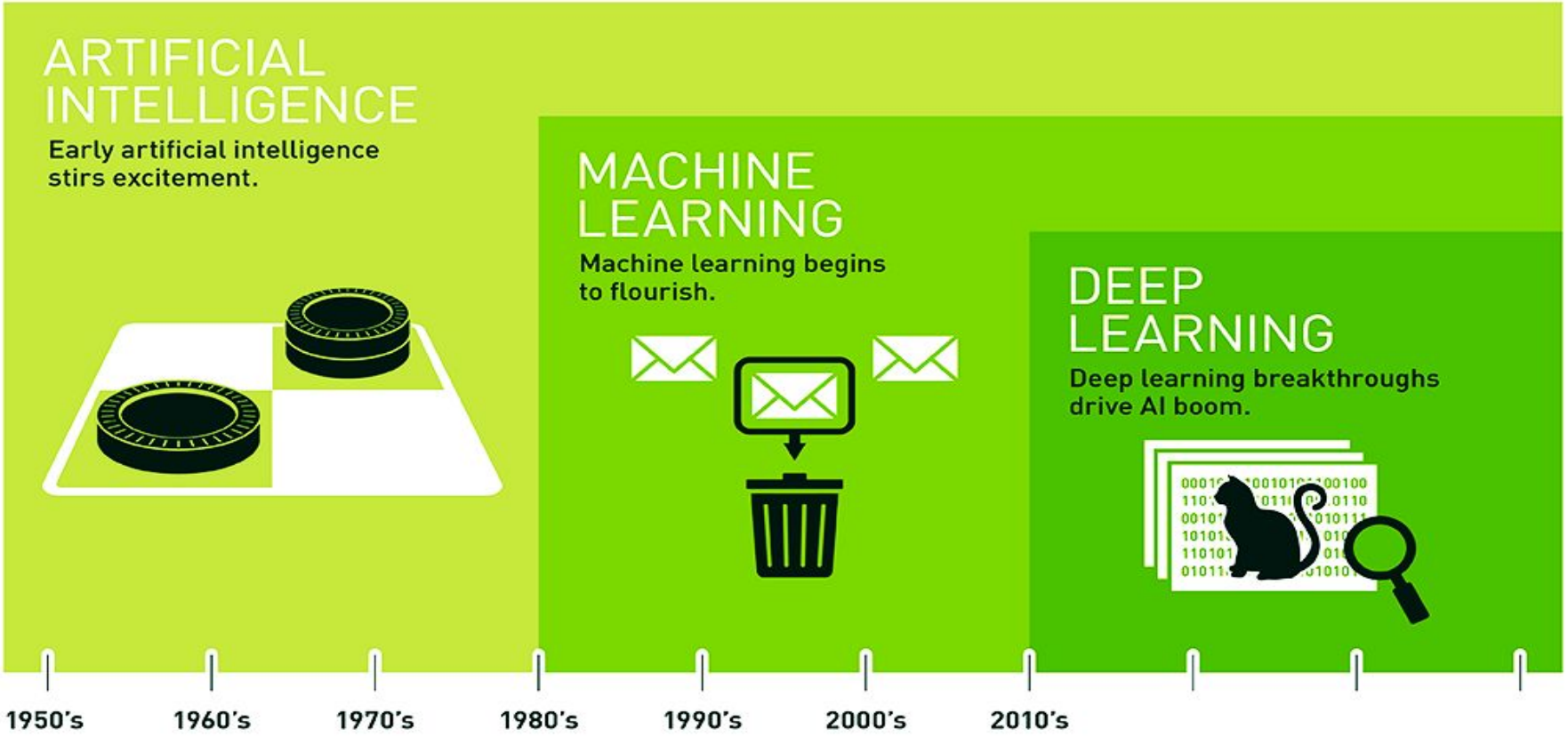
Deep Learning Motivation

By Venkateshwar Reddy Jambula



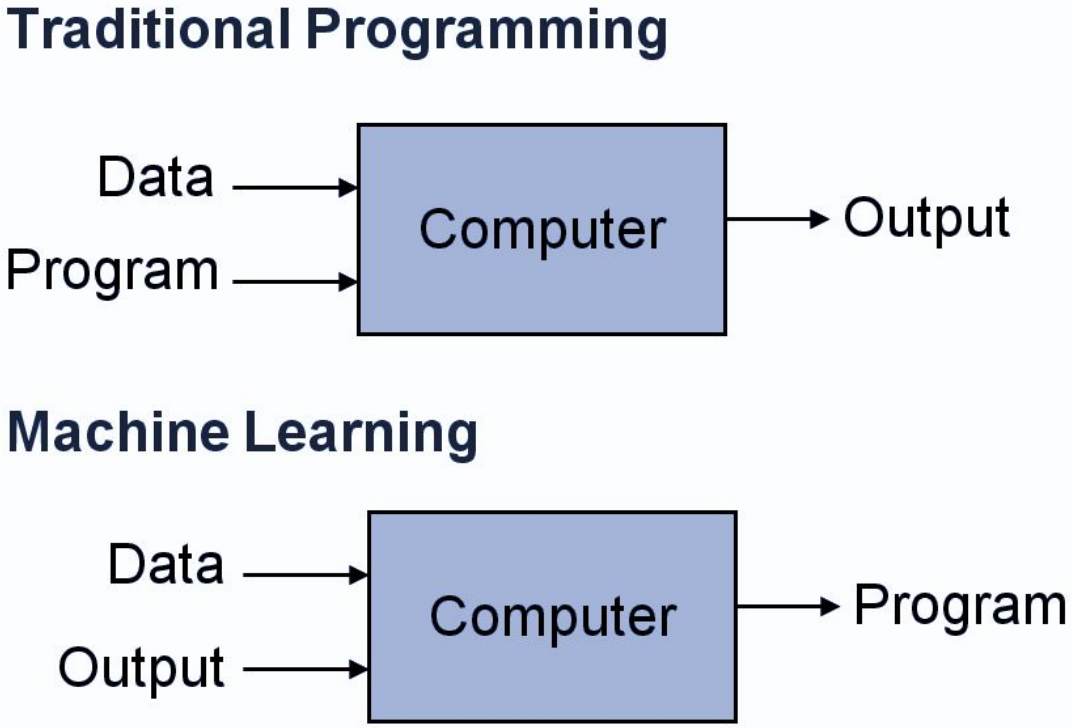
Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Machine Learning :

Machine Learning at its most basic is the practice of using algorithms to parse data, learn from it, and then make a determination or prediction about something in the world. So rather than hand-coding software routines with a specific set of instructions to accomplish a particular task, the machine is “trained” using large amounts of data and algorithms that give it the ability to learn how to perform the task.

Machine learning came directly from minds of the early AI crowd, and the algorithmic approaches over the years included decision tree learning, inductive logic programming. clustering, reinforcement learning, and Bayesian networks among others.

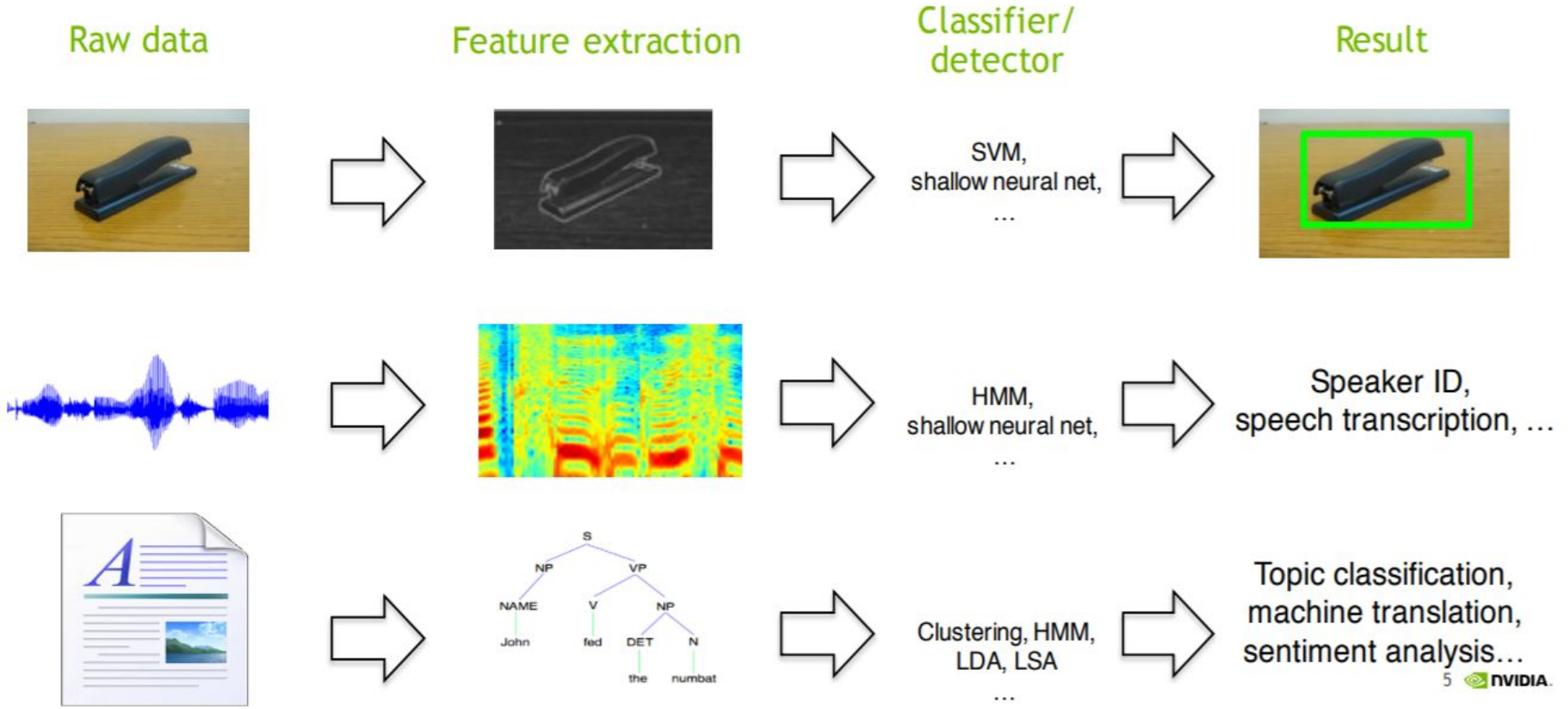
Types of Machine Learning



1. Supervised
2. Unsupervised
3. Semi-Supervised
4. Reinforcement

Traditional machine perception

Hand crafted feature extractors



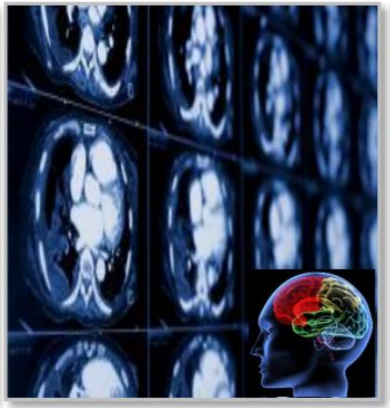
What is Deep Learning?

When you hear the term deep learning, just think of a large deep neural net. Deep refers to the number of layers typically and so this kind of the popular term that's been adopted in the press.

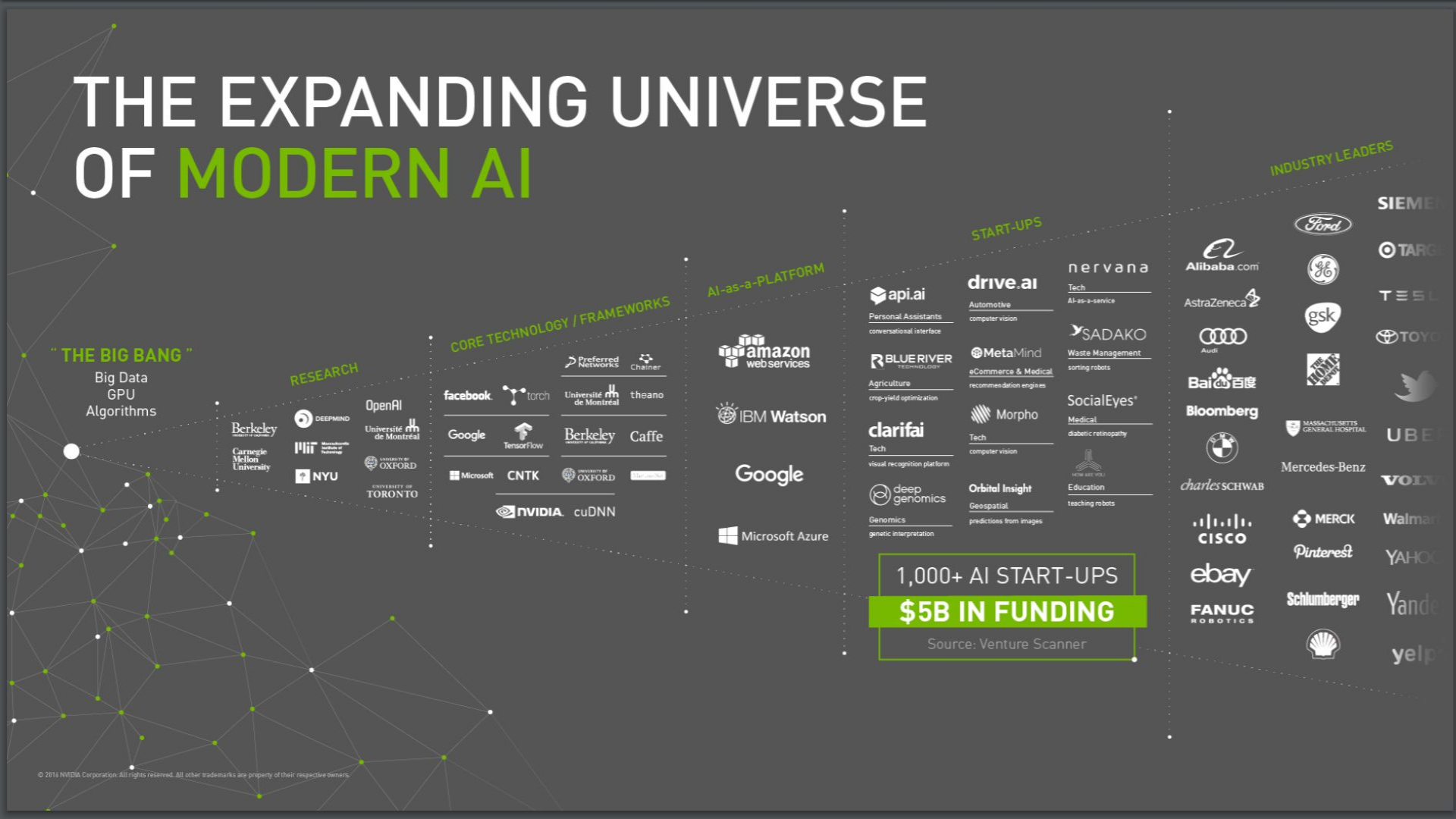
Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features

Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. Automatically learning features at multiple levels of abstraction allow a system to learn complex

DEEP LEARNING IS SWEEPING ACROSS INDUSTRIES

Internet Services	Medicine	Media & Entertainment	Security & Defense	Autonomous Machines
				
➢ Image/Video classification ➢ Speech recognition ➢ Natural language processing	➢ Cancer cell detection ➢ Diabetic grading ➢ Drug discovery	➢ Video captioning ➢ Content based search ➢ Real time translation	➢ Face recognition ➢ Video surveillance ➢ Cyber security	➢ Pedestrian detection ➢ Lane tracking ➢ Recognize traffic signs

THE EXPANDING UNIVERSE OF MODERN AI



A NEW COMPUTING MODEL

Algorithms that Learn from Examples

Expert Written Computer Program

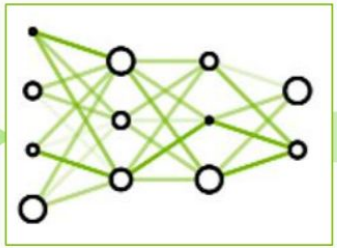
car

vehicle

coupe

Traditional Approach

- Requires domain experts
- Time consuming
- Error prone
- Not scalable to new problems



Deep Neural Network

car

vehicle

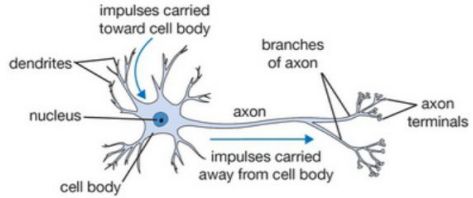
coupe

Deep Learning Approach

- ✓ Learn from data
- ✓ Easily to extend
- ✓ Speedup with GPUs

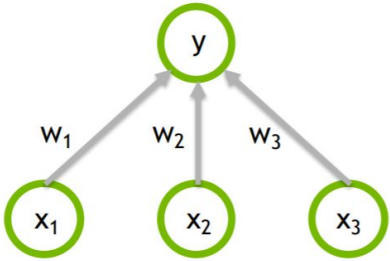
Artificial Neuron

Biological neuron



From Stanford cs231n lecture notes

Artificial neuron

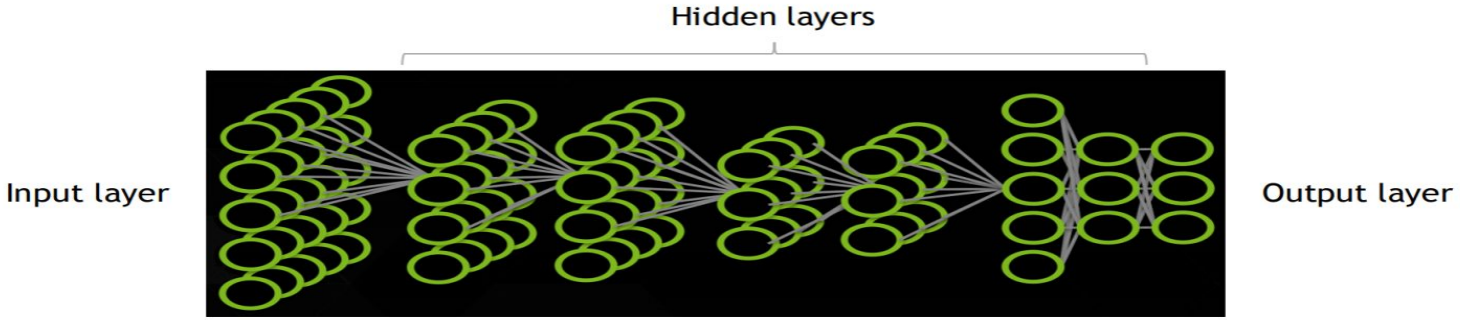


$y=F(w_1x_1+w_2x_2+w_3x_3)$

$F(x)=max(0,x)$

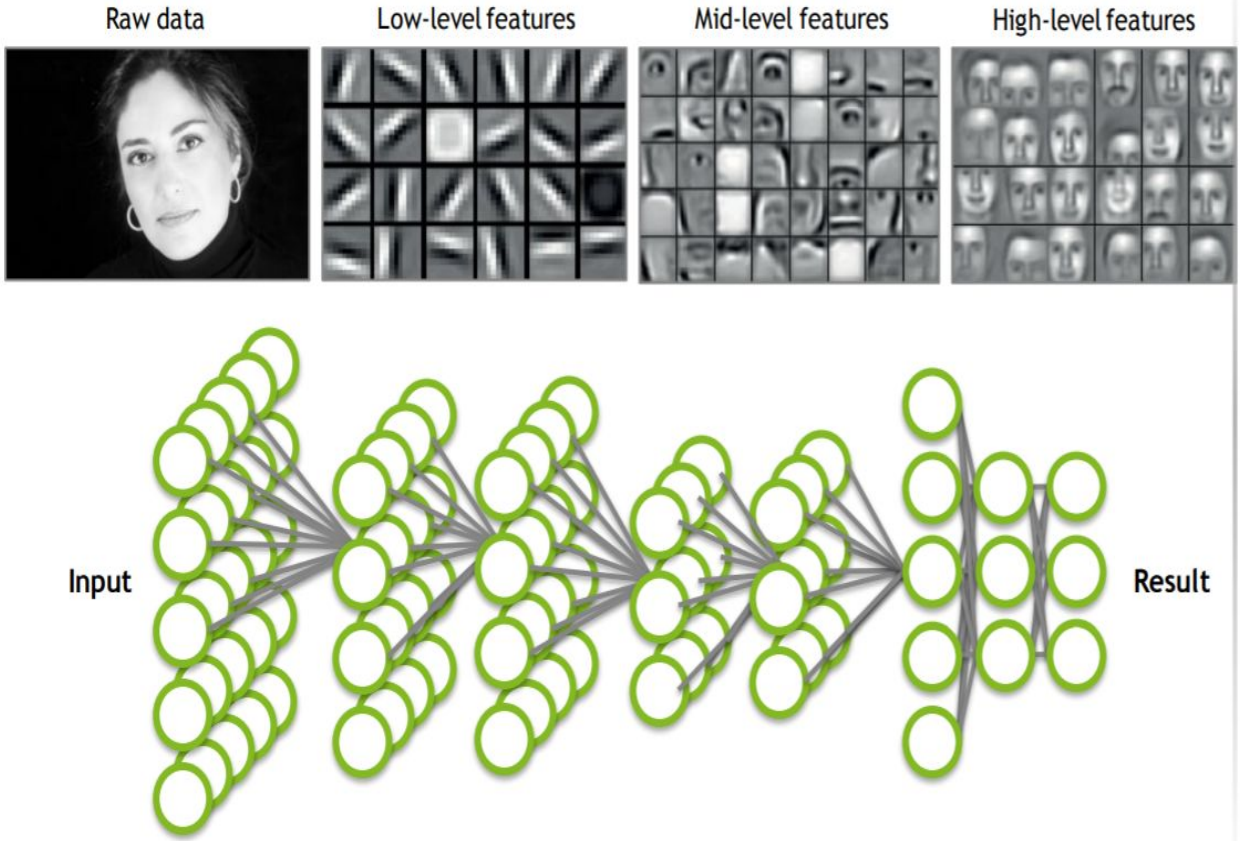
Artificial neural network

A collection of simple, trainable mathematical units that collectively learn complex functions



Given sufficient training data an artificial neural network can approximate very complex functions mapping raw data to output decisions

Deep neural network (dnn)



Application components:

Task objective
- e.g. Identify face

Training data
- 10-100M images

Network architecture
- ~10 layers
- 1B parameters

Learning algorithm
- ~30 Exaflops
- ~30 GPU days

Deep learning approach

Train:

Errors

MODEL

Dog
Cat
Raccoon

Deploy:

MODEL

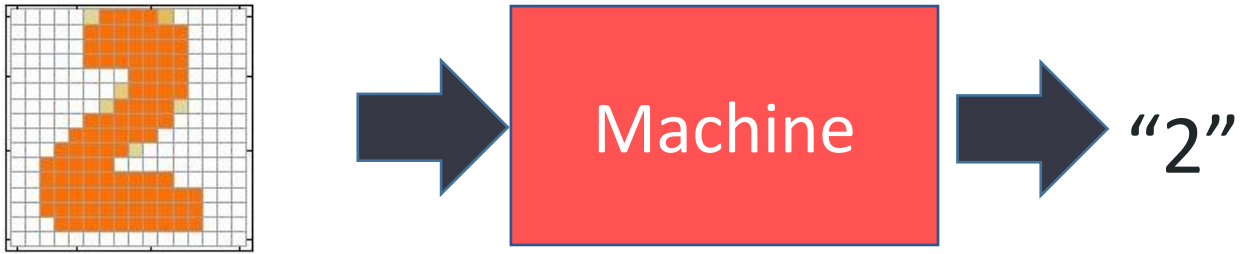
Dog

Deep learning benefits

- Robust
  - No need to design the features ahead of time – features are automatically learned to be optimal for the task at hand
  - Robustness to natural variations in the data is automatically learned
- Generalizable
  - The same neural net approach can be used for many different applications and data types
- Scalable
  - Performance improves with more data, method is massively parallelizable

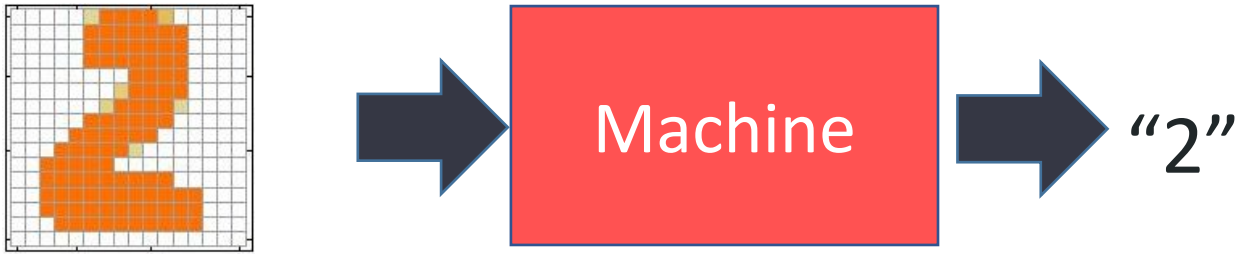
Example Application

- Handwriting Digit Recognition



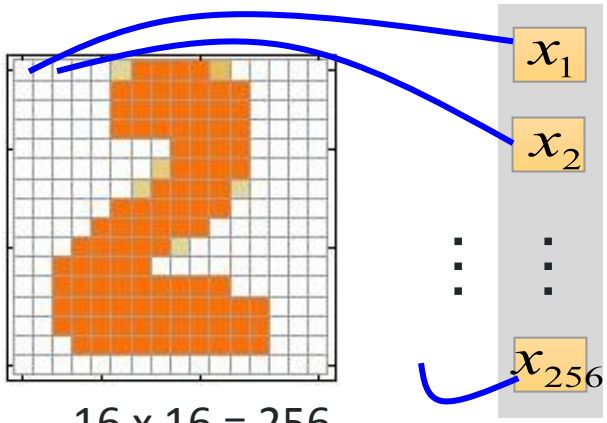
Example Application

- Handwriting Digit Recognition



Handwriting Digit Recognition

Input

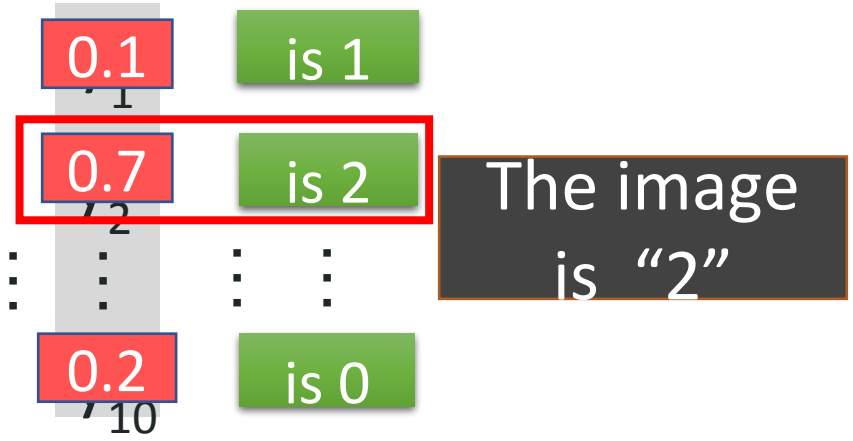


16 x 16 = 256

Ink → 1

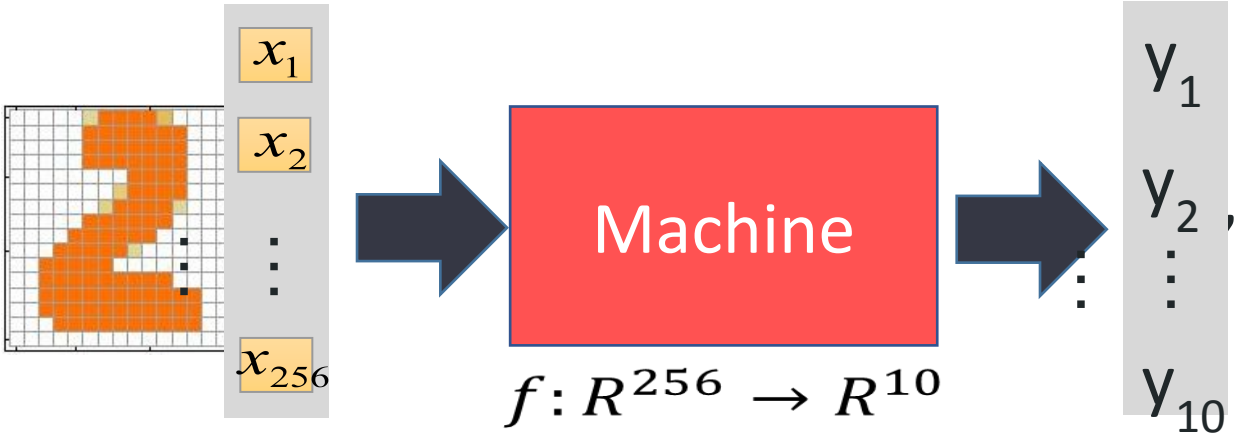
No ink → 0

Output



Example Application

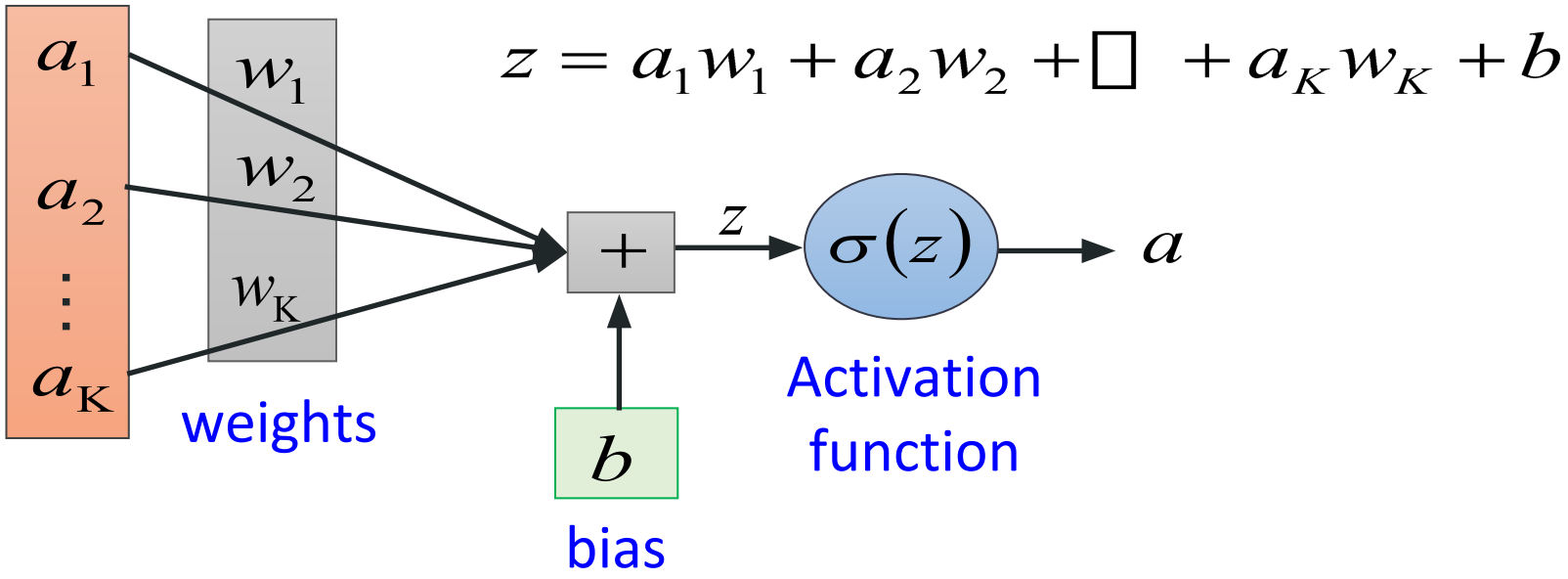
- Handwriting Digit Recognition



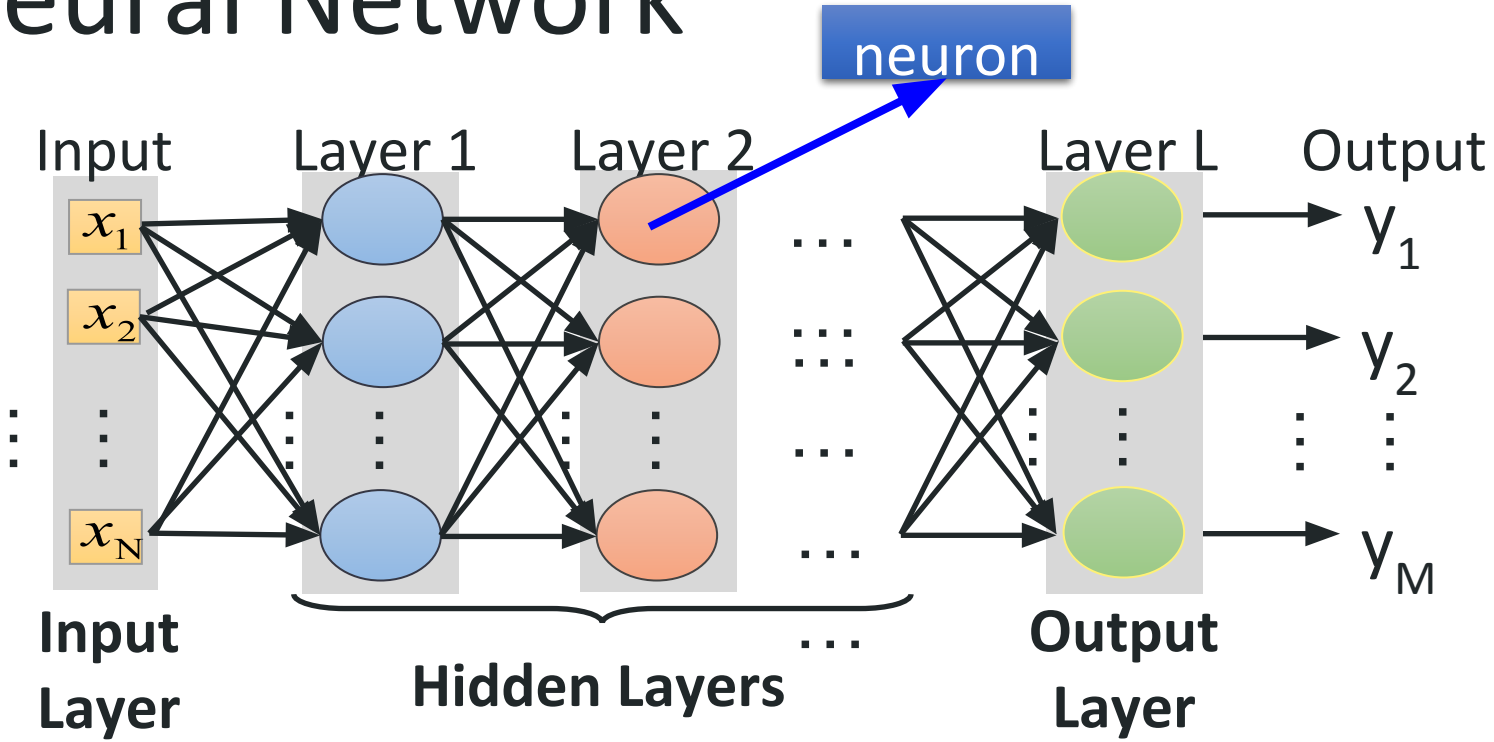
In deep learning, the function f is represented by neural network

Element of Neural Network

Neuron $f: R^K \rightarrow R$

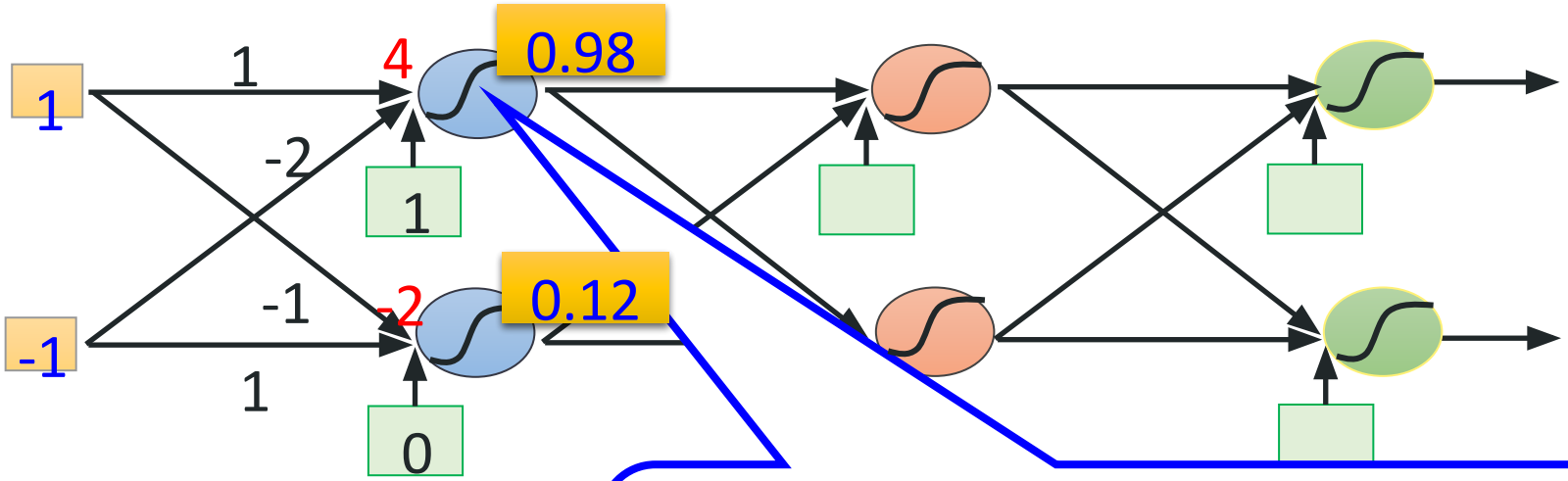


Neural Network



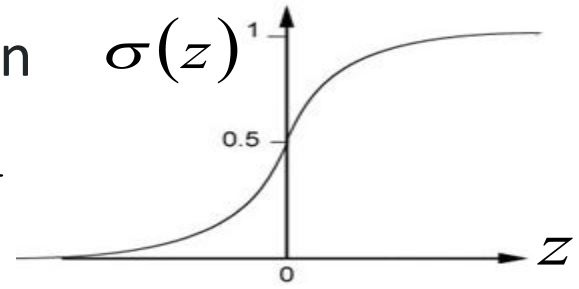
Deep means many hidden layers

Example of Neural Network

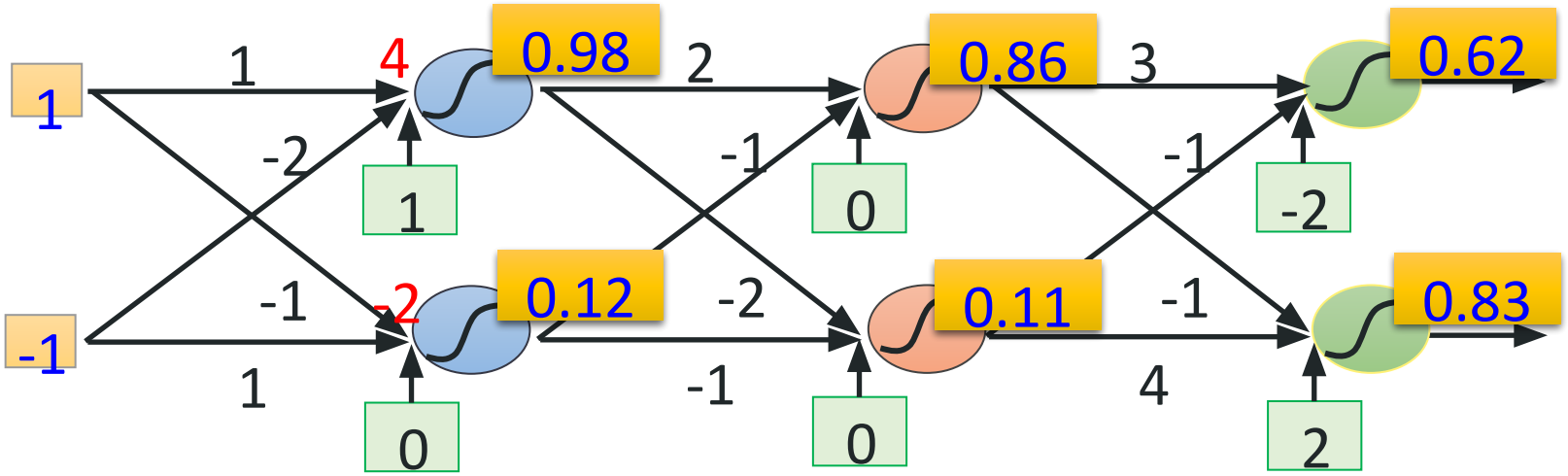


Sigmoid Function

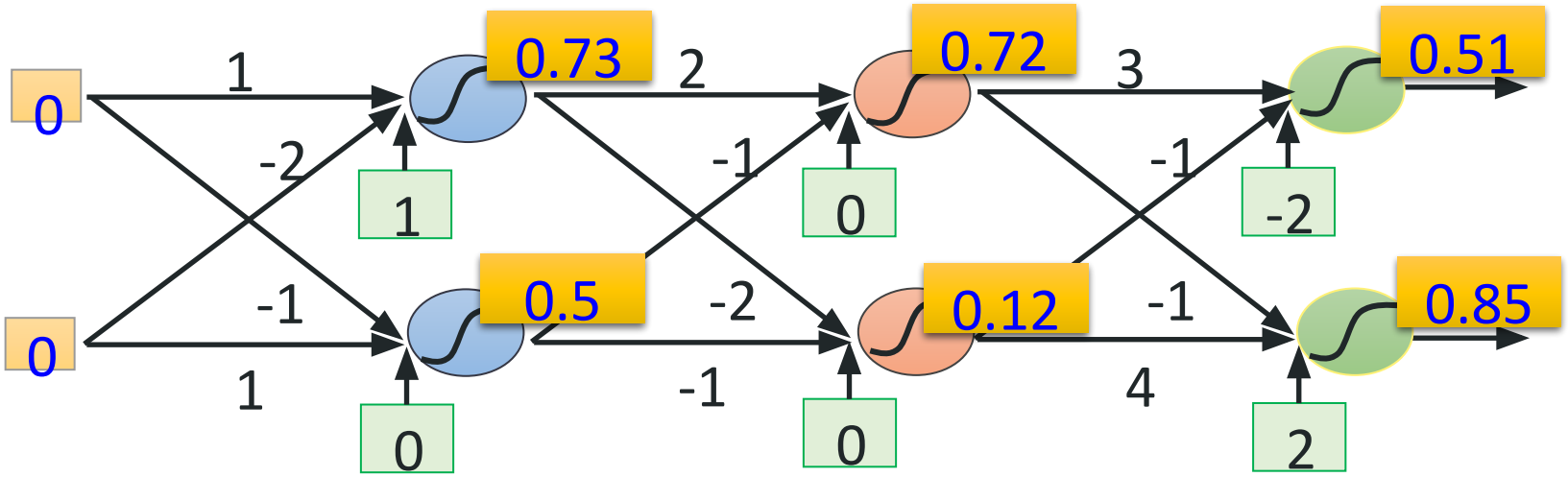
$$\sigma(z) = \frac{1}{1+e^{-z}}$$



Example of Neural Network



Example of Neural Network

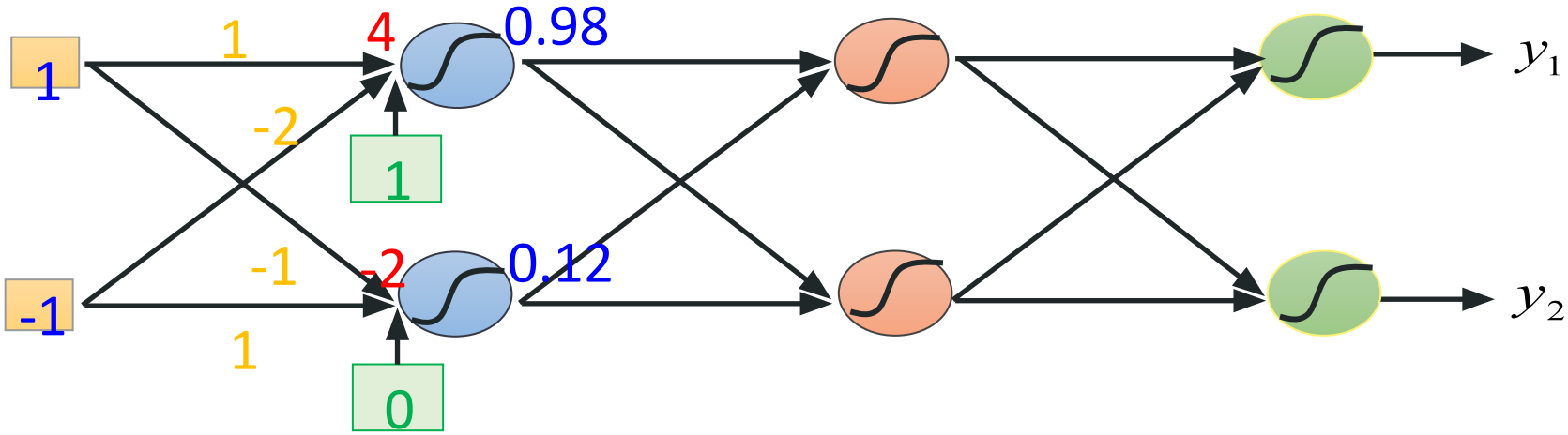


$f: R^2 \rightarrow R^2$

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

Different parameters define different function

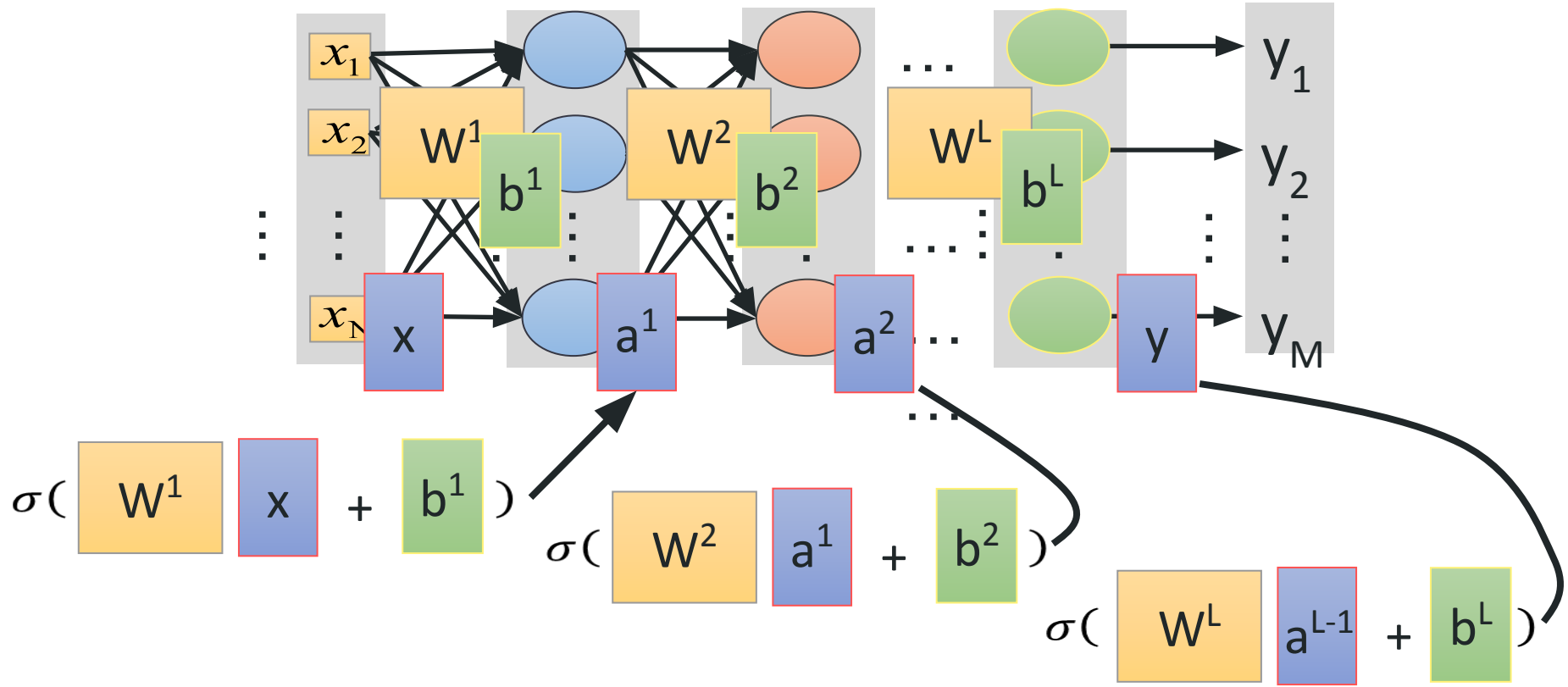
Matrix Operation



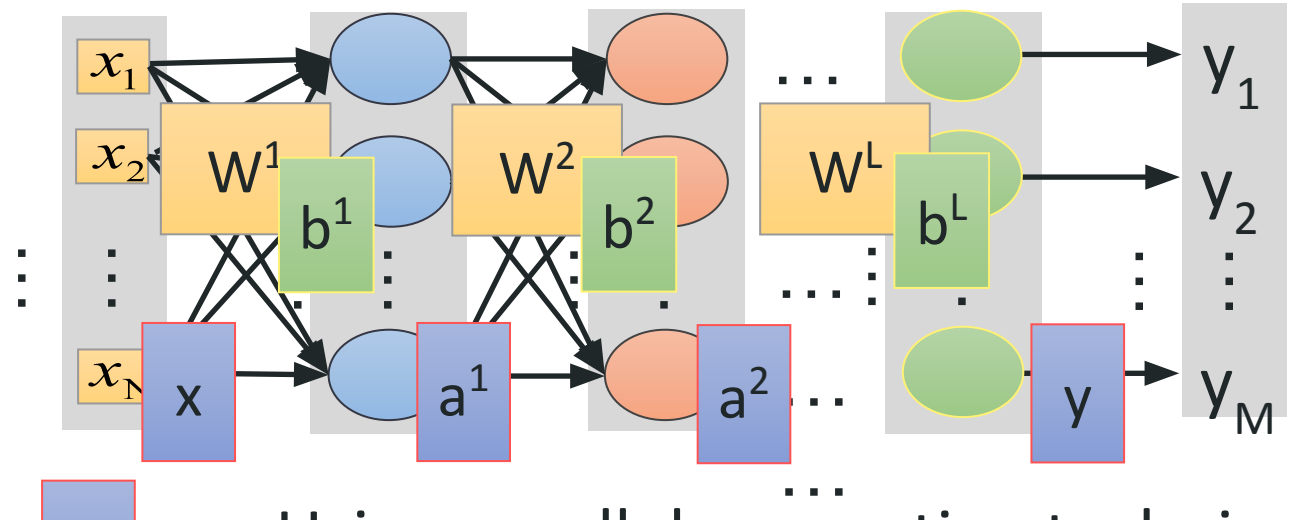
$$\sigma\left(\begin{bmatrix}1 & -2\\ -1 & 1\end{bmatrix}\begin{bmatrix}1\\ -1\end{bmatrix}+\begin{bmatrix}1\\ 0\end{bmatrix}\right)=\begin{bmatrix}0.98\\ 0.12\end{bmatrix}$$

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

Neural Network



Neural Network



$$y = f(x)$$

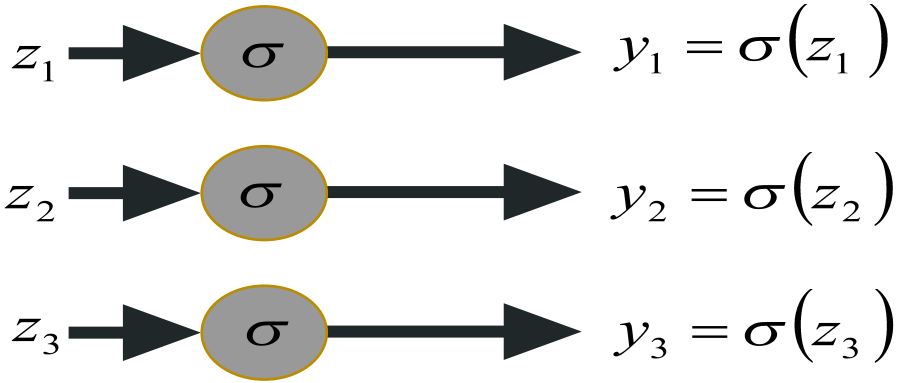
Using parallel computing techniques to speed up matrix operation

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

Softmax

- Softmax layer as the output layer

Ordinary Layer



In general, the output of network can be any value

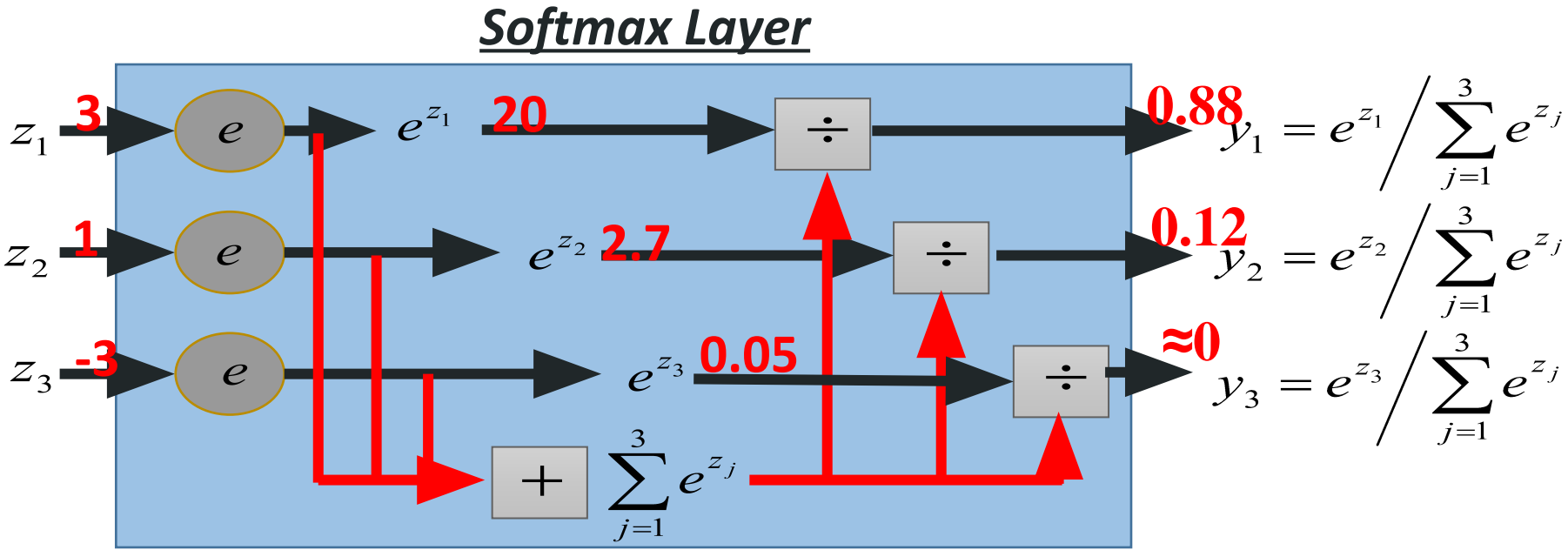
May not be easy to interpret

Softmax

- Softmax layer as the output layer

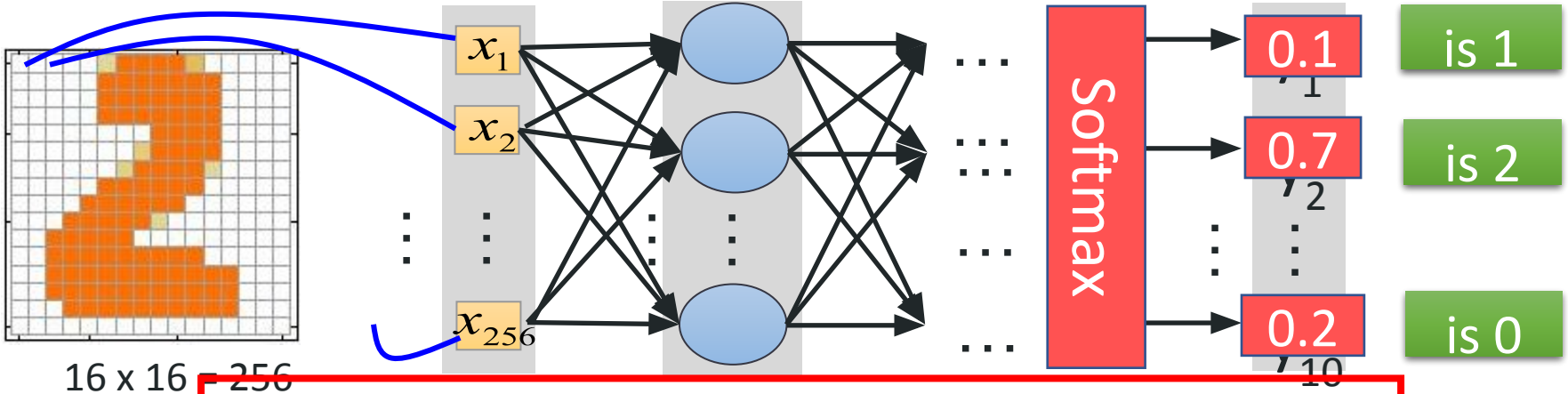
Probability:

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



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 .....

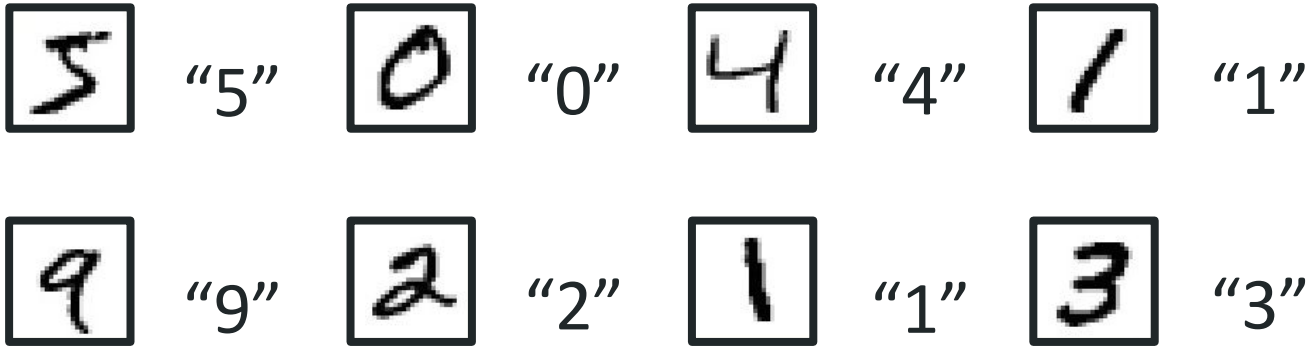
Input: value

Input: y_2 has the maximum value

How to let the neural network achieve this

Training Data

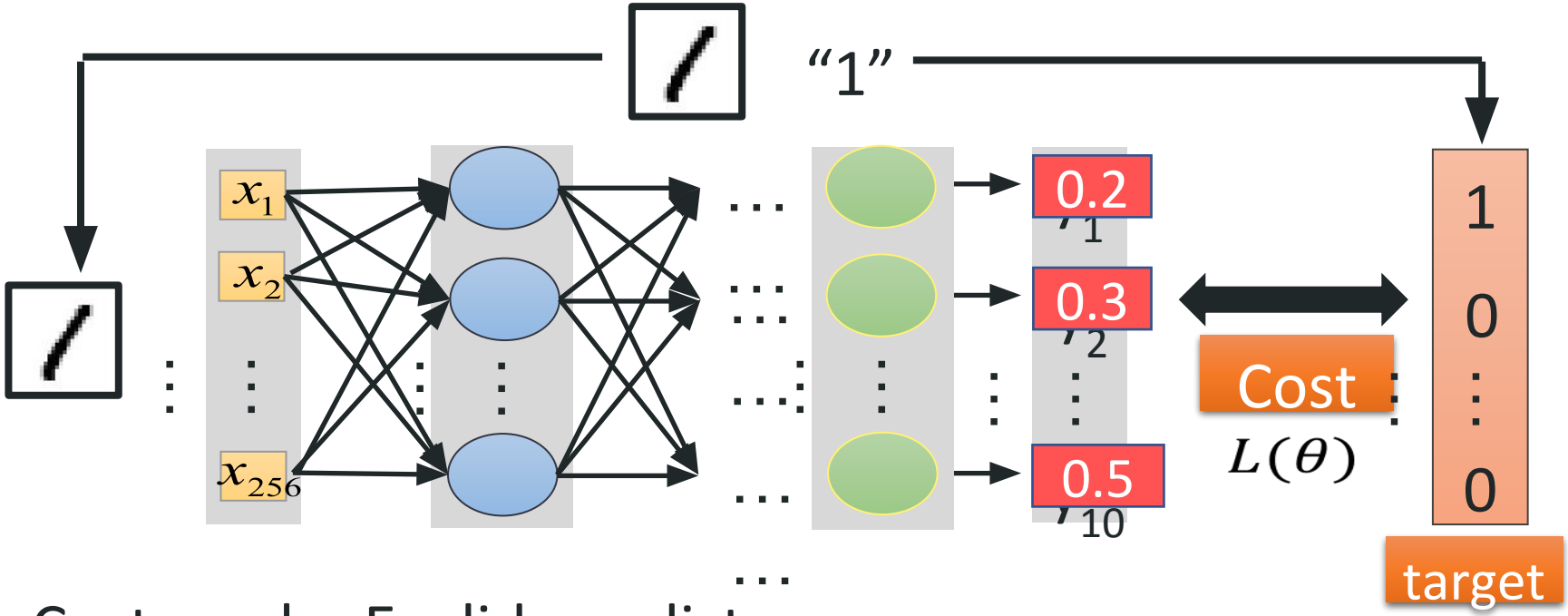
- Preparing training data: images and their labels



Using the training data to find the network parameters.

Cost

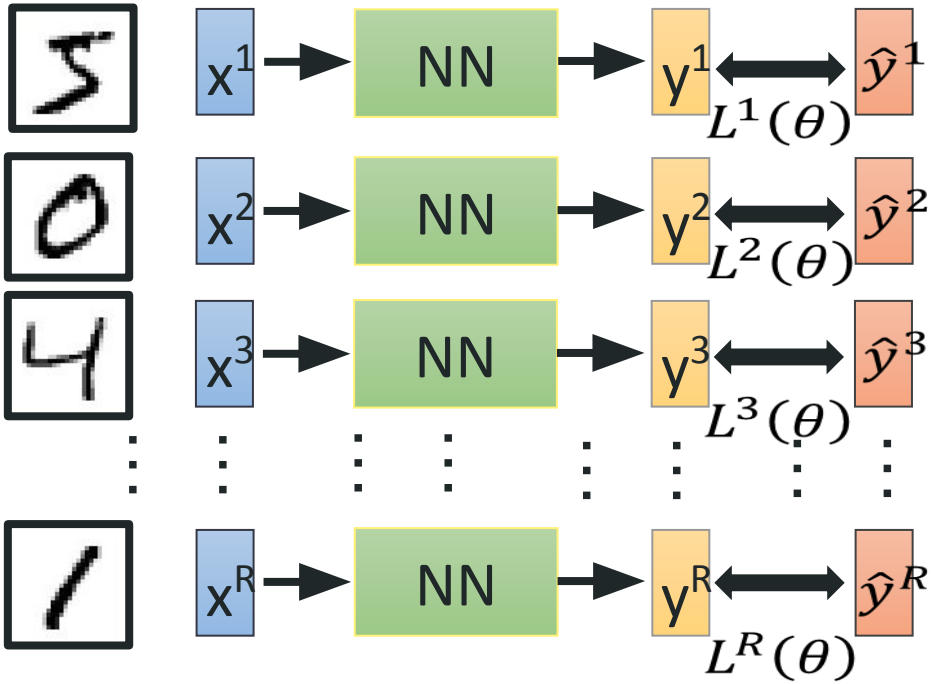
Given a set of network parameters θ, each example has a cost value.



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

Total Cost

For all training data …



Total Cost:

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

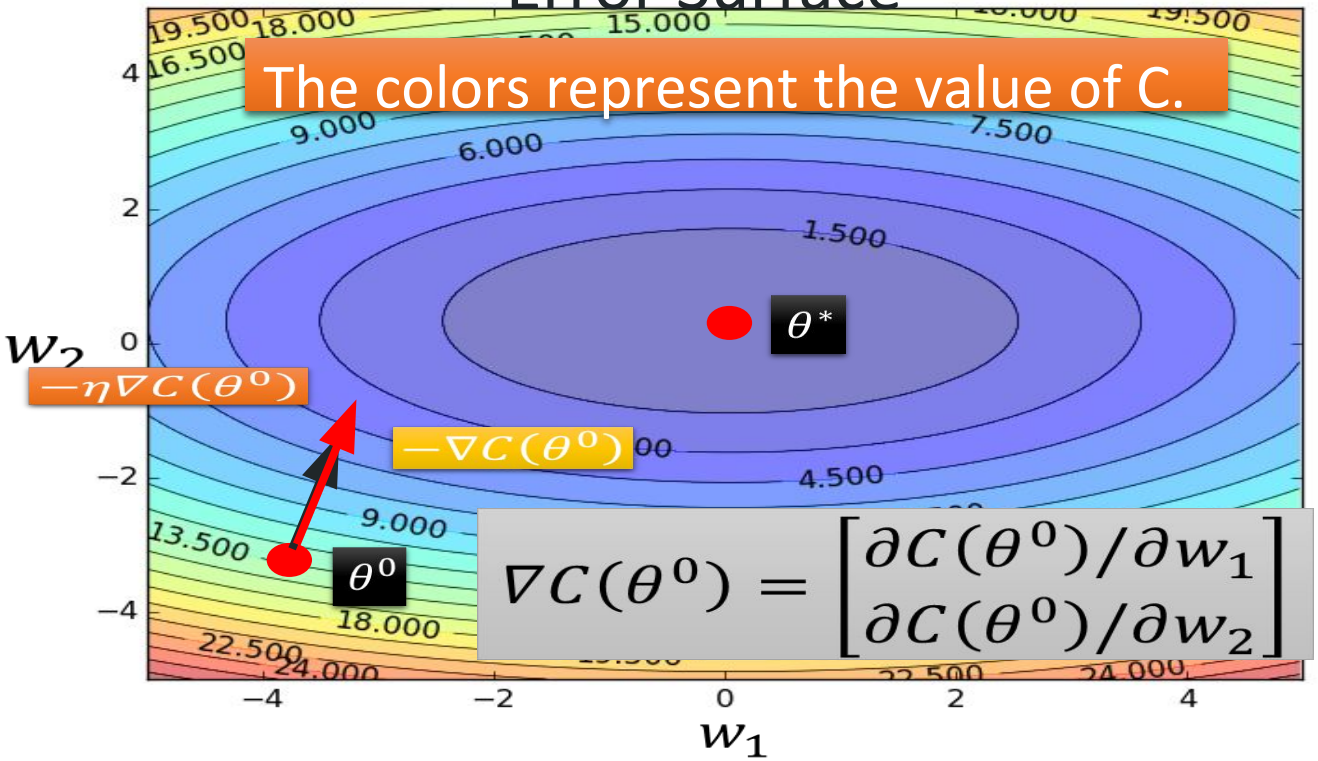
How bad the network parameters θ is on this task

Find the network parameters θ^* that minimize this value

Gradient Descent

Assume there are only two parameters w_1 and w_2 in a network. $\theta = \{w_1, w_2\}$

Error Surface



Randomly pick a starting point θ^0

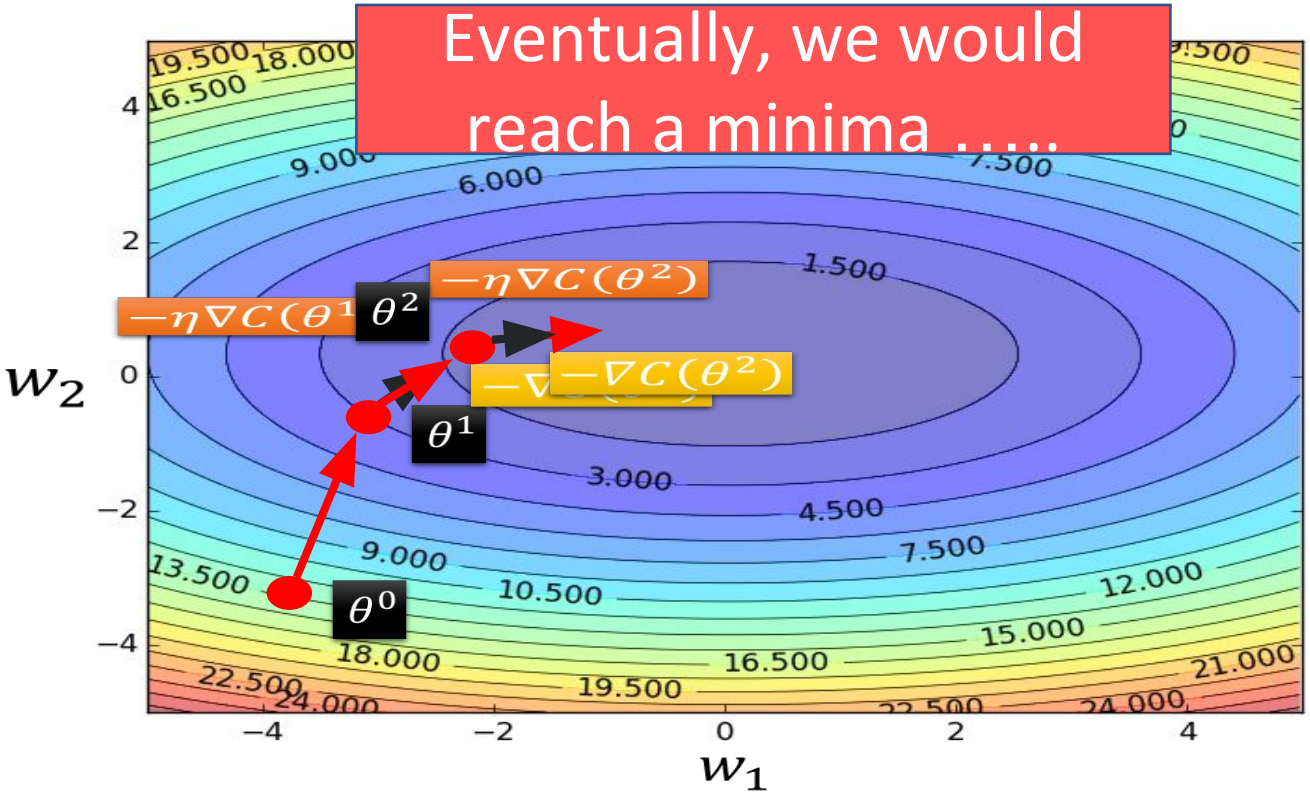
Compute the negative gradient at θ^0

→ $-\nabla C(\theta^0)$

Times the learning rate η

→ $-\eta\nabla C(\theta^0)$

Gradient Descent



Randomly pick a starting point θ^0

Compute the negative gradient at θ^0

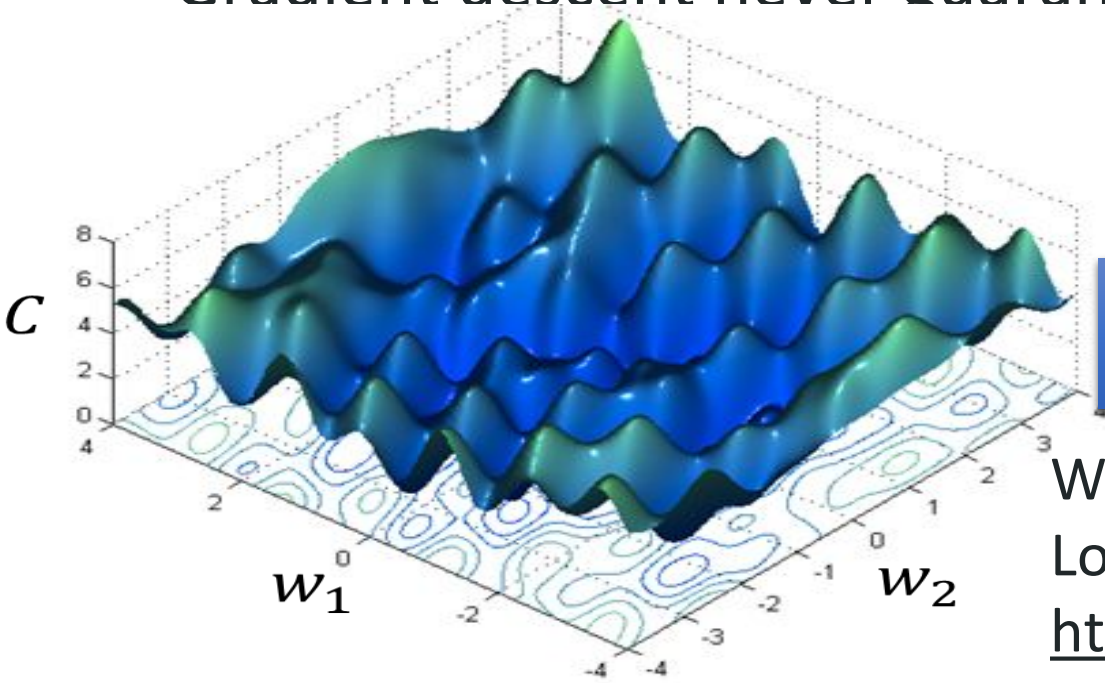
→ $-\nabla C(\theta^0)$

Times the learning rate η

→ $-\eta\nabla C(\theta^0)$

Local Minima

- Gradient descent never guarantee global minima



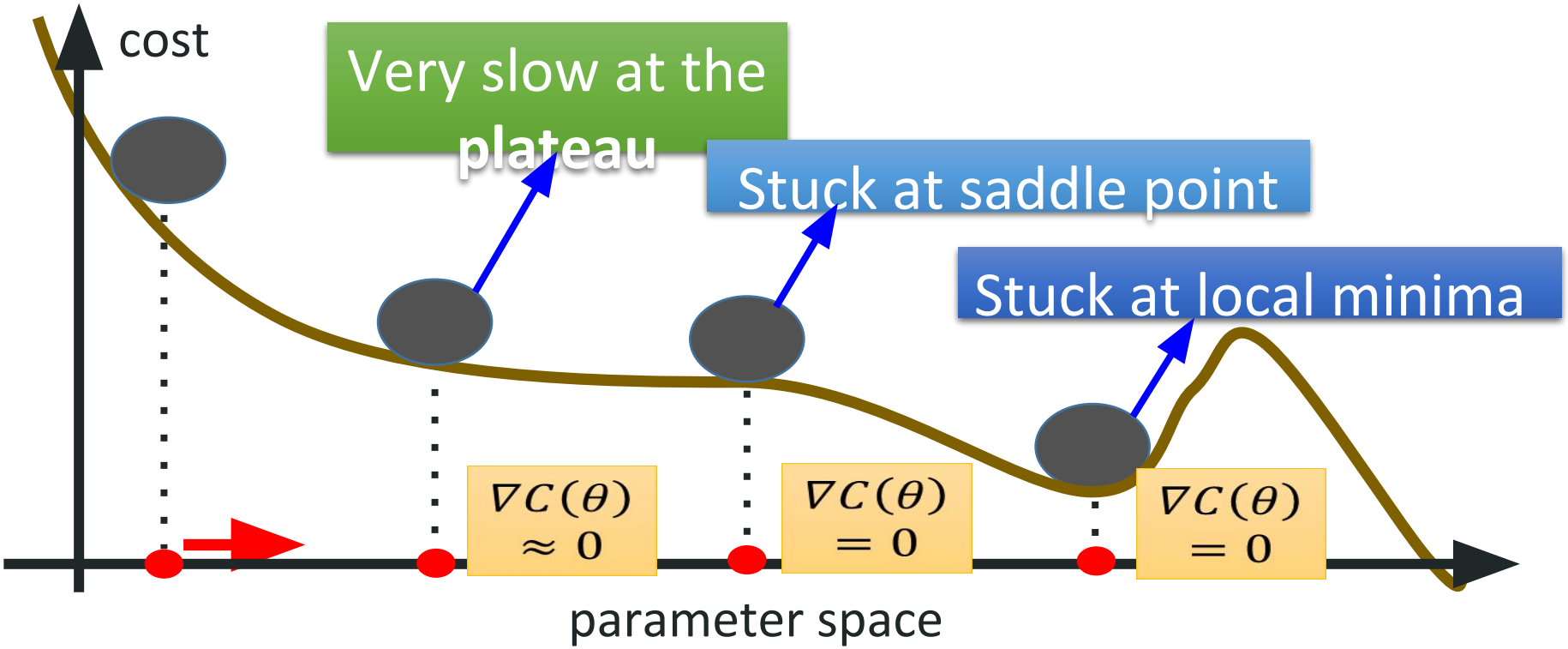
Different initial point θ^0

Reach different minima, so different results

Who is Afraid of Non-Convex Loss Functions?

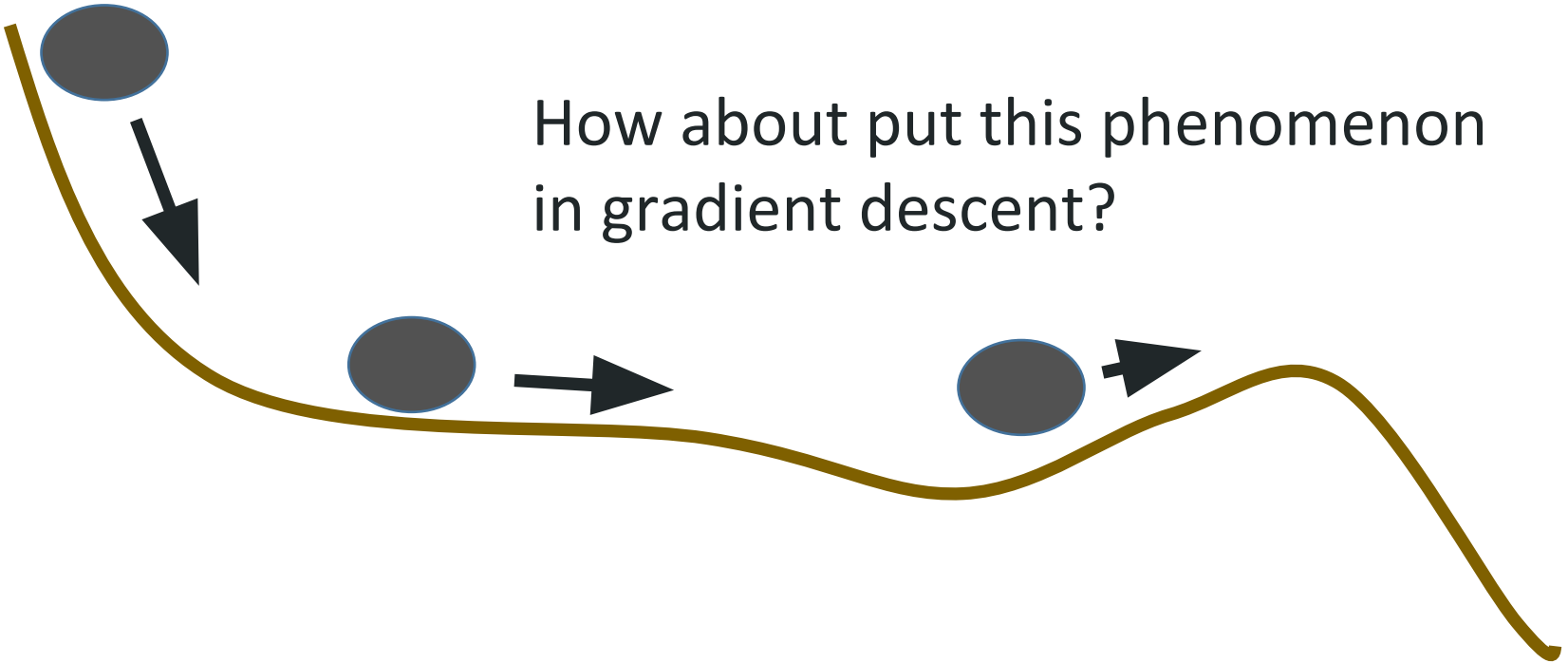
http://videolectures.net/eml07_lecun_wia/

Besides local minima ……



In physical world ……

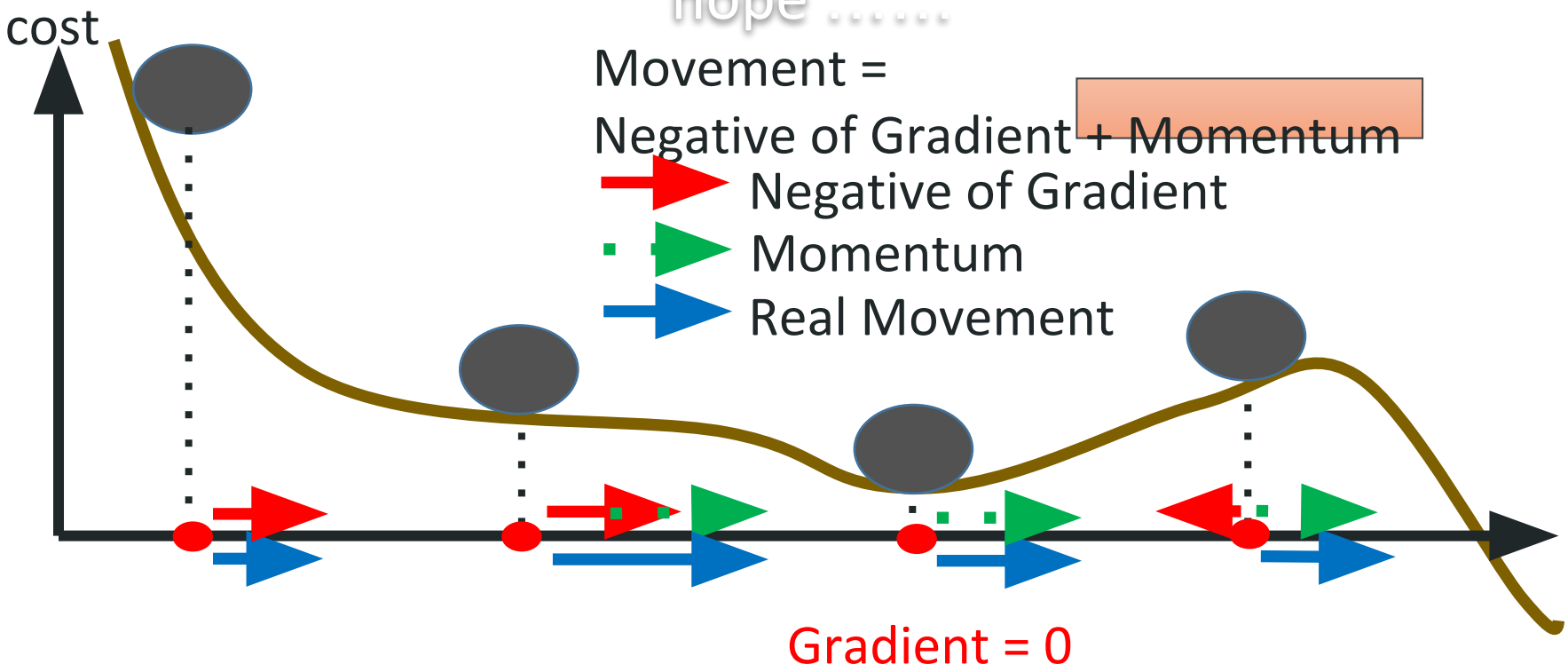
- Momentum



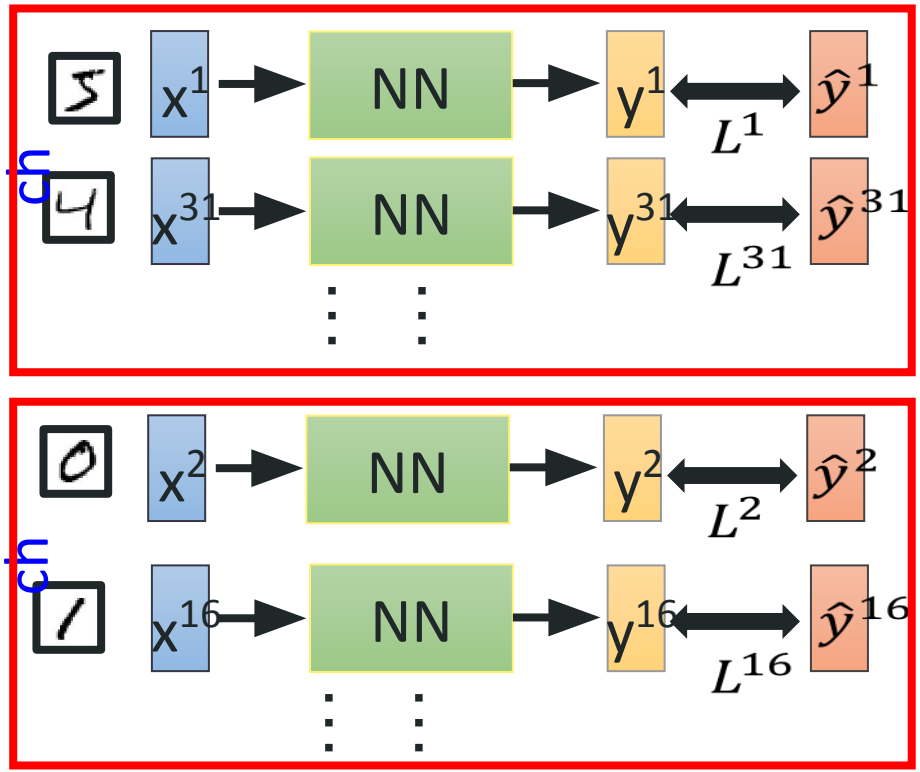
How about put this phenomenon in gradient descent?

Momentum

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



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

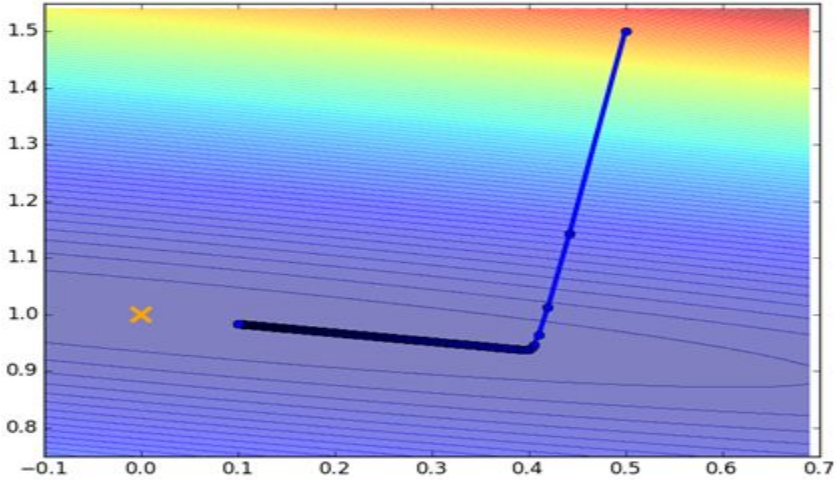
$$C = L^2 + L^{16} + \cdots$$

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

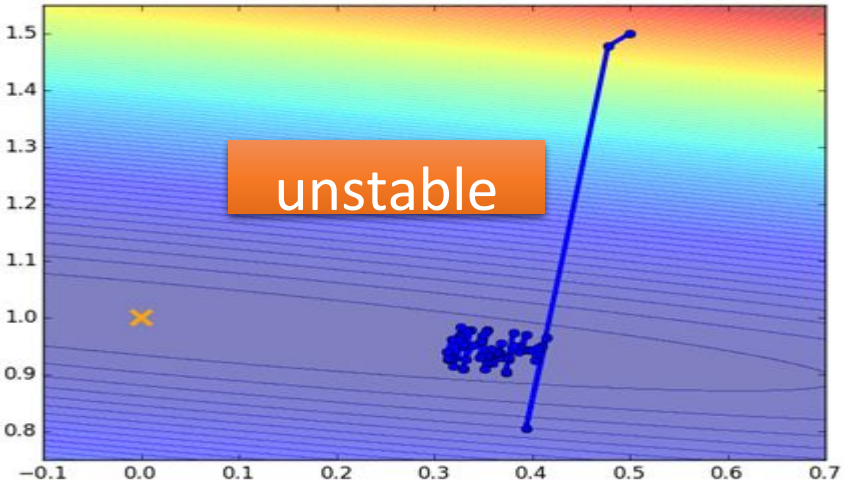
C is different each time when we update parameters!

Mini-batch

Original Gradient Descent



With Mini-batch

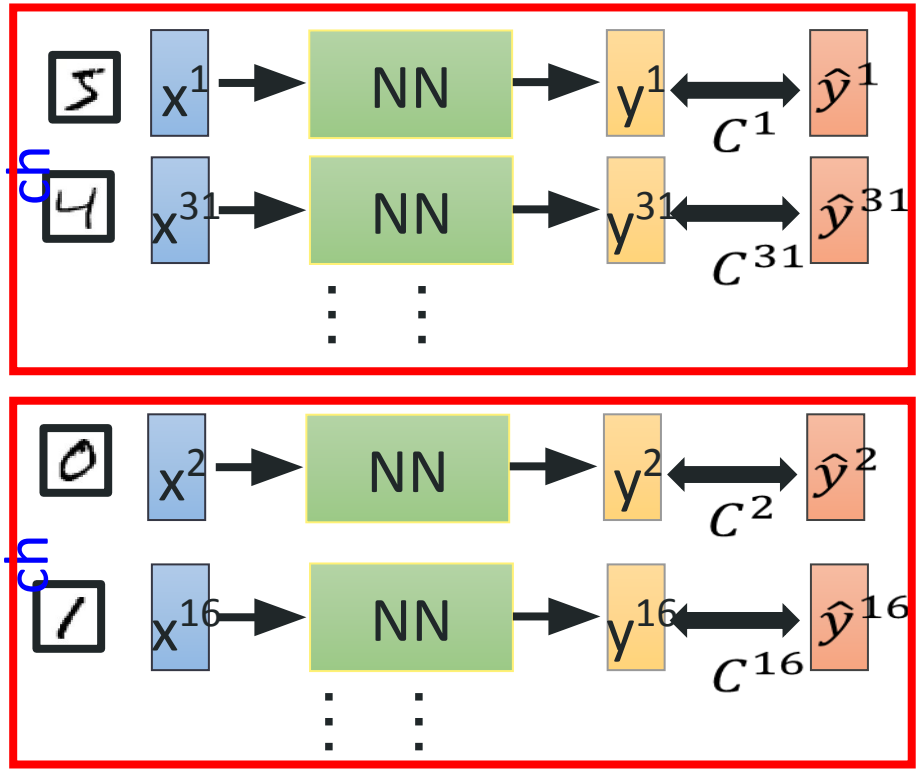


The colors represent the total C on all training data.

Mini-batch

Faster

Better!



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

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

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

- Pick the 2nd batch

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

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

- Until all mini-batches have been picked

one epoch

Repeat the above process

Back propagation principle

The back propagation algorithm is a generalization of the *delta rule* for training multilayer networks (MLN). This algorithm updates the weights w_i of the network by means of successive iterations, that minimize the cost function of the error $\boldsymbol{E}$.

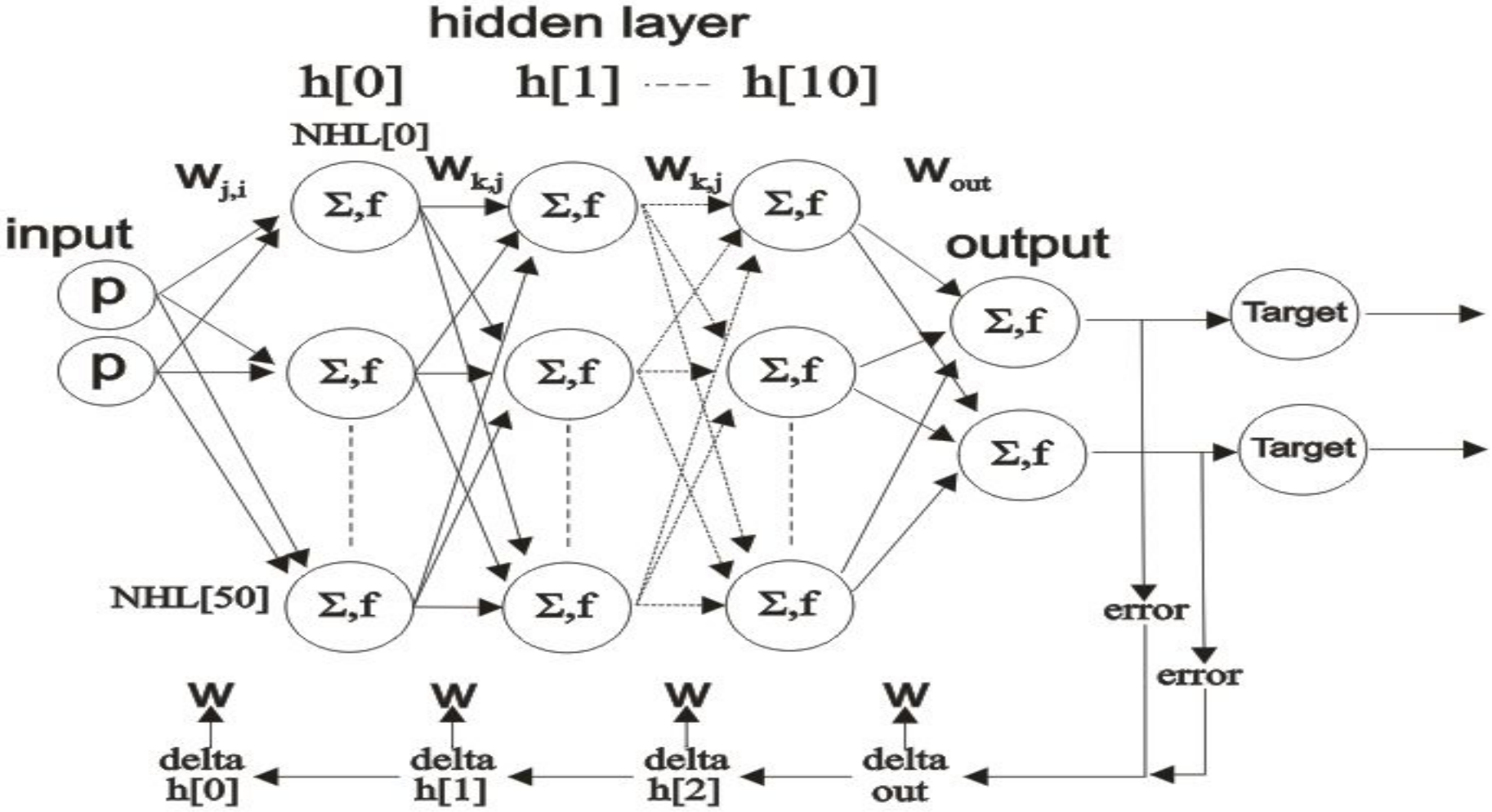
The minimization of the error is obtained using the gradient of the cost function, which consists of the first derivative of the function with respect to all the weights w_i namely:

$$\frac{\partial}{\partial w_i}(E)$$

On the basis of this gradient the weights will be updated with the following mechanism:

$$w_i = w^0{}_i - \eta \frac{\partial}{\partial w_i}(E)$$

where w_i are the weights updated , $w^0{}_i$ are the random weights that initiate the process of adjustment and η is the learning rate.


hidden layer
h[0]
h[1]
h[10]
NHL[0]
NHL[50]
$w_{j,i}$
$w_{k,j}$
$w_{k,j}$
w_{out}
input
p
p
output
Σ,f
Target
Target
error
error
w
delta h[0]
delta h[1]
delta h[2]
delta out

Applications Automotive


head of a giraffe
legs of a zebra
red and white sign
white tennis shoes
hands holding a phone
front wheel of a bus
STOP


man wearing a black shirt
red shirt on a man
elephant is standing
elephant is brown
large green trees
roof of a building
trunk of an elephant
green trees in the background
rocks on the ground
leg of an elephant
ball is white
ground is visible
leg of an elephant
ground is brown
elephant is standing
shadow on the ground

Tell the audience what you expect to happen...

References

https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/

http://machinelearningmastery.com/what-is-deep-learning/

https://www.cs.princeton.edu/courses/archive/spring16/cos495/slides/Intro_to_course.pdf