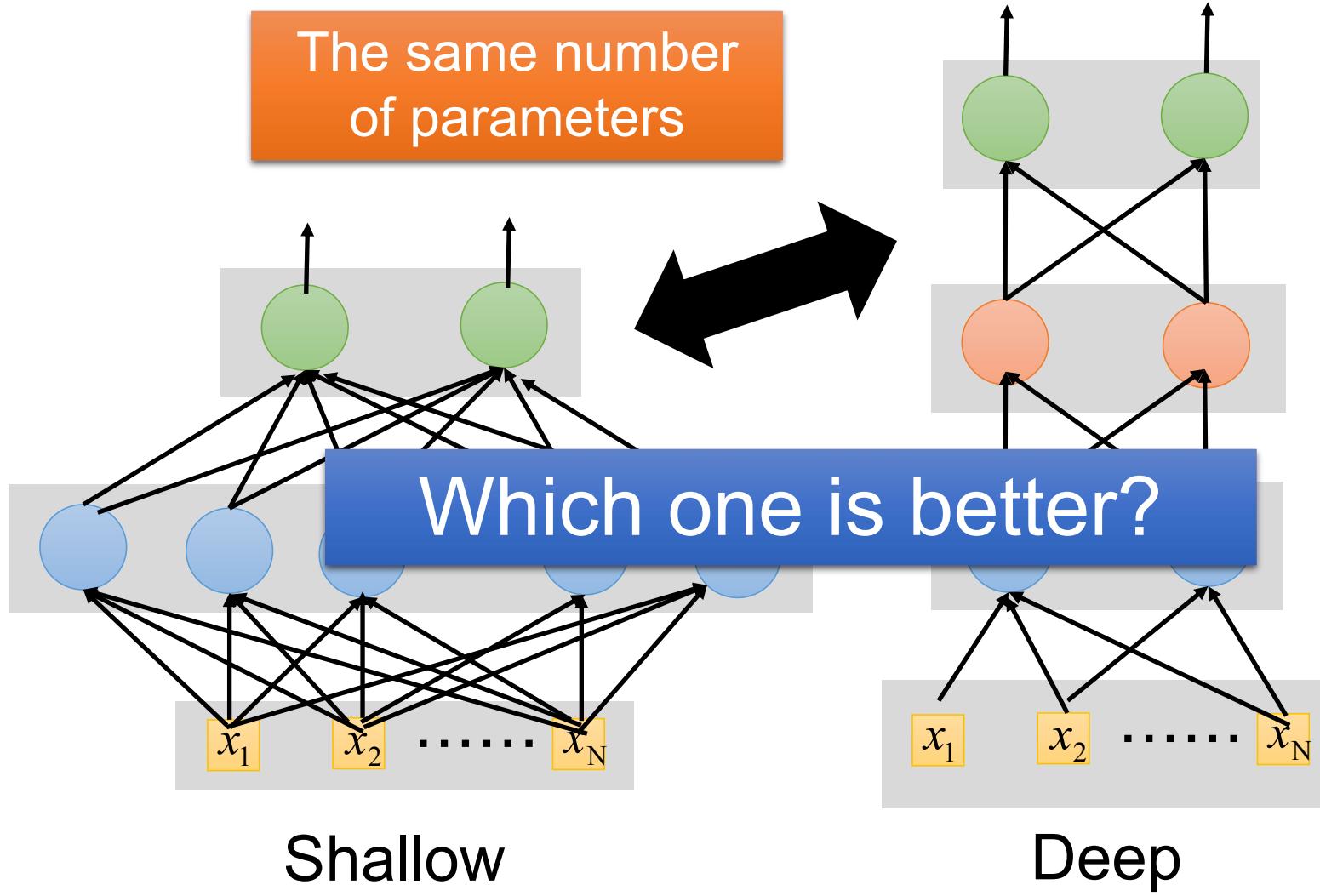
Neural Networks, 1989, 2( 5):359-366.



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

Why “Deep” neural network not “Fat” neural network?

# Fat + Short v.s. Thin + Tall



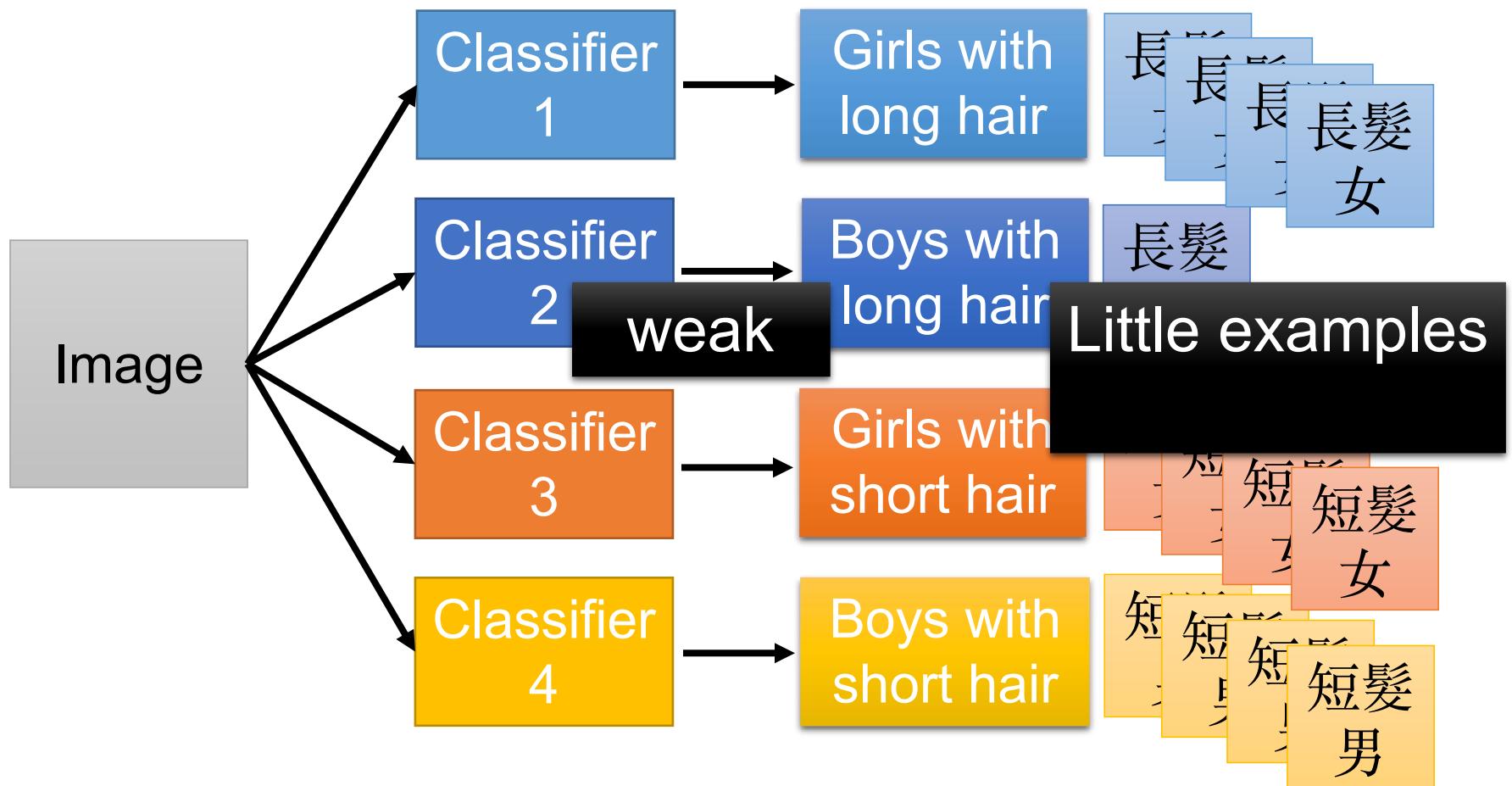
# Fat + Short v.s. Thin + Tall

Layer X Size	Word Error Rate (%)	Layer X Size	Word Error Rate (%)
1 X 2k	24.2		
2 X 2k	20.4		
3 X 2k	18.4		
4 X 2k	17.8		
5 X 2k	17.2	1 X 3772	22.5
7 X 2k	17.1	1 X 4634	22.6
		1 X 16k	22.1

Seide, Frank, Gang Li, and Dong Yu. "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks." *Interspeech*. 2011.

# Why Deep?

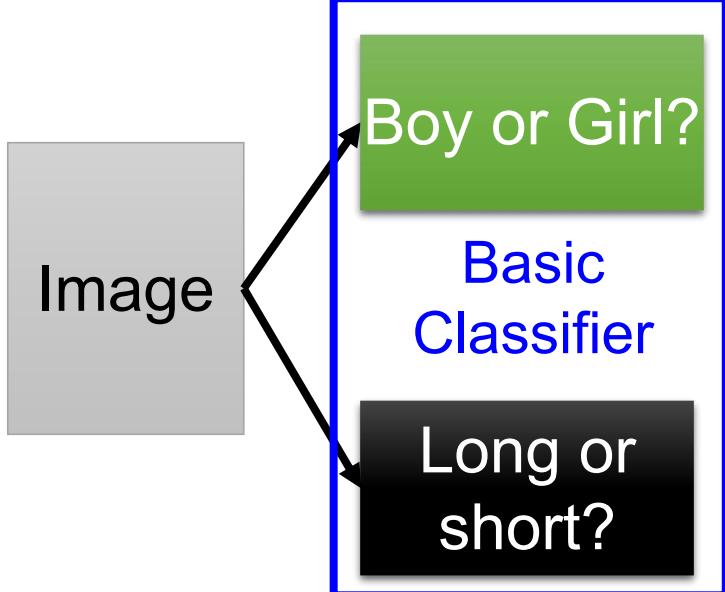
- Deep → Modularization



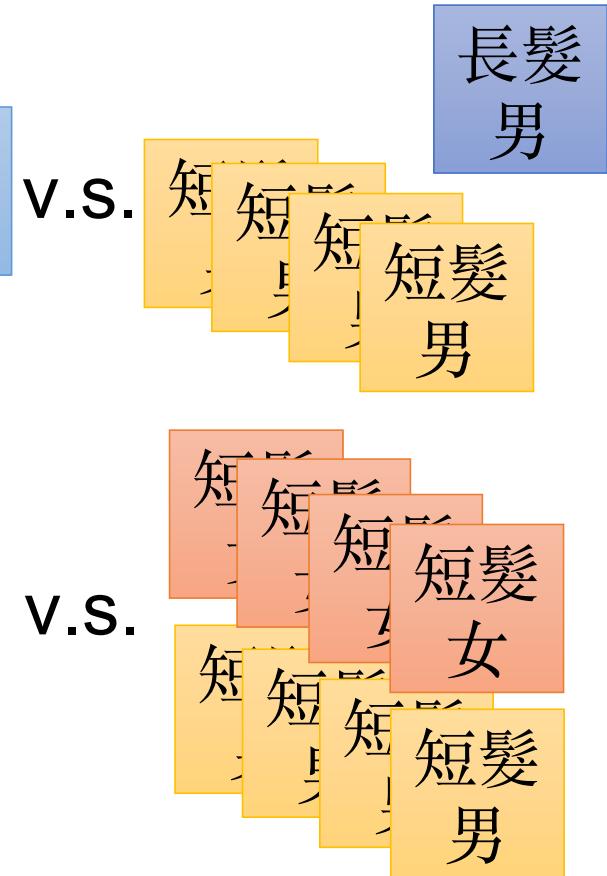
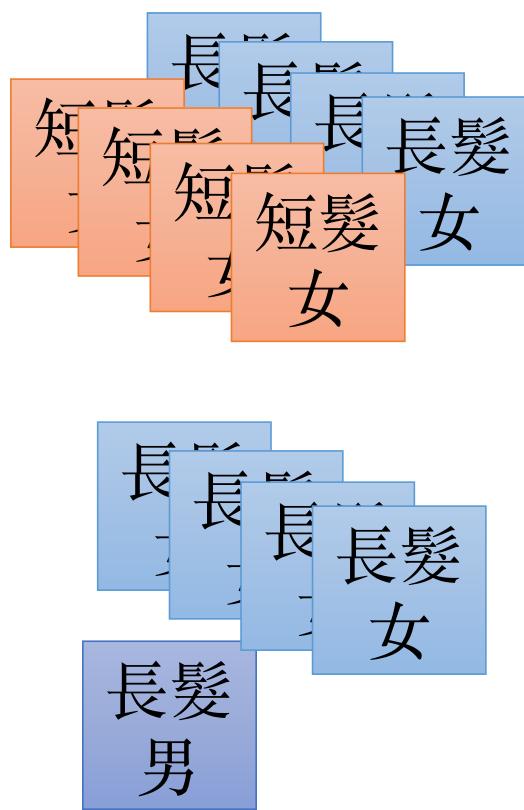
# Why Deep?

Each basic classifier can have sufficient training examples.

- Deep → Modularization



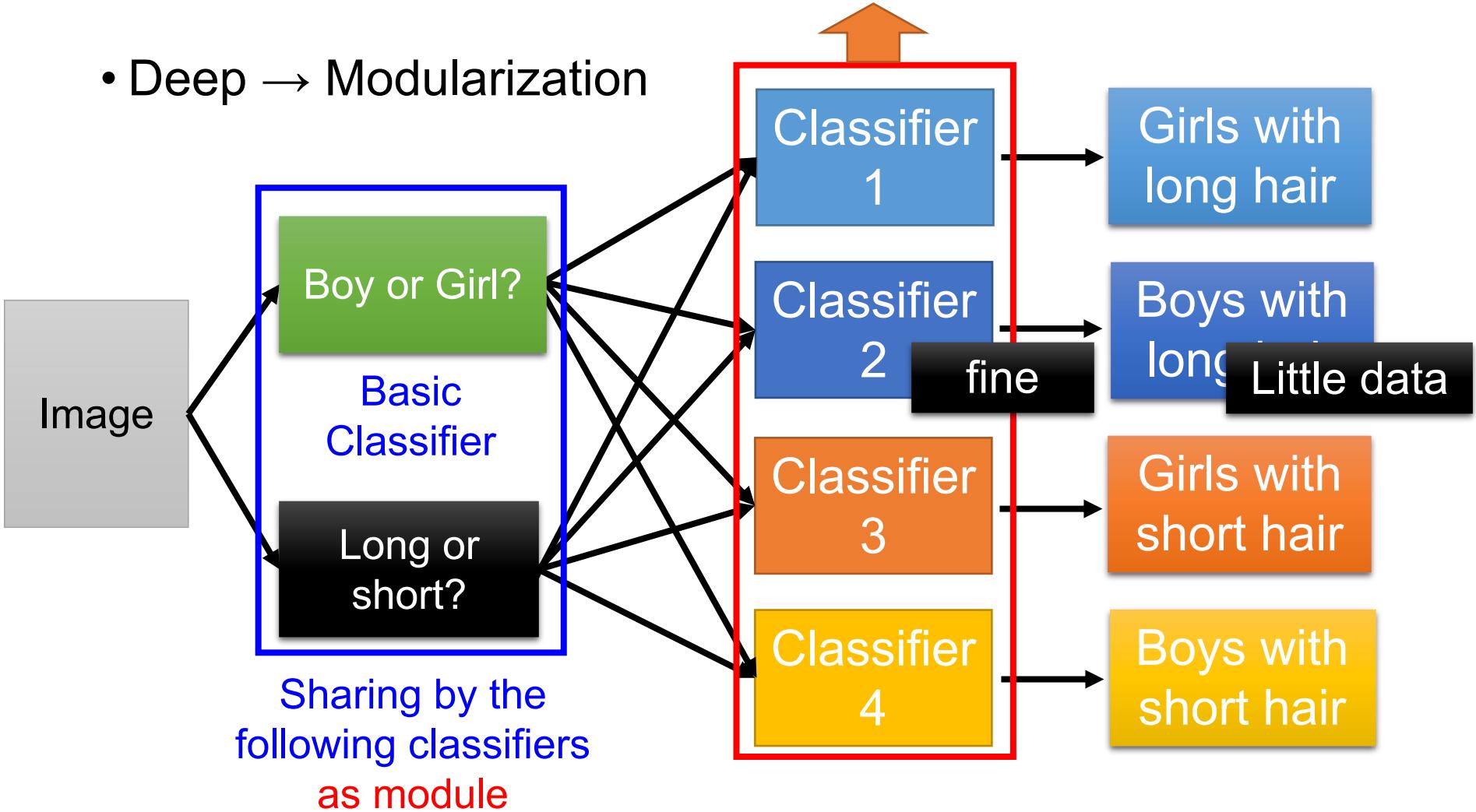
Classifiers for the attributes



# Why Deep?

can be trained by little data

- Deep → Modularization

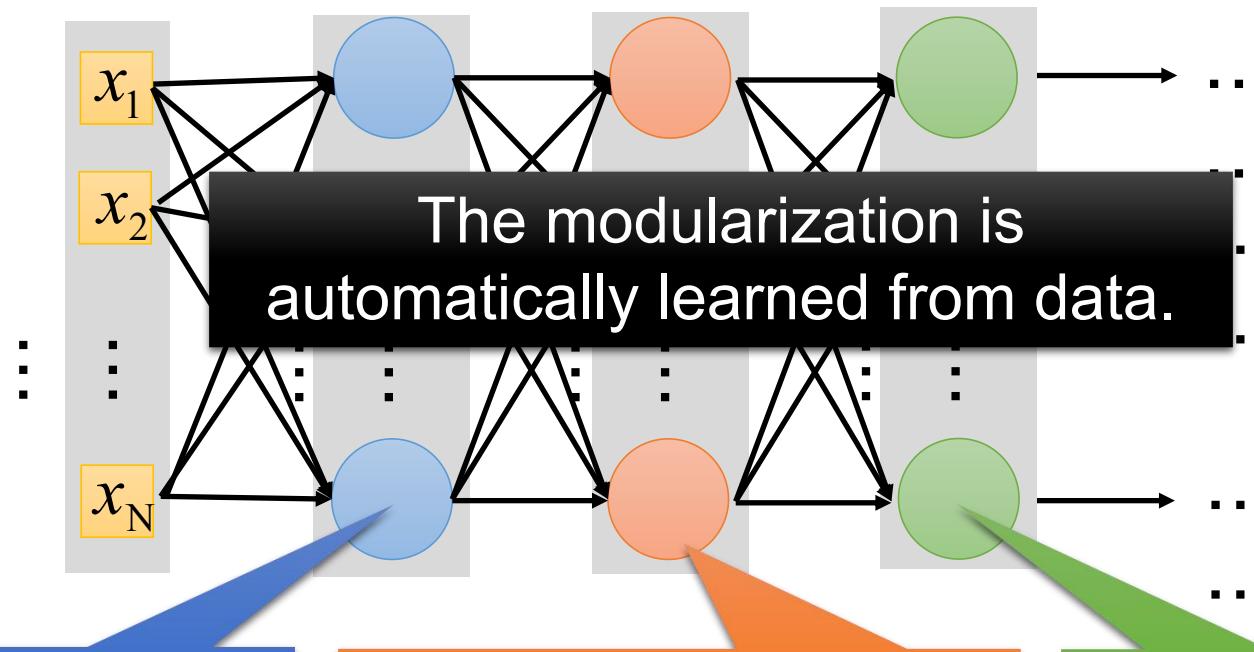


# Why Deep?

Deep Learning also works on small data set like TIMIT.

- Deep → Modularization

→ Less training data?



The most basic classifiers

Use 1<sup>st</sup> layer as module to build classifiers

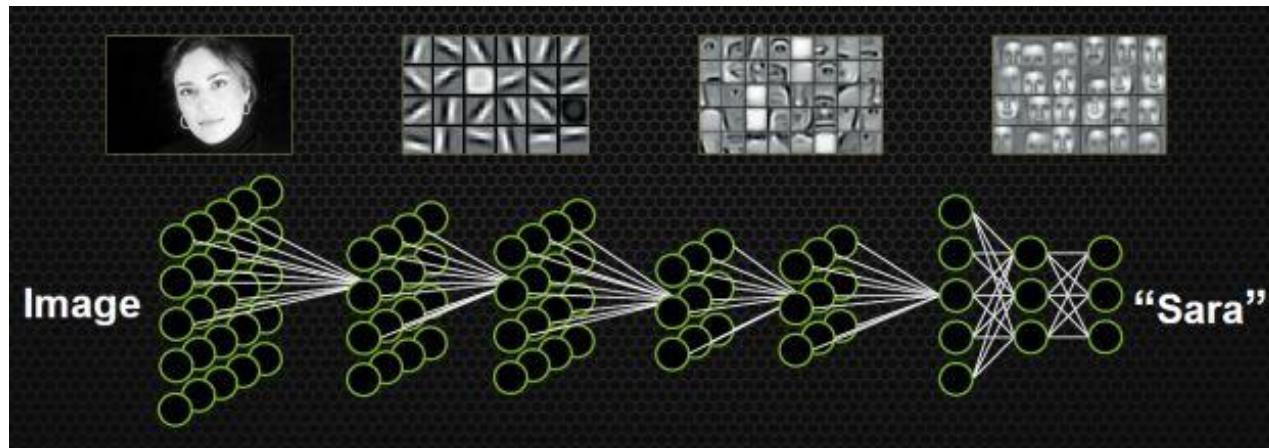
Use 2<sup>nd</sup> layer as module .....

# Why Deep?

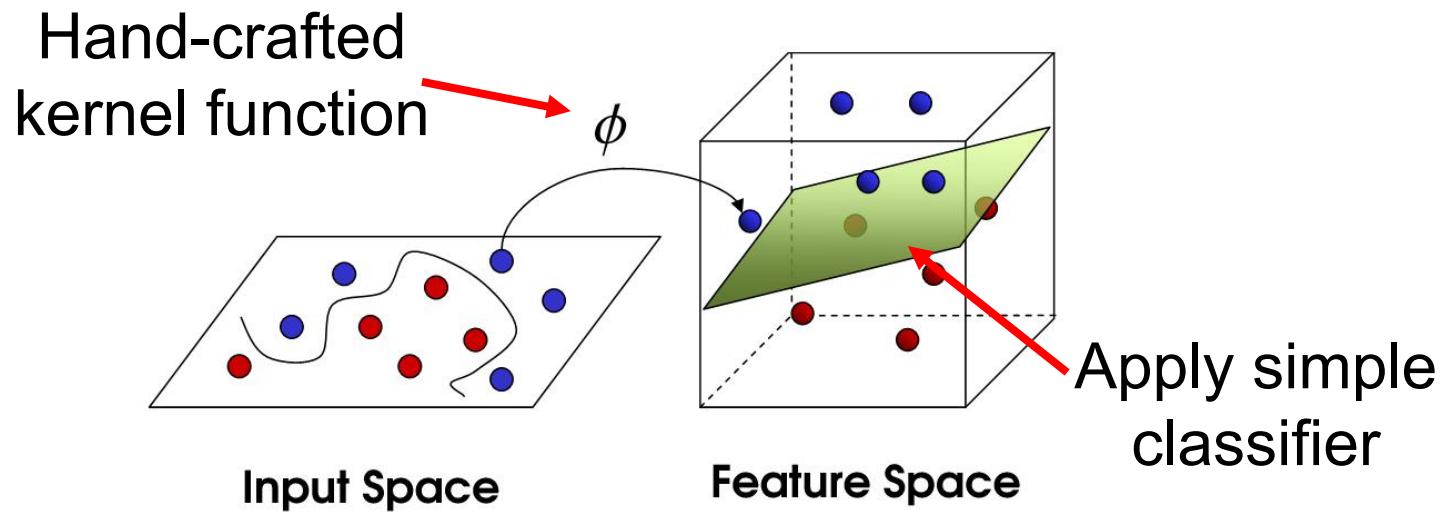
The deep neural network need **less parameters** to describe a function.

- That means that we need **less data** to determine the function.
- It can effectively **reduce the risk of overfitting**.

In the deep neural network, the deeper layers can pick up **more complex features** to help us complete the task.

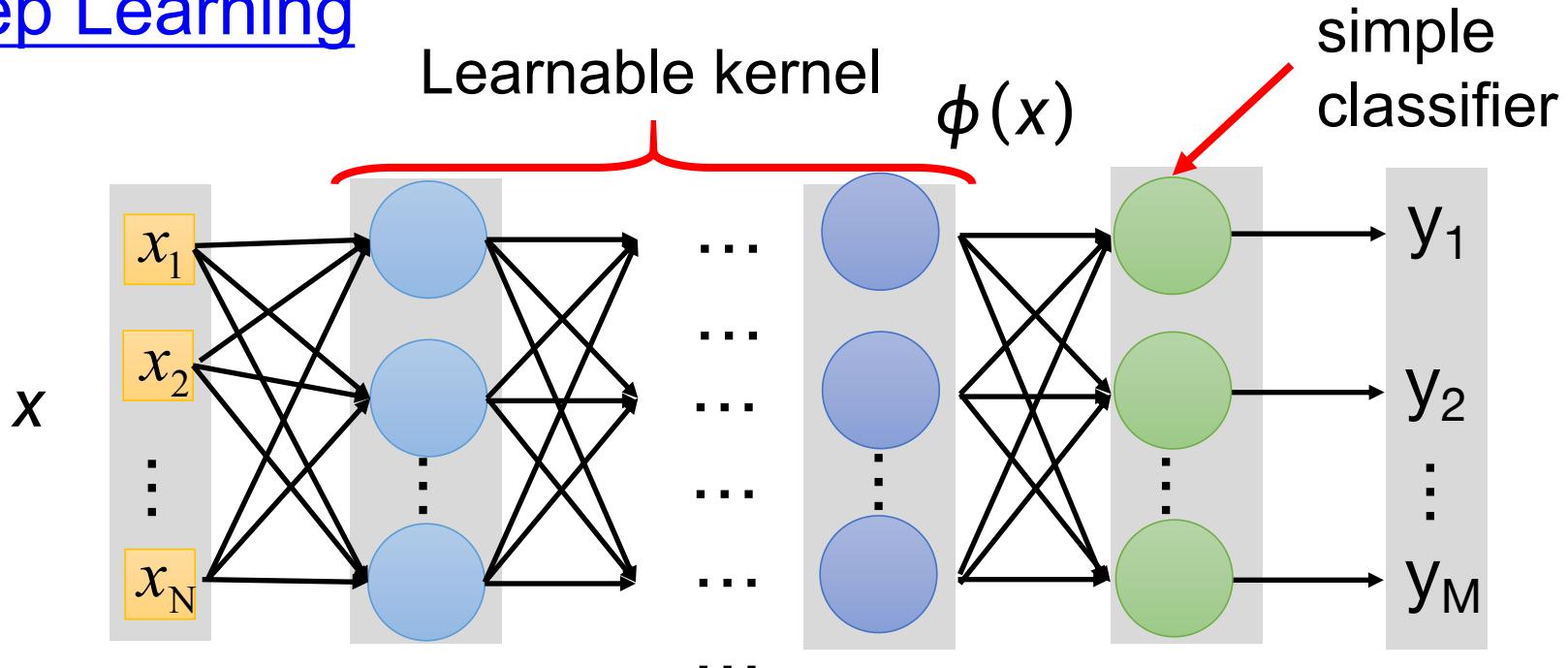


## SVM

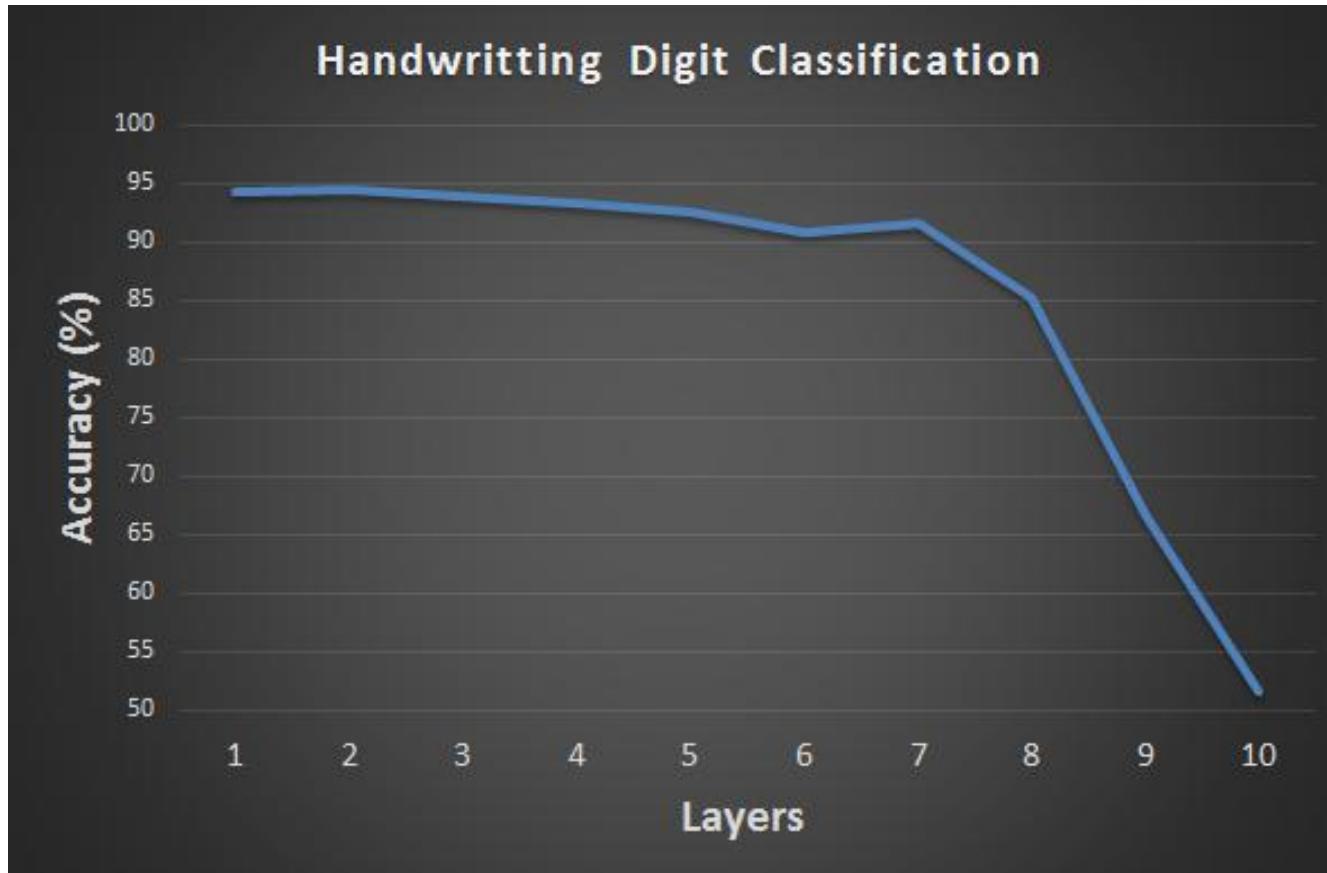


Source of image: [http://www.gipsa-lab.grenoble-inp.fr/transfert/seminaire/455\\_Kadri2013Gipsa-lab.pdf](http://www.gipsa-lab.grenoble-inp.fr/transfert/seminaire/455_Kadri2013Gipsa-lab.pdf)

## Deep Learning



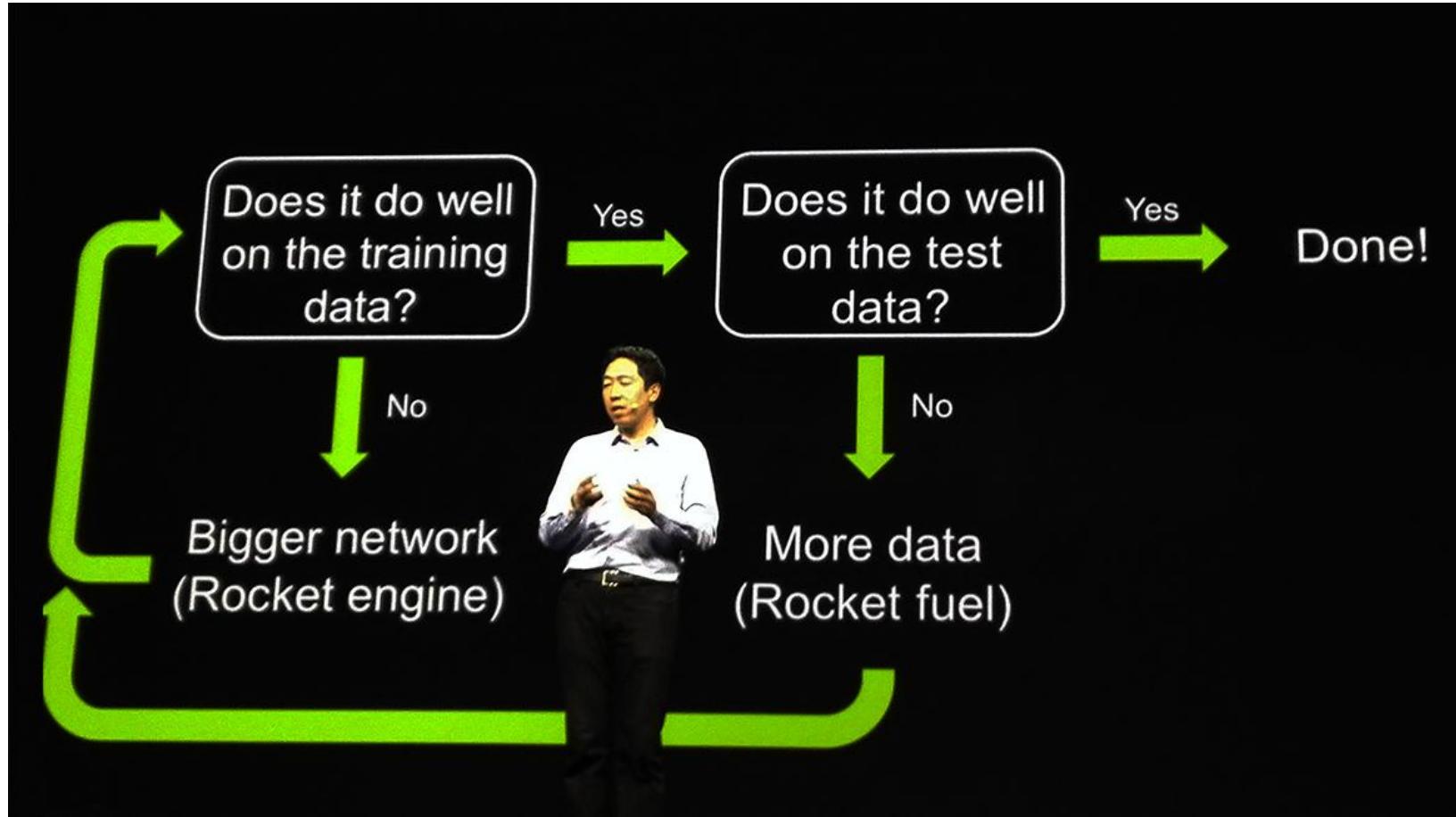
# Hard to get the power of deep ...



Before 2006, deeper usually does not imply better.

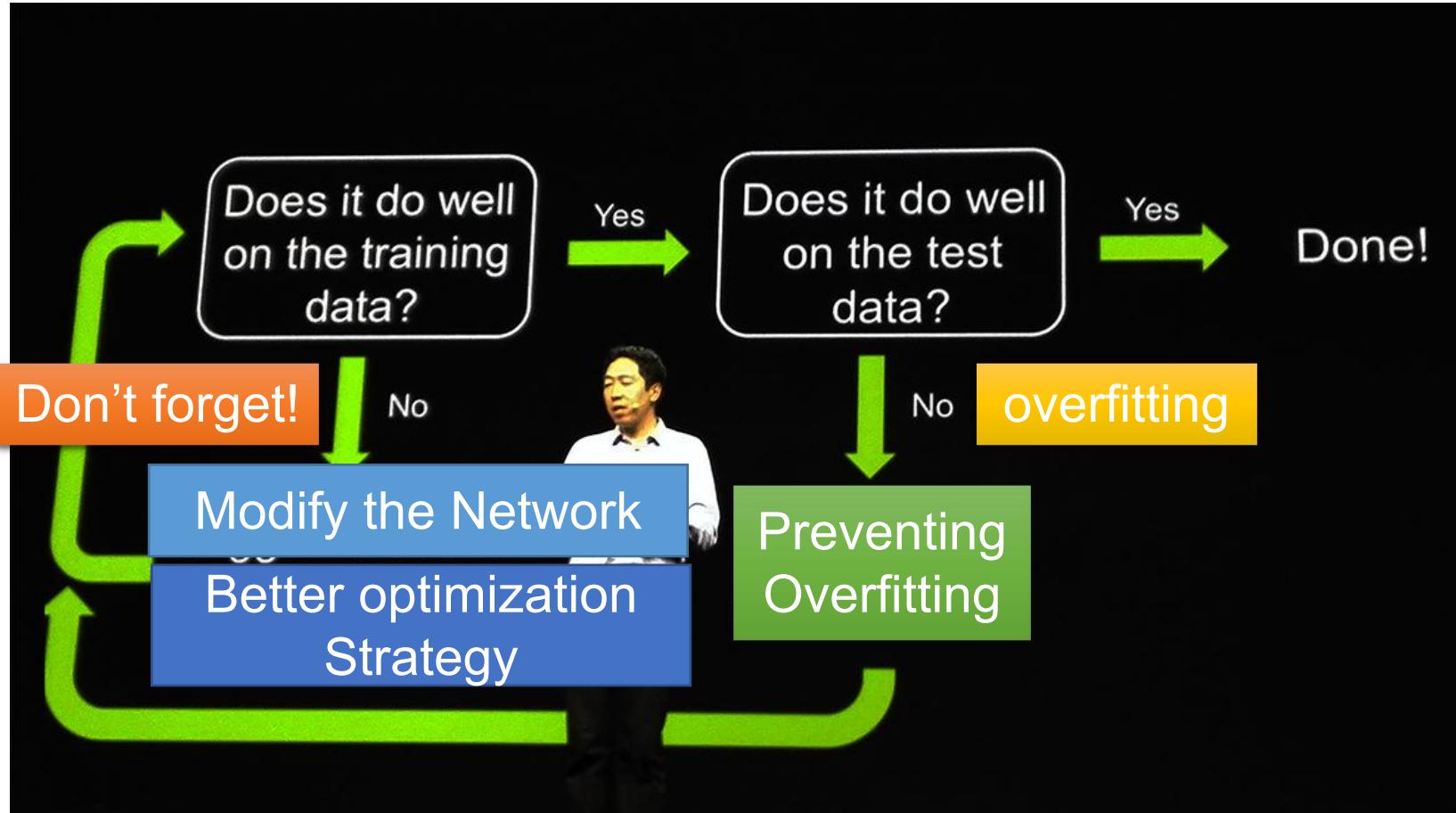
# Recipe for Learning

## Tips for Training DNN



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

# Recipe for Learning



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

# Recipe for Learning

## Modify the Network

- New activation functions, for example, ReLU or Maxout

## Better Optimization Strategy

- Loss function/Mini-batch/Optimizer (Adaptive learning rates)

## Prevent Overfitting

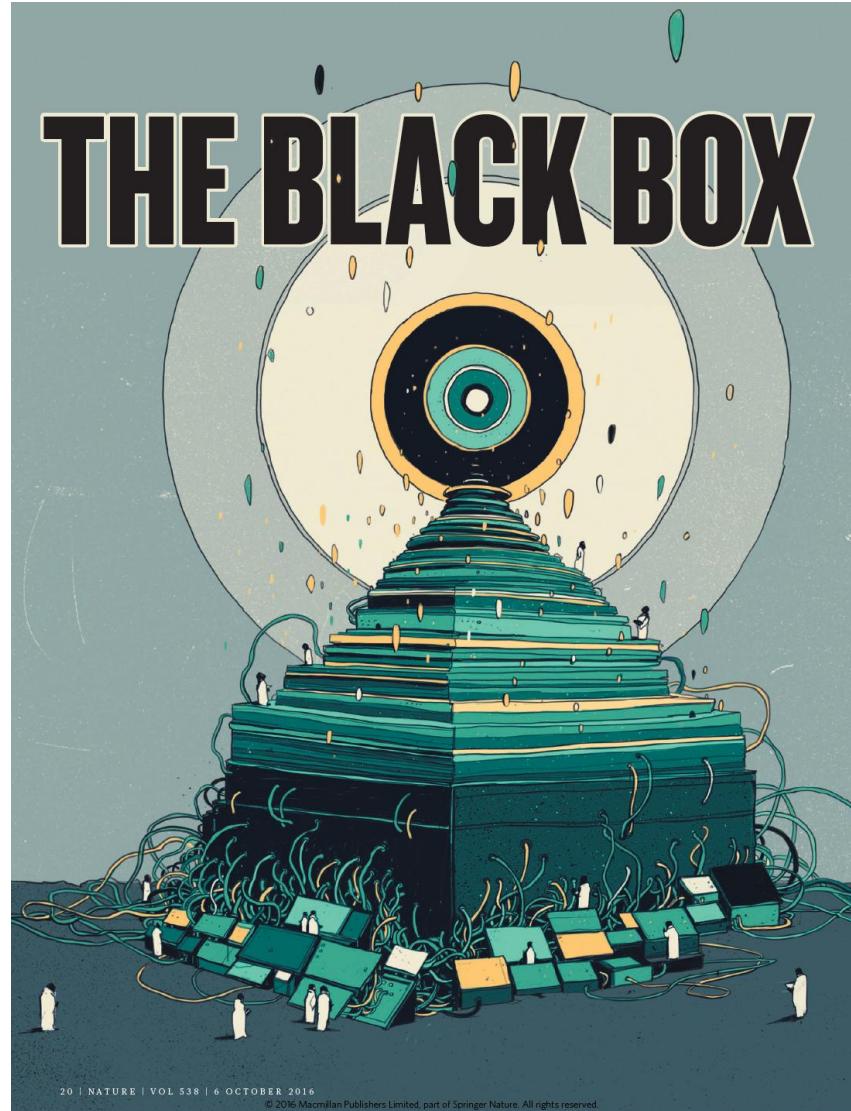
- Increasing training data/Early stopping  
Regularization/Dropout

We get a “perfect” deep learning model, which works well.

*“The problem is that the knowledge gets baked into the network, rather than into us.”*

Nature 538, 20–23 (2016)

# The Black Box of Deep Learning



# Demystifying deep learning

Artificial Intelligence / Machine Learning

---

## The Dark Secret at the Heart of AI

No one really knows how the most advanced algorithms do what they do. That could be a problem.

by **Will Knight**

Apr 11, 2017

---

We can build these models, but we don't know how they work.

How well can we get along with machines that are **unpredictable and inscrutable**?

# Easy to Be Attacked



$+ .007 \times$



$=$



$\mathbf{x}$   
“panda”  
57.7% confidence

$\text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y))$   
“nematode”  
8.2% confidence

$\mathbf{x} +$   
 $\epsilon \text{sign}(\nabla_{\mathbf{x}} J(\theta, \mathbf{x}, y))$   
“gibbon”  
99.3 % confidence

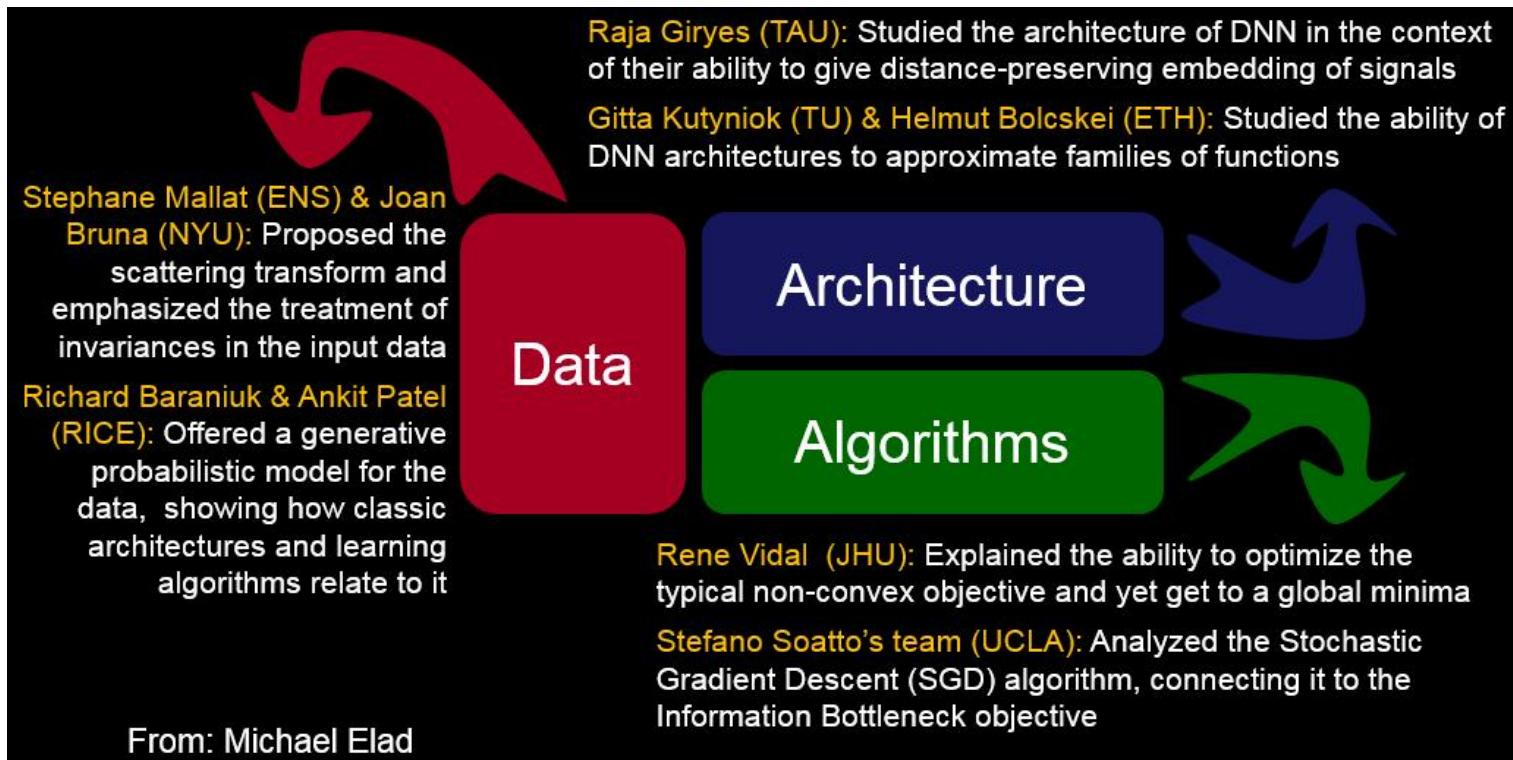
# Larger and Larger DNN Models

- The number of parameters of DNN is bigger and bigger.
- For example, the number of parameters of **Switch Transformer** has reached the trillion level. There are some problems when training so huge network.
- Requirement of a lot of annotation data
- Difficulty of model training

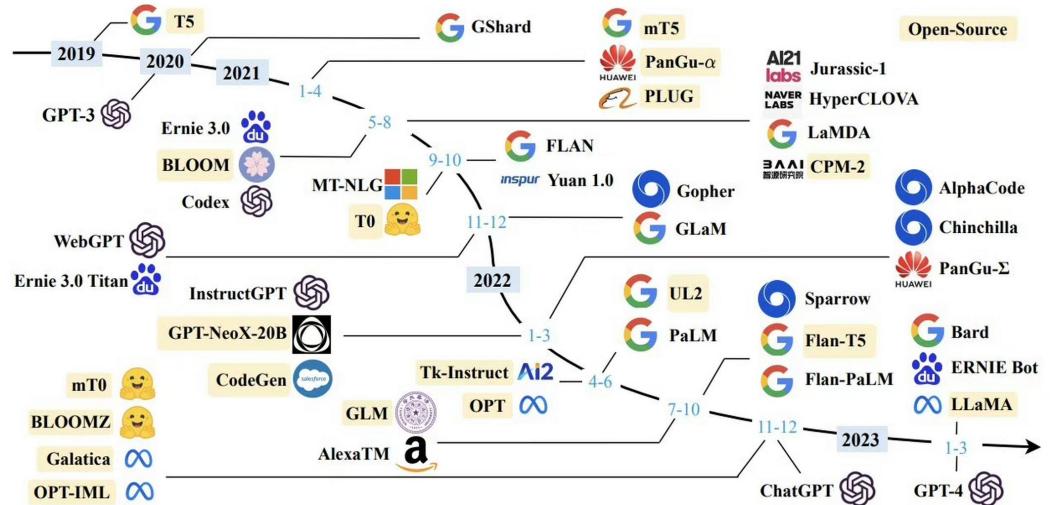
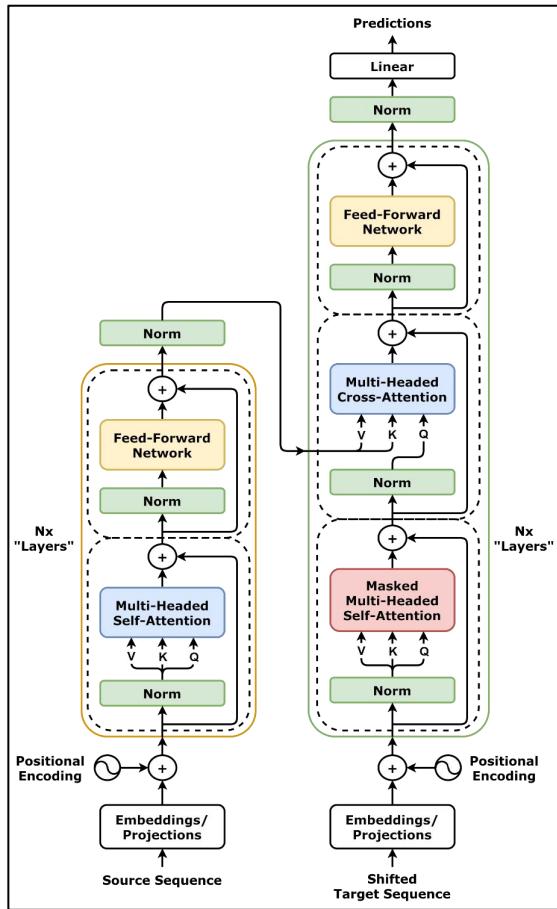
# Deep Learning Theory

- **Representation:** Why are deep neural networks better than shallow ones?
- **Optimization:** Why is SGD (Stochastic Gradient Descent) good at finding minima and what are good minima?
- **Generalization:** Why is it that we don't have to worry about overfitting despite overparameterization?

# Deep Learning Theory



# Generative Pre-trained Transformer (GPT): Large Language Models



代表性大预言模型发展时间脉络 (>10B)

<https://doi.org/10.48550/arXiv.2303.18223>

Attention is All you Need. NIPS, 2017.

# Thanks

[zsh@amss.ac.cn](mailto:zsh@amss.ac.cn)



群聊：深度学习理论2024



该二维码7天内(10月24日前)有效，重新进入将更新