深度学习讨论班

黄雷

2016-11-29

内容大纲

- 1.深度学习介绍
  - 神经网络的历史
  - 深度学习的应用
- 2.多层感知机(multi-layer perceptron machine)
  - 前向神经网络（feedforward neural network）
- 3.卷积神经网络（Convolution neural networks）
- 4.递归神经网络 （Recursive neural networks）
- 5.利用神经网络的针对具体问题建模
  - 如何设计损失函数
  - 是否采用端到端学习
- 6.训练神经网络的实用技巧
  - 如何有效地训练
  - 如何提高模型的泛化能力

深度学习知识层面

深度学习应用实践层面

课程的目标

- 1.在知识层面
  - 基本的术语
  - 三类经典的神经网络
- 2.应用实践层面
  - 会针对具体的问题，利用深度学习建模
  - 编程实践，基于torch平台（深度学习平台）

相关资料

- 深度学习课程
  - 牛津大学，Nando de Freitas，https://www.cs.ox.ac.uk/people/nando.defreitas/machinelearning/
  - Coursera, Geoffrey Hinton , Neural Network for Machine Learning
  - 斯坦福大学，Fei-Fei Li，CS231

相关资料

- 领域顶级会议
  - ICLR (International Conference on Learning Representation)
  - ICML
  - CVPR, ACL, IJCAI, AAAI.

相关资料

- 实验室ftp: fileserver.nlsde.buaa.edu.cn/Public/Study/DeepLearning/

Deep learning introduction

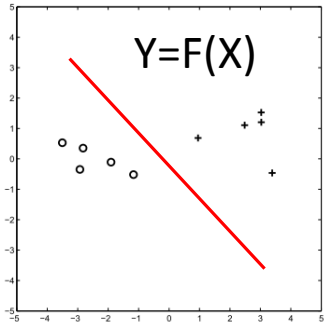
Presented by Lei Huang

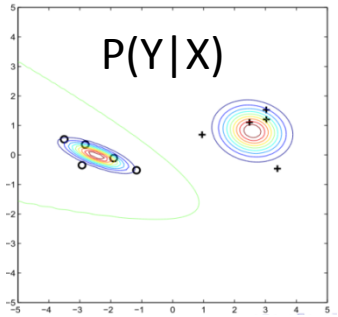
November 29, 2016

Outline

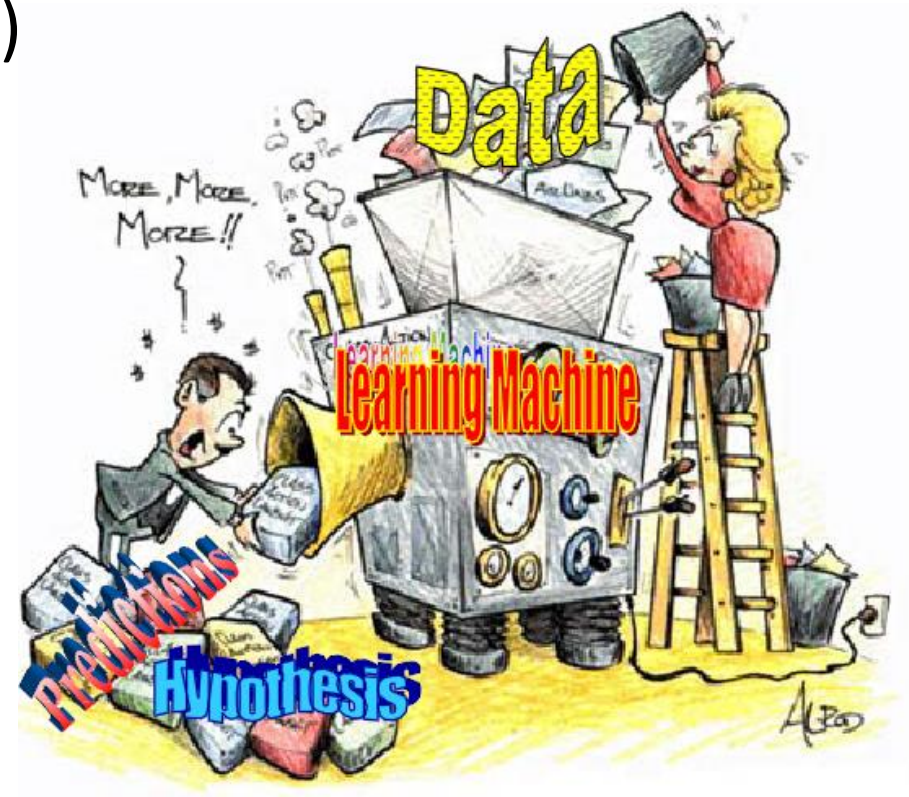
- Basic Concept
  - Machine learning
  - Neural network
  - Deep network
- History of neural network
  - Perceptron
  - BackPropagation
  - Deep learning
- Application

Machine learning





- dataset D={X, Y}
  - Input: X
  - Output: Y
  - Learning: Y=F(X) or P(Y|X)
- Goal
  - Automatically detect patterns in data
  - Use the uncovered patters to predict future data
- Fitting and Generalization



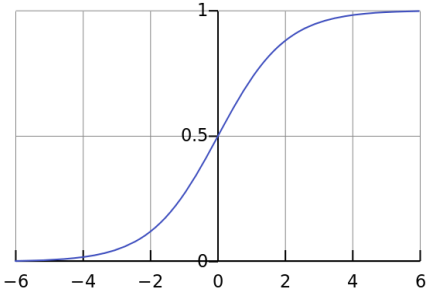
Machine learning

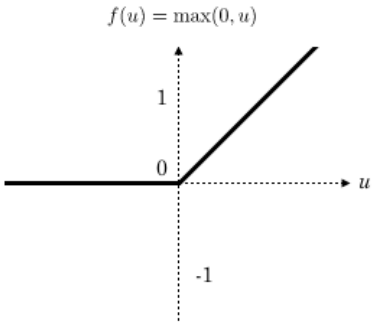
- Types: view of data
  - Supervised Learning(监督学习)
    - D={X, Y}
    - Learning: Y=F(X) or P(X,Y)
  - Unsupervised Learning(非监督学习)
    - dataset D={X, X}
    - Learning: X=F(g(X)) , use G(x) as representation

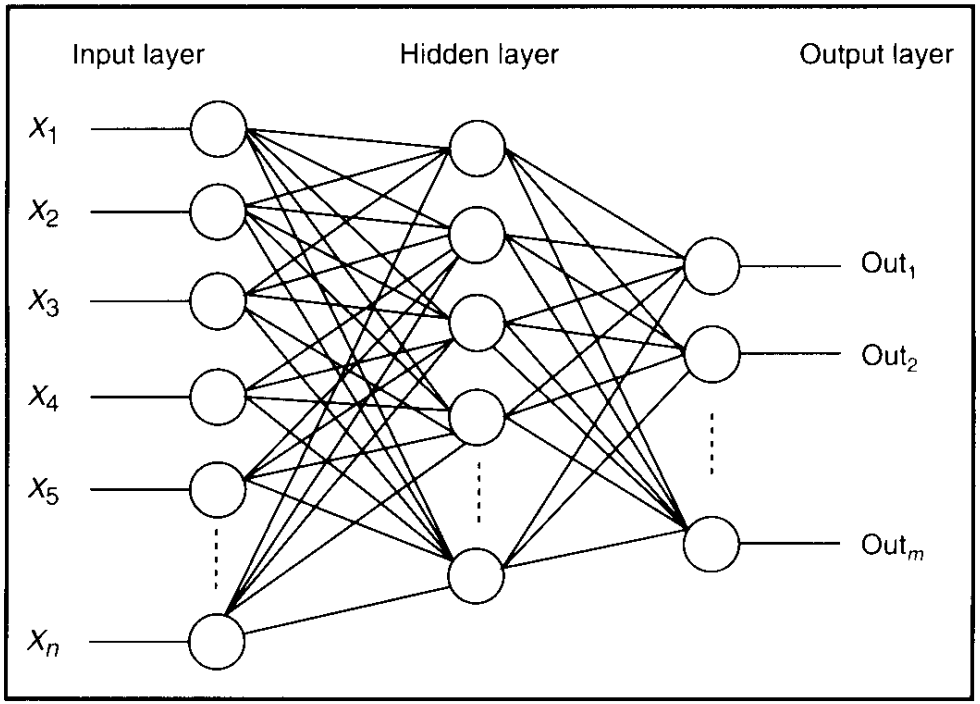
- Types: view of models
  - Non-parametric model (非参模型)
    - Y=F(X; $x_1, x_2 \ldots x_n$)
  - Parametric model(参数化模型)
    - Y=F(X; θ)

Neural network

- Neural network
  - Y=F(X)=$f_T(f_{T-1}(\dots f_1(X)))$
  - $f_i(x) = g(Wx + b)$
- Nonlinear activation
  - sigmod
  - Relu

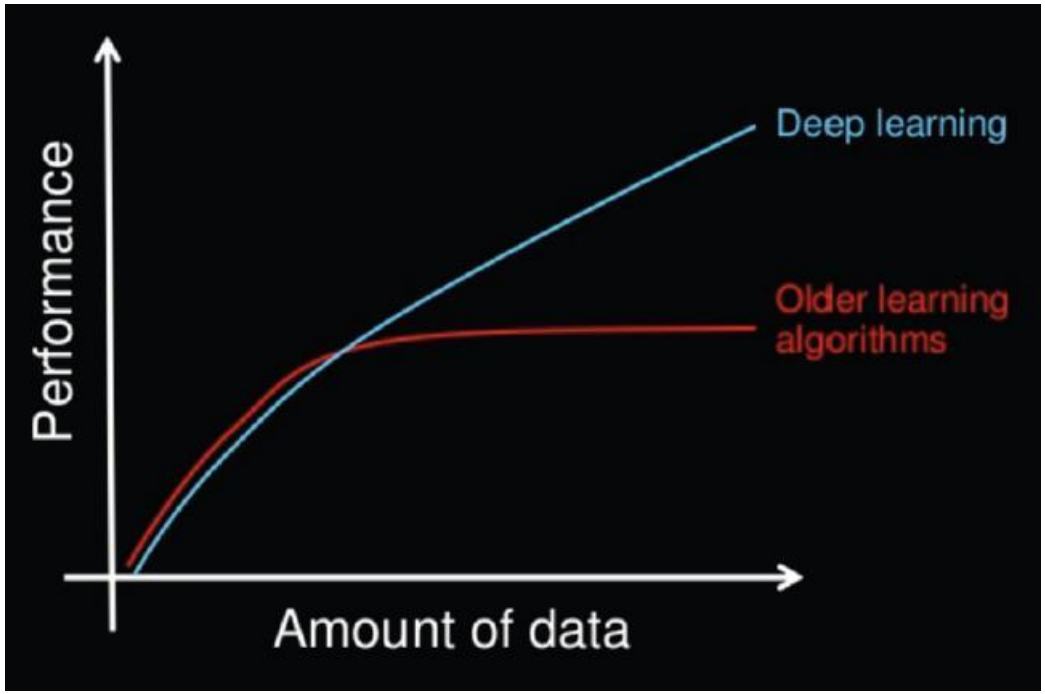






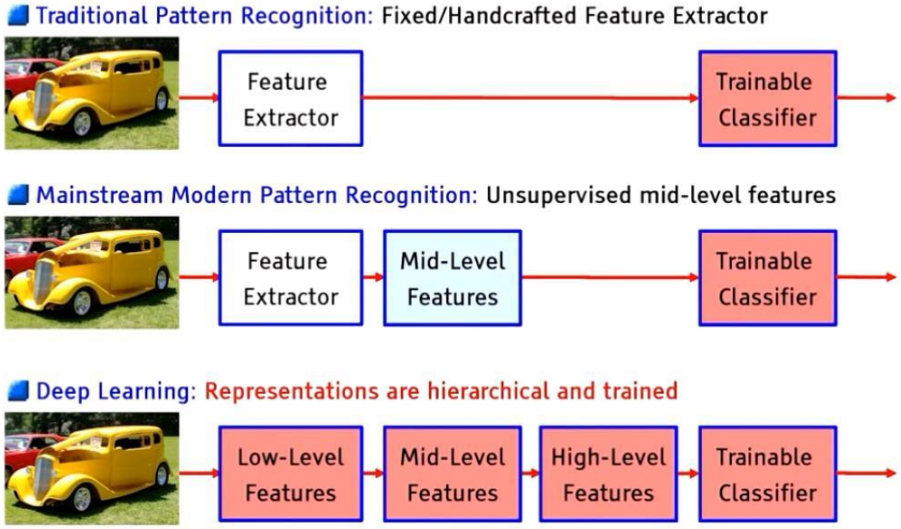
Deep neural network

- Why deep?
  - Powerful representation capacity(函数表达能力)



Key properties of Deep learning

- End to End learning （端到端学习）
  - no distinction between feature extractor and classifier



- “Deep” architectures:
  - cascade of simpler non-linear modules

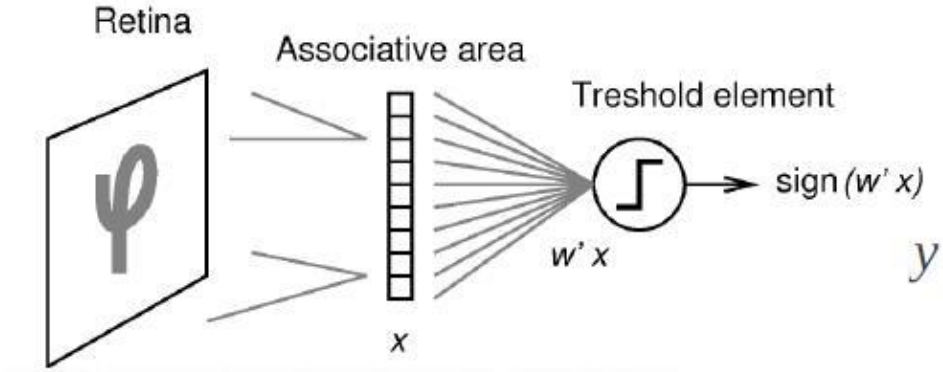
Outline

- Basic Concept
  - Machine learning
  - Neural network
  - Deep network
- History of neural network
  - Perceptron
  - BackPropagation
  - Deep learning
- Application

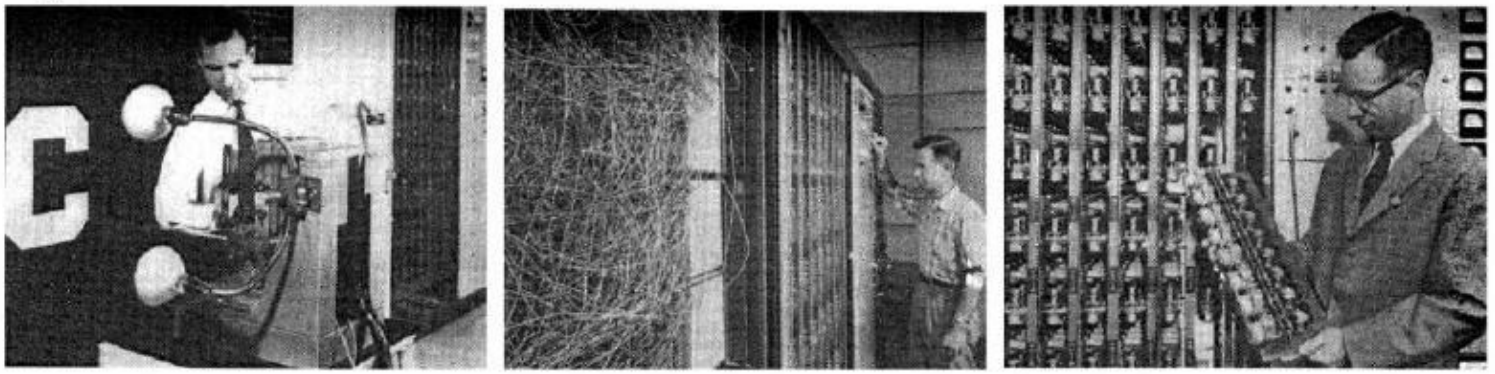
The Perceptron

- 1957, Frank Rosenblatt, Perceptron（感知机）

A simple simulated neuron with adaptive "synaptic weights"
- Computes a weighted sum of inputs
- Output is +1 if the weighted sum is above a thresold, -1 otherwise.



$$y = sign\left(\sum_{i=1}^{N} W_i X_i + b\right)$$

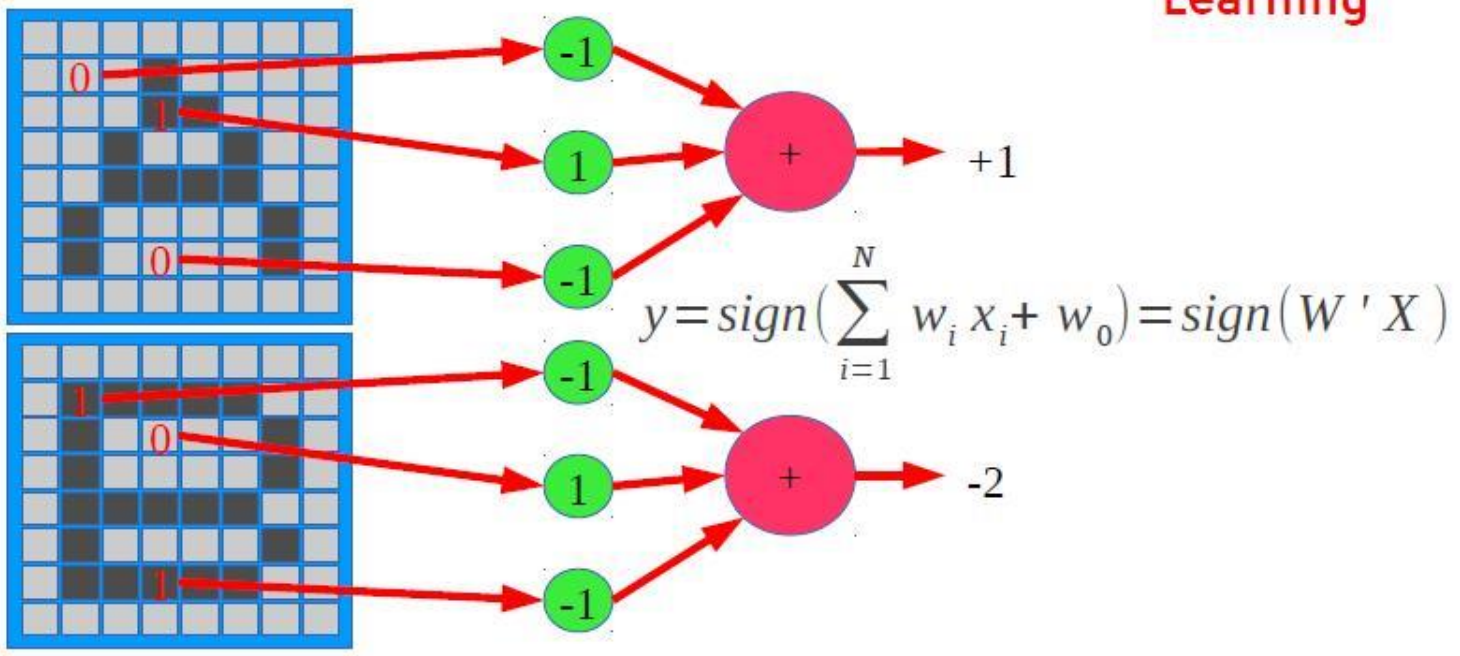


Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

- Example: classifying letters "A" from "B"
- Learning: find the weight values that produce +1 for A and -1 for B
- Training set: $(X^1,Y^1),(X^2,Y^2),.....,(X^P,Y^P)$
- Example: (**A**,+1),(**B**,-1),(**A**,+1),(**B**,-1),(**A**,+1),(**B**,-1),.....

Supervised Learning

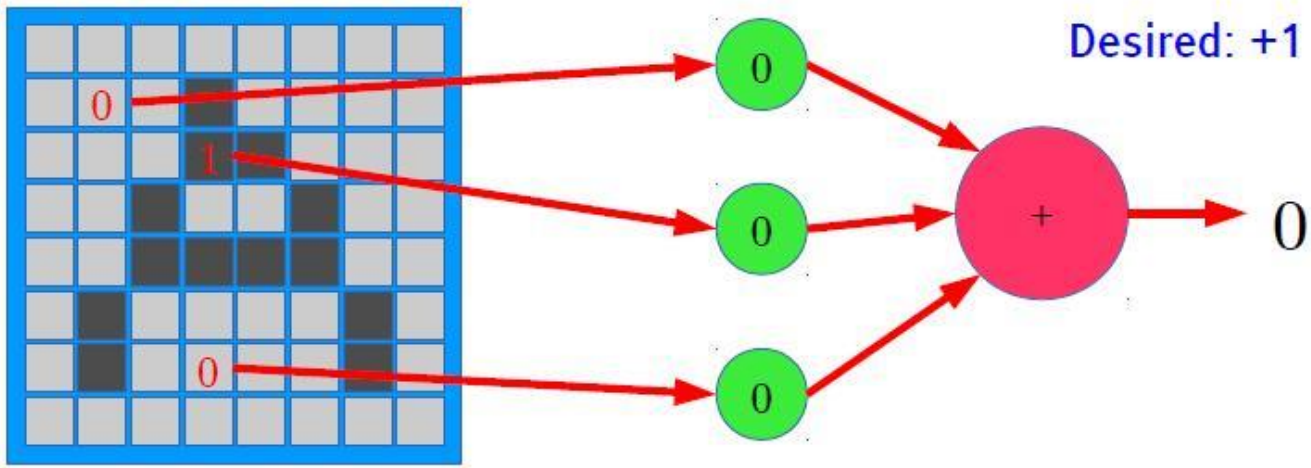


$$y = sign\left(\sum_{i=1}^{N} w_i x_i + w_0\right) = sign(W' X)$$

Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

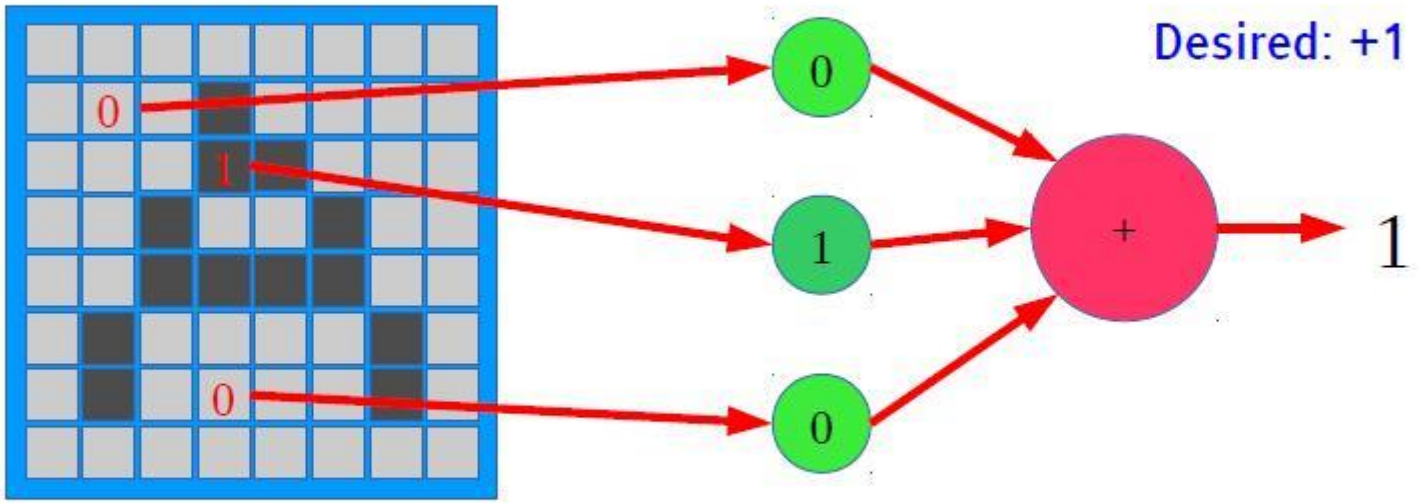
- Learning: adjusting the weights so as to obtain the desired result
  - Initially, the weights are 0.



Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

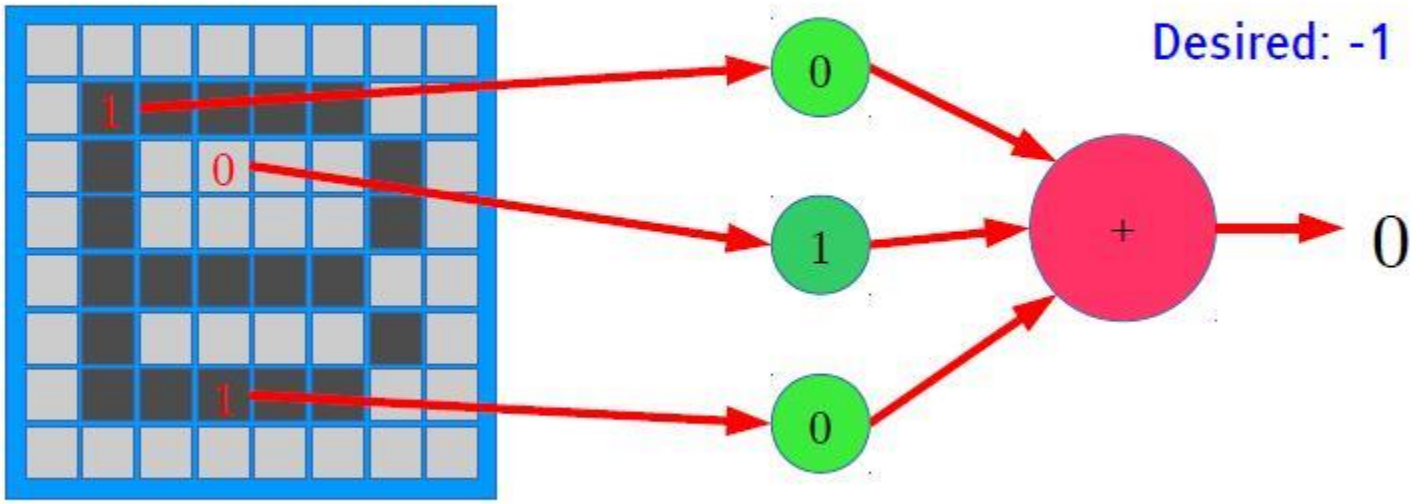
- Adjusting the weights when the the output is incorrect
  - If the desired output is +1, add pixel values to the weights (Hebbian learning)



Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

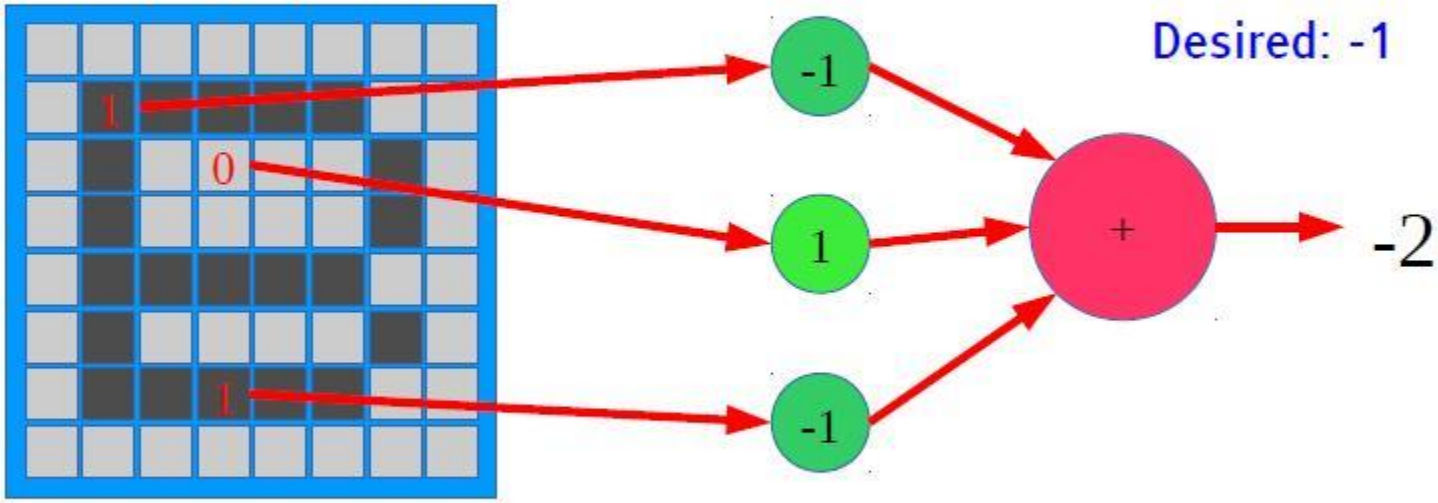
- Adjusting the weights when the the output is incorrect
  - If the desired output is -1, subtract pixel values from the weights.



Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

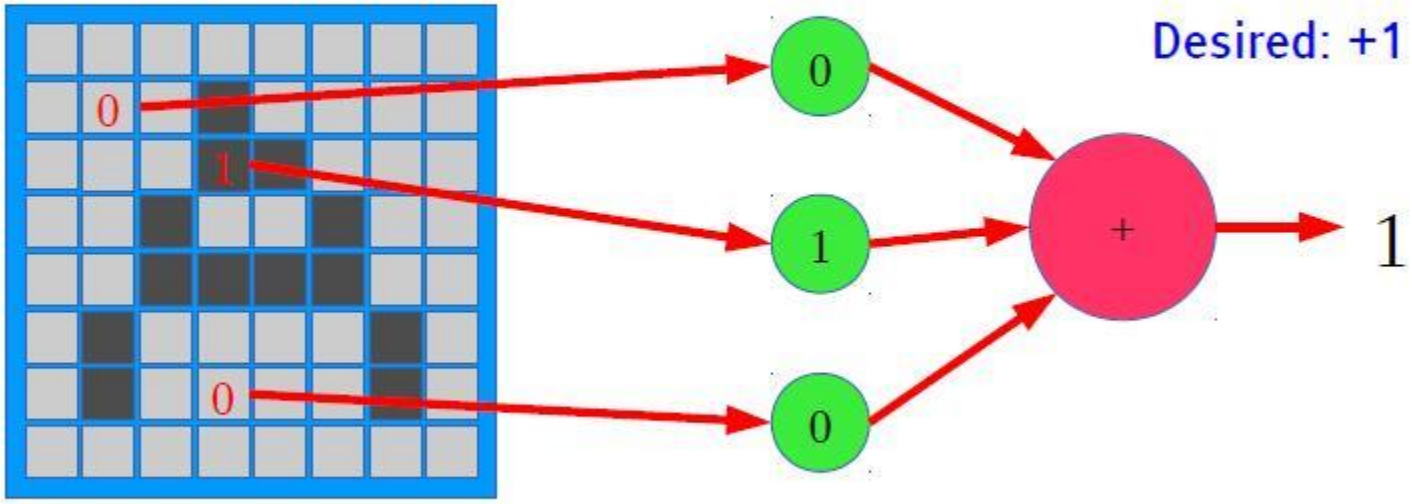
- Adjusting the weights when the the output is incorrect
  - If the desired output is -1, subtract pixel values from the weights.



Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

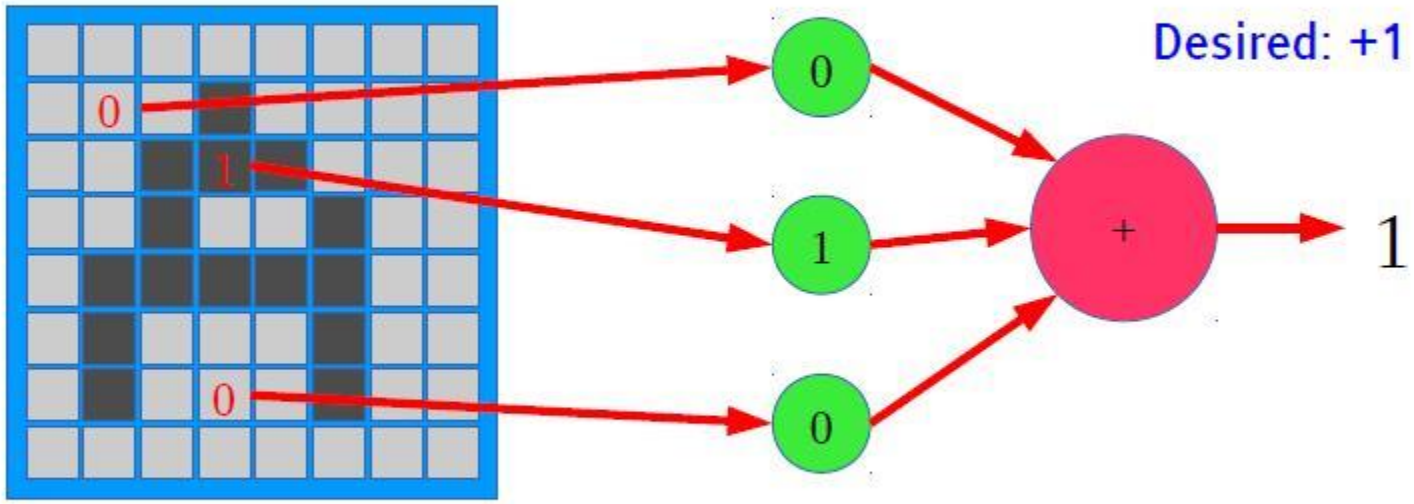
- Write if the writing style varies?



Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

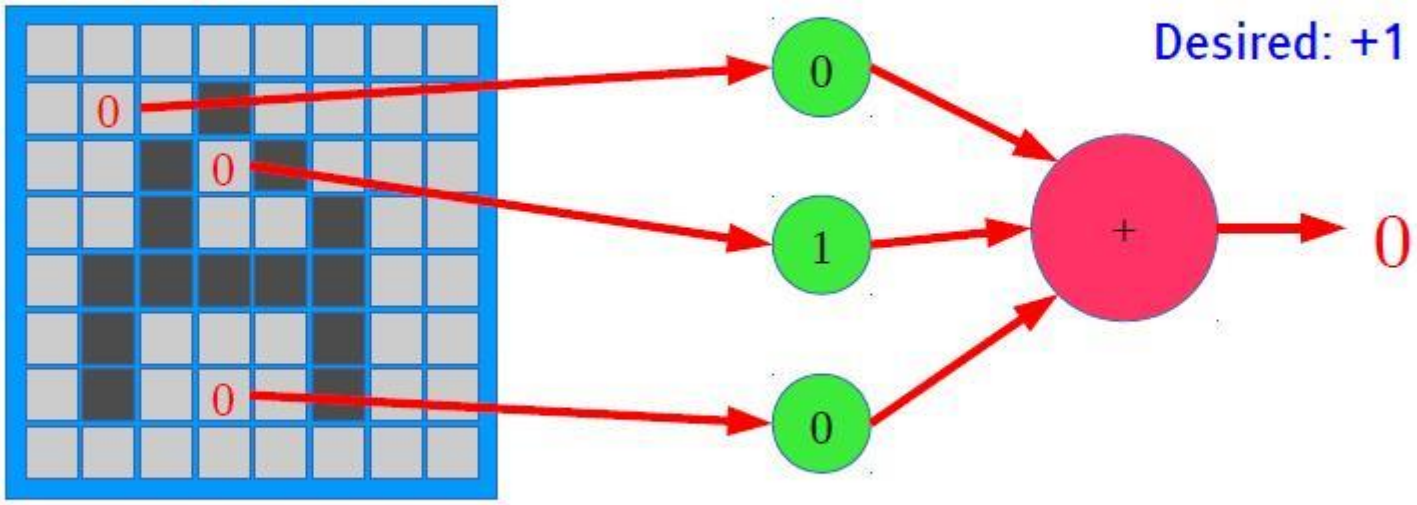
- What if the writing style varies?



Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

- What if the writing style varies?
  - The output may become incorrect

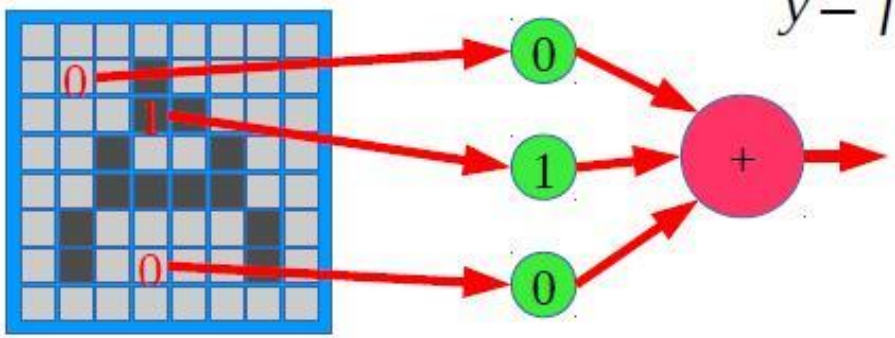


Source: Intelligence artificielle
Yann Le cun, 2015-2016

The Perceptron

- Training set: $(X^1,Y^1),(X^2,Y^2),.....,(X^P,Y^P)$
- Take one sample (X^k,Y^k), if the desired output is +1 but the actual output is -1
  - Increase the weights whose input is positive
  - Decrease the weights whose input is negative
- If the desired is -1 and actual is +1, do the converse.
- If desired and actual are equal, do nothing

$$w_i(t+1) = w_i(t) + (y_i^p - f(W'X^p))x_i^p$$

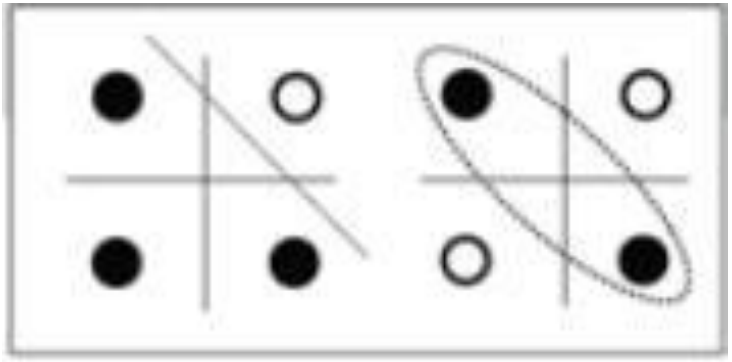


$$y = f\left(\sum_{i=1}^{N} w_i x_i + w_0\right) = f(W'X)$$

Source: Intelligence artificielle
Yann Le cun, 2015-2016

AI Winter

- 1969, Minsky, 《perceptron》
  - XOR
  - Two layers, computation impossible.



Conclusion for first phase

- Training by iteration
  - Inference, calculate f(x)
  - Compare the difference between the f(x) and y
  - Adjust the weights. (gradient based)

Source: Intelligence artificielle
Yann Le cun, 2015-2016

Second: Back Propagation

- 1986, Backpropagation（反向传播）
  - Calculate gradient efficiently O(d^2)
- Routine of training
  - Forward
  - Back-propagation
  - Update weights based on gradients

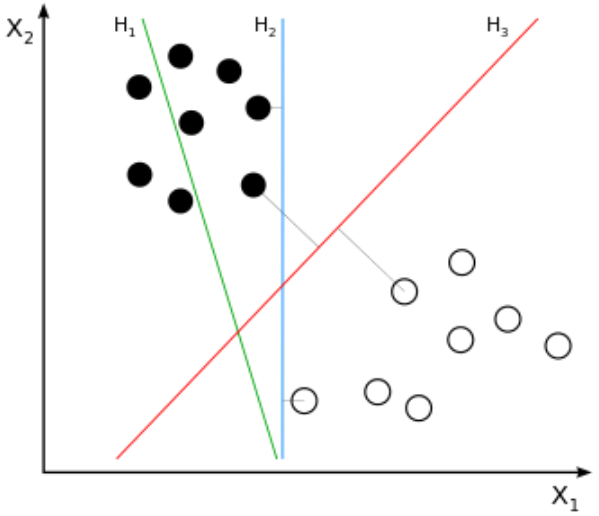
$$x \xrightarrow[w_1]{} h_1 \xrightarrow[w_2]{} \cdots \xrightarrow[w_n]{} h_n \xrightarrow[w_{n+1}]{} y \longrightarrow l(y)$$

$$\frac{\partial h_1}{\partial x} \leftarrow \frac{\partial h_2}{\partial h_1} \leftarrow \cdots \leftarrow \frac{\partial y}{\partial h_n} \leftarrow \frac{\partial l(y)}{\partial y}$$

$$\frac{\partial h_1}{\partial w_1} \quad \cdots \quad \frac{\partial h_n}{\partial w_n} \quad \frac{\partial y}{\partial w_{n+1}}$$

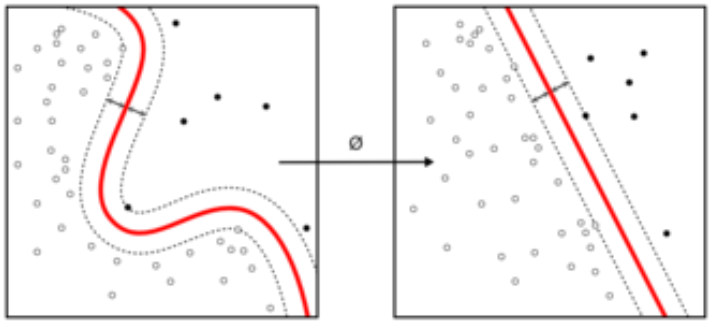
Beaten by SVM

- 1990s, Vapnik, support vector machines.（支持向量机）
  - y=f(WX+b)
  - Globally optimization
  - High efficiency, just one layer



- Loss function
  - Function + loss

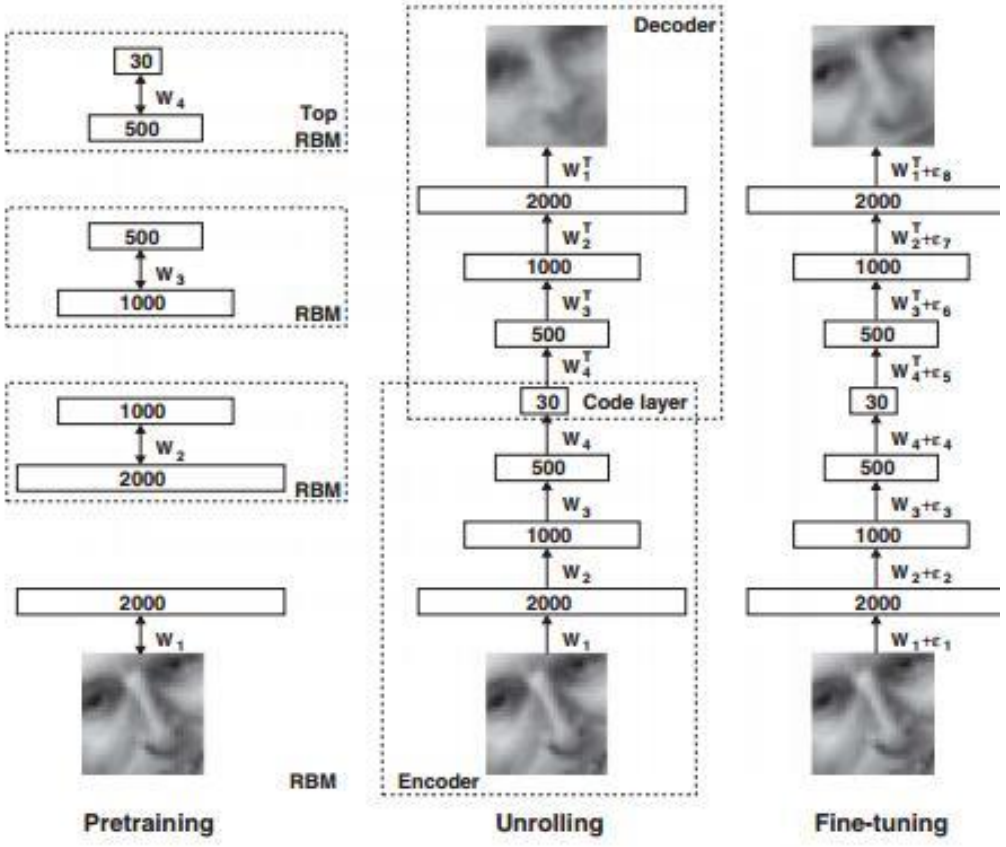
$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0,1-y_i(\vec{w}\cdot\overrightarrow{x_i}-b)\right)\right]+\lambda\|\vec{w}\|^2$$



Kernel trick for nonlinear

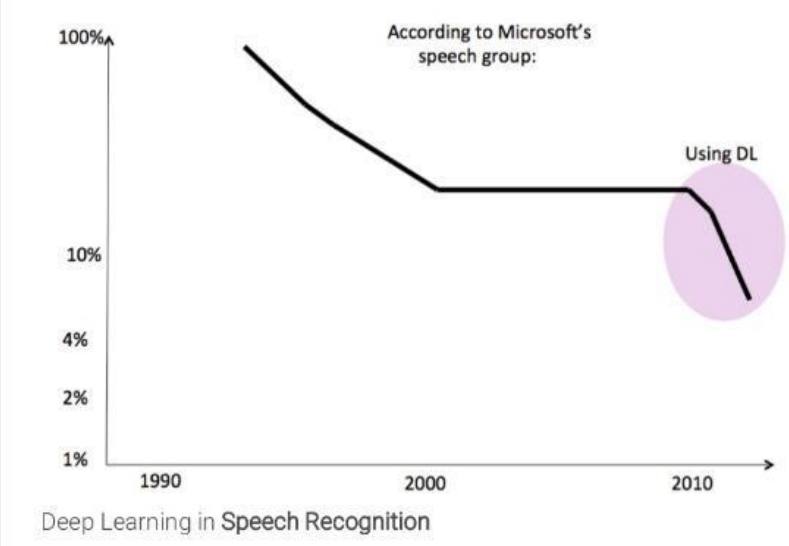
Third: deep learning

- 2006, Geoffery Hinton
  - Deep Belief network
    - Pre-training
    - Fine-tuning



Third: deep learning

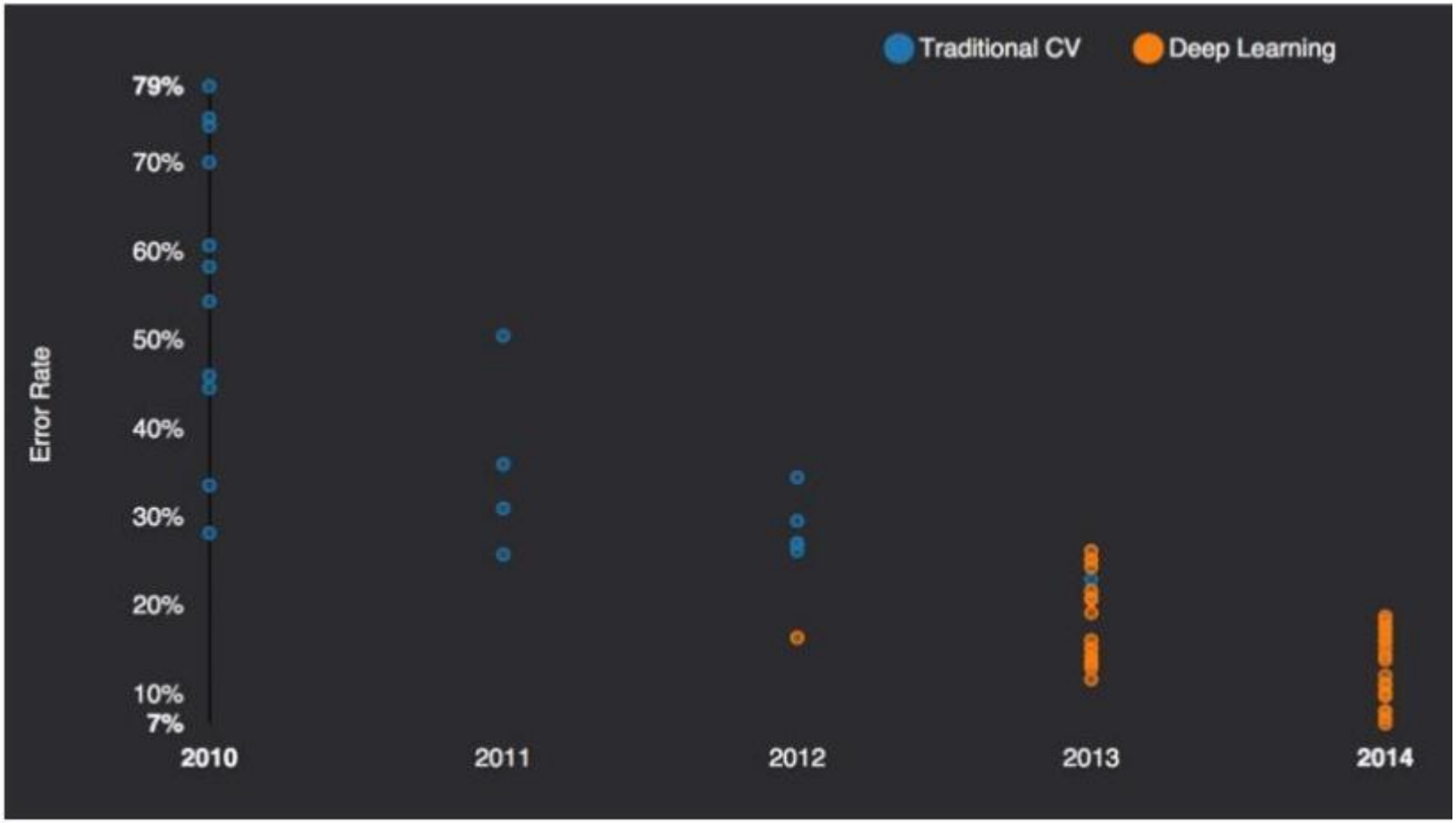
- 2011, audio



The task	Hours of training data	Deep neural network	Gaussian Mixture Model	GMM with more data
Switchboard (Microsoft Research)	309	18.5%	27.4%	18.6% (2000 hrs)
English broadcast news (IBM)	50	17.5%	18.8%	
Google voice search (android 4.1)	5,870	12.3% (and falling)		16.0% (>>5,870 hrs)

Third: deep learning

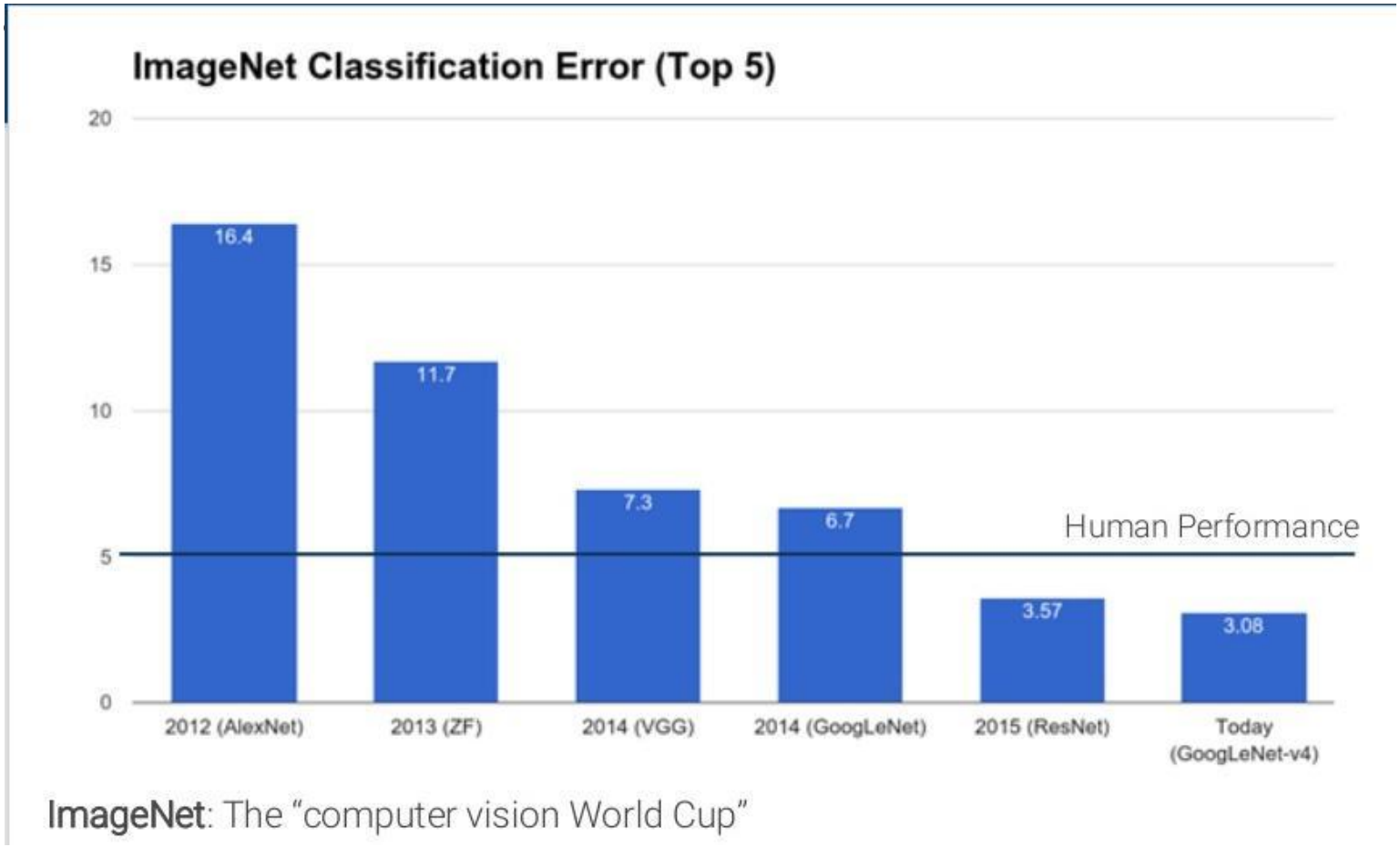
- 2012,imageNet
- .



ImageNet: The "computer vision World Cup"

Third: deep learning

- 2012,imageNet
- ..



Why deep learning grow so fast?

- Big Data
- More Powerful and cheaper machine
- Open Source
  - Code: git-hub
  - Paper: arxiv

项目	型号	价格
CPU	Intel i7-6850K	¥4,699
主板	华硕（ASUS）X99-DELUXE II 主板	¥3,995
机箱	乔思伯（JONSBO）W2 黑色 全塔式机箱	¥519
电源	美商海盗船（USCorsair）额定650W RM650x 电源	¥869
内存	金士顿(Kingston)骇客神条 Fury系列 DDR4 2400 64G (16GBx4)	¥2,459
散热器	猫头鹰（NOCTUA）NH-D9L CPU散热器	¥484
显卡	EVGA GTX 1070 8G SC GAMING ACX 3.0 Black Edition	¥3,199
SSD	闪迪(SanDisk) 加强版 480G 固态硬盘	¥680
合计		¥16,904

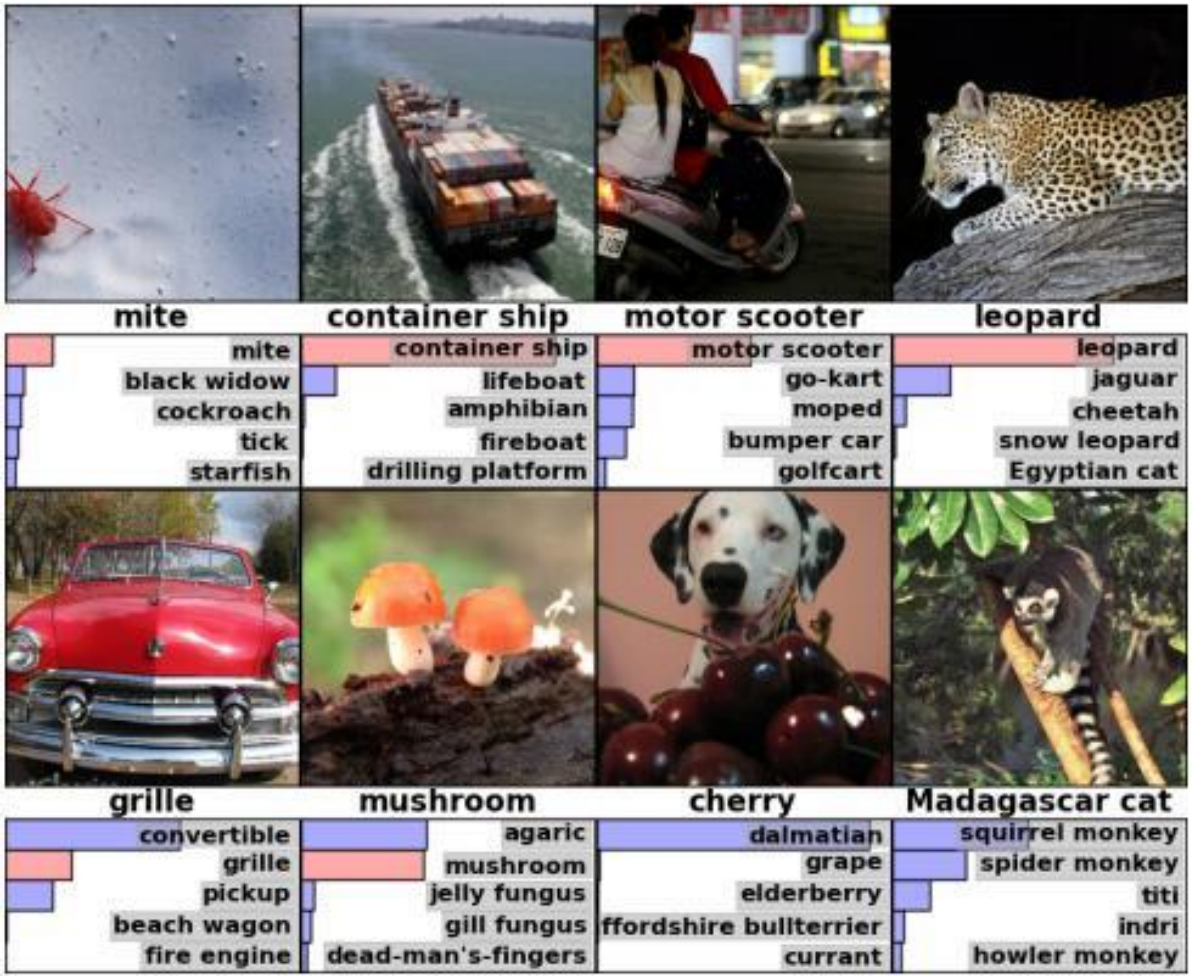
Source: 程序员的深度学习入门指南
费良宏, 2016

Outline

- Basic Concept
  - Machine learning
  - Neural network
  - Deep network
- History of neural network
  - Perceptron
  - BackPropagation
  - Deep learning
- Application

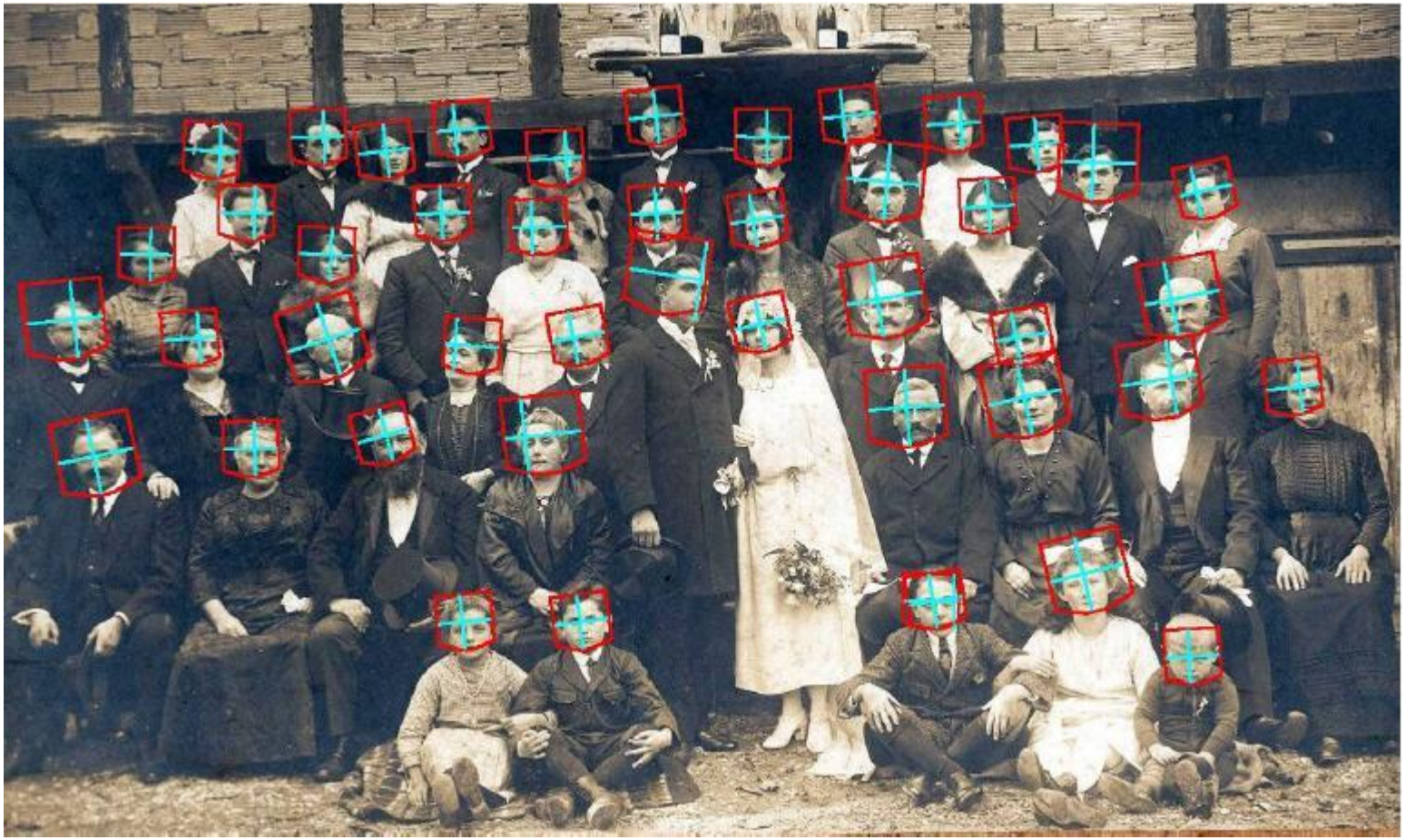
Application

- **Object Classification**



Application

- **Object detection**

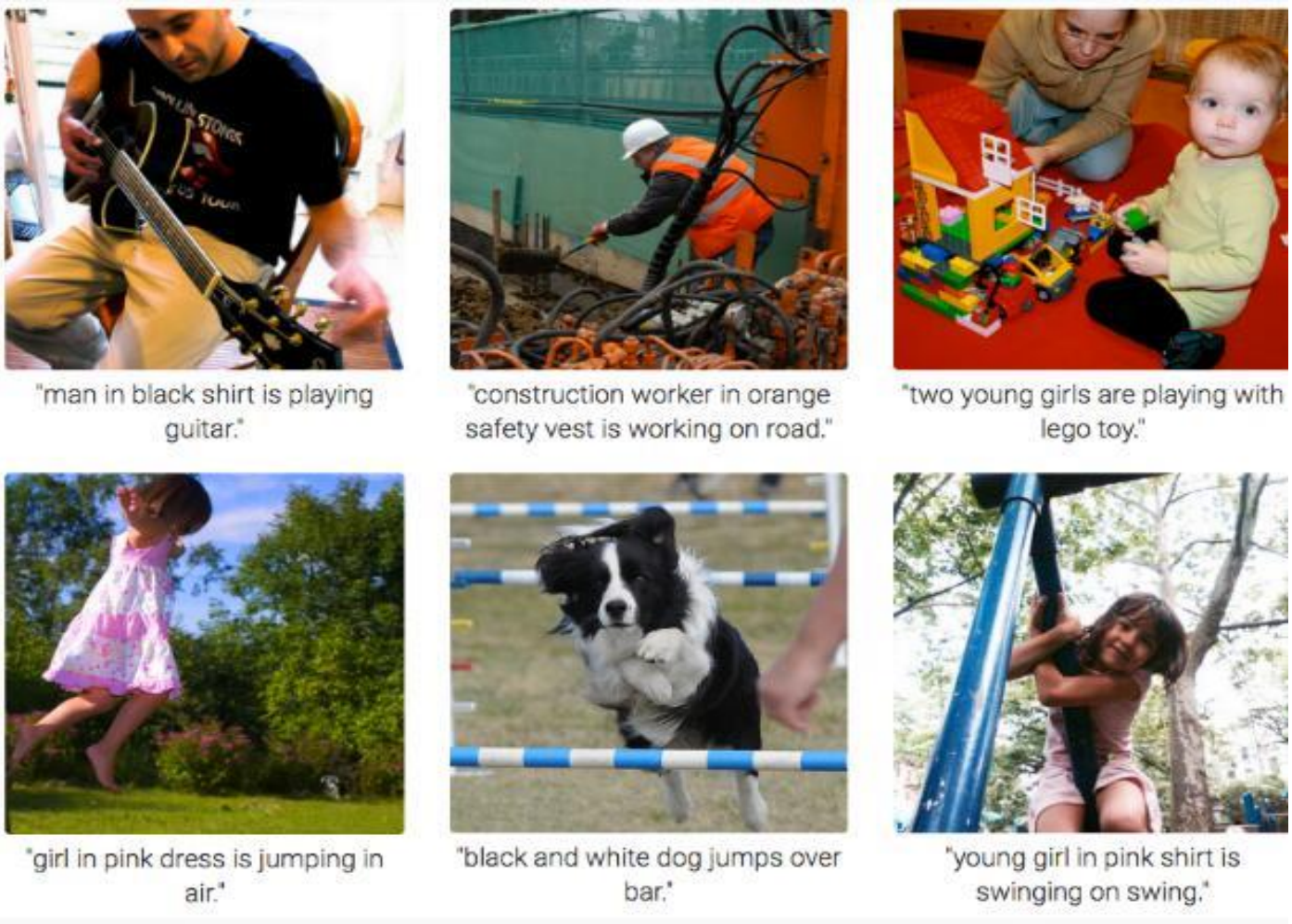


Application

- **Scene Parsing**

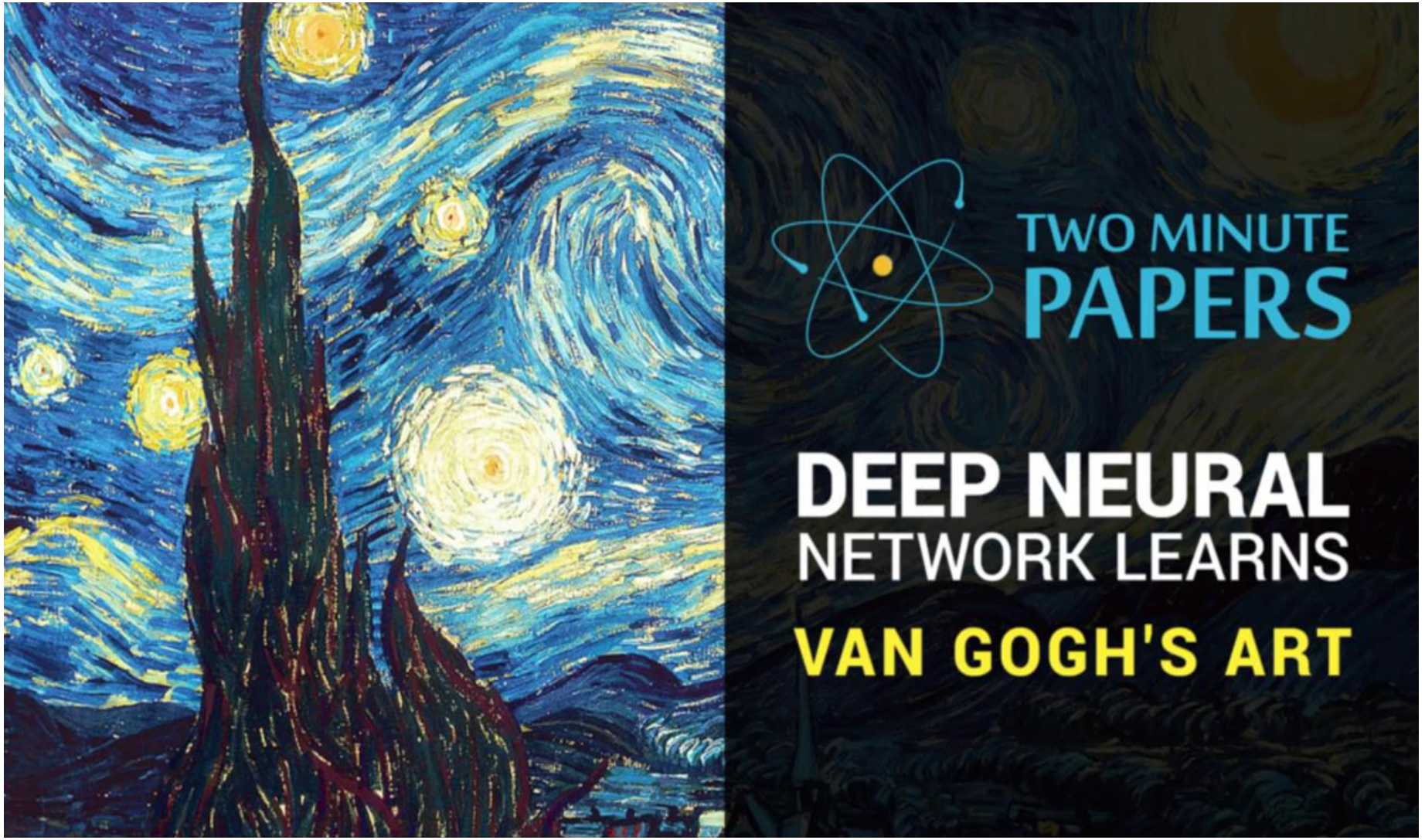
Application

- **Automatic Image Caption Generation**



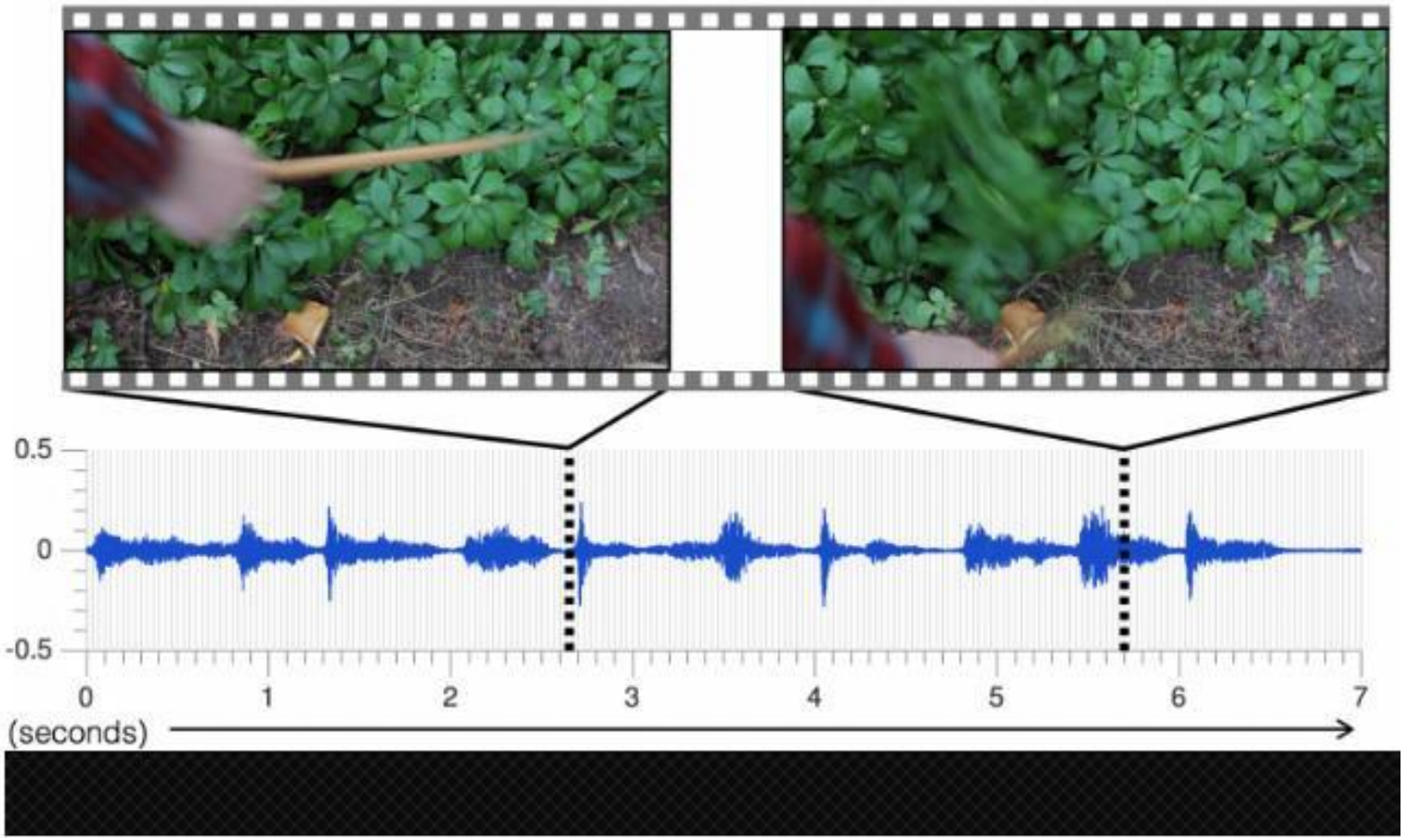
Application

- **Artistic style learning on images**



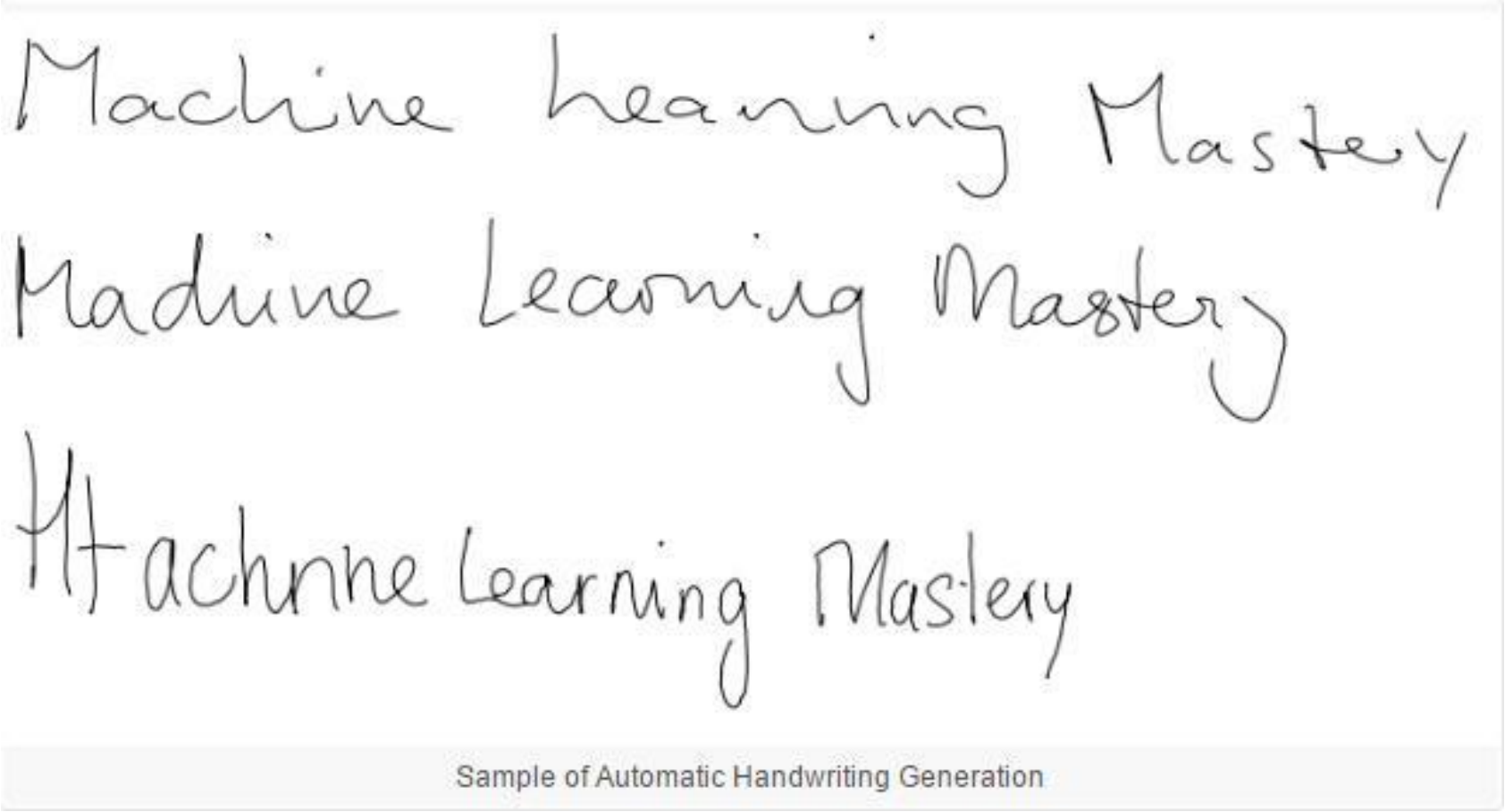
Application

- **Automatically Adding Sounds To Silent Movies**



Application

- **Automatic Handwriting Generation**



Sample of Automatic Handwriting Generation

Application

- **Automatic Text Generation**
  - Shakespeare
  - Wikipedia articles (including the markup)
  - Algebraic Geometry (with LaTeX markup)
  - Linux Source Code

- **http://karpathy.github.io/2015/05/21/rnn-effectiveness/**

Application

- **Alpha Go**

Q&A