Deep Learning and TensorFlow

Dapeng Hu, Qinglong Tian, Haozhe Zhang, Min Zhang

April 23, 2017

STAT 580 Statistical Computing
Department of Statistics
Iowa State University

Content

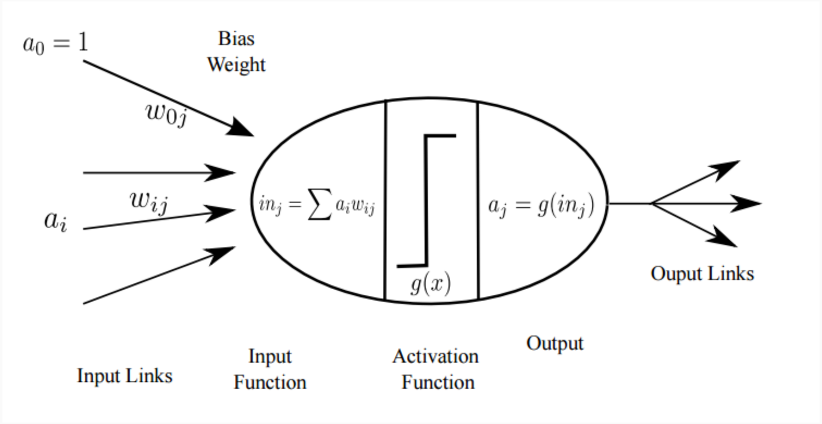
History

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

Neural Network

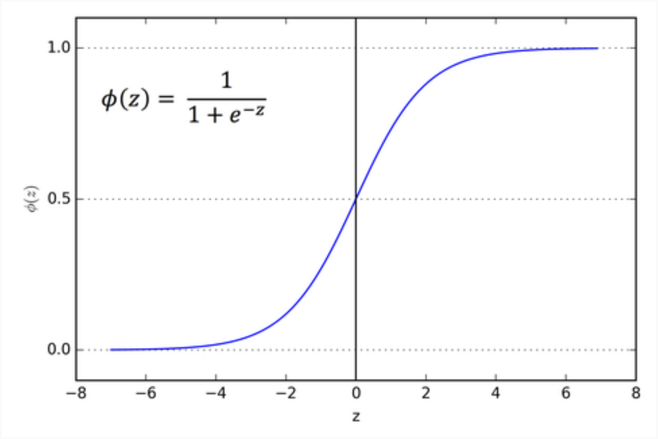
A Neuron Model



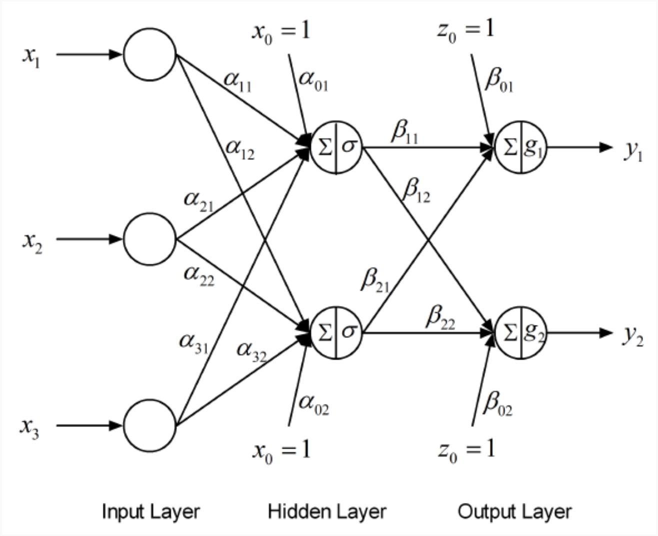
A Neuron Model

- 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?

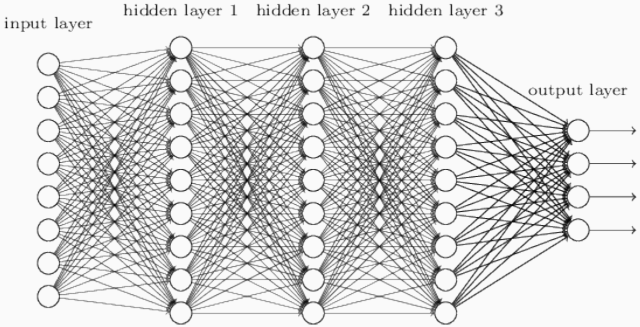
A Neuron Model



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:

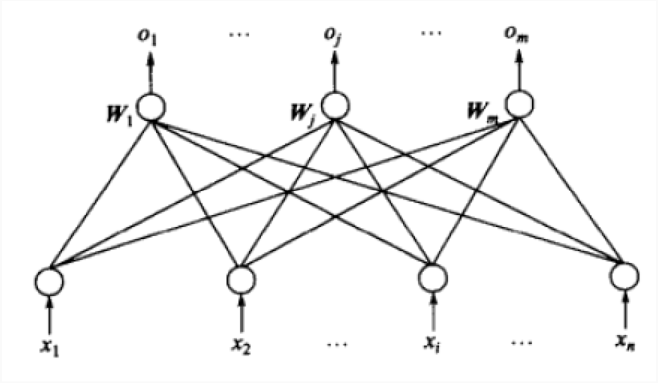
$$r = \{d_j - f(\boldsymbol{W}_j^T \boldsymbol{X})\} f'(\boldsymbol{W}_j^T \boldsymbol{X}) \approx (d_j - o_j) f'(\text{net}_j)$$

- The weights will be adjusted as:

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

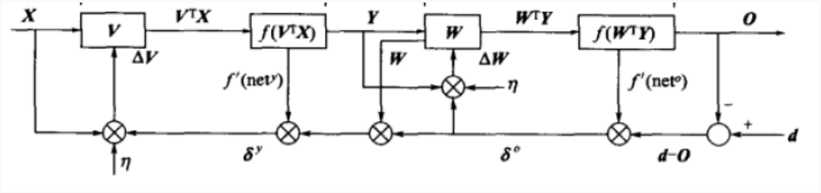
Backpropagation Algorithm

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



Backpropagation Algorithm

The flow of signal in BP algorithm:

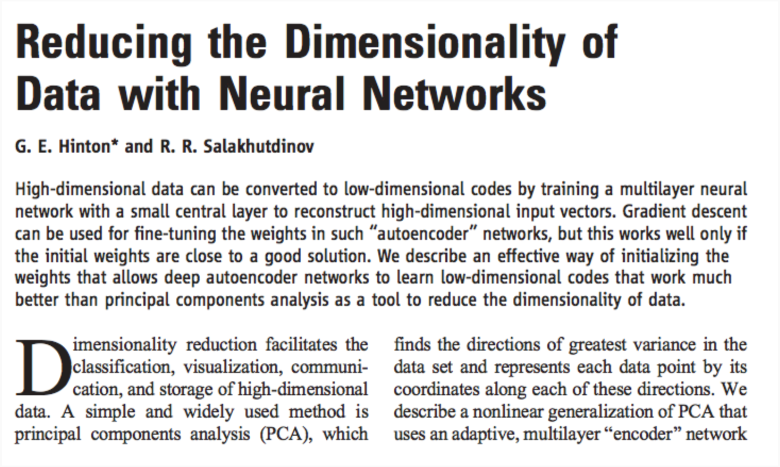


More Hidden Layers?

- 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

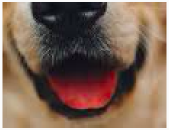
High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial 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

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



CNN

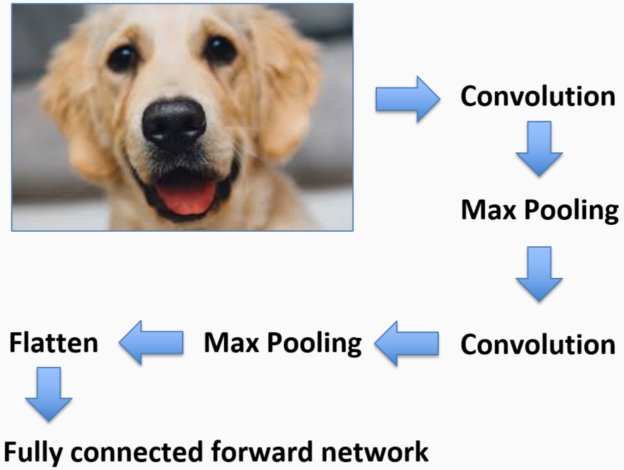
The same pattern appears in different regions.

CNN

Subsampling the pixels will not change the object.

How does convolution work?

The whole Convolutional neural network:



How does convolution work?

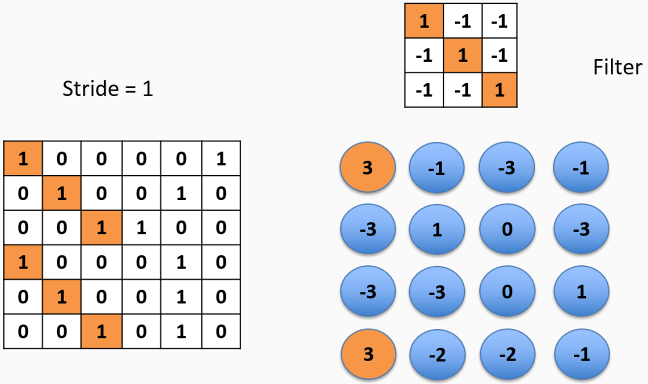
1	-1	-1
-1	1	-1
-1	-1	1

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

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

How does convolution work?



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

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

How does convolution work?

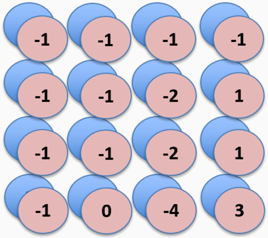
Stride = 1

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

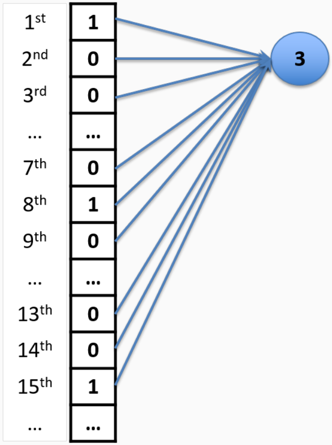


Feature map

Connection between CNN and Fully Connected Neural Network

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

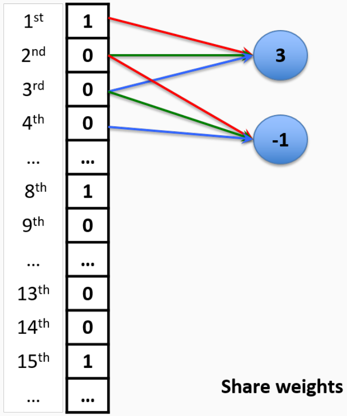
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0



Connection between CNN and Fully Connected 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



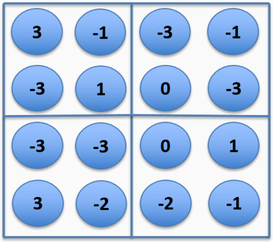
How does max pooling work?

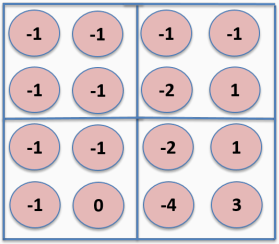
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

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

Filter 2





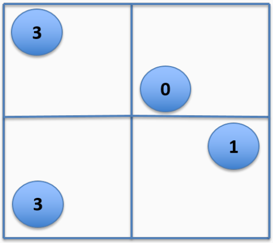
How does max pooling work?

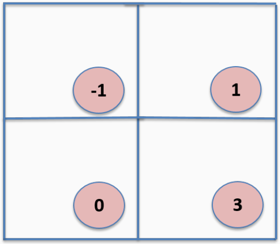
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

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

Filter 2

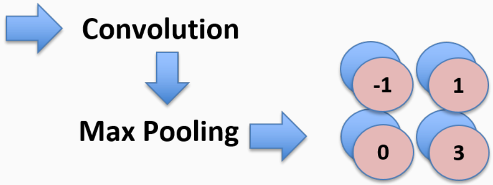




After convolution and max pooling

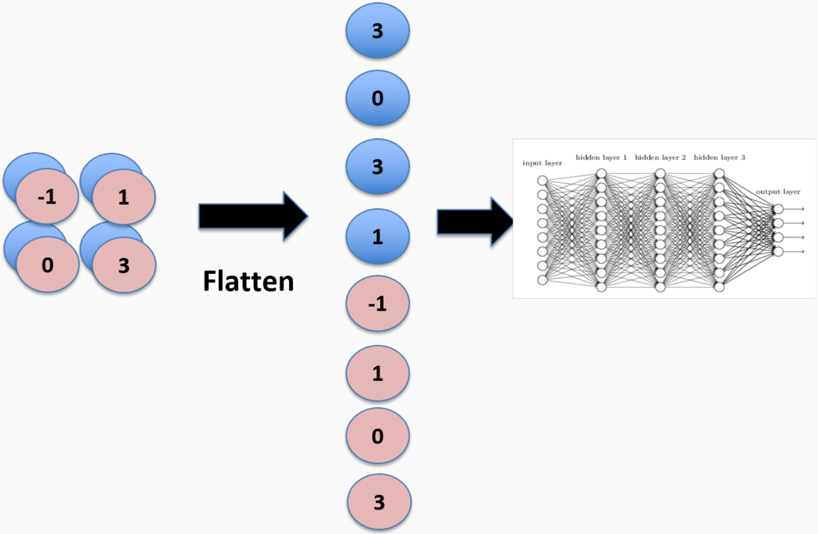
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Smaller image 2x2



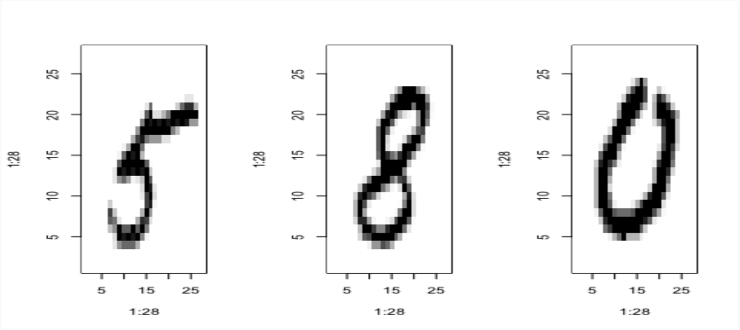
Each filter is a channel

How does Flatten work?



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.



TensorFlow

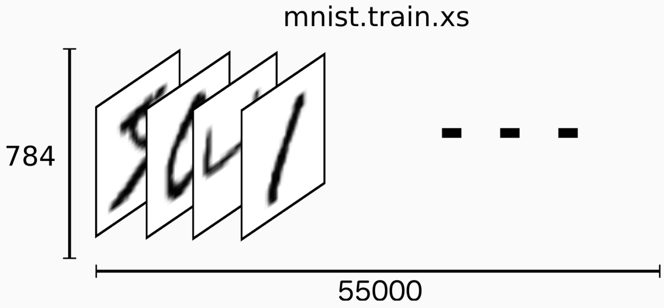
TensorFlow

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.

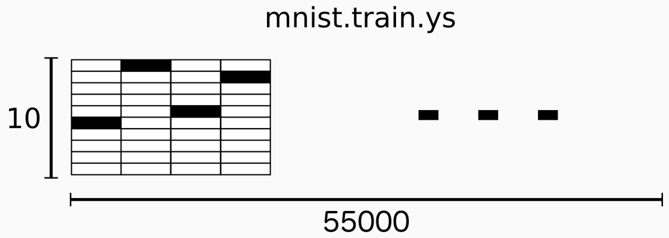
MNIST

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



MNIST

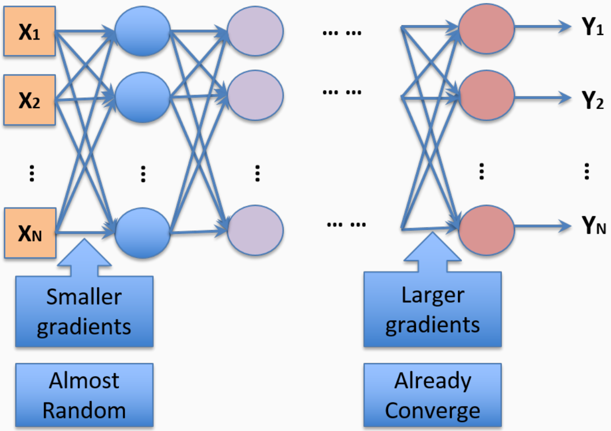
The label is "one-hot vectors".
For example, 3 would be $[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]$.
mnist$train$labels is a tensor with shape (55000L, 10L).



MNIST

Why do we have 2 layers, not more?

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

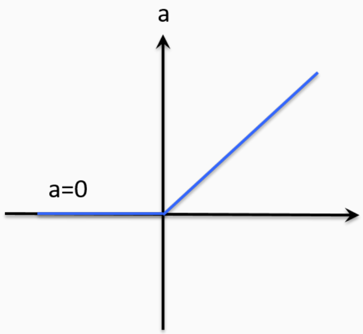


MNIST

Activation function: Rectified Linear Unit (ReLU)

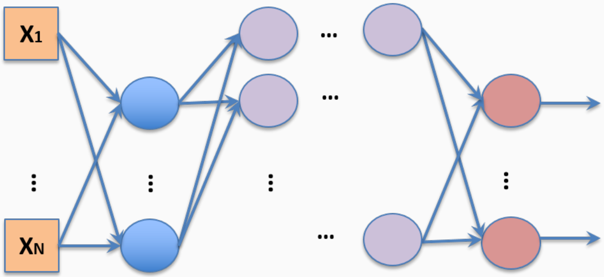
Reason:

- 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.



Style Transfer

Style Transfer, A.K.A. Deep Style

Raymond Wong

Snedecor Hall

$$\begin{tikzcd}
\text{CNN} \arrow[d] & & \text{CNN} \arrow[d] \\
\text{Content} & \text{CNN} \arrow[l] \arrow[r] & \text{Style}
\end{tikzcd}$$

Style Transfer

Style Transfer, A.K.A. Deep Style