Python Machine Learning: The 11st Book Circle

Deep Neural Network Algorithm II: Convolutional Neural Network, Recurrent Neural Network

Jianan Liu

Gothenburg

July, 5, 2017

Disclaimer

All opinions and statements in this presentation are mine and do not in any way represent the company
Any comment or correction of error is welcome, please contact me
chisyliu@hotmail.com

Acknowledgement

Acknowledgement

Special thanks to YJango in Zhihu, his article(in Chinese) in Zhihu provides elegent introduction to CNN. He also produces lots of nice figure which makes CNN to be understood easily and I borrowed some of them in our presentation. I also would like to thank Tiancong Zheng from machine learning group in Zenuity for his recommendation of books regarding modern deep neural network. As an engineer who has statistical signal processing/wireless communication academic background, most of knowledge in neural network learned from university were from Simon Haykin's papers and book which are more related to classical feedforward neural network and statistical adaptive filter, the books Tiancong refereed are extremely good complementary resource for me to get deep insight of modern deep learning.

The 11st Book Circle of Python Machine Learning

In this presentation belongs to **algorithm** part of the book

- The context in this circle is NOT included in the book
- If you want to know more about Deep Neural Network, I recommend reading book Deep Learning from Ian Goodfellow, Yoshua Bengio and Aaron Courville
- We will try to go through basic mathematics behind Convolutional Neural Network(CNN), Recurrent Neural Network(time RNN).
- All of us need to debug the python code, in order to get practice of implementing machine learning algorithm
- A complete resources for deep neural network and machine learning could be found here ▸ Link

Overview

Outline for Section 1

Convolutional Neural Network

From this slide we introduce Convolutional Neural Network. Here are some comments right before we start

- Actually lots of topics regarding deep learning, e.g. convolutional neural network, have weaker/less connection to perfect/beautiful mathematical expression/derivation. They are more relative to intuitive and heuristic designing, especially on the designing of network architecture/structure. It makes the deep neural network problem does NOT look like algorithm design but more logical and mechanism design(At least it is like that as compared with classical machine learning algorithms, adaptive/statistical signal processing and Bayesian filters)

Convolutional Neural Network

- So we more or less give up mathematical expression/derivation but go for logical reasoning or logically explanation for convolutional neural network
- An easy understood resource regarding convolutional neural network could refer to Ujjwal Karn's blog "An Intuitive Explanation of Convolutional Neural Networks" Link
- A better description of convolutional neural network could be found in note "Module 2: Convolutional Neural Networks" of Feifei Li's course "Convolutional Neural Networks for Visual Recognition" in Standford University Link
- Since these materials have extremely good contents on CNN, I might sometimes just refer to the contents of them directly with denoting

Convolutional Neural Network

- There are more videos and materials could be good to look at, e.g. Last lecture of Jianbo Shi's Vision Intelligence and Machine Learning Course at University of Pennsylvania/Edx ▸ Link which talks CNN for machine vision, Lex Fridman's Deep Learning for Self-Driving Cars at MIT ▸ Link where introduces what is CNN and how the CNN can be designed for self-driving car, and also "Deep Learning School 2016: Individual Talks" ▸ Link where discusses how the CNN applies for computer vision
- If you still remembered we discussed how many high techniques have been invented in Bell Lab in the SVM section on the 2nd circle of our machine learning presentation, here we have to mention one more important invention, LeNet, the first successful applications of Convolutional Networks were developed by Yann LeCun in 1990s in Bell lab as well!

Intro CNN for Image Recognition

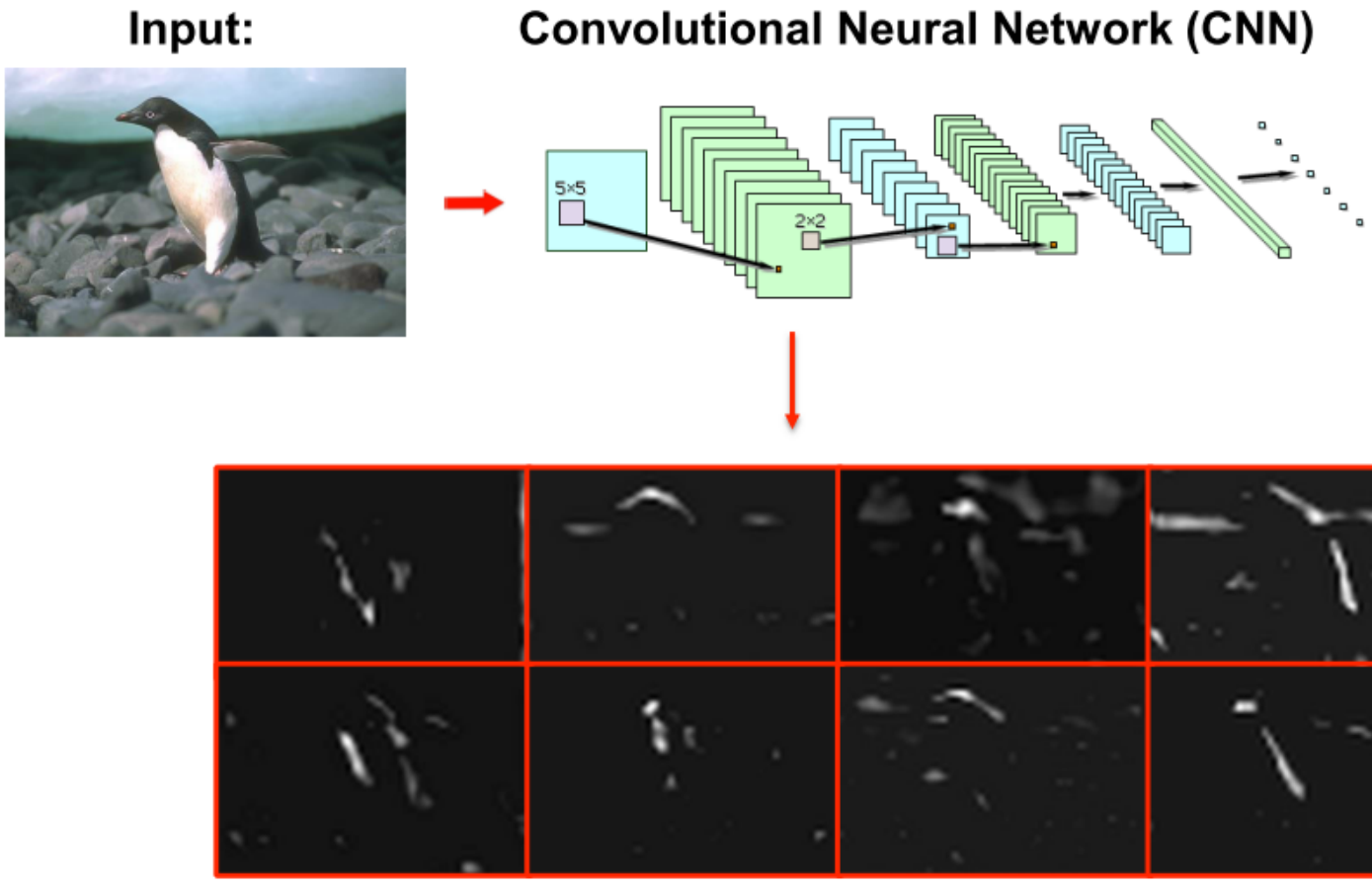
Architecture of CNN

Architecture of CNN

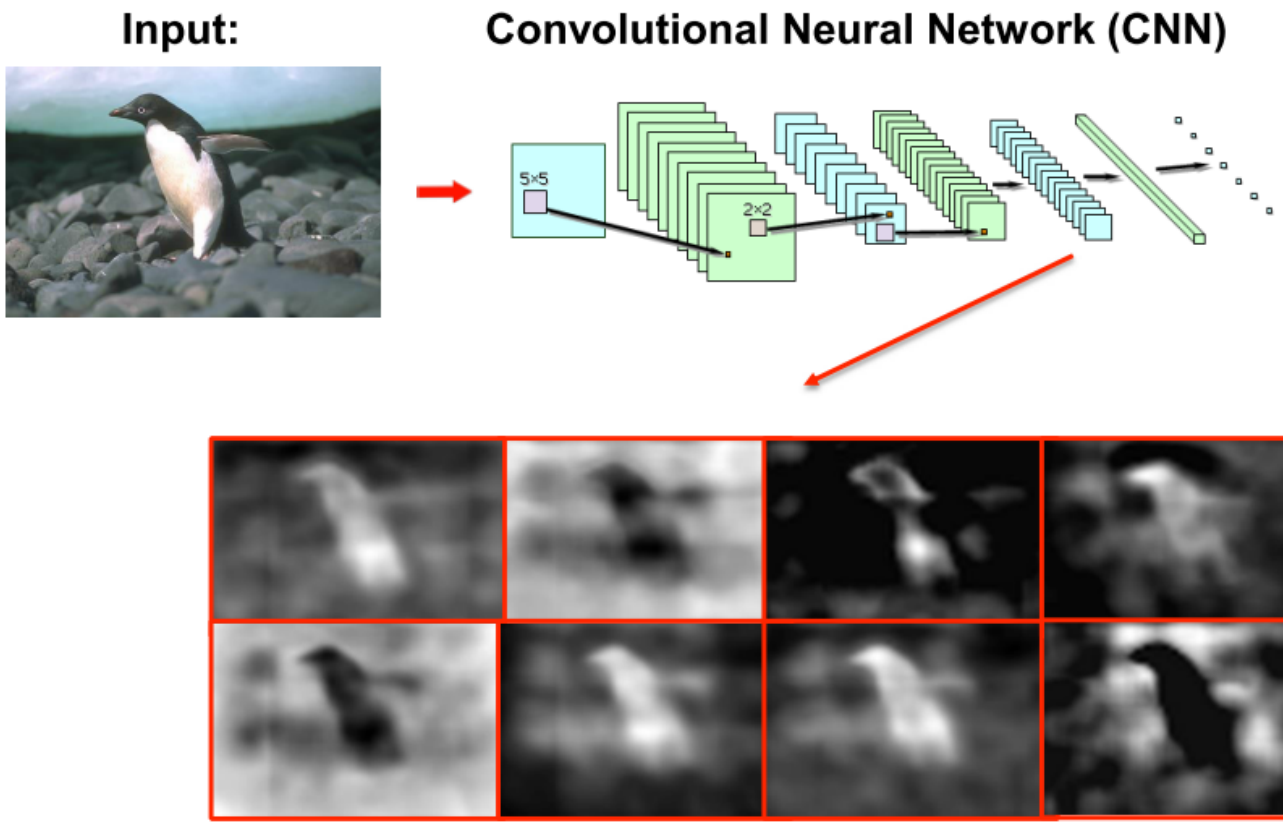
Architecture of CNN

- Now, let's borrow 3 pictures from Jianbo Shi's Vision Intelligence and Machine Learning Course at University of Pennsylvania/Edx(See reference) to show/illustrate what are the functionality of neurons in early convolutional layers of CNN, functionality of neurons in deeper/later convolutional layers of CNN, and functionality of different filters on each convolutional layer

Architecture of CNN

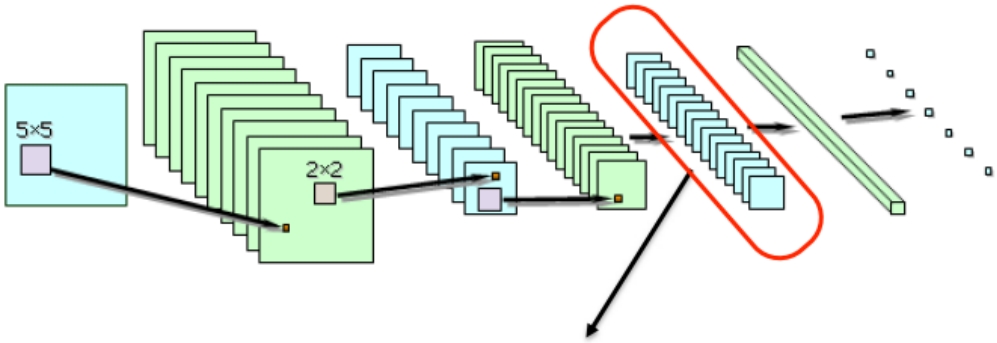


Architecture of CNN

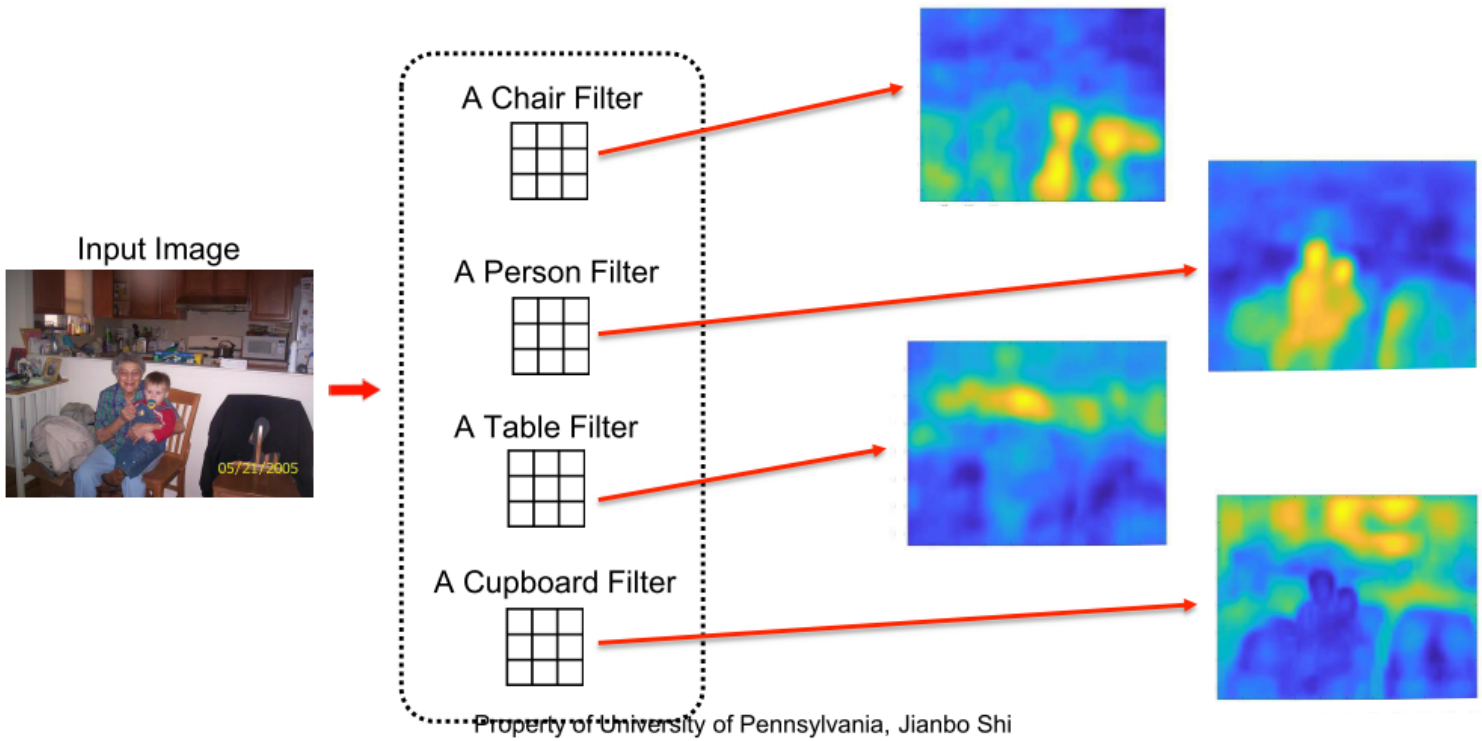


Deep layers learn to detect high-level object structures and their parts.

Architecture of CNN

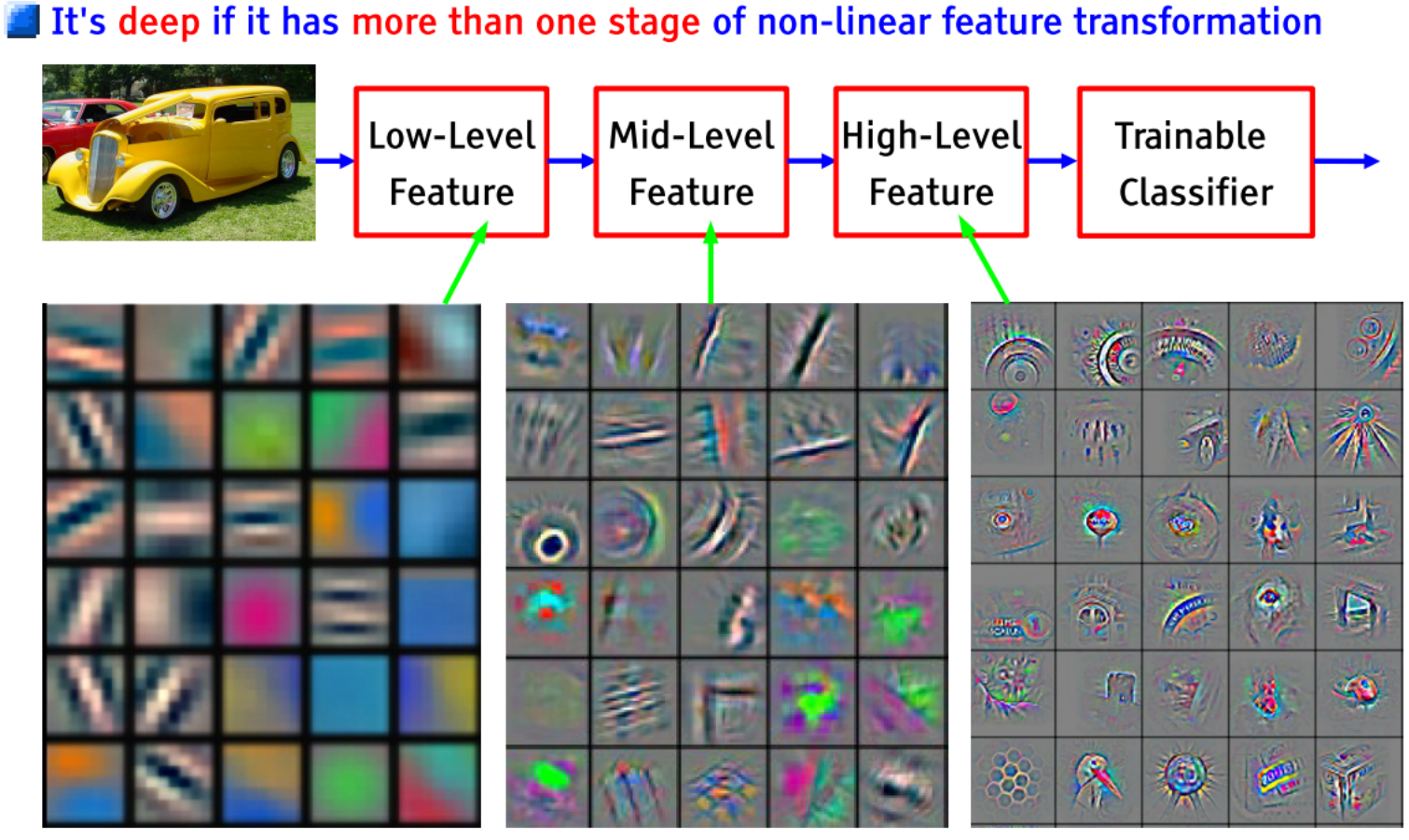


A Closer Look inside the Convolutional Layer



Architecture of CNN

Another figure from Yann LeCun's deep learning at New York University also shows the same idea



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

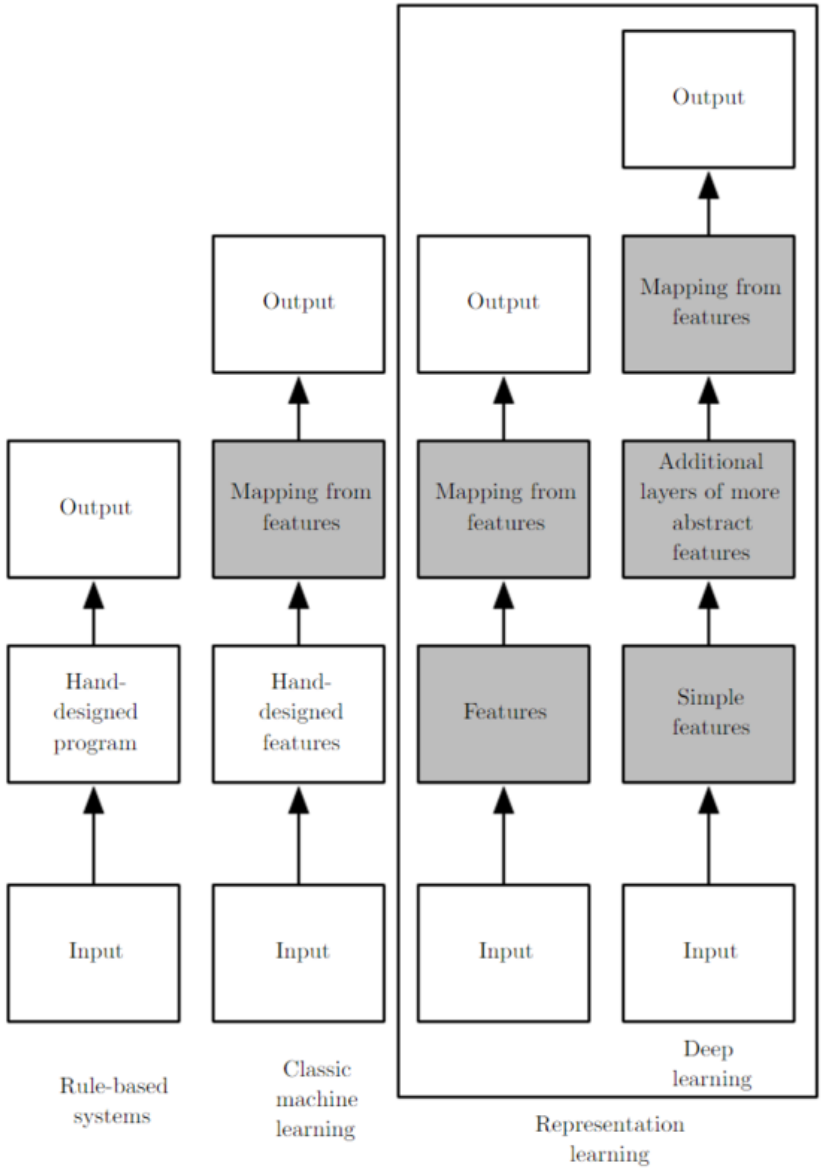
Architecture of CNN

From the architecture of CNN we noticed the biggest difference between CNN and traditional machine learning algorithms are:

- 1. We leave the feature extraction/selection to machine(Neural Network itself) rather extract features manually
- 2. In CNN we also gather together lower level detailed features/patterns to make high level/abstract/"near to the object" features

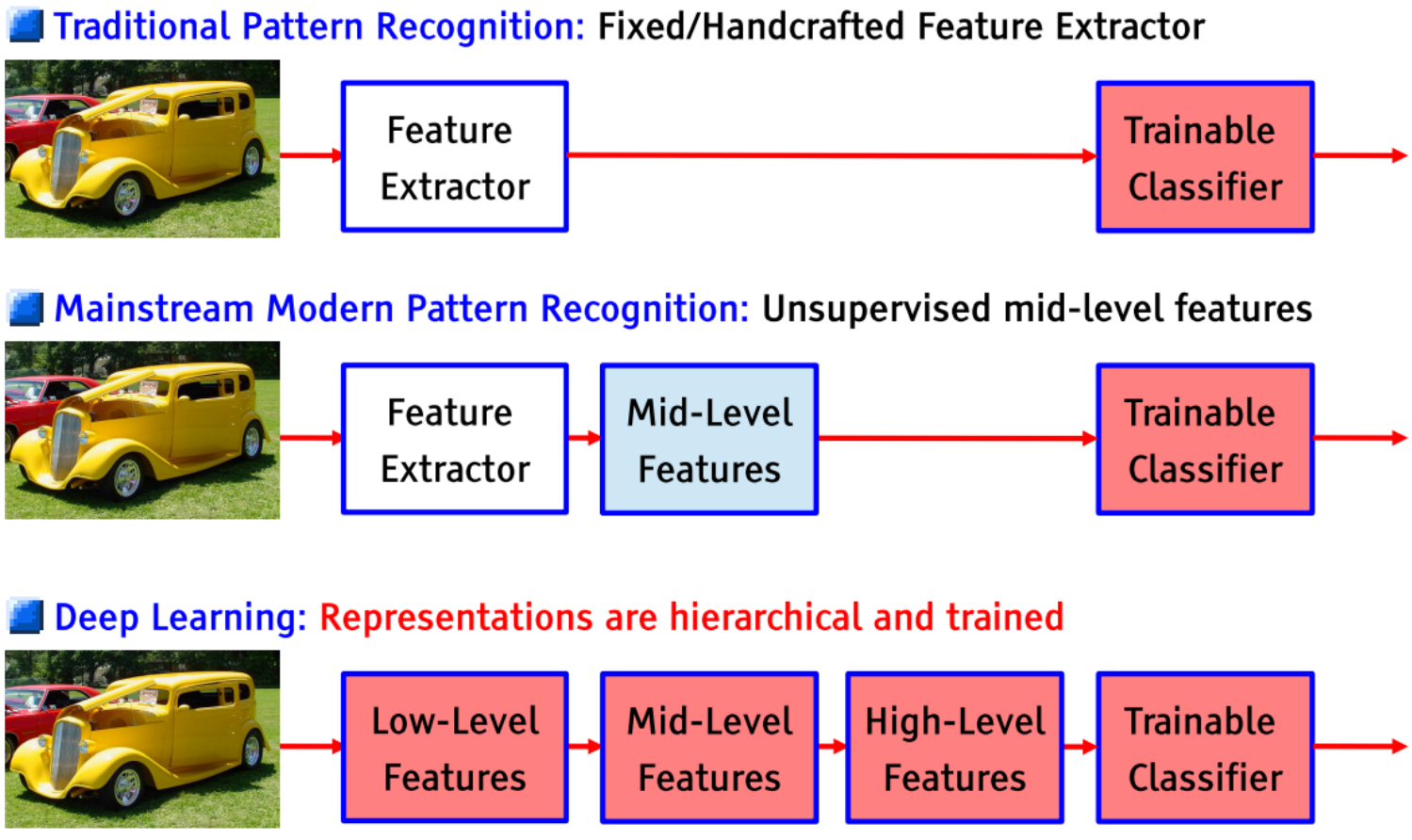
We can summarize these differences by using the figure borrowed from Ian Goodfellow Yoshua Bengio, Aaron Courville's book Deep Learning, on next page

Architecture of CNN



Architecture of CNN

- Another figure from Yann LeCun's deep learning at New York University also shows the same idea regarding the difference between traditional ML and deep learning



How to Train CNN

Outline for Section 2

Recurrent Neural Network

-
-

-
-

-
-
-

-

Note

-

-
-

1: Initialize the all elements w_d in weight **w** to 0
2: **for** t in T **do**
3: Shuffle(); // randomly sort the L samples once
4: **for** l in L **do**
5: **for** d in D **do**
6: computer $w_d(t+1) = w_d(t) + \eta(t)(y^{(l)} - \mathbf{x}^{(l)}\mathbf{w})x_d^{(l)}$
7: d = d + 1
8: **end for**
9: l = l + 1
10: **end for**
11: t = t + 1
12: **end for**

- $\eta(t)$ means η is a function of t

Outline for Section 3

Tensorflow

- Here is the collection of some python codes which uses TensorFlow to implement deep neural network as examples
  - Link
-

Outline for Section 4

Conclusion

Conclusion

Conclusion

References

 Geoff Hinton, Neural Networks for Machine Learning Course, University of Toronto

 Hung-Yi Lee, Machine Learning Course, National Taiwan University

 Richard Socher, Deep Learning for Natural Language Processing, Stanford University

 Fei-fei Li, Andrej Karpathy, Justin Johnson, Convolutional Neural Networks for Visual Recognition Course, Stanford University

 Ujjwal Karn, An Intuitive Explanation of Convolutional Neural Networks ▸ Link

 Jianbo Shi, Vision Intelligence and Machine Learning, University of Pennsylvania/Edx ▸ Link

 Deep Learning School 2016: Individual Talks ▸ Link

References

 Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning ▸ Link

 Michael Nielsen, Neural Network and Deep Learning ▸ Link

 Simon Haykin, Neural Network-A Comprehensive Foundation

 Simon Haykin, Neural Network and Learning Machine

Question?