Neural Network and TensorFlow

Dapeng Hu, Qinglong Tian, Haozhe Zhang, Min Zhang

April 23, 2017

STAT 580 Statistical Computing
Department of Statistics
Iowa State University

Content

1. History of Neural Network
2. Neural Network
3. Convolutional Neural Network
4. TensorFlow

History of Neural Network

History of Neural Networks

- The 1940s: The Beginning of Neural Networks
- The 1950s and 1960s: The First Golden Age of Neural Networks
- The 1970s: The Quiet Years
- The 1980s: Renewed Enthusiasm

The 1940s: The Beginning of Neural Networks

McCulloch and Pitts, 1943

- Employed logic and mathematical notion of computation to explain how neural mechanism might realize mental functions.
- Commonly regarded as the inception of artificial neural networks.

The 1940s: The Beginning of Neural Networks

McCulloch and Pitts, 1943

- Employed logic and mathematical notion of computation to explain how neural mechanism might realize mental functions.
- Commonly regarded as the inception of artificial neural networks.

The 1940s: The Beginning of Neural Networks

Hebb's Learning Rule:

- Connectionism
- A method of determining how to alter the weights between model neurons.

to work with another Russian emigré, Leonid Andreyev. Hebb conditioned dogs and became less impressed with Pavlovian techniques. After much soul-searching as to whether he should continue in psychology, he decided in 1934 to burn his boats, borrow money and go to Chicago to continue his doctoral research under Karl S. Lashley.

The elder scientist was to exert a profound influence on Hebb's approach, above all in his emphasis on physiology. Lashley had never doubted that to understand behavior one must first understand the brain. As a lab boy in 1910, he had salvaged slides of a frog brain from the trash heap and tried to find in the neural connections some clue to frog behavior. Lashley performed experiments to detect memory traces in the brain, inventing techniques for making brain lesions and measuring their location and extent. By around 1930 he had become convinced that memories could not be stored in a single region of the brain but must be spread throughout. In 1934, when Hebb went to Chicago, Lashley was concentrating on the study of vision.

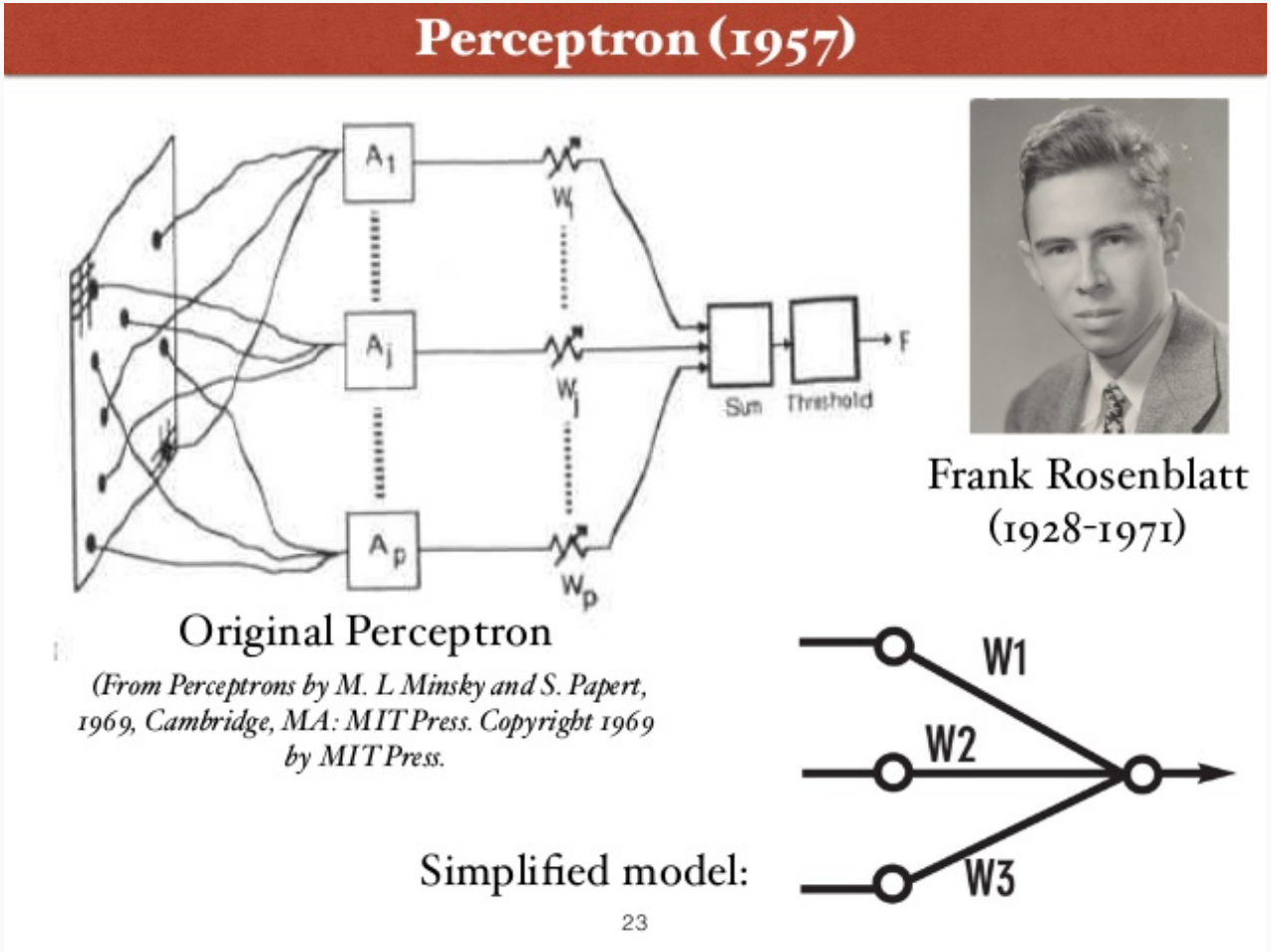
A year later Lashley was offered a professorship at Harvard University and managed to take Hebb along. Hebb had to start his research from scratch, and having only enough money for one more year, he sought an experiment that could support a thesis no matter how it came out. He contrived to adapt his interest in the nature-nurture question to Lashley's vision project by investigating the effects of early experience on the development of vision in the rat.

Contrary to the empiricist ideas of his master's thesis, Hebb found that rats reared in complete darkness could distinguish the size and brightness of patterns as accurately as rats reared normally. This finding indicated that the organization of the visual system was innate and independent of environmental cues, a view coinciding with that of the Gestalt school, to which Lashley was sympathetic [see "The Legacy of Gestalt Psychology," by Irvin Rock and Stephen Palmer; SCIENTIFIC AMERICAN, December 1990]. What Hebb did not notice, although the results were included in a paper he published at the time, was that the dark-reared rats took much longer than normal rats to learn to distinguish vertical from horizontal lines. Only many years later, after he had again changed his ideas about the relative importance of innate and learned mechanisms, did he appreciate the significance of this result.

Hebb received his Ph.D. from Harvard in the middle of the Depression, when there were no jobs in physiological psychology to be had. He therefore stayed on for a year as a teaching assistant, a post that enabled him to continue his work with Lashley. In 1937 there was still no improvement in the job market, but Hebb's luck held out. His sister was taking her Ph.D. in physiology at McGill and heard that Wilder Penfield, a surgeon who had just established the Montreal Neurological Institute there, was looking for someone to study the consequences of brain surgery on the behavior of patients. She passed

The 1950s and 1960s: First Golden Age of Neural Networks

Perceptron (Rosenblatt's perceptron), 1957



The 1970s: The Quiet Years

Marvin Minsky

- "Perceptron", 1969.
- "Fatal flaw" of the perceptron; inability to solve non-linear problems.

The 1970s: The Quiet Years

Kunihiko Fukishima

- Neocognitron

Teuvo Kohonen

- Self-Organizing Map (SOM) algorithm

The 1980s: Renewed Enthusiasm

John Hopfield

- Hopfield Neural Network, 1982

Geoffrey Hinton, etc.

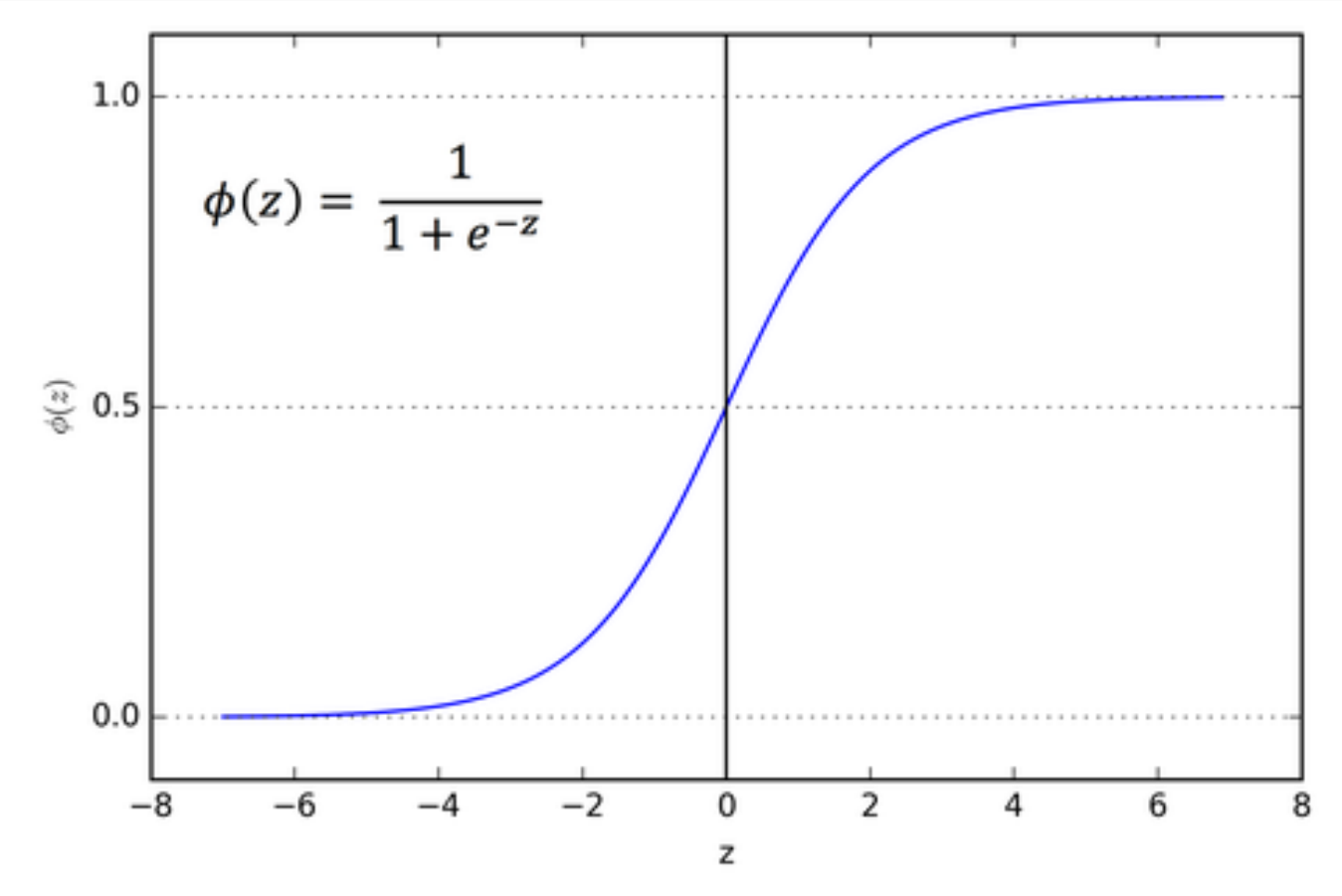
- Blotzmann Machine
- Back-propagation (BP) algorithm

Neural Network

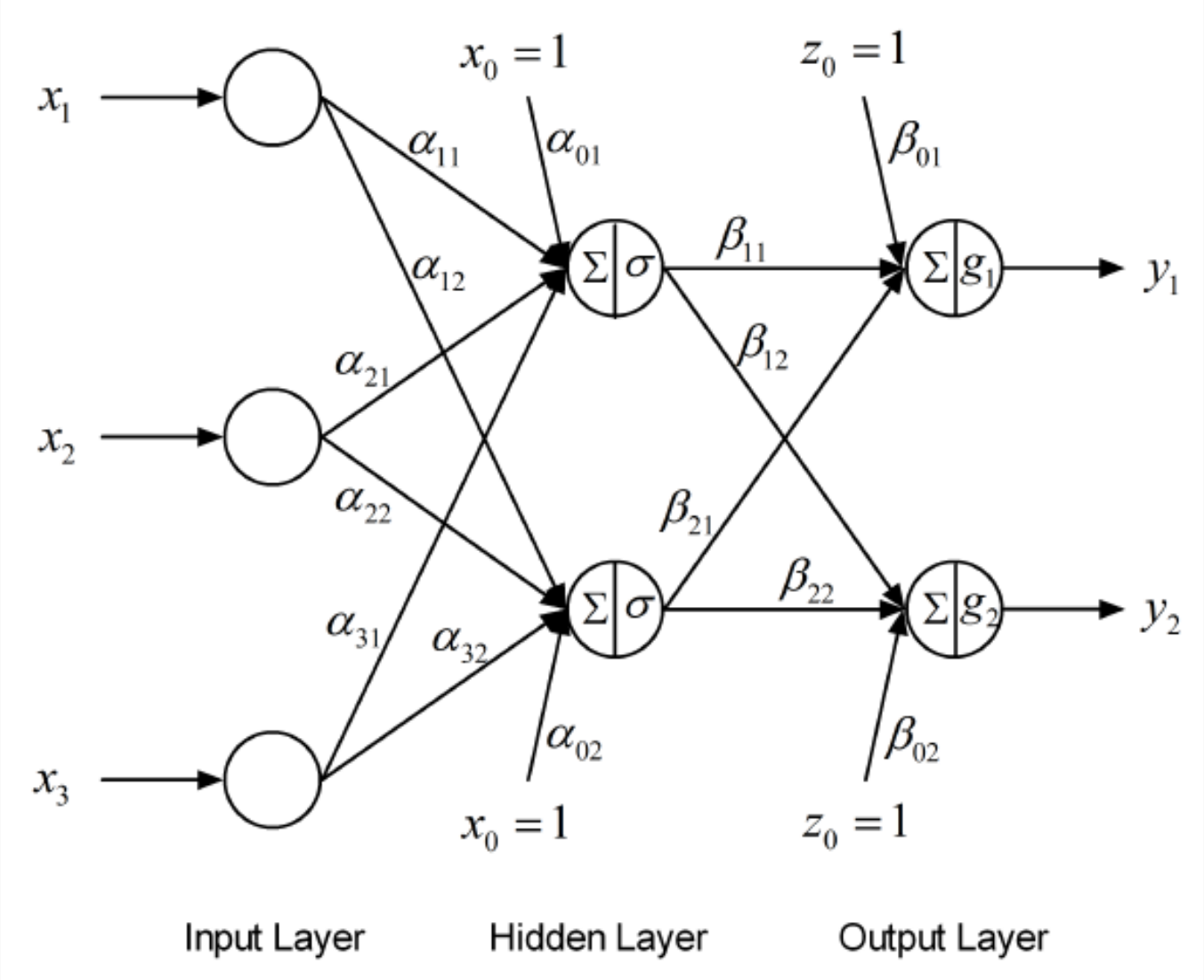
A Neuron Model

- The weights of the neuron are the parameters of the model.
- The input is computed as a weighted sum of the inputs.
- The activation function determines the output and can be different functions.
- The output is obtained by applying the activation function to the input.
- How does the neuron work?

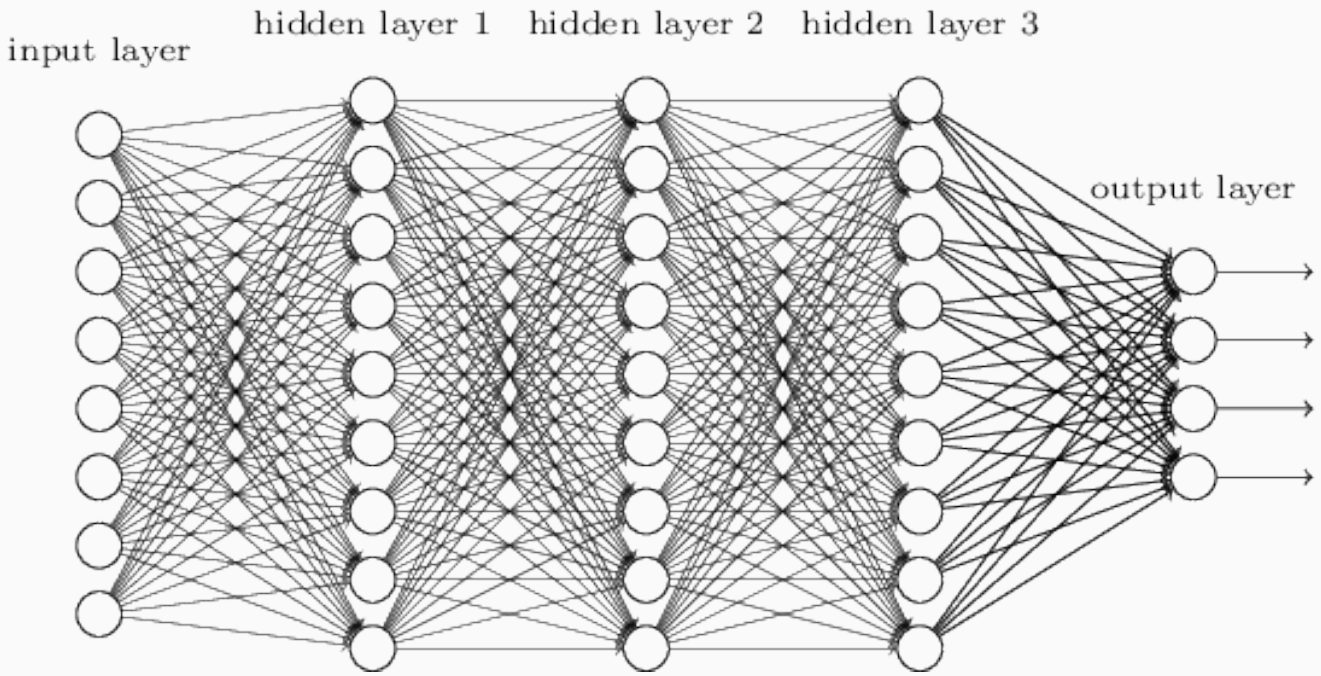
A Neuron Model



Single-layer Neural Network



Multiple-layer Neural Network



Delta Learning Rule

- Based on gradient descent algorithm.
- Only applicable for continuous activation function.
- The learning signal r is called delta and defined as:

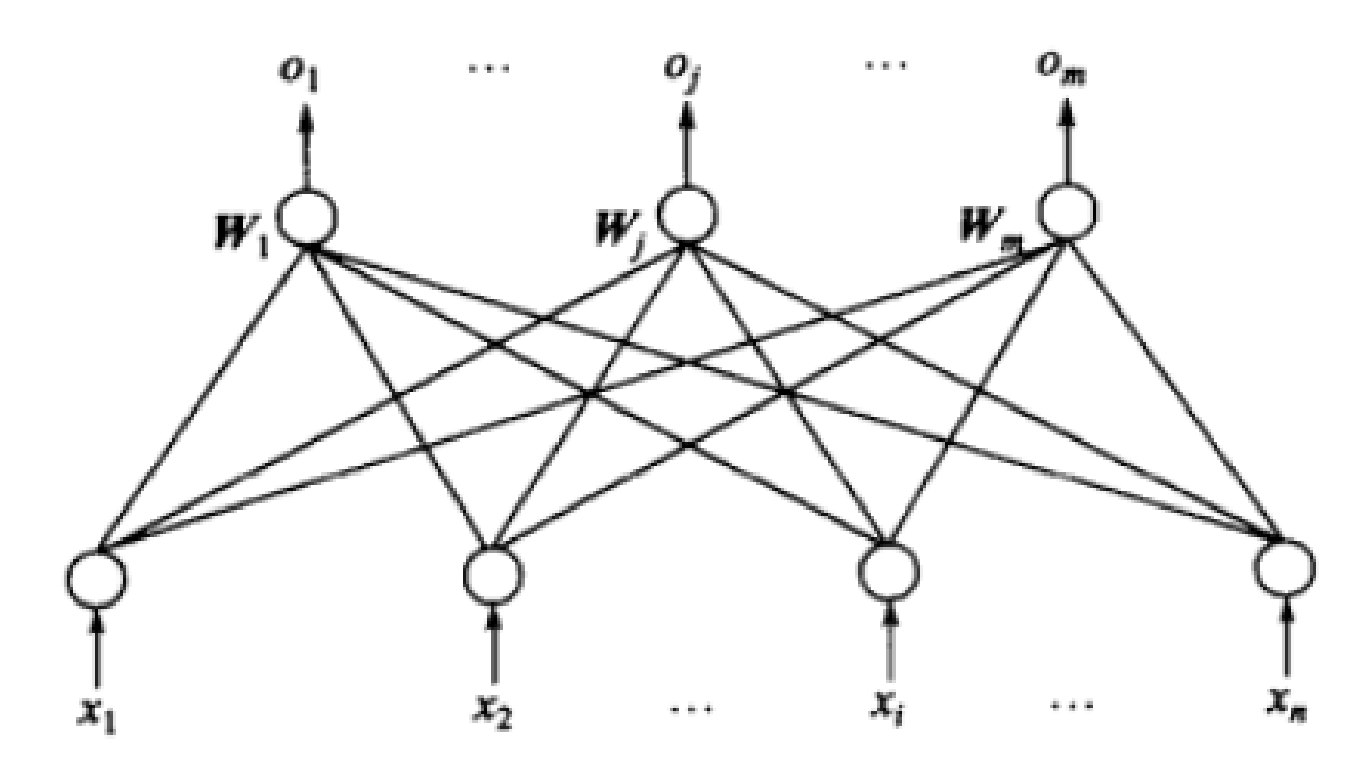
$$r = \{d_j - f(\boldsymbol{W}_j^T \boldsymbol{X})\} f'(\boldsymbol{W}_j^T \boldsymbol{X}) \approx (d_j - o_j) f'(\text{net}_j)$$

- The weights will be adjusted as:

$$\Delta \boldsymbol{W}_j = \eta (d_j - o_j) f'(\text{net}_j) \boldsymbol{X}$$

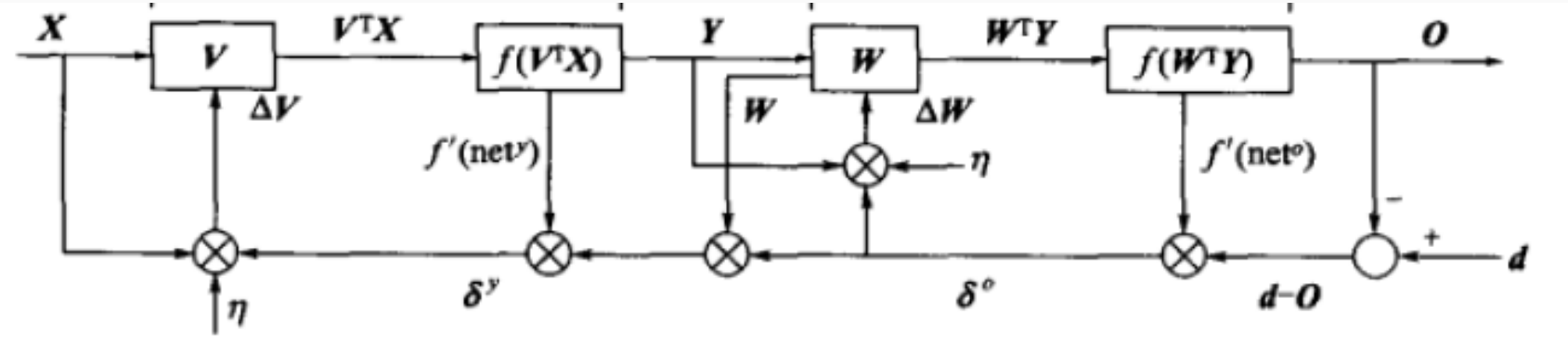
Backpropagation Algorithm

- BP algorithm is a process of training a neural network
- Based on delta learning rule



Backpropagation Algorithm

The flow of signal in BP algorithm:



More Hidden Layers?

- The signal is getting weaker
- Non-convex problem
- Slow Convergence
- Weakness of Gradient Descent: Local Optima

Breakthrough

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science, 313(5786), 504-507.

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. We describe a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network

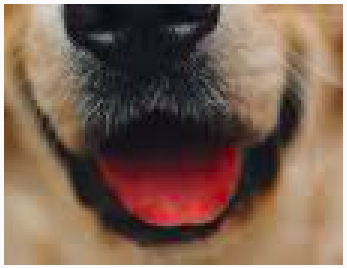
Convolutional Neural Network

CNN


THE LEAD
JAKE TAPPER
FIRST ON CNN
ACTRESS FROM CRUZ AD GIVES FIRST TV INTERVIEW
CNN
4:20 PM ET
THE LEAD

Why CNN?

- Fully connected neural network can bring us too many parameters.
- Some patterns are much smaller than the whole image.



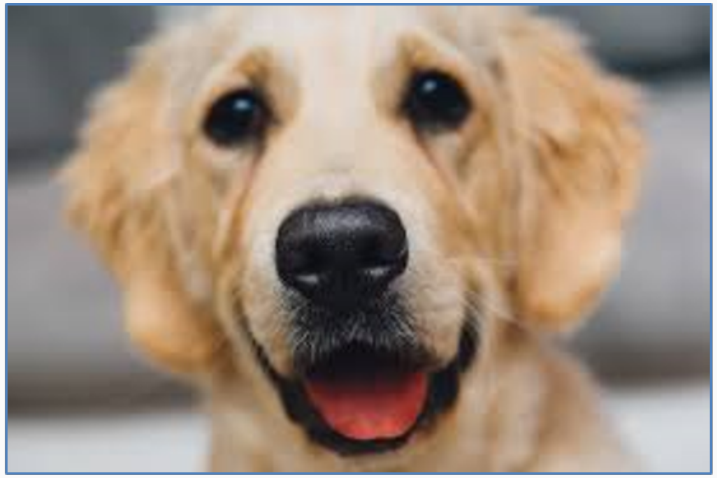
Why CNN?

The same pattern appears in different regions.

Why CNN?

Subsampling the pixels will not change the object.

Whole Structure



Convolution

Max Pooling

Convolution

Max Pooling

Flatten

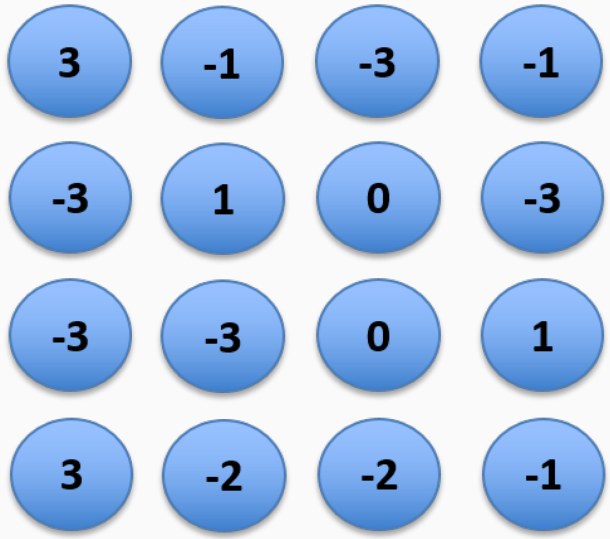
Fully connected forward network

How convolution works?

Stride = 1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

1	-1	-1
-1	1	-1
-1	-1	1



How convolution works?

Stride = 1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

1	-1	-1
-1	1	-1
-1	-1	1

Filter

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

How convolution works?

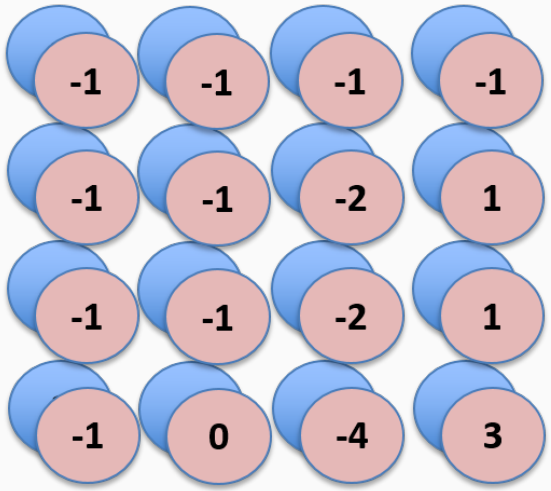
Stride = 1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

1	-1	-1
-1	1	-1
-1	-1	1

-1	1	-1
-1	1	-1
-1	1	-1

2nd filter

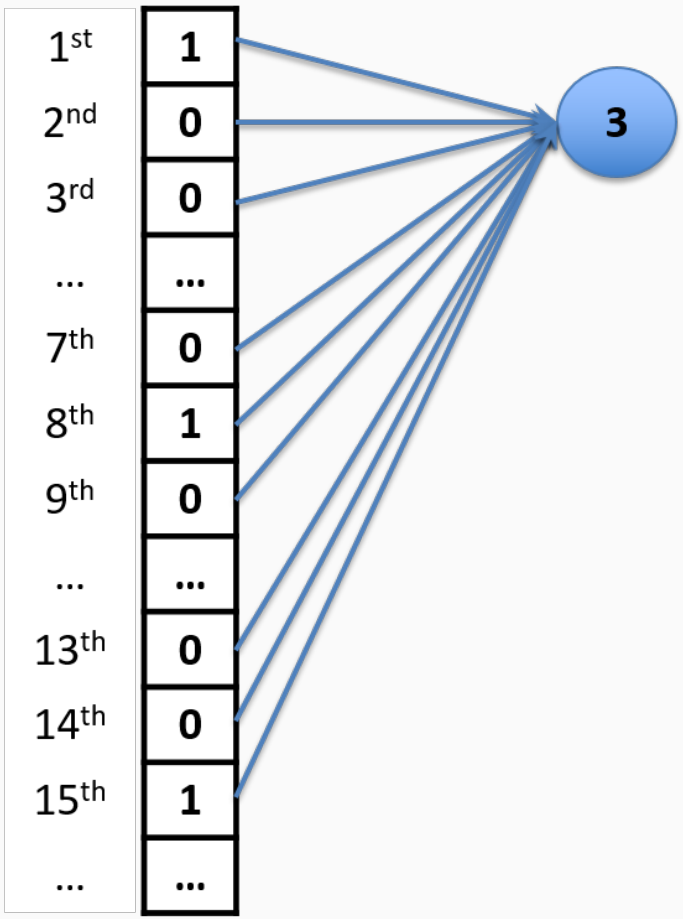


Feature map

Connection between CNN and Neural Network

1	-1	-1
-1	1	-1
-1	-1	1

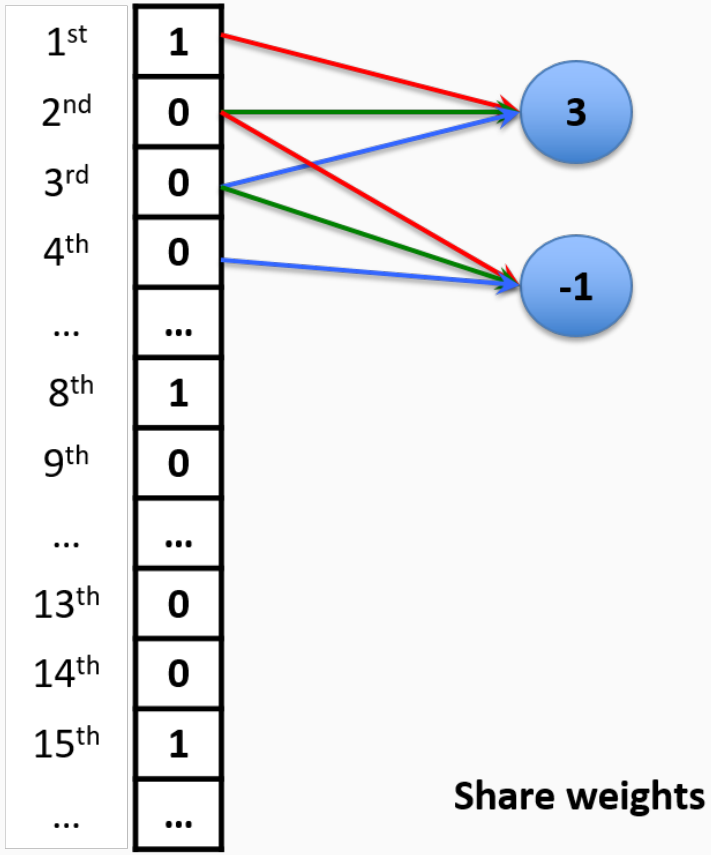
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0



Connection between CNN and Neural Network

1	-1	-1
-1	1	-1
-1	-1	1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0



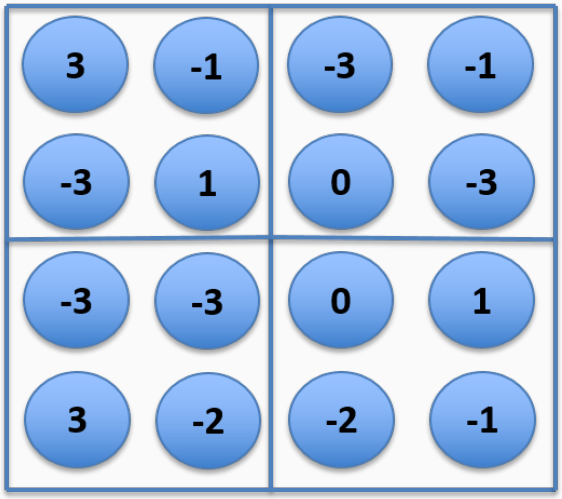
Max pooling

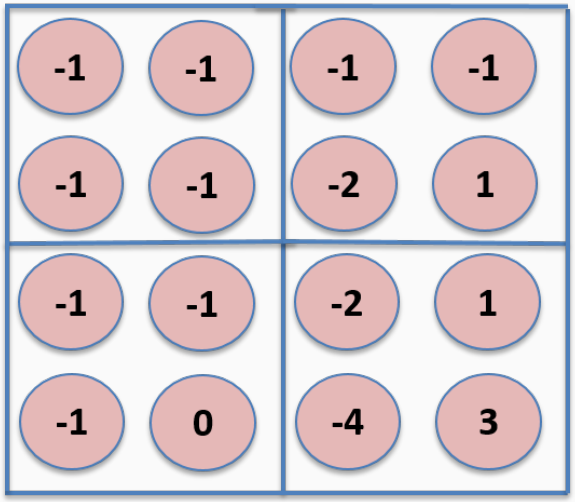
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2





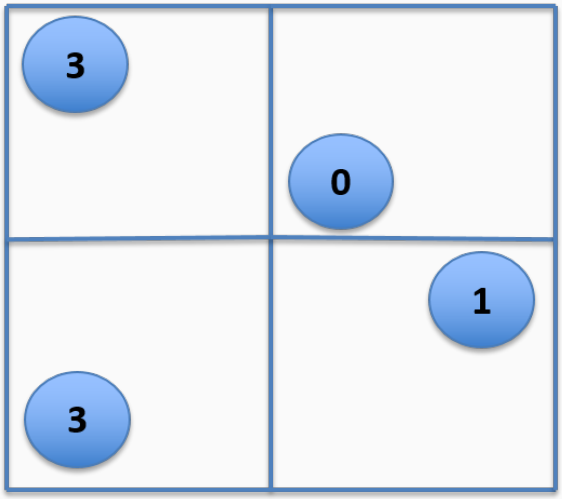
Max pooling

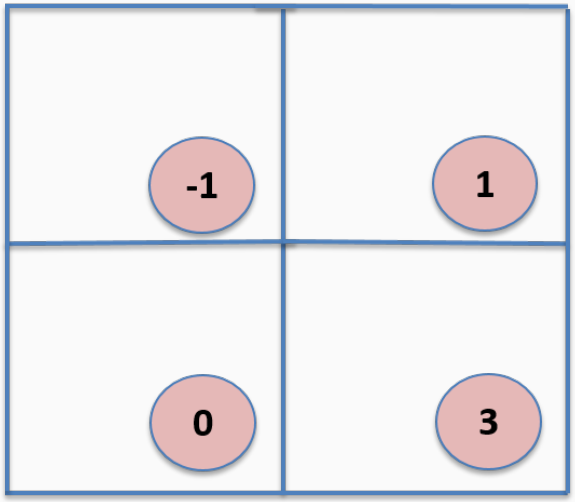
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2



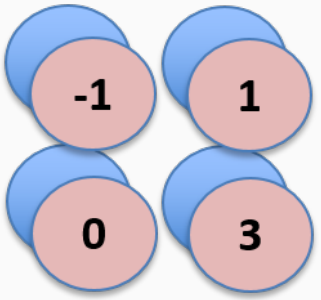


1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Convolution

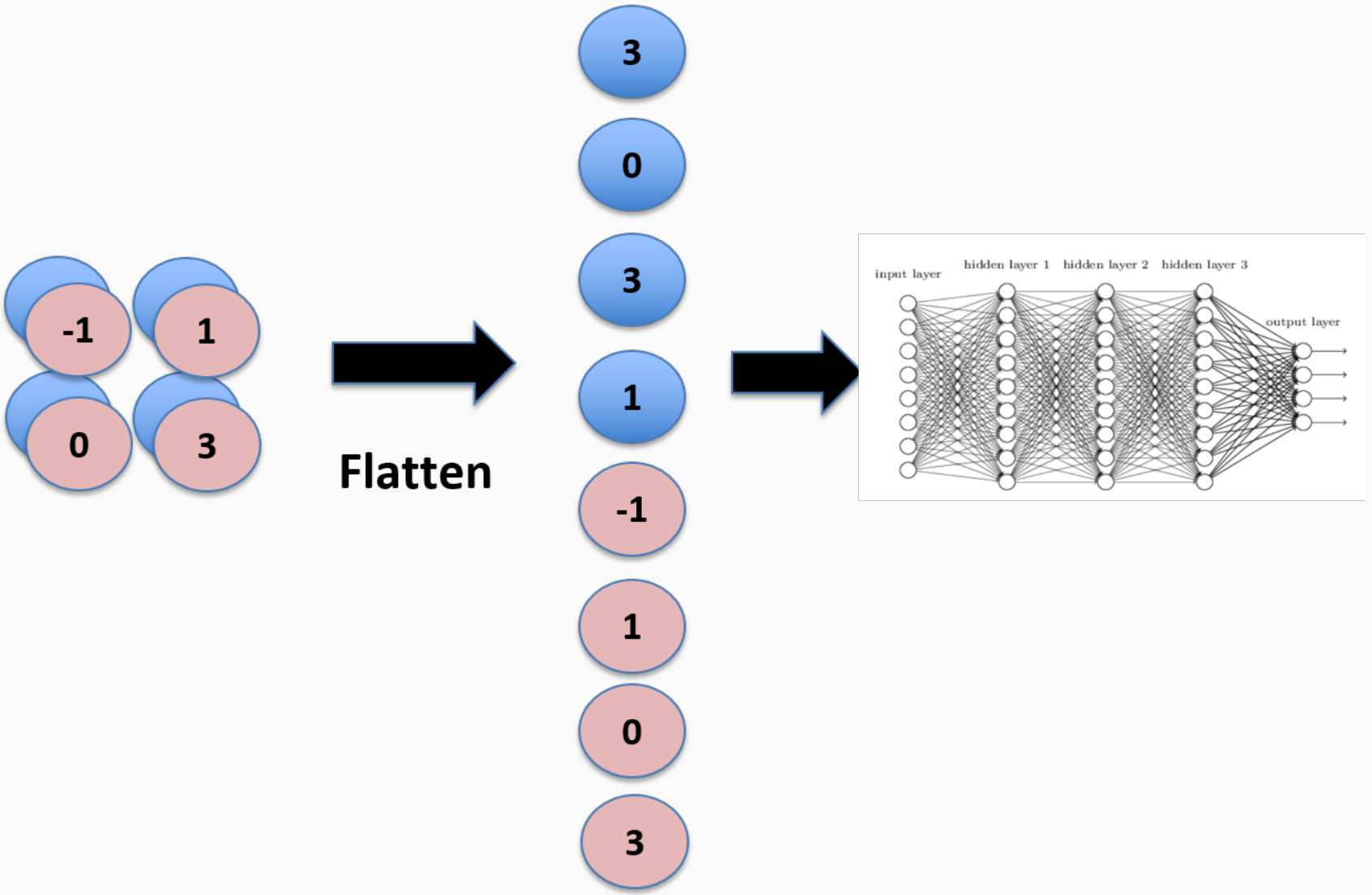
Max Pooling

Smaller image 2x2



Each filter is a channel

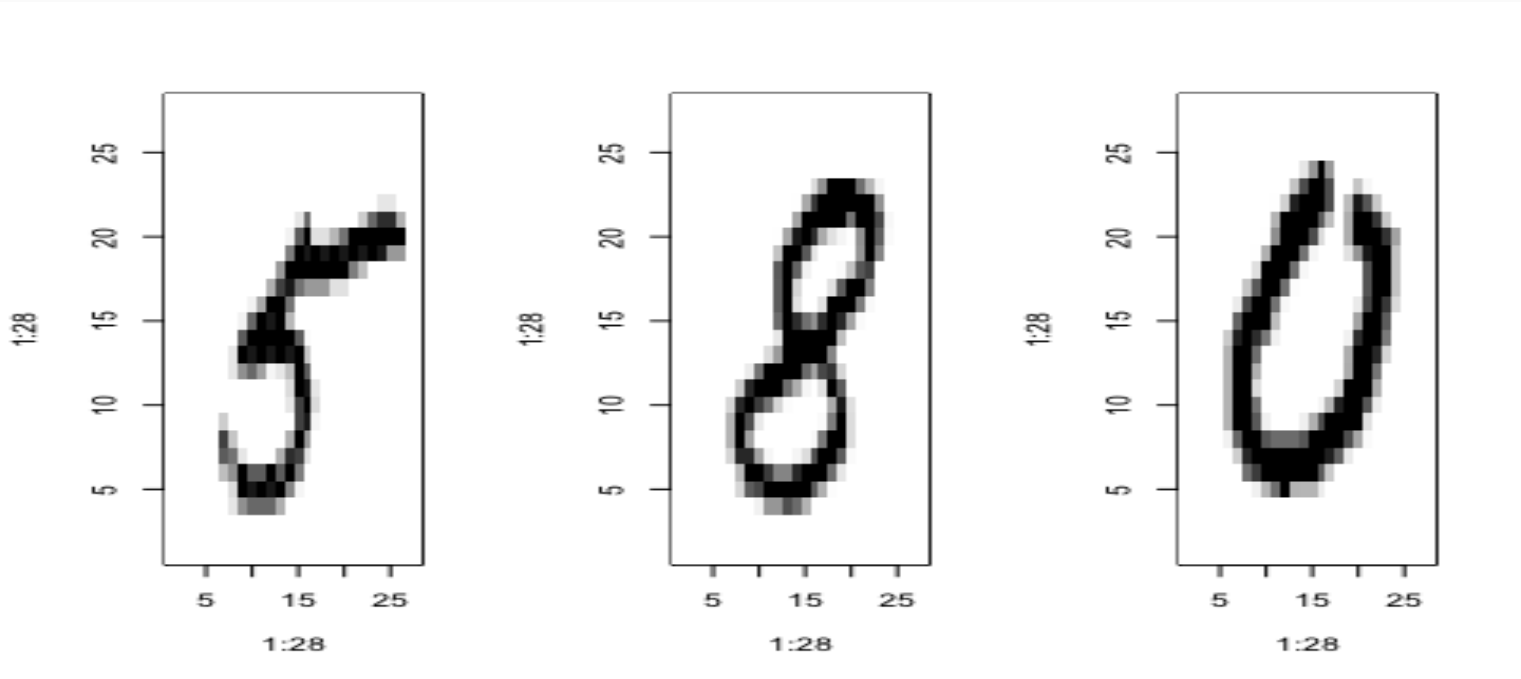
Flatten


-1
1
0
3
Flatten
3
0
3
1
-1
1
0
3
input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer

Example of CNN: MNIST

The MNIST dataset contains images of handwritten digits like below. Each image is 28 pixels by 28 pixels. It also includes labels for each image, telling us which digit it is. Next, we are going to train a model to look at images and predict which digits they are.

First Convolutional Network Demo from 1993



TensorFlow

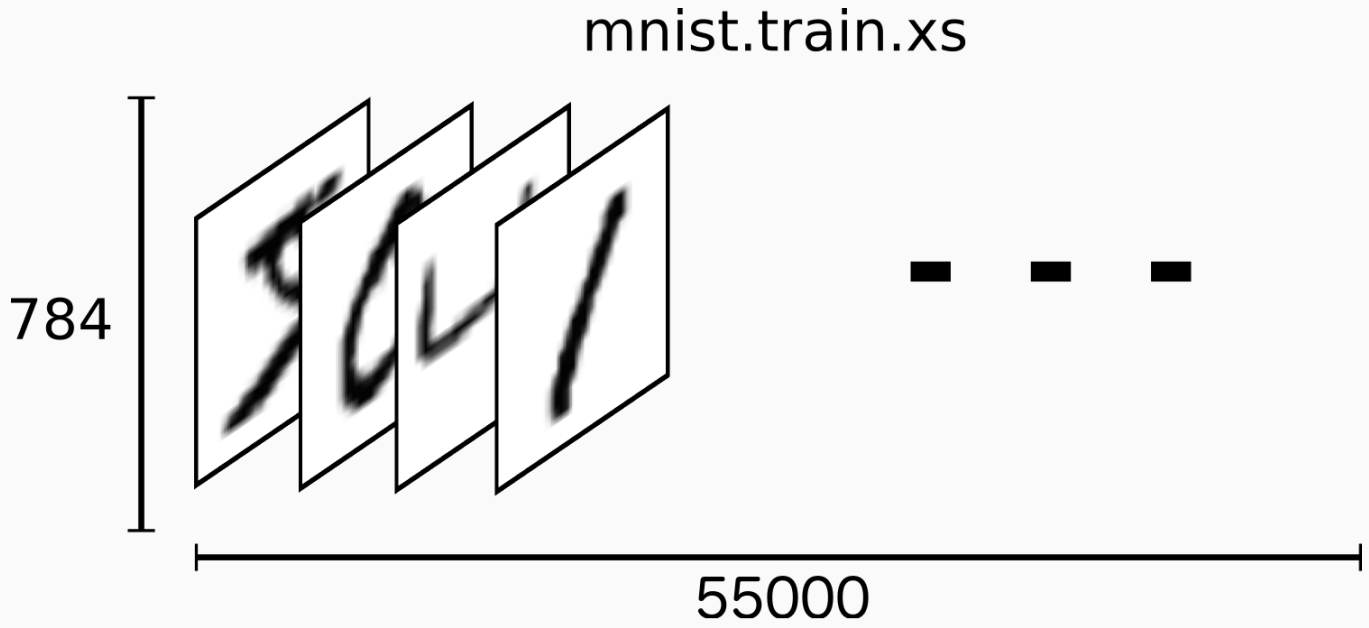
TensorFlow

About TensorFlow:

- Tensorflow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, which the graph edges represent the multidimensional data arrays (tensors) communicated between them.
- It allows you to deploy computation to one or more CPUs or GPUs in d desktop, server, or mobile device with a single API. The Tensorflow package provide access to the complete Tensorflow API from within R.

MNIST

mnist$train$image is a tensor (an n-dimensional array) with shape (55000L, 784L).

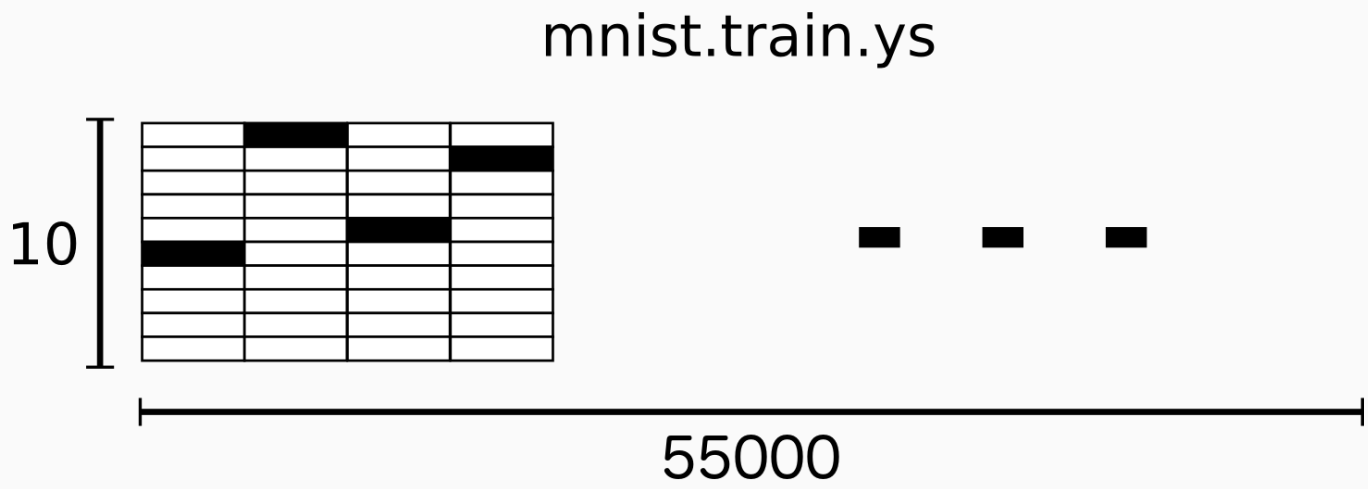


MNIST

The label is “one-hot vectors”.

For example, 3 would be $[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]^{\cdot}$

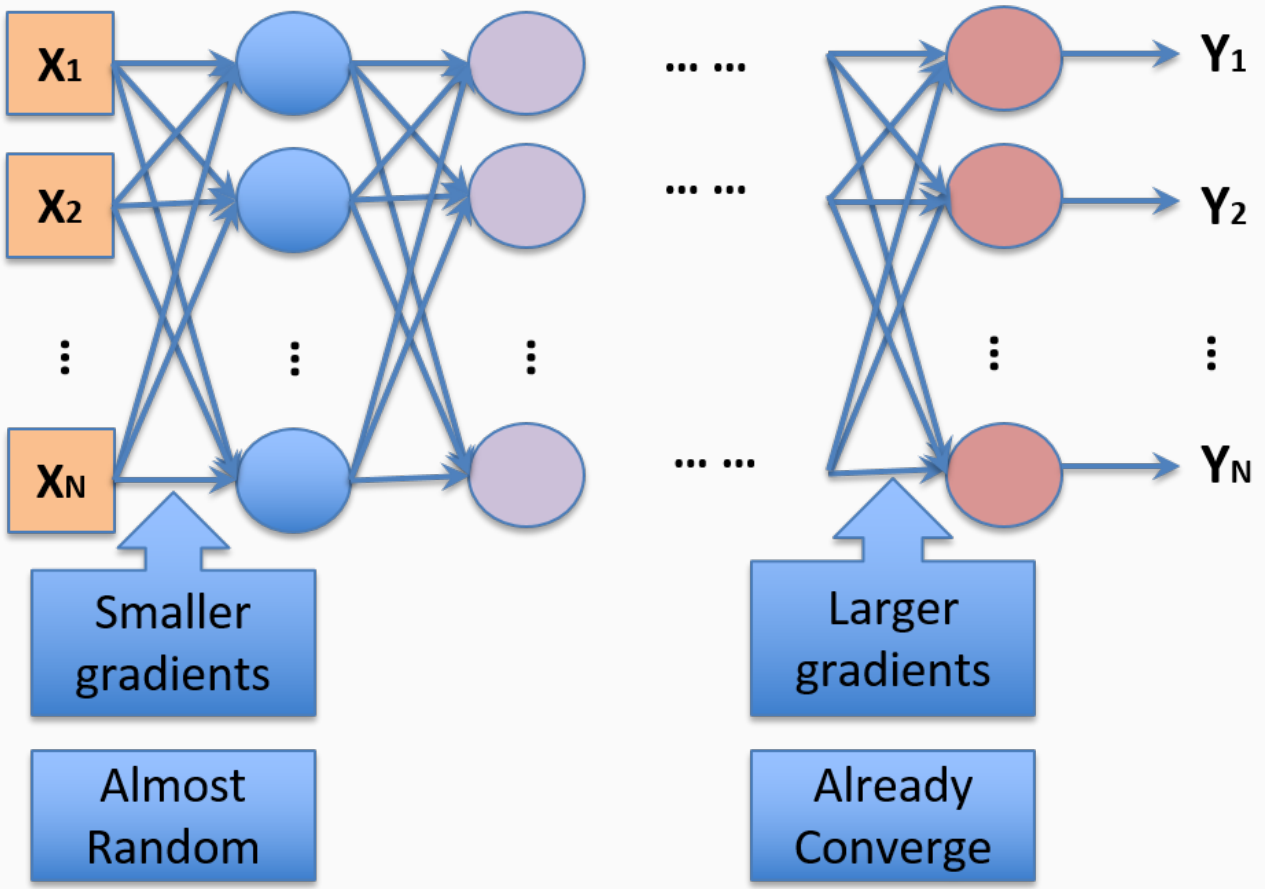
mnist$train$labels is a tensor with shape (55000L, 10L).



MNIST

Why do we have 2 layers, not more?

Vanishing Gradient Problem $\longrightarrow$ Deeper does not usually imply better.

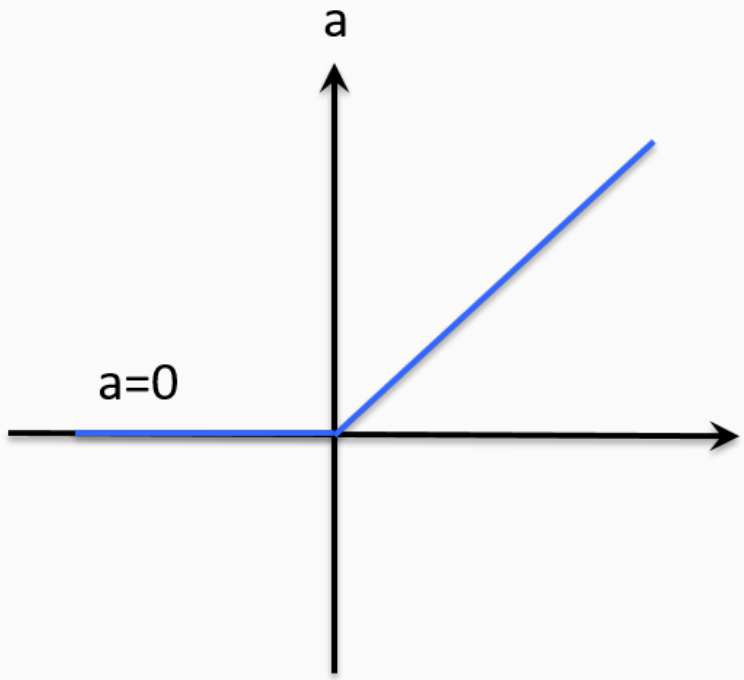


MNIST

Activation function: Rectified Linear Unit (ReLU)

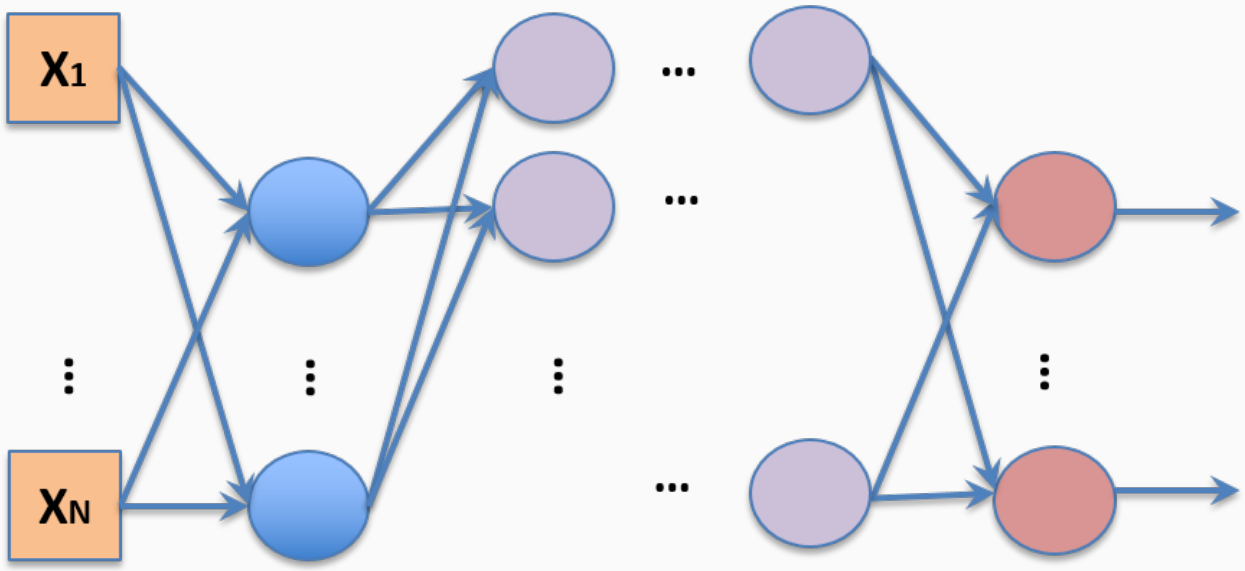
Reason:

- Infinite sigmoid with different biases
- Able to handle vanishing gradient problem
- Fast to compute
- Biological reason



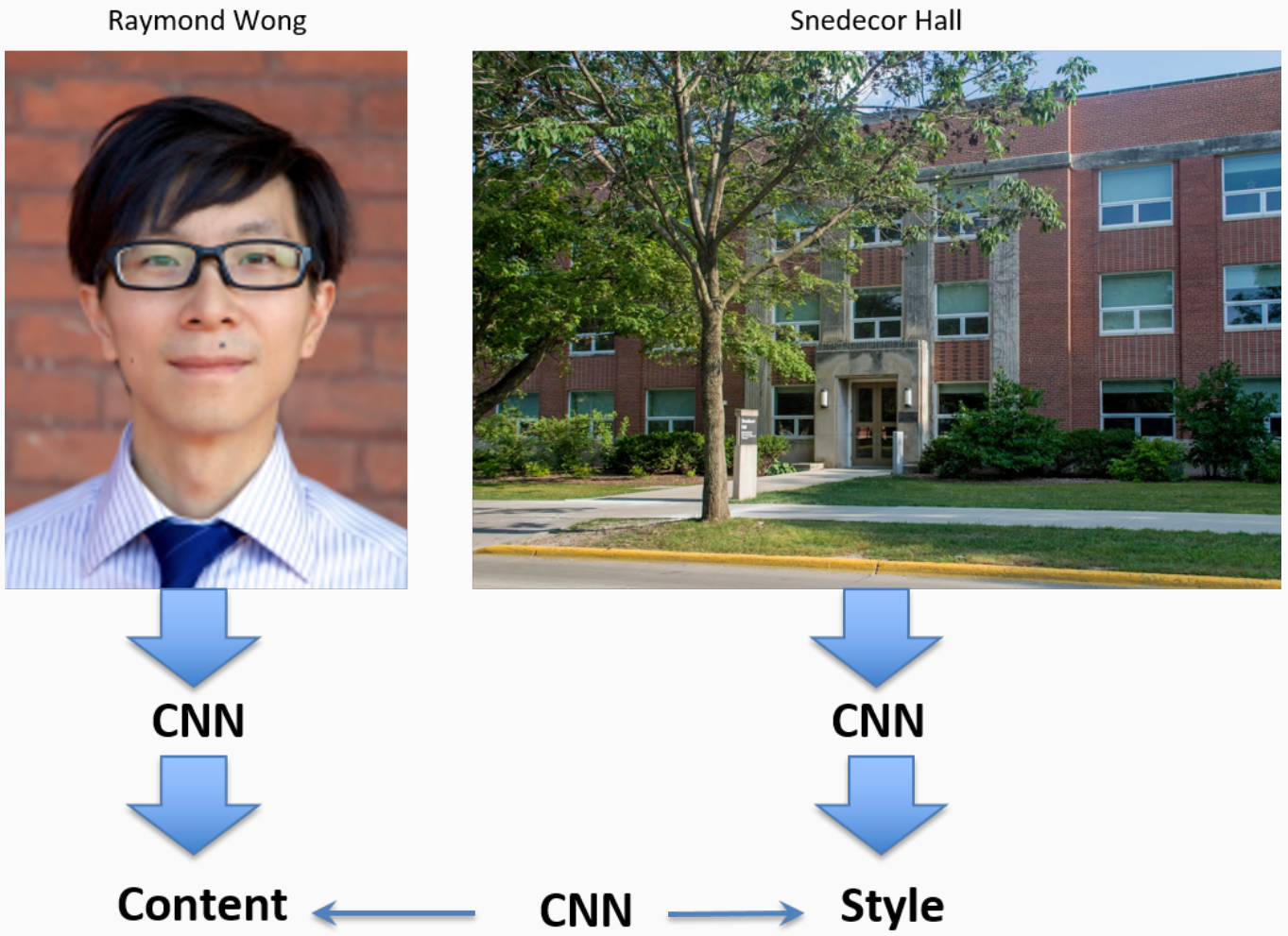
MNIST

Dropout: when you do training, each time before updating the parameters, each neuron has p% to dropout.



Thinner structure

Style Transfer, A.K.A. Deep Style



Style Transfer, A.K.A. Deep Style