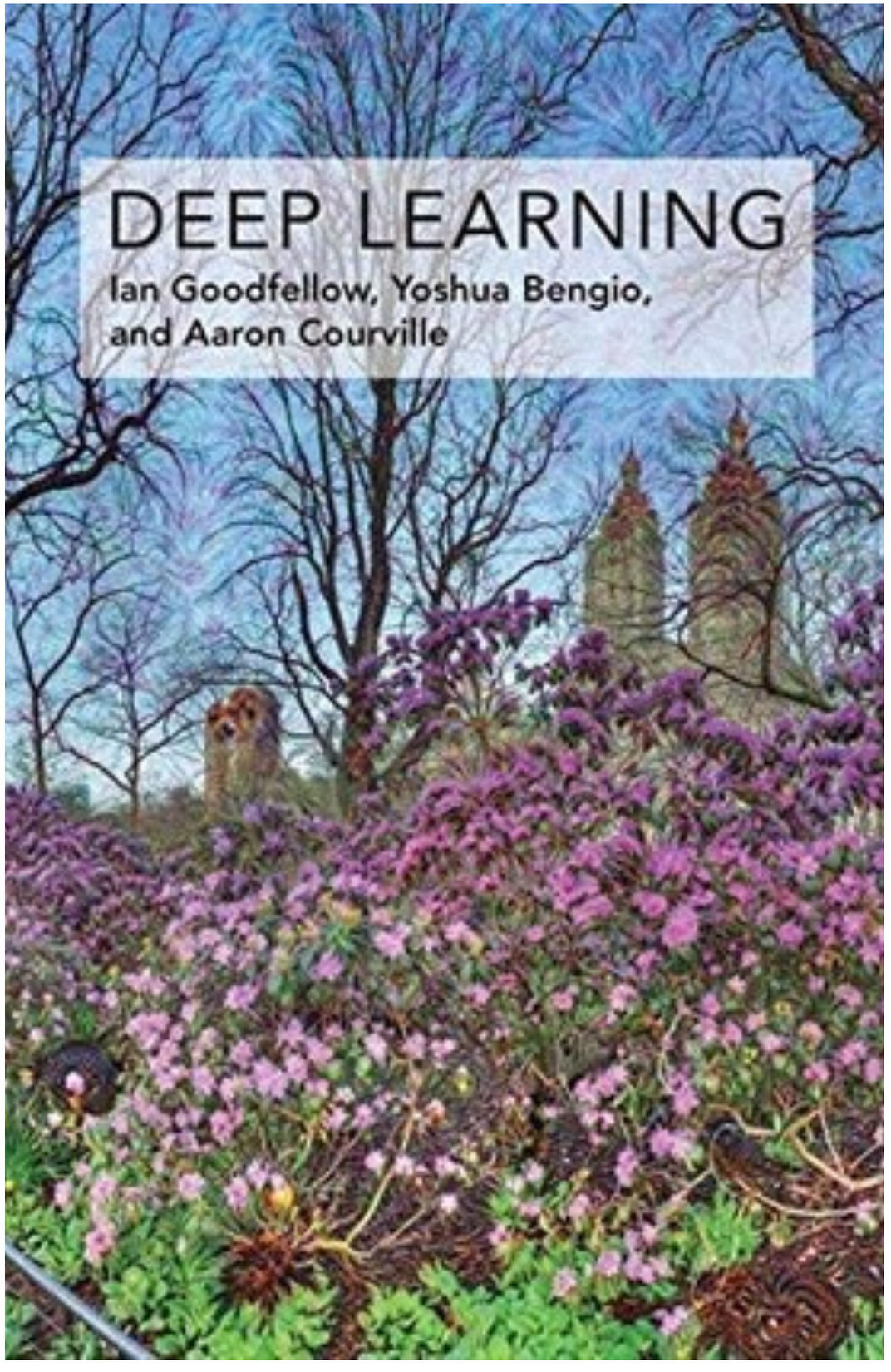
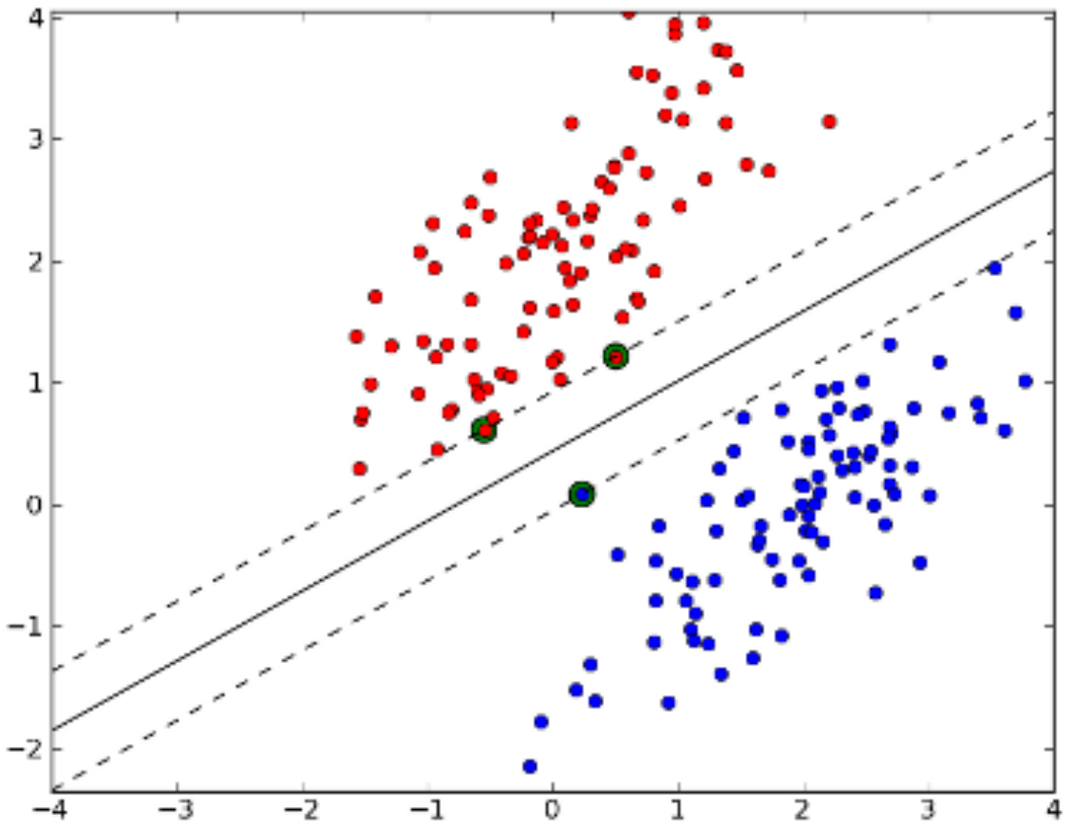
深度学习基础


DEEP LEARNING
Ian Goodfellow, Yoshua Bengio,
and Aaron Courville

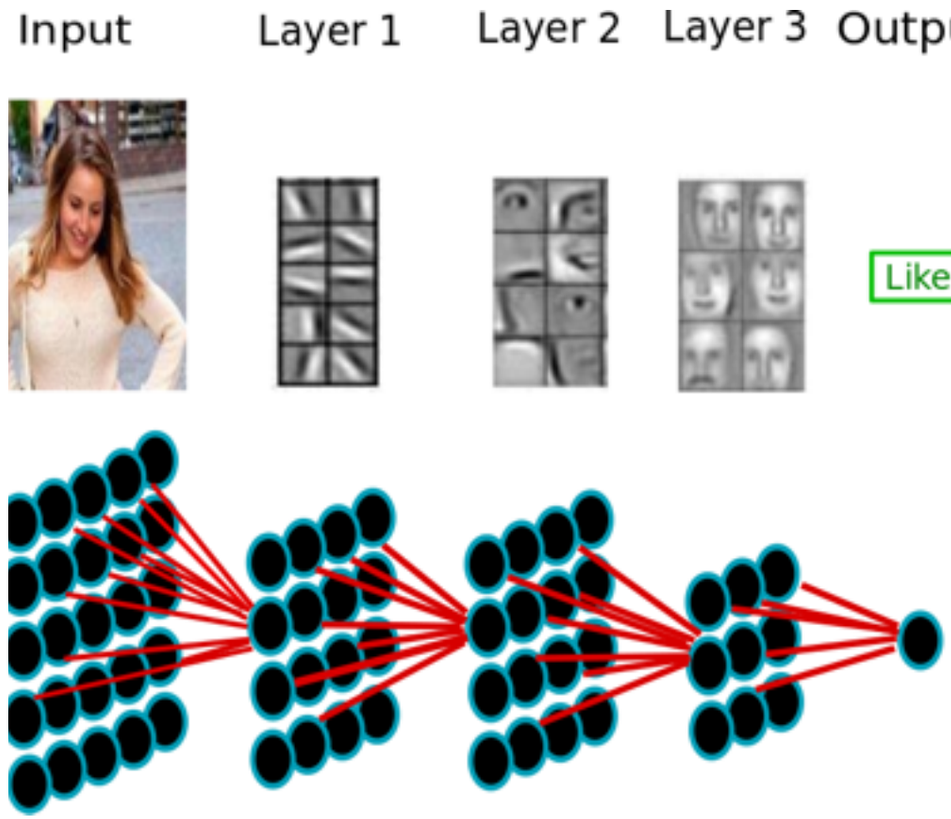
- 深度学习参考资料:
- 英文资料:
- 1. Deep Learning by Yoshua Bengio, Ian Goodfellow and Aaron Courville
- 2. Deep Learning by Microsoft Research
- 3. Stanford University 2016 CS231n: Convolutional Neural Networks for Visual Recognition by Fei-Fei Li （http://cs231n.stanford.edu/ ）
- 中文资料:
- 1. 神经网络与深度学习讲义20151211， 邱锡鹏
- 2. 神经网络与机器学习（第3版）

深度学习介绍

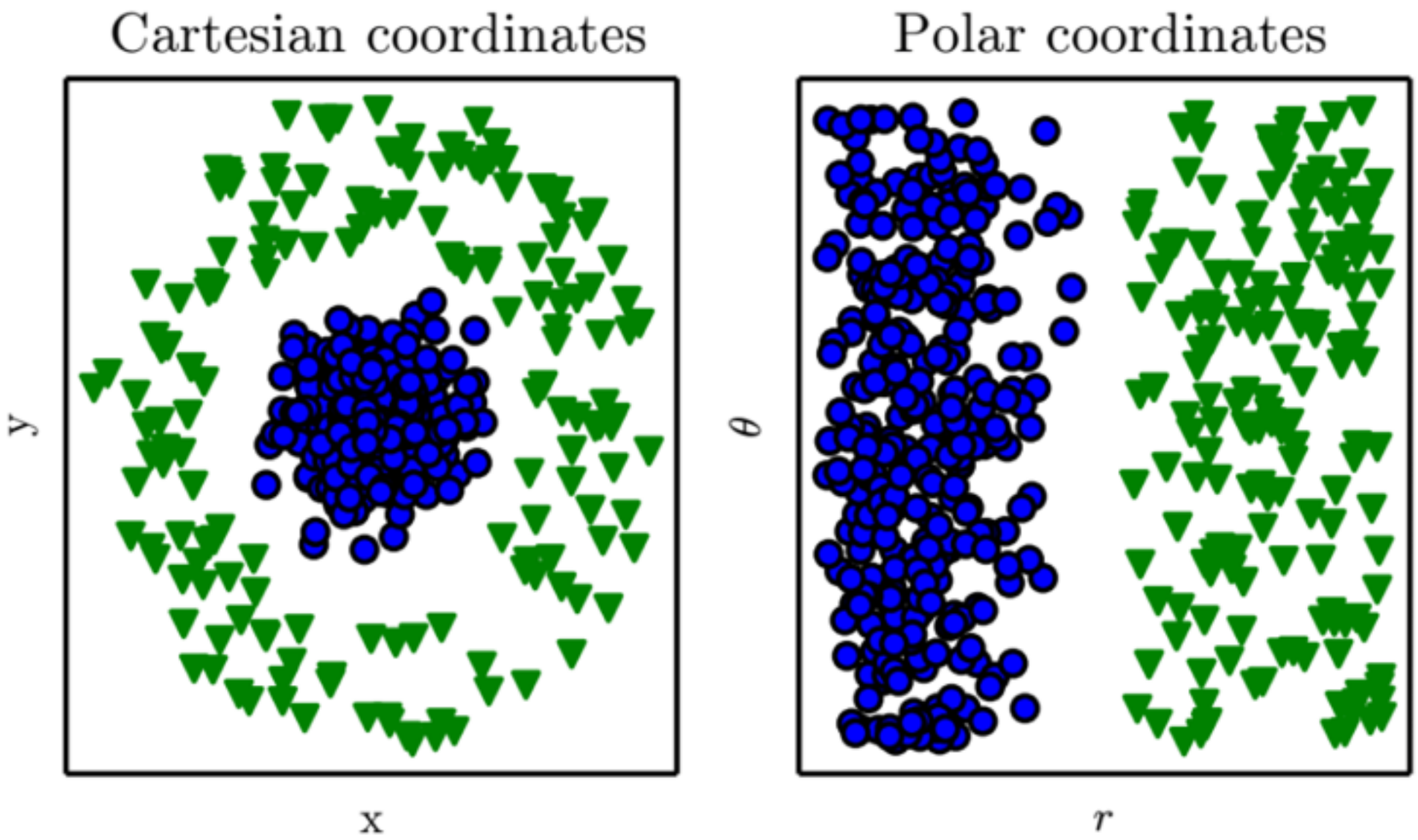
Shallow Learning



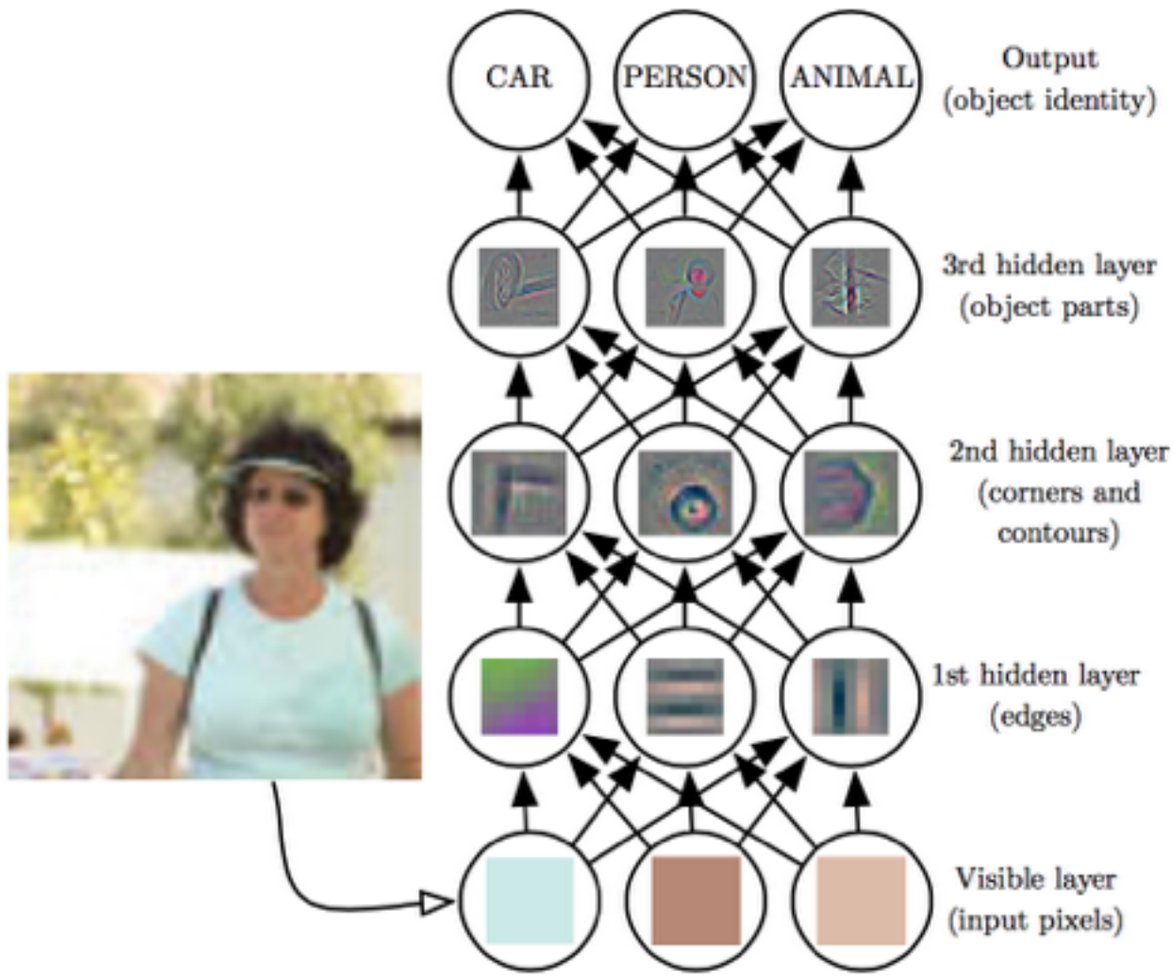
Deep Learning



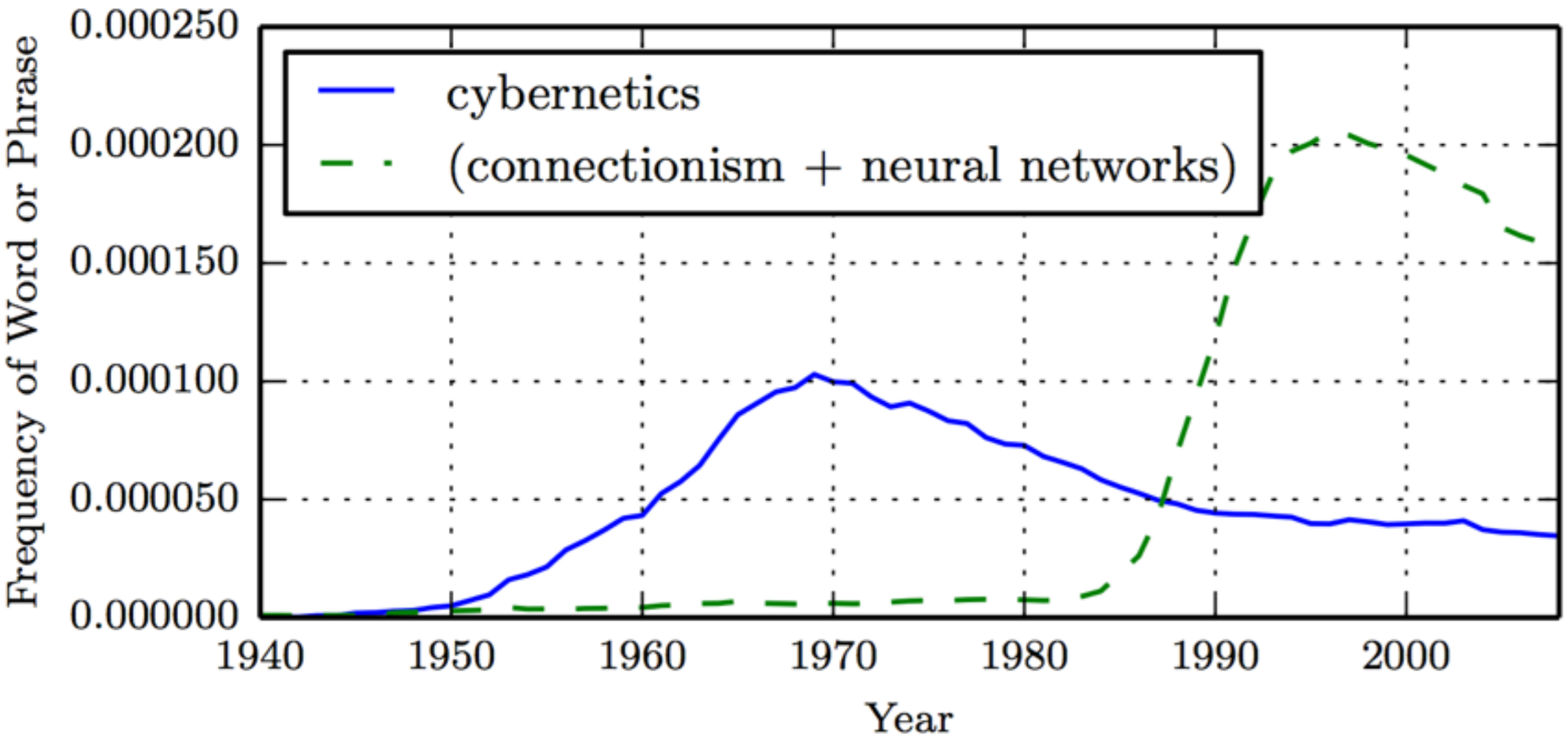
特征表示



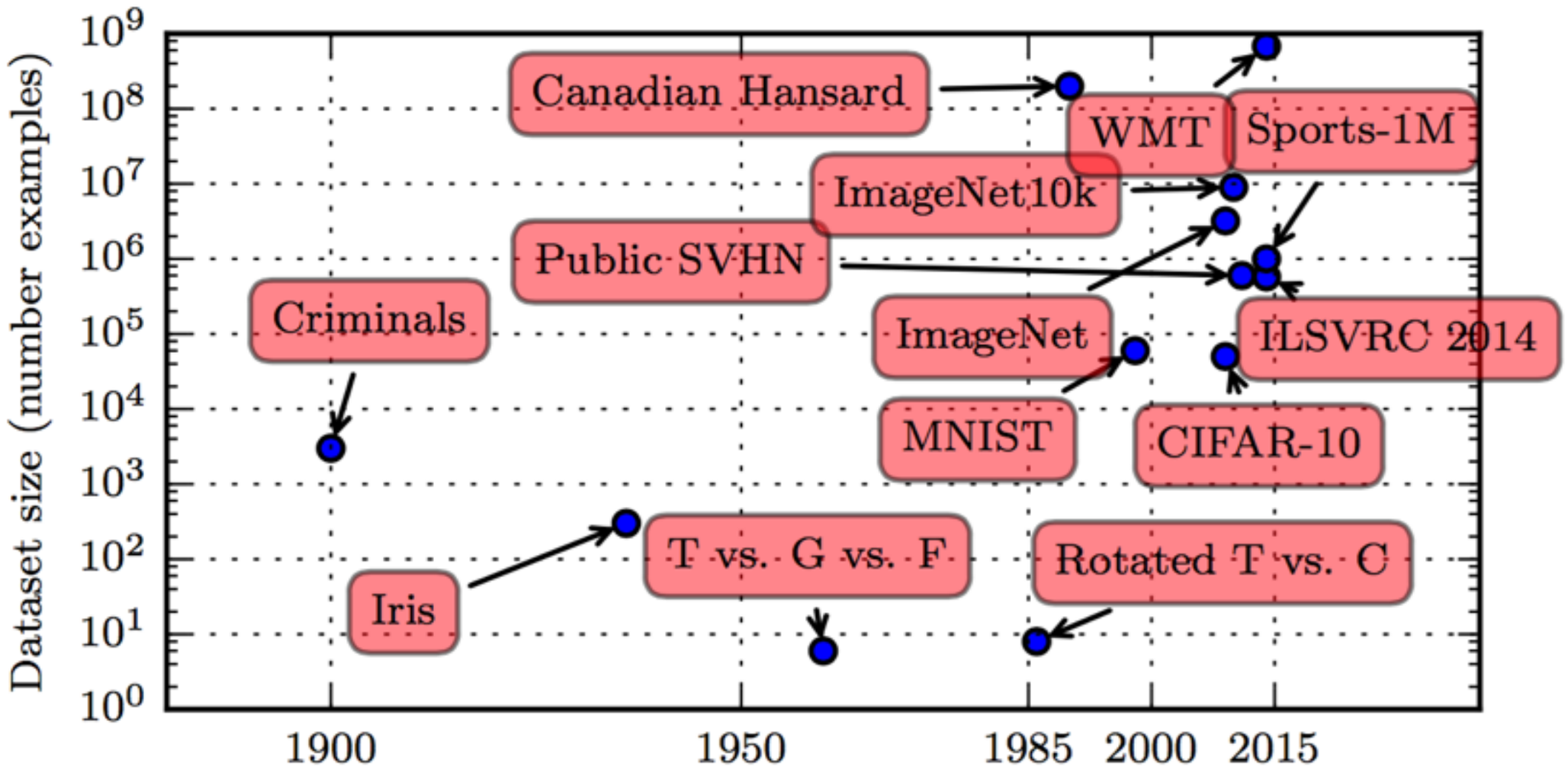
深度表达


CAR
PERSON
ANIMAL
Output
(object identity)
3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)

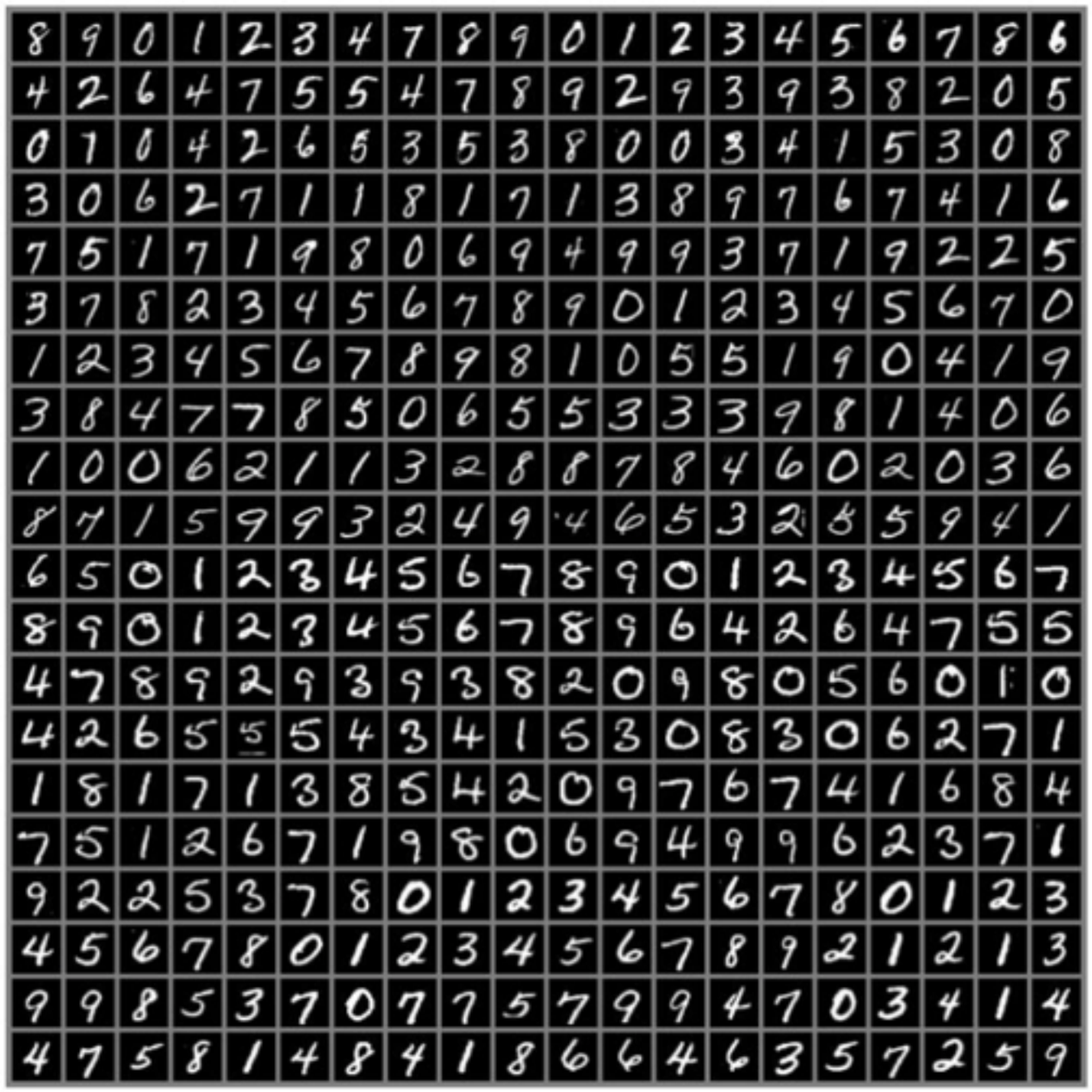
深度学习的发展历史


cybernetics
(connectionism + neural networks)
Frequency of Word or Phrase
0.000250
0.000200
0.000150
0.000100
0.000050
0.000000
1940
1950
1960
1970
1980
1990
2000
Year

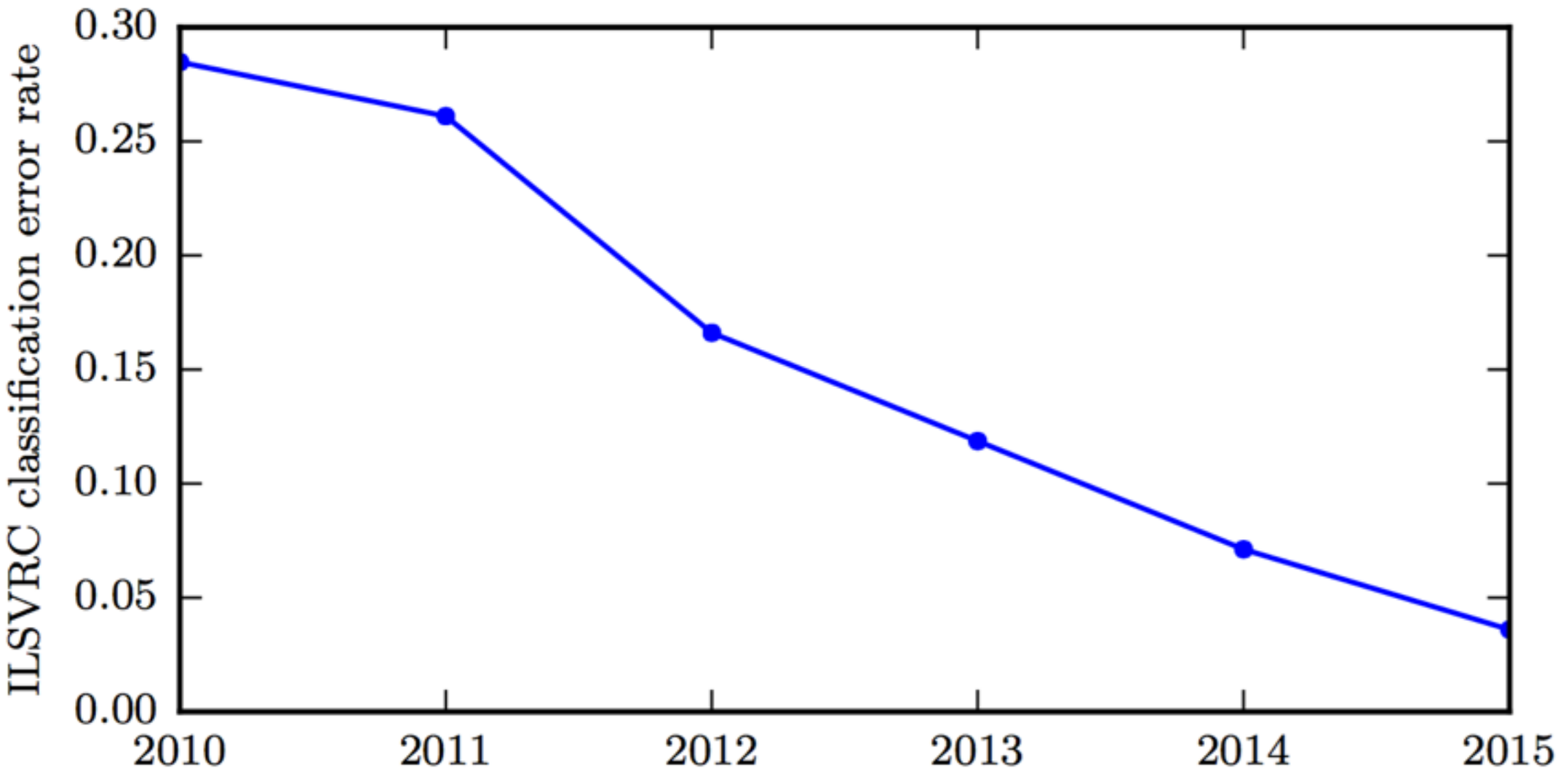
增长的数据集


Dataset size (number examples)
10^9
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
1900
1950
1985
2000
2015
Canadian Hansard
WMT
Sports-1M
ImageNet10k
Public SVHN
Criminals
ImageNet
ILSVRC 2014
MNIST
CIFAR-10
Iris
T vs. G vs. F
Rotated T vs. C

The MNIST Dataset



增长的识别率（错误率在下降）



深度学习的主要应用

- 1.人脸识别

- 2.以图搜图

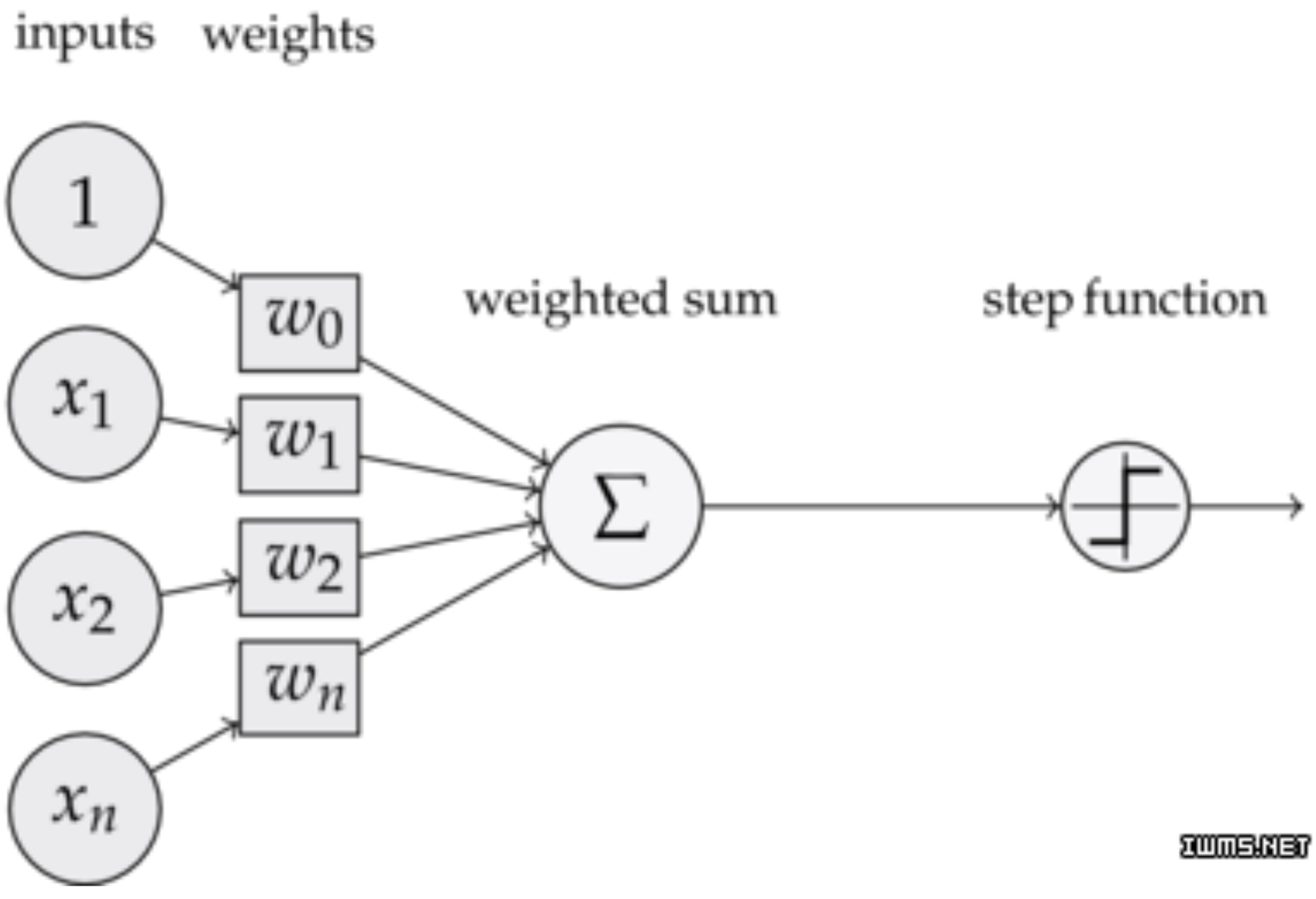

Google
圖片
搜尋圖片
廣告服務
商業解決方案
Google 完全手冊
© 2013 - 隱私權與條款

- 3.街景识别

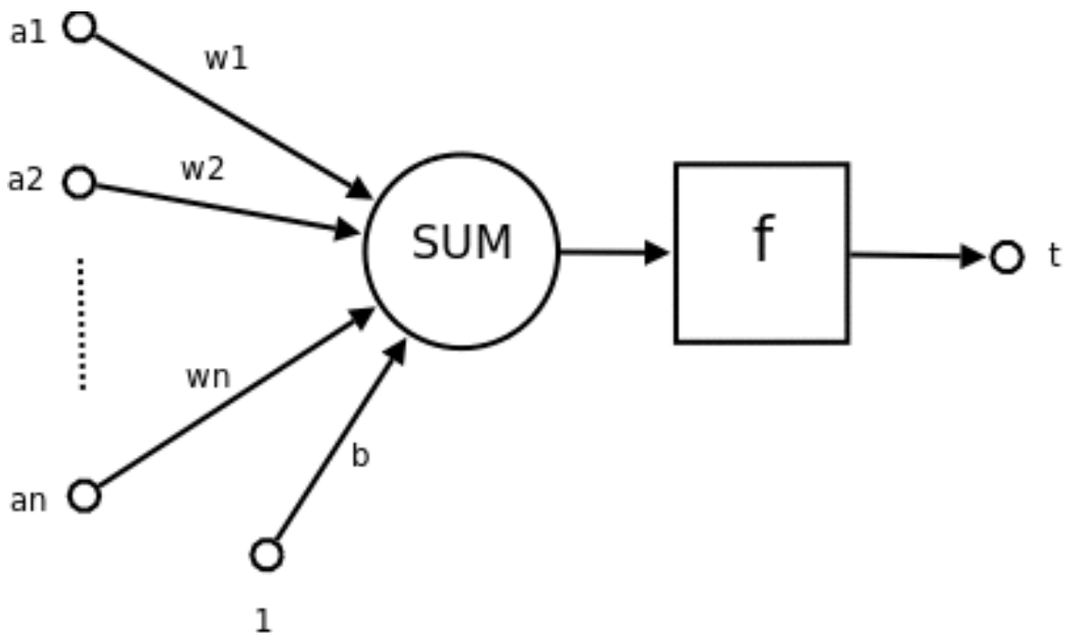

New York (UCC = +0.10)
2007
2014
New York (UCC = +0.01)
2007
2014
(b) Street Blocks with No Significant Change in Streetscore
New York (UCC = +6.28)
2007
2014
New York (UCC = +7.59)
2007
2014
(c) Street Blocks with Significant Improvement in Streetscore
Detroit (UCC = -4.70)
2007
2014
Detroit (UCC = -6.54)
2007
2014
(d) Street Blocks with Significant Decline in Streetscore

- 4.服饰搭配

感知器（神经元）


inputs
weights
1
w_0
x_1
w_1
w_2
x_2
w_n
x_n
weighted sum
Σ
step function
IWMS.NET

- 感知器（神经元）



- a1~an为输入向量的各个分量
- w1~wn为神经元各个突触的权值
- b为偏置
- f为传递函数，通常为非线性函数。t为神经元输出

数学表示 $t = f(\vec{W'}\vec{A} + b)$

- $\vec{W}$为权向量，$\vec{W'}$为$\vec{W}$的转置
- $\vec{A}$为输入向量
- b为偏置
- f为传递函数

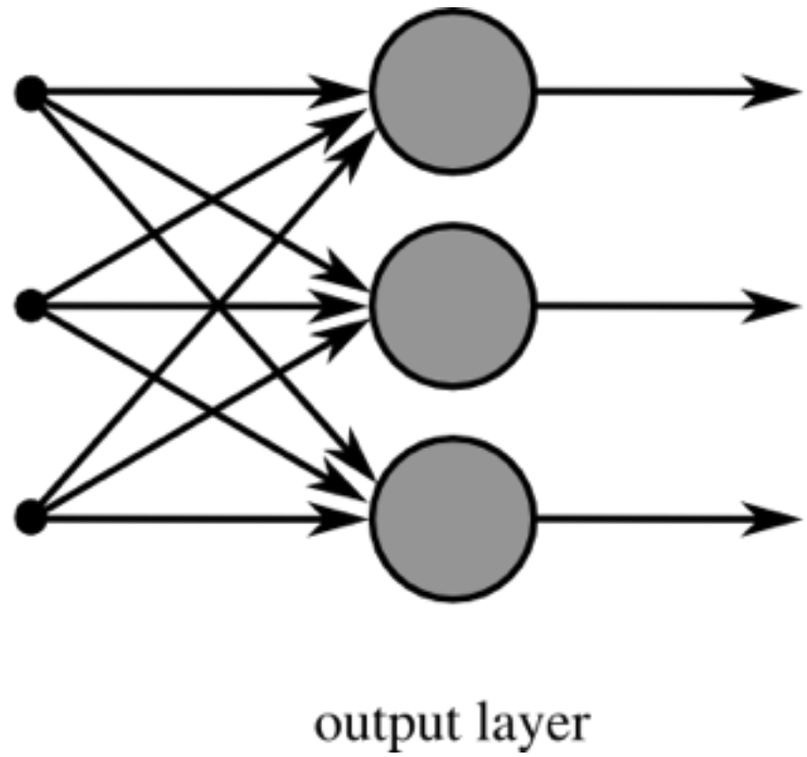
可见，一个神经元的功能是求得输入向量与权向量的内积后，经一个非线性传递函数得到一个标量结果。

单个神经元的作用：把一个n维向量空间用一个超平面分割成两部分（称之为判断边界），给定一个输入向量，神经元可以判断出这个向量位于超平面的哪一边。

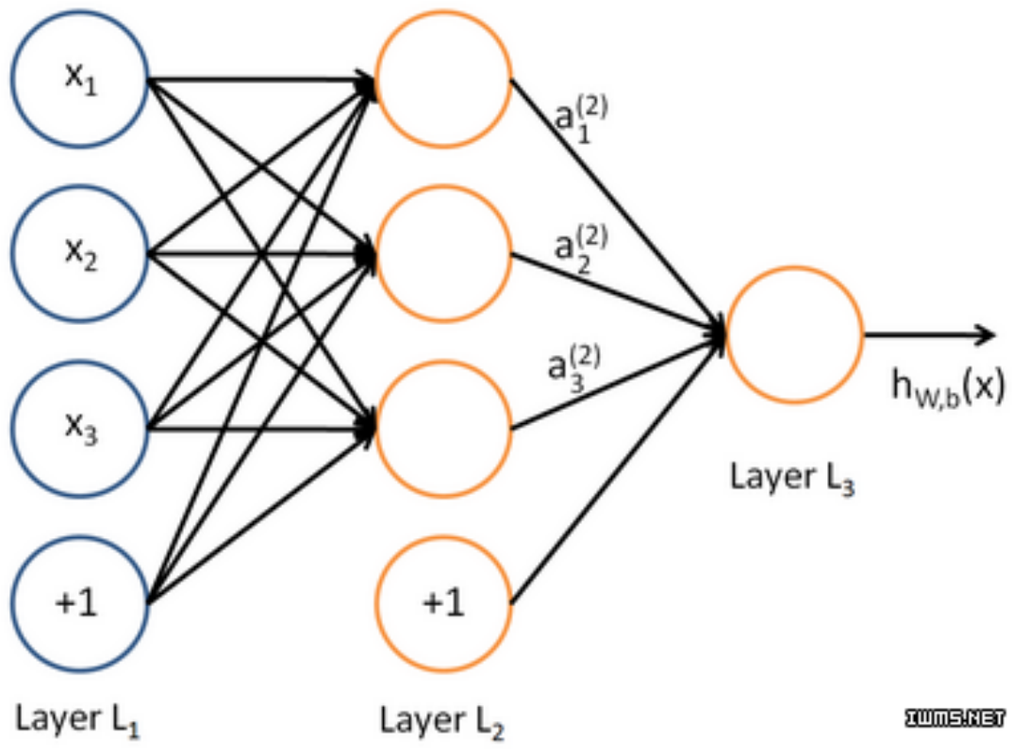
神经网络

- 单层神经网络

- 是最基本的神经元网络形式，由有限个神经元构成，所有神经元的输入向量都是同一个向量。由于每一个神经元都会产生一个标量结果，所以单层神经元的输出是一个向量，向量的维数等于神经元的数目。



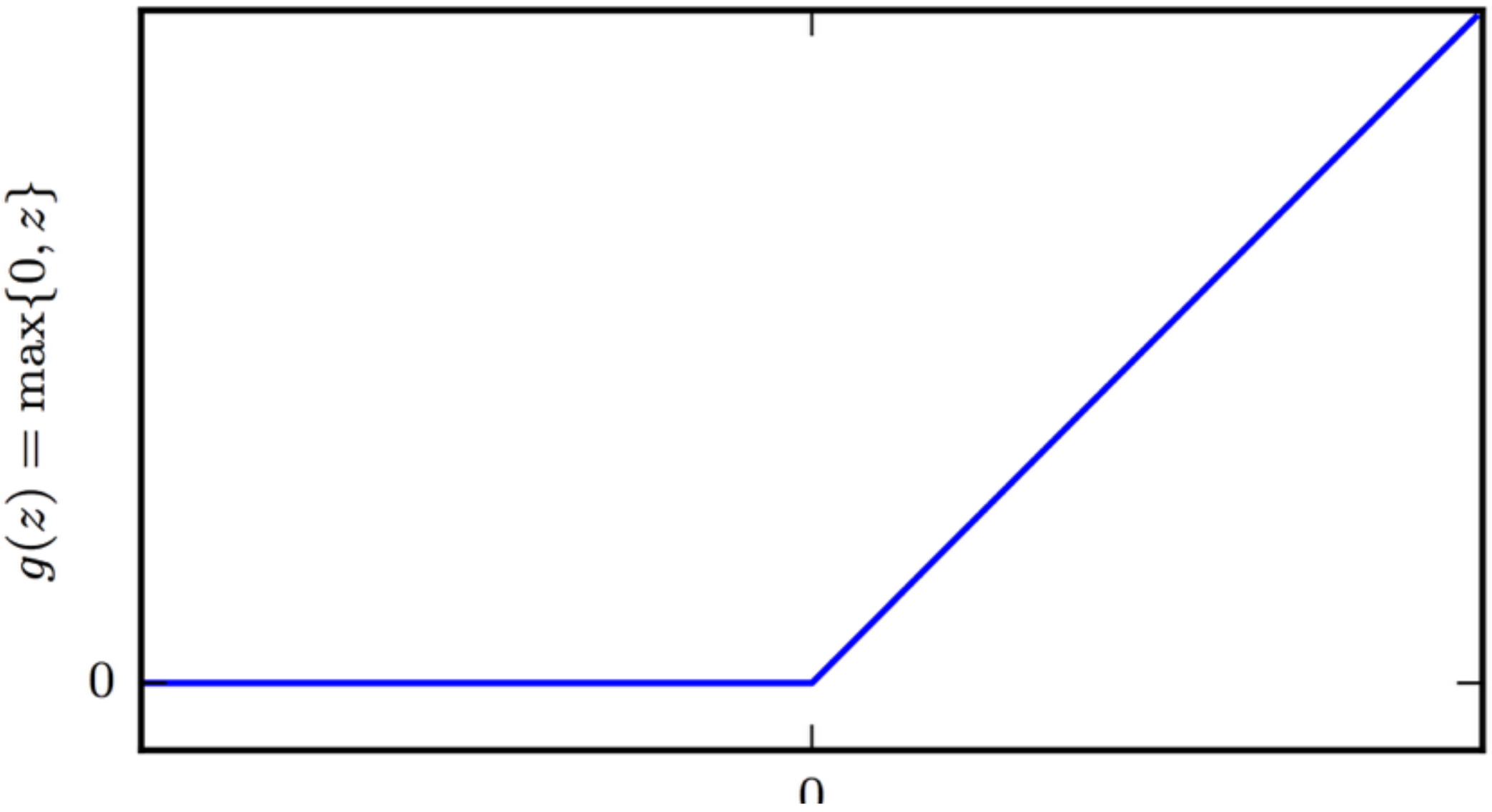
- 多层神经网络



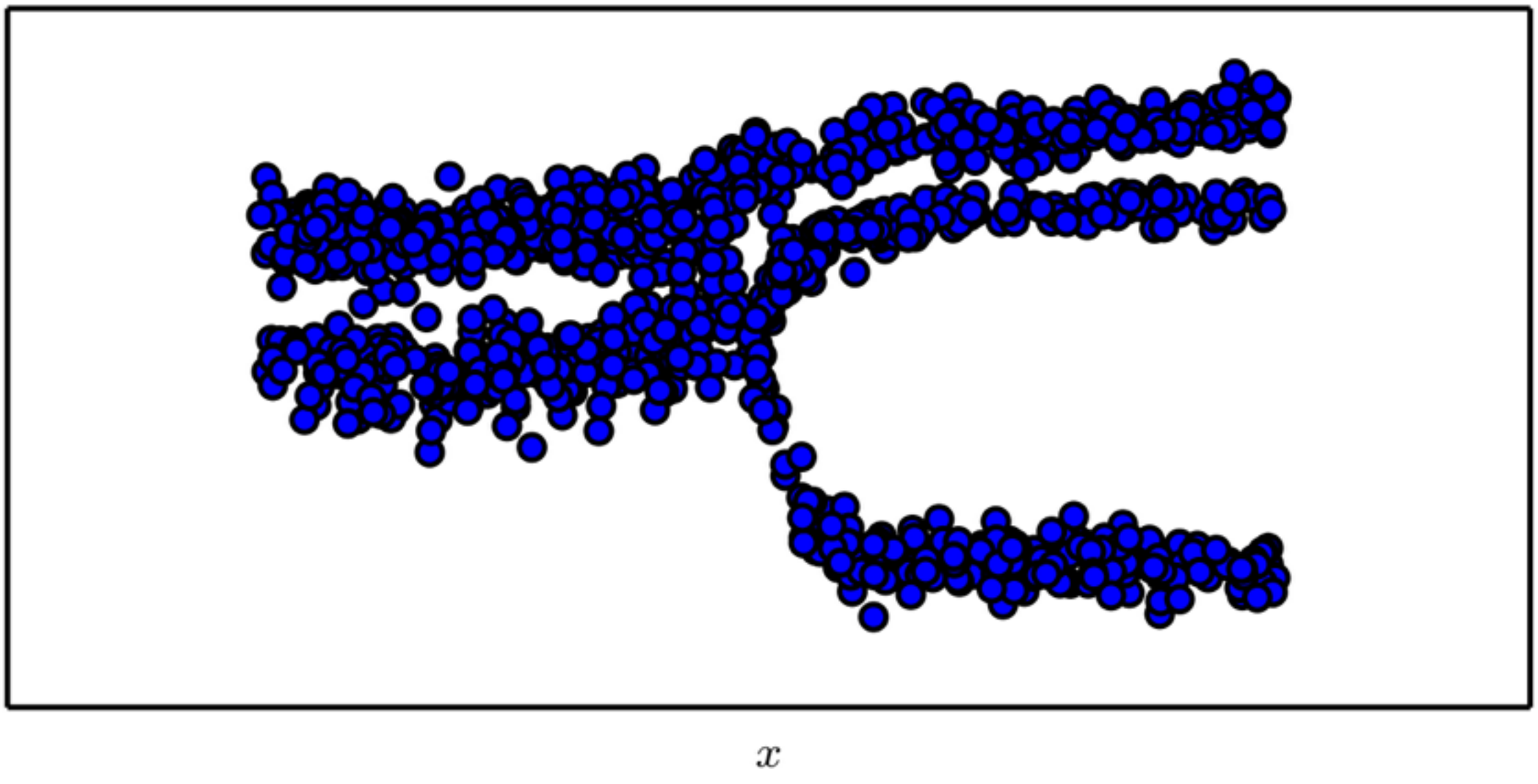
前馈神经网络


神经网络
Neural Network
人工神经网络
ANN
生物神经网络
前馈神经网络
反馈神经网络
单层神经网络
Perceptron
两层神经网络
MLP
多层神经网络
DL
普通深度网络
DNN
深度信念网络
DBN
卷积神经网络
CNN

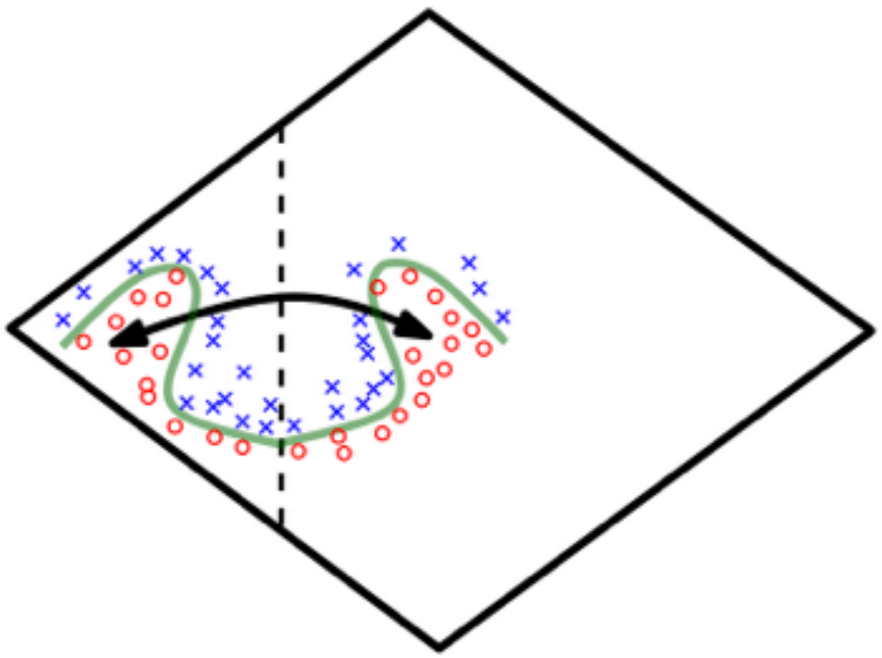
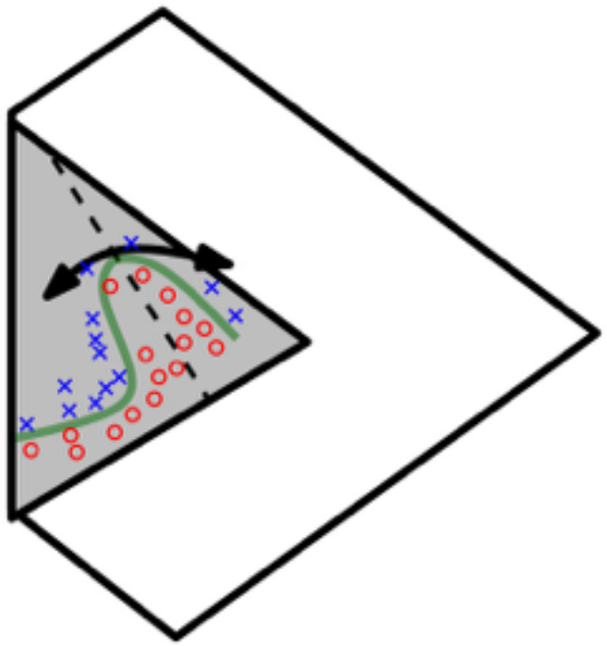
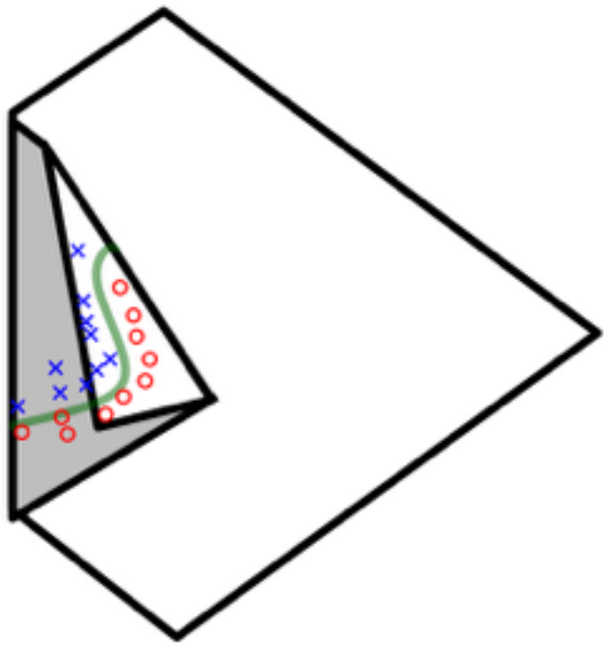
修正线性激活



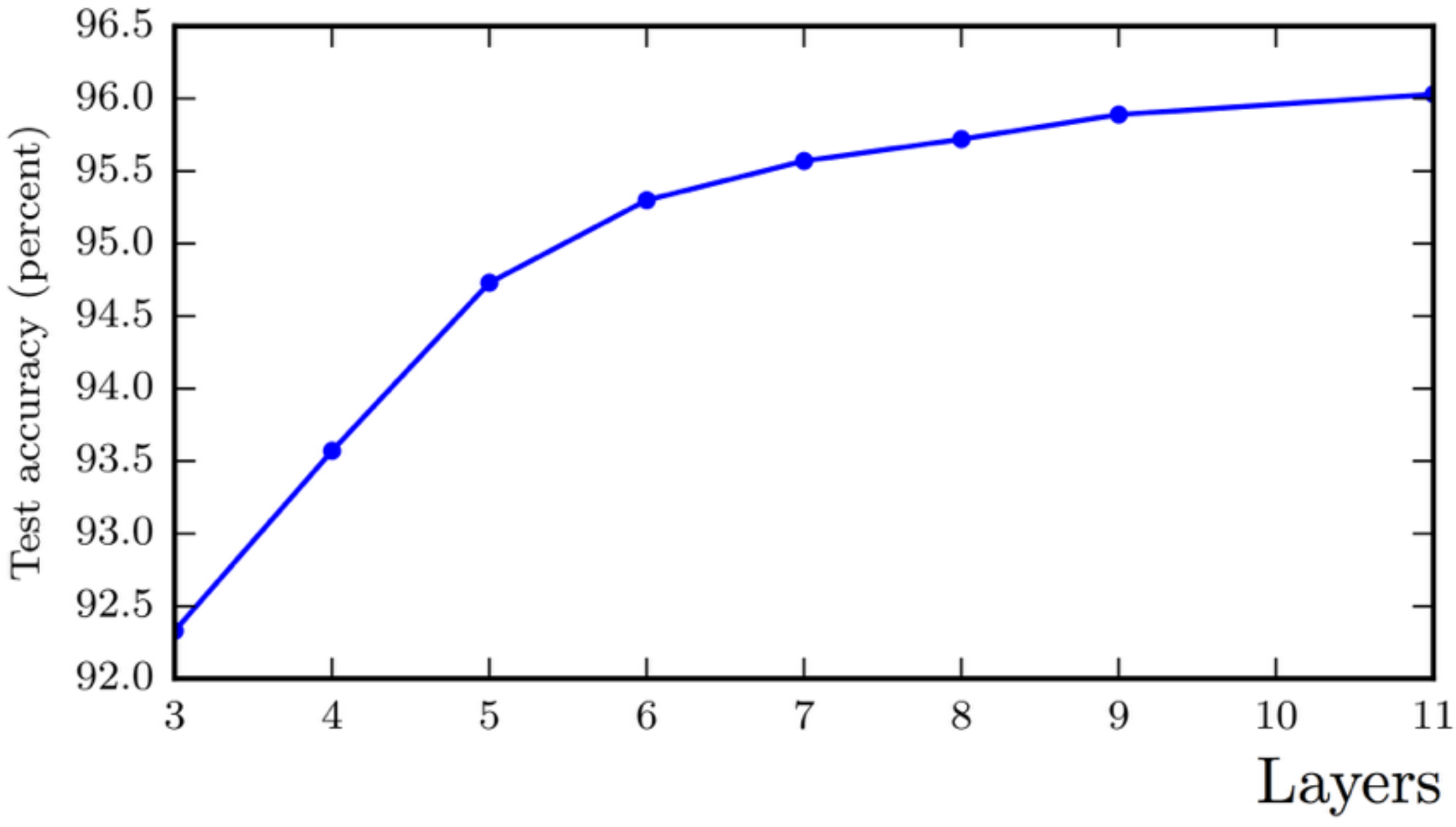
混合密度输出



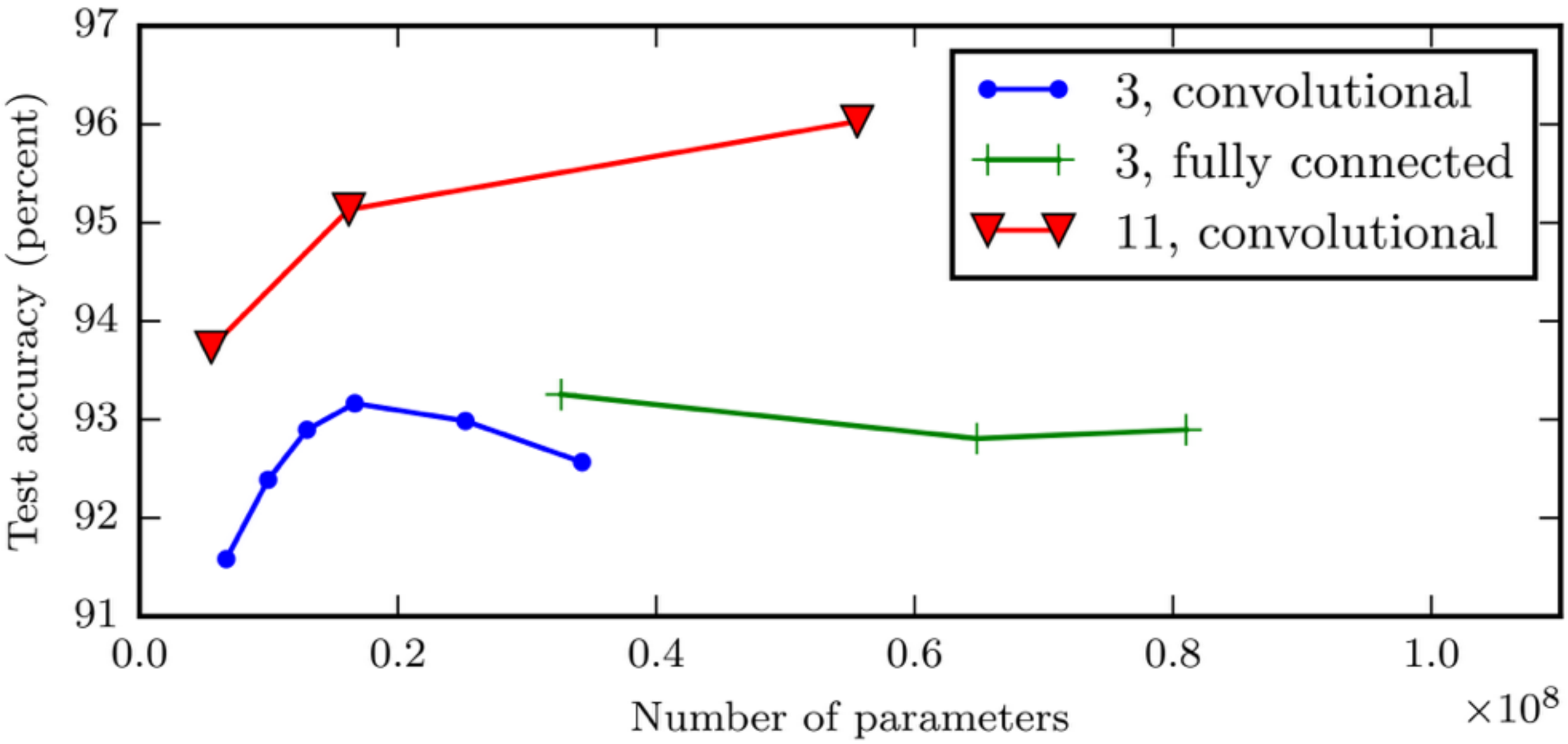
多层次学习的优势

更深的网络

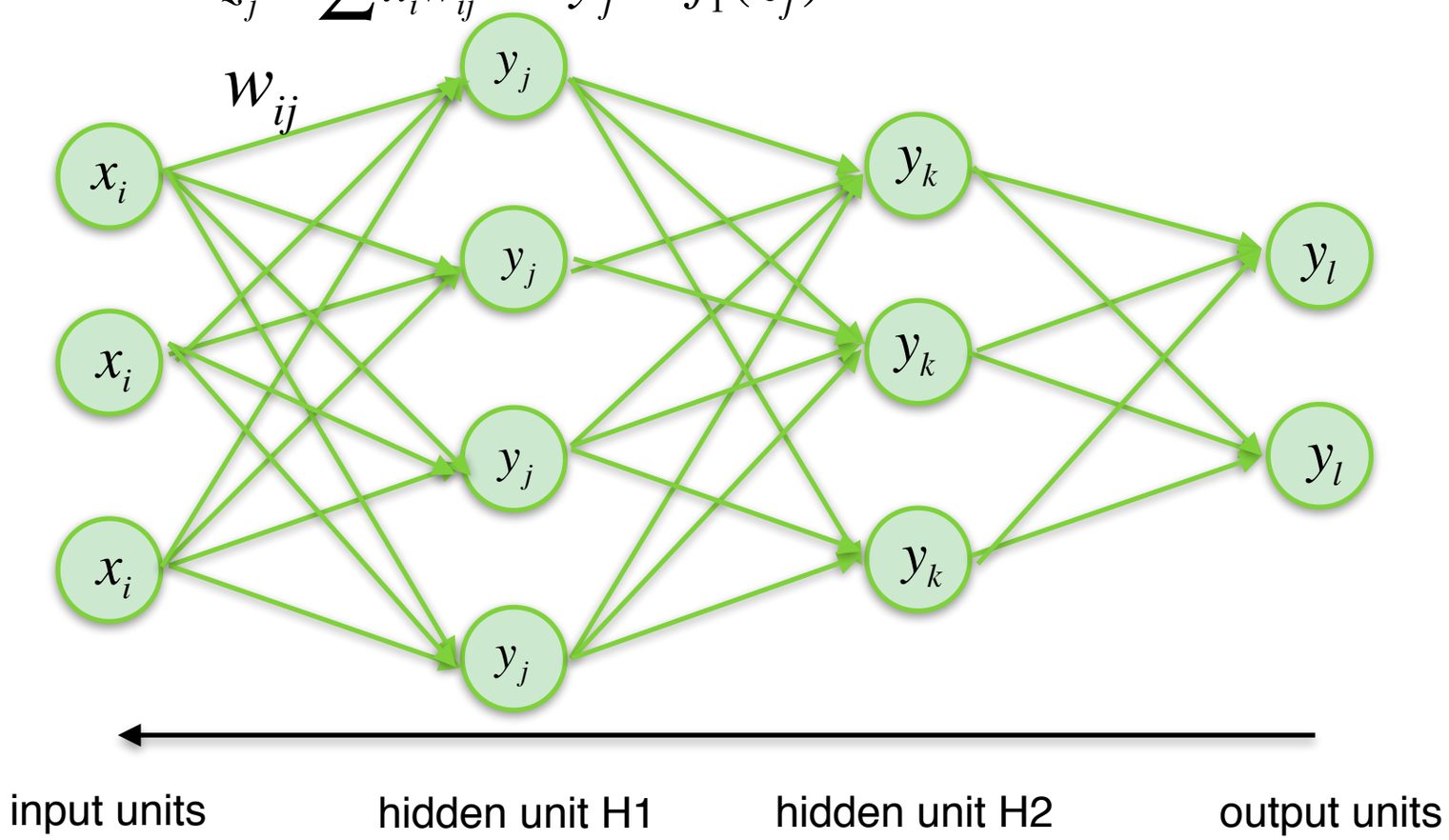

96.5
96.0
95.5
95.0
94.5
94.0
93.5
93.0
92.5
92.0
Test accuracy (percent)
3
4
5
6
7
8
9
10
11
Layers

浅层模型更容易过拟合


97
96
95
94
93
92
91
Test accuracy (percent)
0.0
0.2
0.4
0.6
0.8
1.0
Number of parameters
×10^8
3, convolutional
3, fully connected
11, convolutional

BP反向传播

$$z_j = \sum x_i w_{ij} \qquad y_j = f_1(z_j)$$

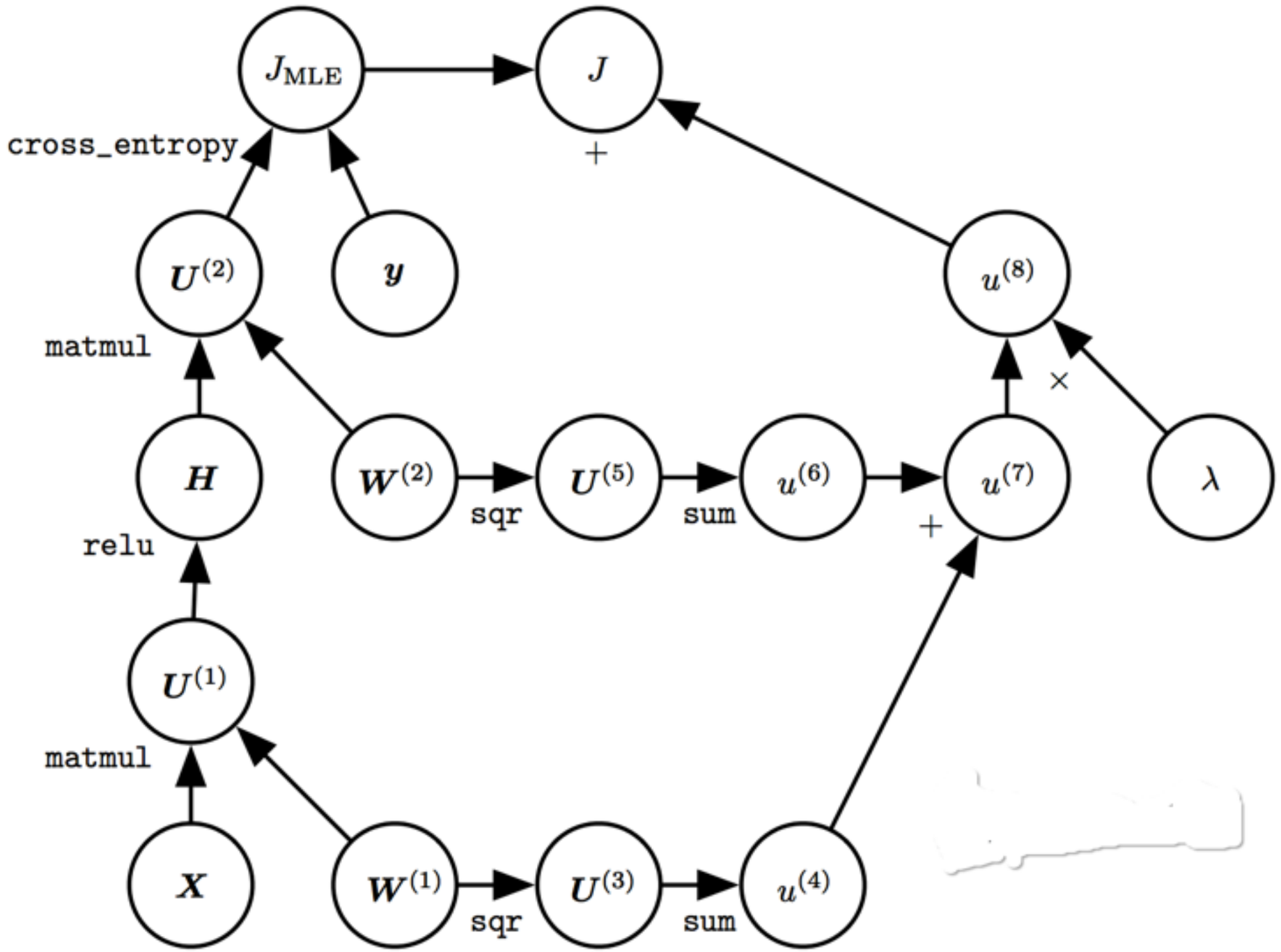


损失函数（假设）：$E = 0.5(y_l - t_0)^2$

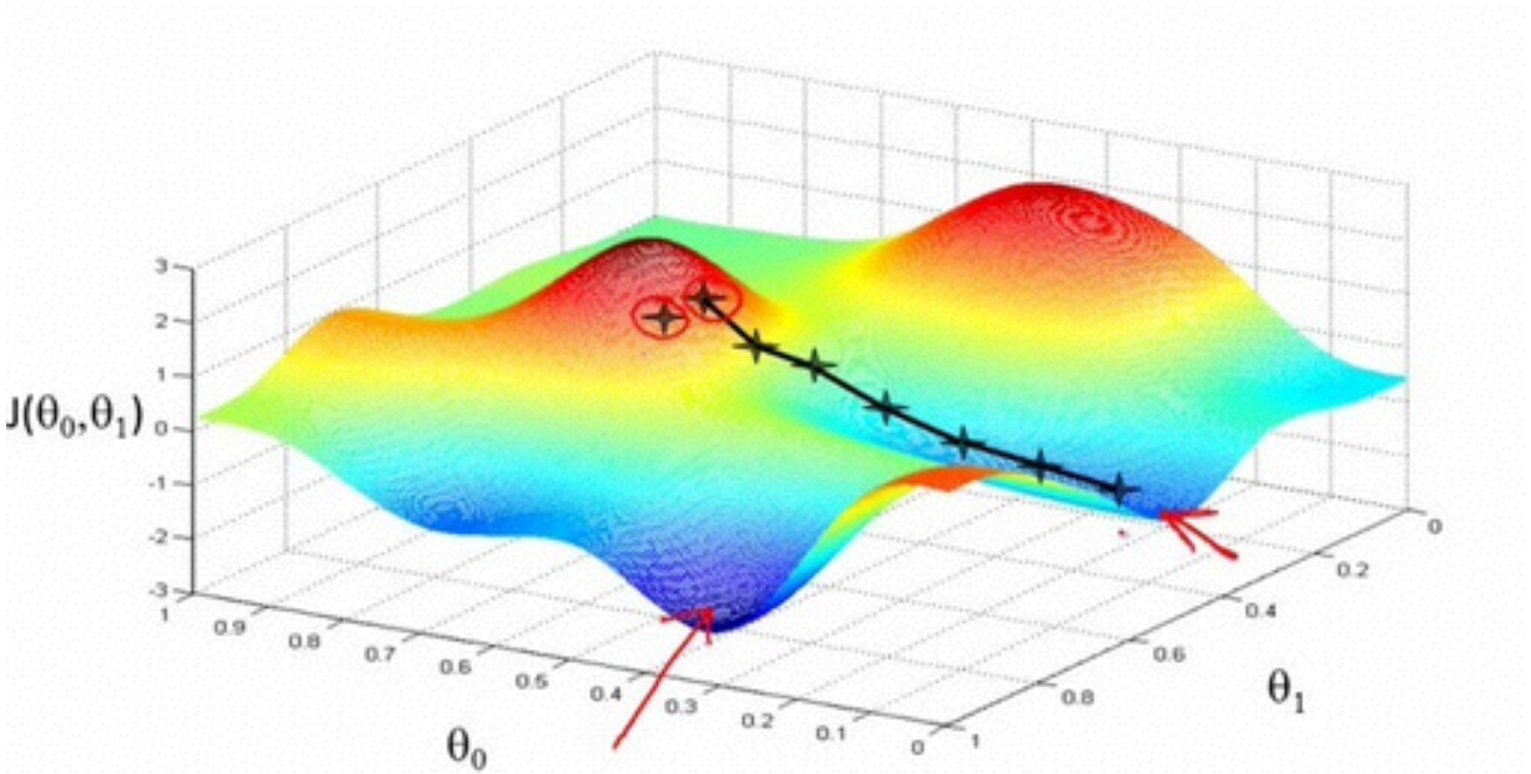
（凸函数）

链式求导：$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial z_j}\frac{\partial z_j}{\partial w_{ij}}$$

损失函数


J_{MLE}
J
cross_entropy
+
$U^{(2)}$
y
$u^{(8)}$
matmul
×
H
$W^{(2)}$
sqr
$U^{(5)}$
sum
$u^{(6)}$
$u^{(7)}$
λ
relu
+
$U^{(1)}$
matmul
X
$W^{(1)}$
sqr
$U^{(3)}$
sum
$u^{(4)}$

梯度下降



如上图所示，得到了局部最优解。x,y表示的是theta0和theta1，z方向表示的是花费函数，很明显出发点不同，最后到达的收敛点可能不一样。当然如果是碗状的，那么收敛点就应该是一样的。

- 梯度下降法的变形形式：
- 1.批量梯度下降法BGD
- 2.小批量梯度下降法MBGD
- 3.随机梯度下降法SGD

6. Hessian矩阵

二阶偏导

- Jacobian和Hessian矩阵是深度学习基于梯度优化算法中两个非常重要的概念, 分别对应一阶偏导和二阶偏导。

- Jacobian矩阵可以帮你找到所有的critical points, 而Hessian矩阵帮你定位发现的临界点(critical points)是什么类型的, 譬如局部最小值, 局部最大值还是鞍点(saddle point)。

- Hessian矩阵

性质1: 如果二阶偏导是连续的, 那么Hessian矩阵是对称的

- 这是个非常好的性质, 因为深度学习领域里面大部分的函数的二阶偏导都是连续的, 所以它们的Hessian矩阵都是对称的. 由于是实对称矩阵, 我们可以把它分解成一些实特征值集合以及一些互相垂直的实特征向量的集合.

- Hessian矩阵

特征值和特征向量

矩阵是一种向量到向量的映射, 矩阵A的特征值λ和特征向量$\vec{x}$具有性质$A\vec{x} = \lambda\vec{x}$, 也就是说矩阵A对特征向量$\vec{x}$的映射只是在$\vec{x}$方向上进行扩展, 不会改变把$\vec{x}$映射到其他方向.
实对称阵的不同特征值对应的特征向量是相互垂直的.

$$A\vec{x} = \lambda\vec{x}$$

$$(A - \lambda I)\vec{x} = \vec{0}$$

由于$\vec{x} \neq \vec{0}$存在, 那么$A - \lambda I$必然不可逆, 所以$|A - \lambda I| = 0$
如果A正定, 那么A所有的特征值都为正, 并且任意$\vec{x}$有$\vec{x}^T A\vec{x} > 0$
如果A半正定, 那么A所有的特征值都非负, 并且任意$\vec{x}$有$\vec{x}^T A\vec{x} \geq 0$

- Hessian矩阵

性质2: $f(\vec{x})$在方向单位向量$\vec{u}$上的导数为$\vec{u}^T \nabla_{\vec{x}} f(\vec{x})$, 在方向$\vec{d}$(单位向量)上的二阶导数为$\vec{d}^T H\vec{d}$.

- 如果$\vec{d}$ 是H的一个特征向量, 那么对应方向的二阶导数为相应的特征值; 如果是其他方向的, 那么二阶偏导为是所有特征值的加权(权值0到1)平均(可以理解为什么所有特征值为正就能断定这是个最小值, 因为所有方向上的二阶偏导都是正), 并且如果$\vec{d}$ 与某个特征值的夹角较小, 那么相应的权值会较大; 最大的特征值对应最大的方向导数, 最小的特征值决定最小的方向导数.

- Hessian矩阵

性质3: 二阶导数可以决定临界点是一个局部最小值, 局部最大值还是鞍点.

- 临界点的一阶导数为0, 当二阶导数大于0, 临界点为局部最小值; 当二阶导数小于0, 临界点为局部最大值; 当临界点为0时, 临界点可以为鞍点或者在一个平坦区域里.

- 推广到多维空间里, 如果临界点的Hessian矩阵是正定的, 那么它是个局部最小值; 如果临界点的Hessian矩阵是负定的, 那么它是个局部最大值; 当临界点的Hessian的特征值有正有负, 那么它可能是个鞍点.