深蓝学院
shenlanxueyuan.com

深度学习：从理论到实践

第一章：深度学习理论（一）

课程大纲

深蓝学院
shenlanxueyuan.com

- ✓ 概述
- ✓ 单层神经网络（单层感知器）
- ✓ 多层神经网络（多层感知器）

前馈神经网络

课程大纲

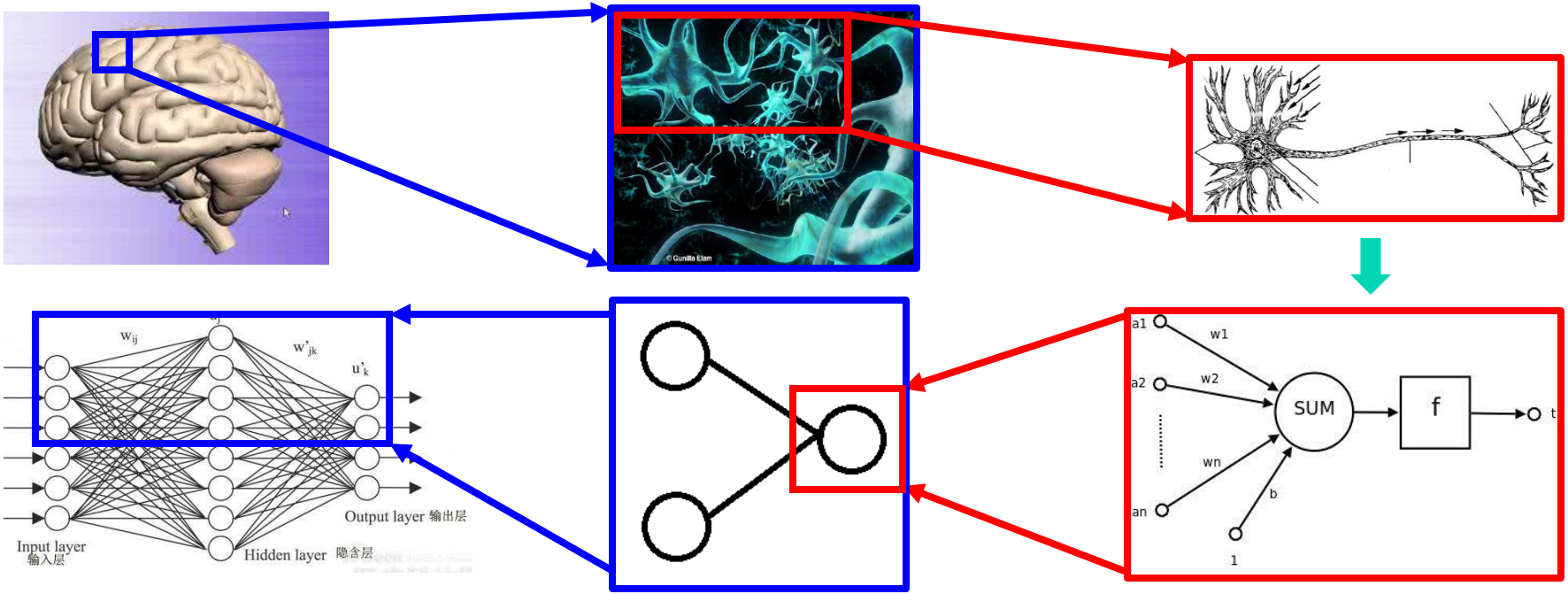
深蓝学院
shenlanxueyuan.com

- ✓ 概述
  - 定义
  - 发展历程
  - 发展现状
  - 人工神经网络基础
- ✓ 单层神经网络（单层感知器）
- ✓ 多层神经网络（多层感知器）

概述—定义

深蓝学院
shenlanxueyuan.com

人脑神经网络 vs 人工神经网络



概述—定义

深蓝学院
shenlanxueyuan.com

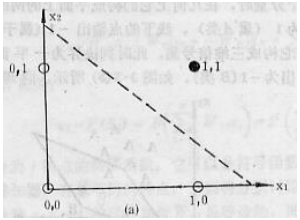
什么是人工神经网络

一种模仿生物神经网络的结构和功能的数学模型

美国神经网络学者Nielsen给出的定义

- 人工神经网络是一个并行、分布式处理结构，由处理单元及其称为连接的无向通道互连而成。
- 这些处理单元具有局部内存，可以完成局部操作，该操作由输入至该单元的信号值和存储在该单元中的信号值来确定。
- 每个处理单元有一个单一的输出连接，输出信号可以是任何需要的数学模型。

概述—发展历程



起源

1940s

MP模型
Hebb学习规则

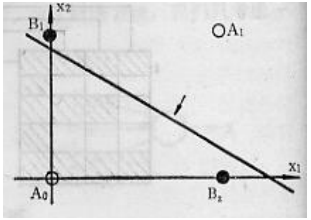
第一次浪潮

1960s

感知器模型

第一次低谷

单层感知器无法处理非线性问题



第二次浪潮

1980s

Hopfield网络
Boltzmann机
BP算法

第二次低谷

深层网络难以训练；
浅层方法取得成功：
SVM （1995），
Boosting （1999）

CNN：LeNet（1998）

第三次浪潮

2000s

DBN（Science 2006，Hinton）

更有效的训练方法
更强大的计算能力
更大量的训练数据

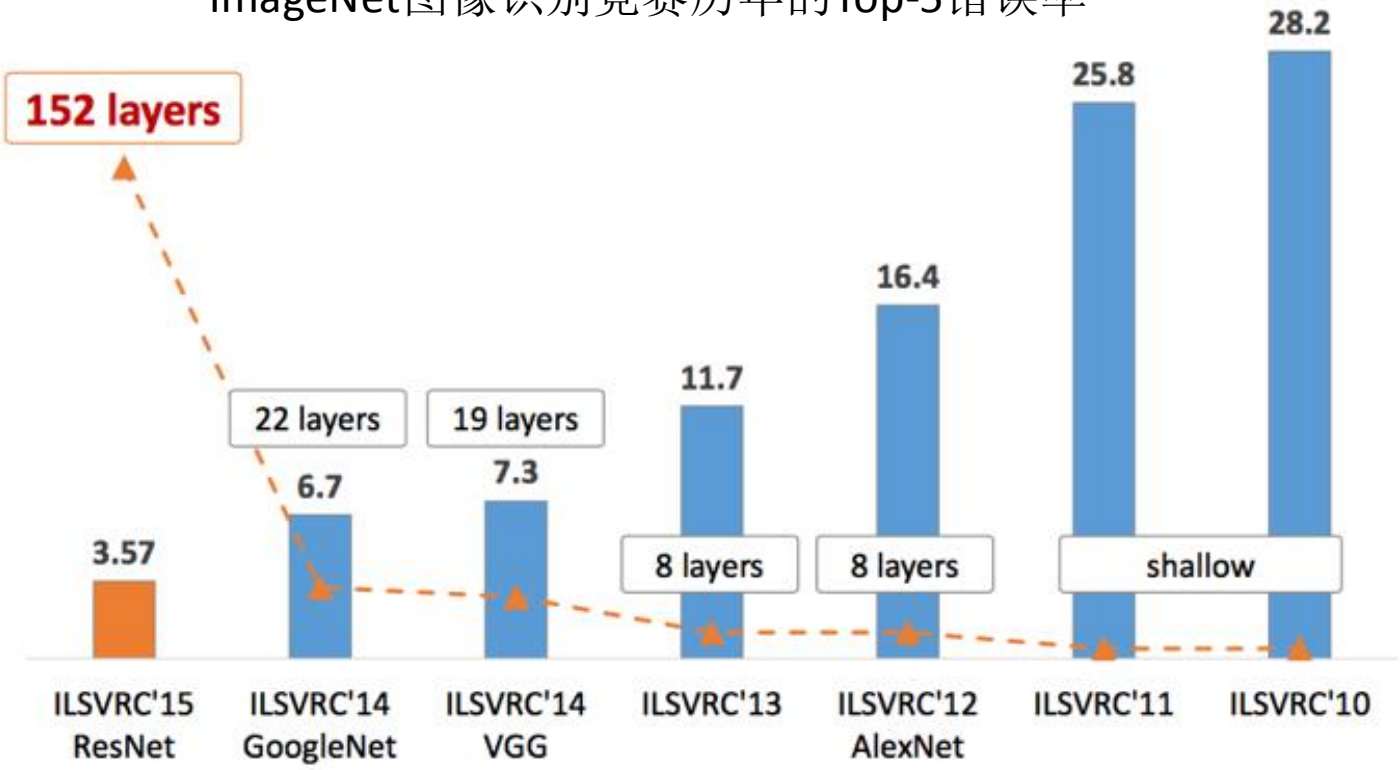
DBN，RBM，DBM
CNN
RNN，LSTM
GAN……

?

AlphaGo
=>
AlphaGo Zero
=>
AlphaZero
=>
……

概述—发展历程

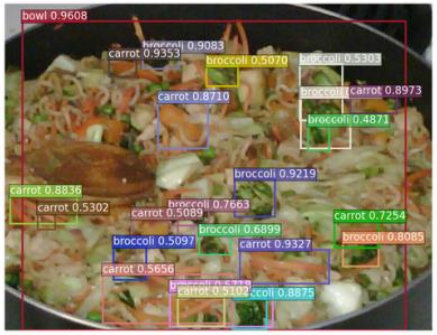
ImageNet图像识别竞赛历年的Top-5错误率

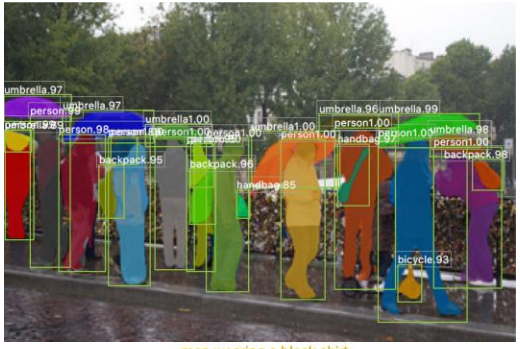


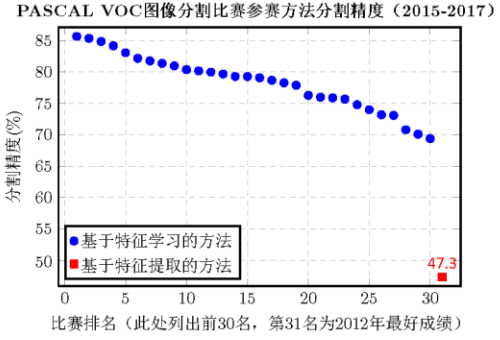
概述—发展现状

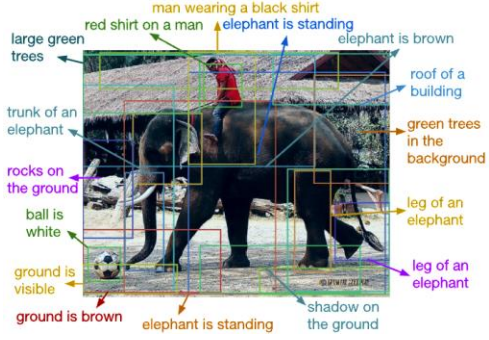
研究热度（仅列举计算机视觉相关方向）

- 图像分类
- 目标检测与识别
- 图像分割
- 图像描述
- 视频处理
- 图像生成









概述—发展现状

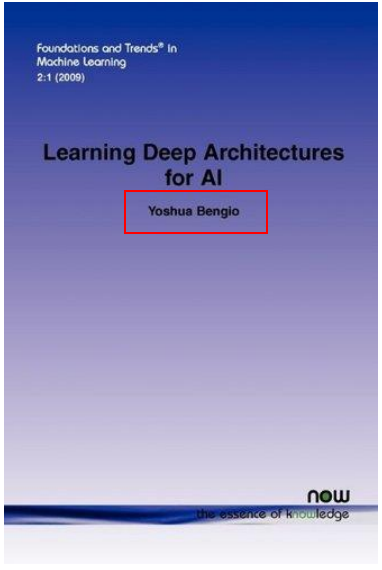
深蓝学院 shenlanxueyuan.com

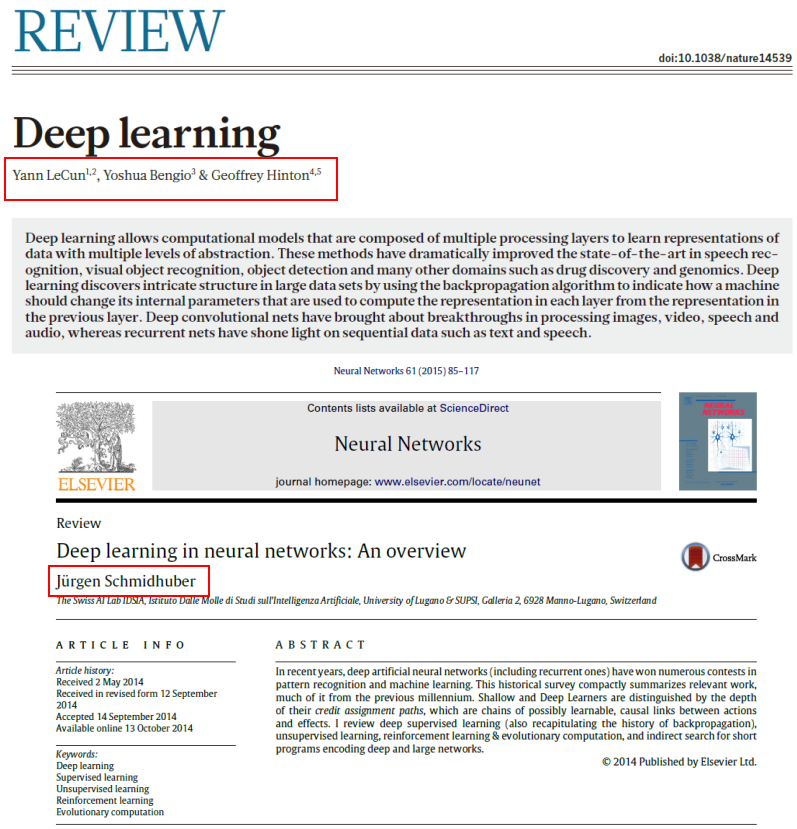
主要研究团队

代表人物	所在单位	代表成果
Geoffrey E. Hinton	University of Toronto	DBN
Yann LeCun	New York University	CNN
Yoshua Bengio	University of Montreal	《Learning Deep Architectures for AI》《Deep learning》
Michael Jordan	University of California, Berkeley	(Andrew Ng的导师)
Jürgen Schmidhuber	Swiss AI Lab IDSIA	LSTM
Ian Goodfellow	Google	GAN,《Deep learning》
Fei-Fei Li	Stanford University	ImageNet
Kaiming He	Facebook AI Research	ResNet, Mask R-CNN, Focal loss

概述—发展现状

由作者列表决定了的必读论著：





REVIEW

doi:10.1038/nature14539

Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

Neural Networks 61 (2015) 85–117

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Review

Deep learning in neural networks: An overview

Jürgen Schmidhuber

The Swiss AI Lab IDSIA, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, University of Lugano & SUPSI, Galleria 2, 6928 Manno-Lugano, Switzerland

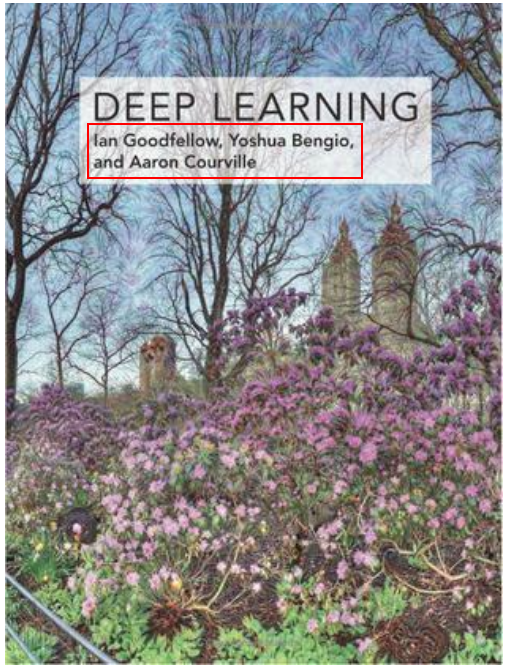
ARTICLE INFO

Article history:
Received 2 May 2014
Received in revised form 12 September 2014
Accepted 14 September 2014
Available online 13 October 2014

Keywords:
Deep learning
Supervised learning
Unsupervised learning
Reinforcement learning
Evolutionary computation

ABSTRACT

In recent years, deep artificial neural networks (including recurrent ones) have won numerous contests in pattern recognition and machine learning. This historical survey compactly summarizes relevant work, much of it from the previous millennium. Shallow and Deep Learners are distinguished by the depth of their *credit assignment paths*, which are chains of possibly learnable, causal links between actions and effects. I review deep supervised learning (also recapitulating the history of backpropagation), unsupervised learning, reinforcement learning & evolutionary computation, and indirect search for short programs encoding deep and large networks.



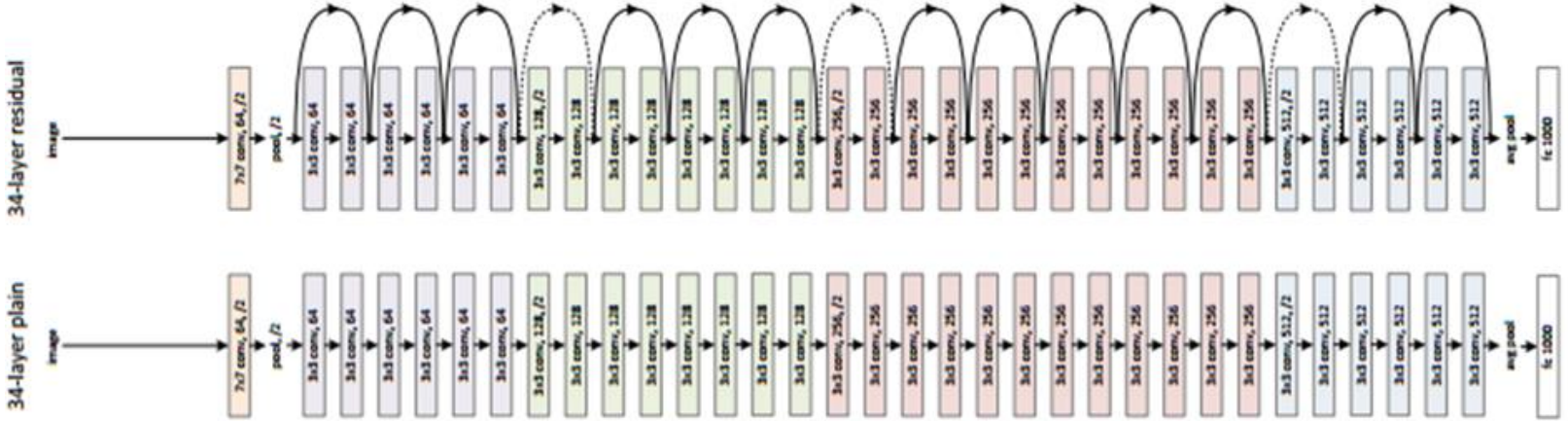
https://github.com/exacity/deeplearningbook-chinese

概述—发展现状

深蓝学院
shenlanxueyuan.com

早期模型（2012–2015）

AlexNet, VGG-Net, ZF-Net, GoogLeNet, FCN, Deeplab, DeconvNet, CRFRNN, SPP-Net, R-CNN, Fast R-CNN, Faster R-CNN, ResNet…



残差网络ResNet

概述—发展现状

深蓝学院
shenlanxueyuan.com

最新模型（2016-2017）

ENet, MobileNet, ShuffleNet, PSPNet, RefineNet, DenseNet (CVPR 2017 best paper), …

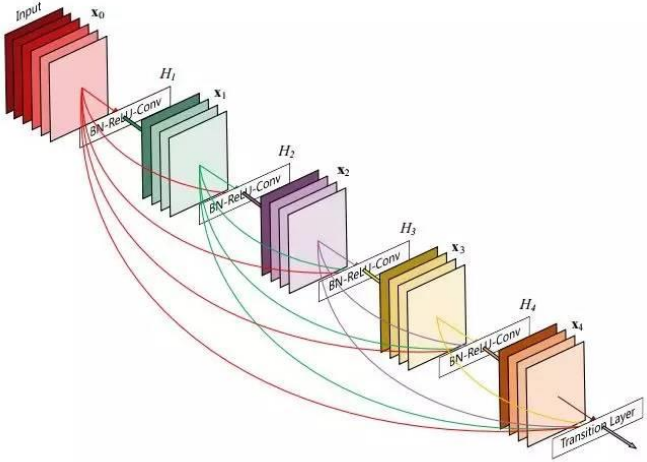
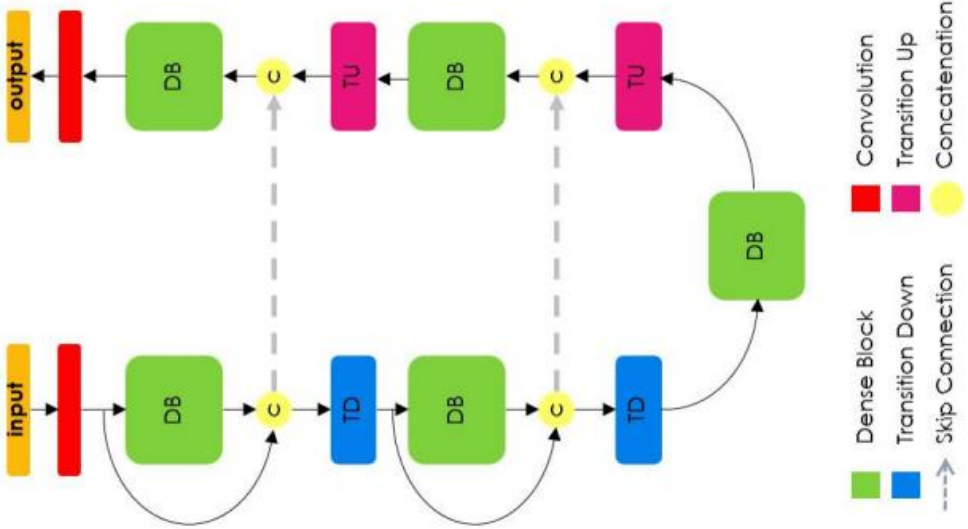


Figure 1. A 5-layer dense block with a growth rate of $k = 4$. Each layer takes all preceding feature-maps as input.



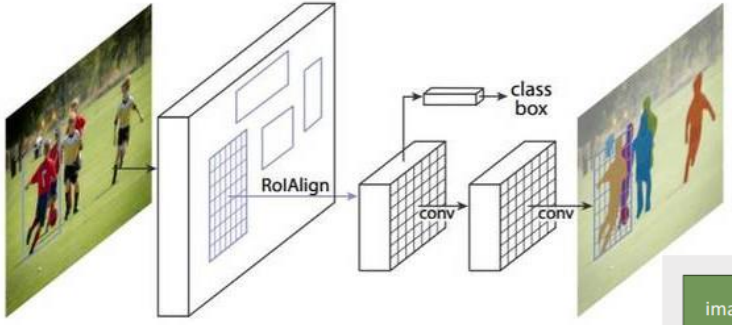
DenseNet and One-Hundred-Layer-Tiramisu

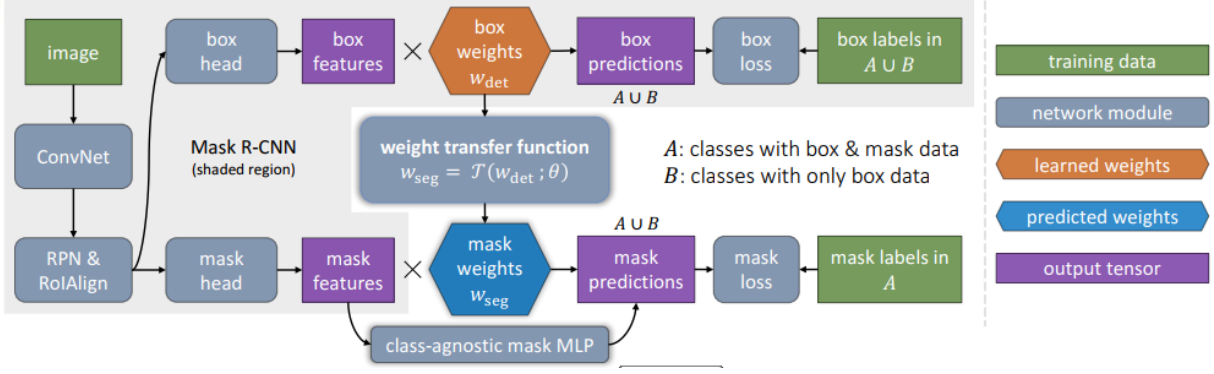
概述—发展现状

深蓝学院 shenlanxueyuan.com

最新模型（2017）

Mask R-CNN (ICCV 2017 best paper), Mask^X R-CNN (learning to segment everything)…



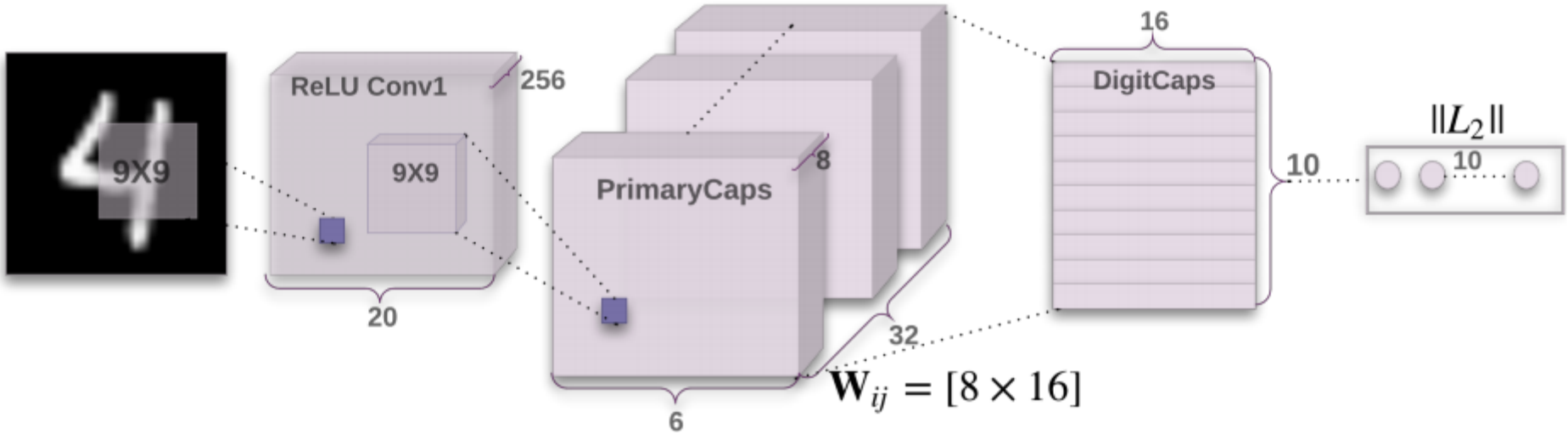


概述—发展现状

深蓝学院 shenlanxueyuan.com

最新模型（2017）

CapsuleNet

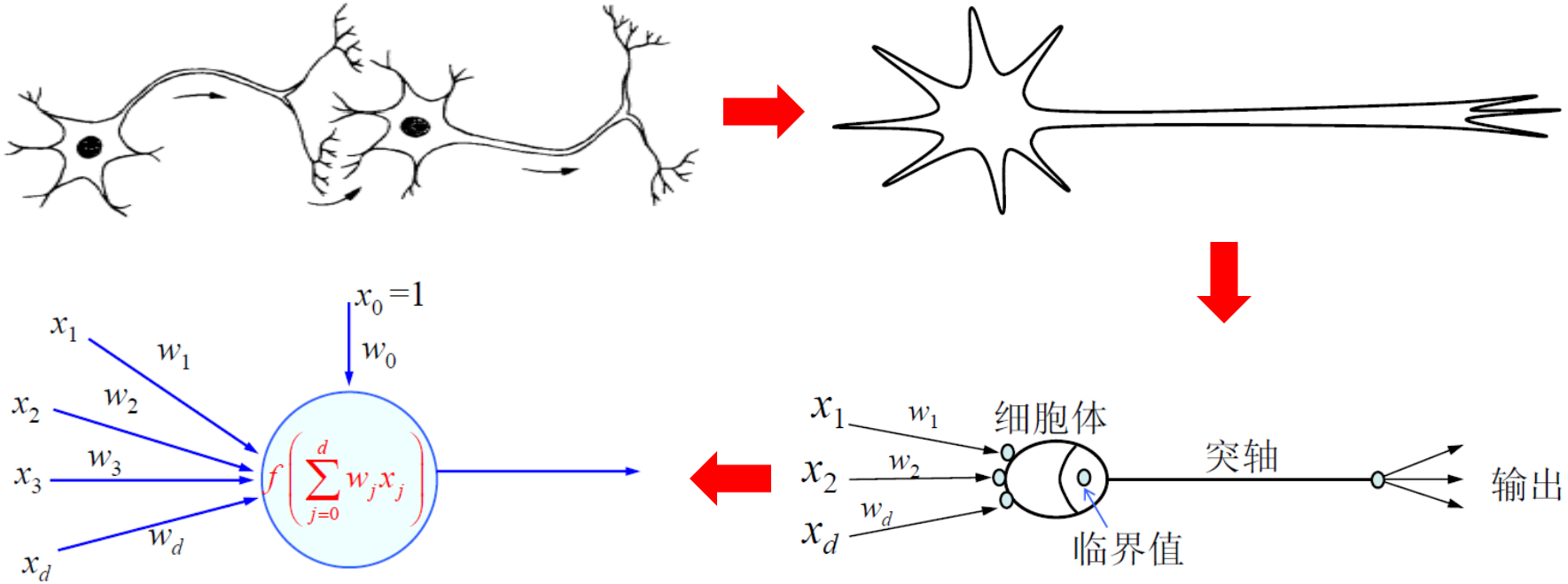


S. Sabour et al., Dynamic Routing Between Capsules. NIPS 2017.

概述—人工神经网络基础

深蓝学院 shenlanxueyuan.com

从神经元到人工神经元



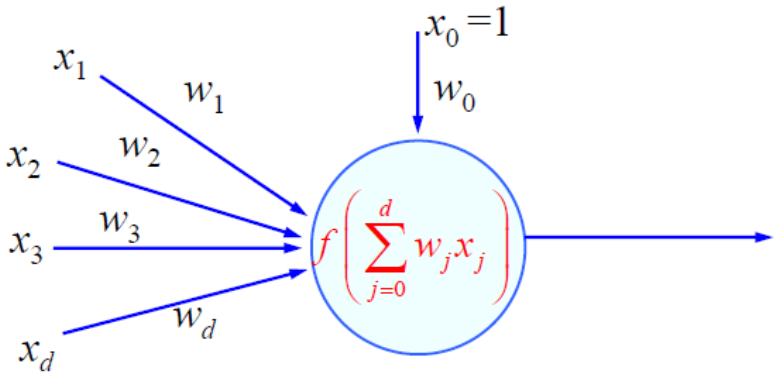
概述—人工神经网络基础

深蓝学院 shenlanxueyuan.com

人工神经元的基本结构和功能

- 对每个输入信号进行处理以确定其强度（**加权**）；
- 确定所有输入信号的组合效果（**求和**）；
- 确定其输出（**激励**）。

加权：w_*
求和：Σ
激励：$f(\cdot)$

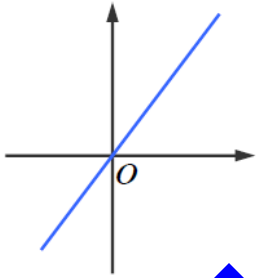


概述—人工神经网络基础

深蓝学院 shenlanxueyuan.com

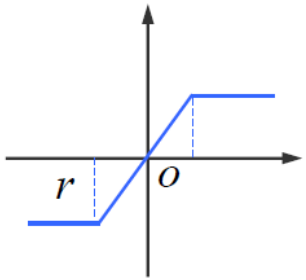
激励函数

- 将可能的无限域变换到指定的**有限范围内**进行输出
- 类似于生物神经元的**非线性**转移特性
- 常用的激励函数：

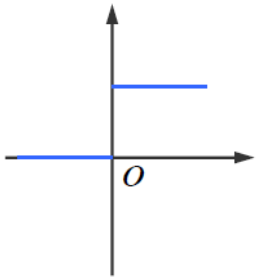


线性函数

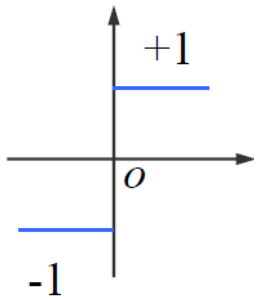
非线性函数



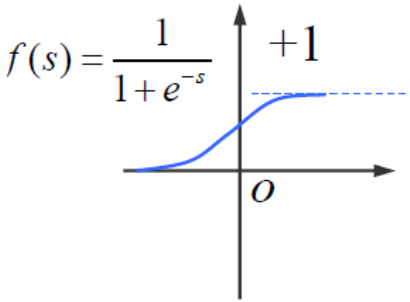
斜坡函数



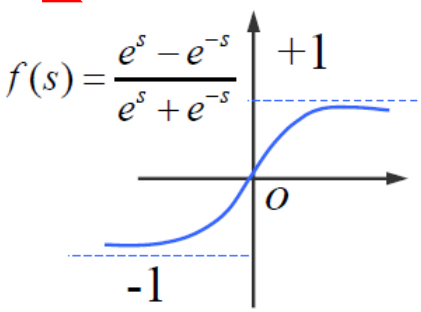
阶跃函数



符号函数



Sigmoid函数



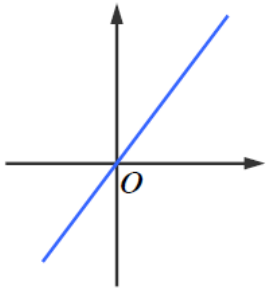
双曲正切函数

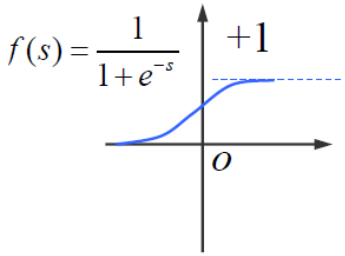
概述—人工神经网络基础

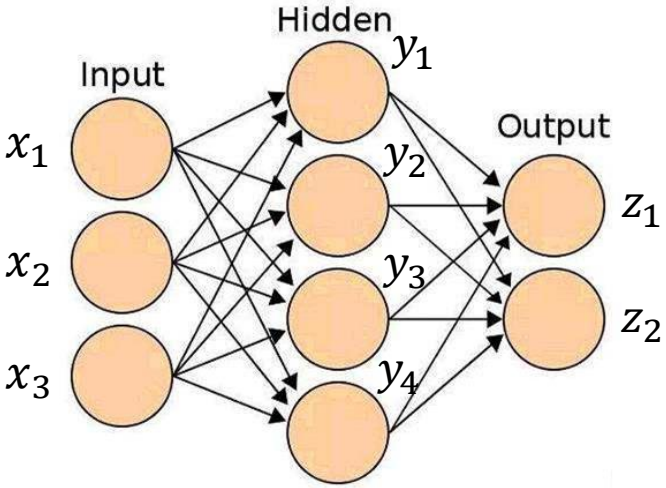
深蓝学院 shenlanxueyuan.com

激励函数——思考

- 为什么通常需要**非线性**的激励函数？







$$y_j = \sum_{i=1}^{m} w_{ij} x_i = W^1 \mathbf{x}$$

$$z_k = \sum_{j=1}^{n} w_{jk} y_j = W^2 \mathbf{y} = W^2 W^1 \mathbf{x} = W^{total} \mathbf{x}$$

两层线性网络等效于单层网络，只是后者的加权矩阵为两个加权矩阵的乘积。

概述—人工神经网络基础

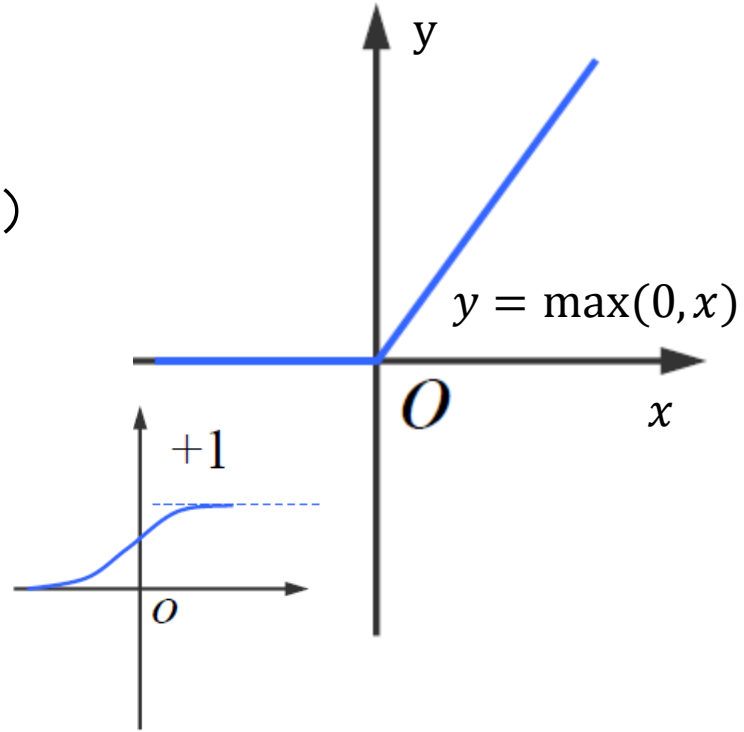
深蓝学院 shenlanxueyuan.com

激励函数——更常用的激励函数

- 修正线性单元Rectified Linear Units（ReLU）
- 分段线性，容易计算和优化
- 于2012年首次提出，在此之前Sigmoid

和双曲正切两种函数最为常用

- 目前ReLU在深度学习中得到广泛应用



Alex Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks. NIPS 2012.

概述—人工神经网络基础

深蓝学院 shenlanxueyuan.com

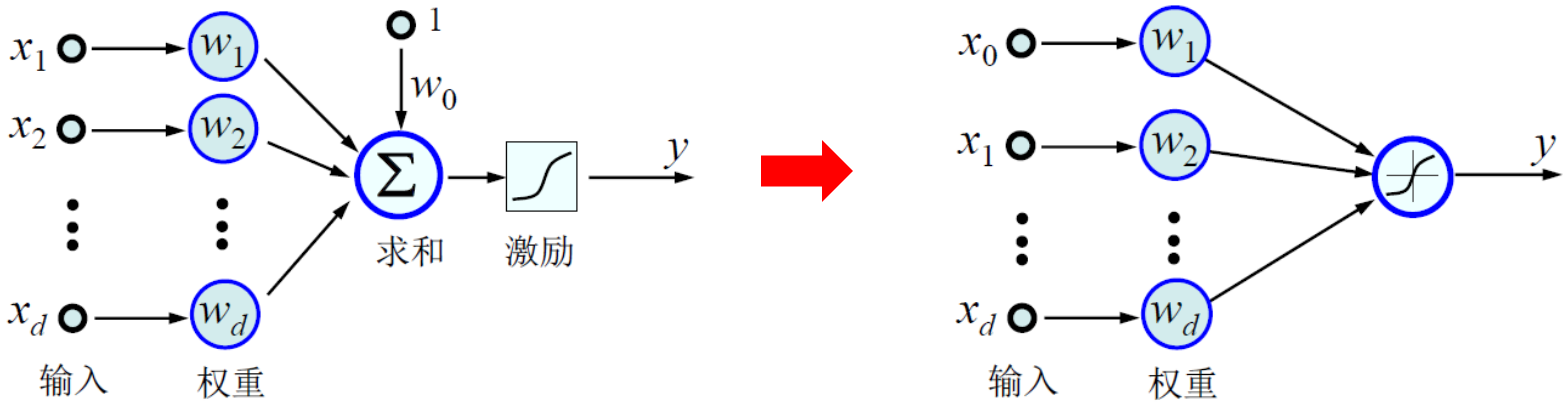
激励函数——更常用的激励函数ReLU

- ReLU本质上是**分段线性**模型，前向计算非常简单，无需指数之类操作；
- ReLU的**偏导也很简单**，反向传播梯度，无需指数或者除法之类操作；
- ReLU**不容易发生梯度消失和发散问题**，Sigmoid和双曲正切函数在两端的时候导数容易趋近于零，多级连乘后梯度更加约等于0；
- ReLU关闭了左边，从而会使得很多的隐层输出为0，即**网络变得稀疏**，起到了类似L1的正则化作用，可以在一定程度上缓解过拟合。

概述—人工神经网络基础

深蓝学院 shenlanxueyuan.com

拓扑结构-单层网络



$$sum = \sum_{i=1}^{d} w_i x_i + w_0$$

$$y = f\left(sum\right)$$

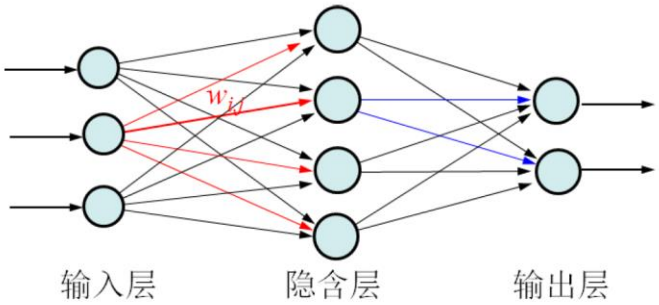
$$sum = \sum_{i=0}^{d} w_i x_i, \quad x_0 = 1$$

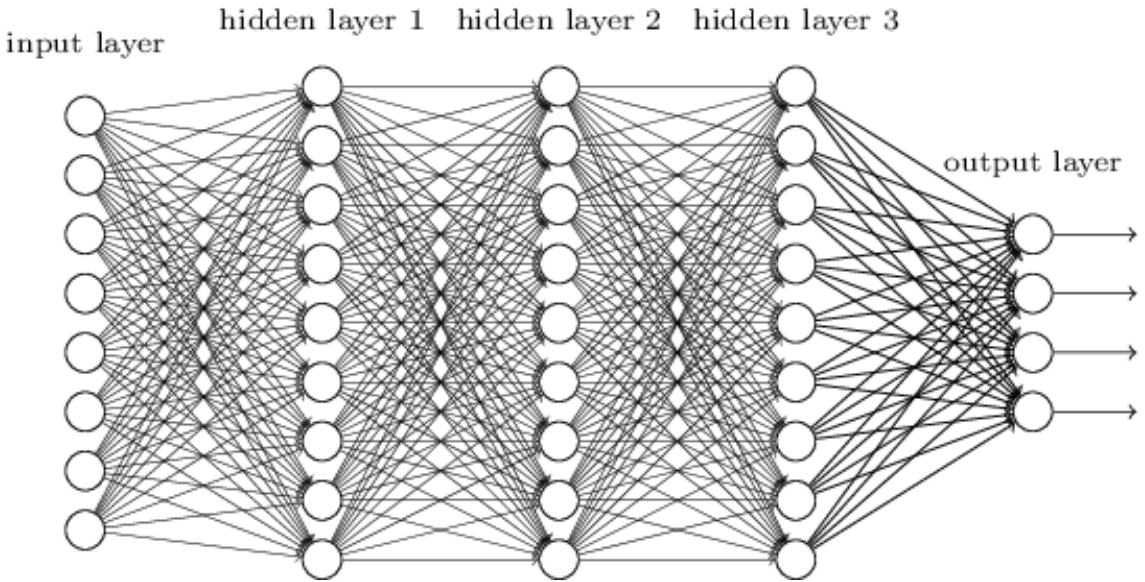
$$y = f\left(sum\right)$$

概述—人工神经网络基础

深蓝学院 shenlanxueyuan.com

拓扑结构-多层网络





注意：

理论上可以有无限多层

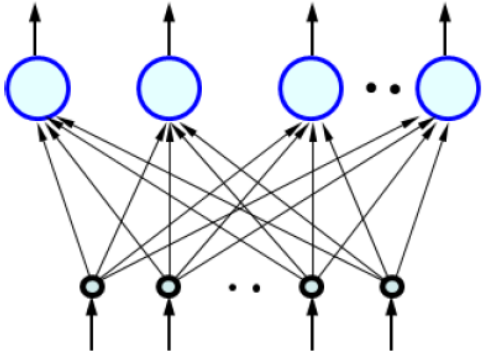
理论上单隐层即可实现任意复杂的非线性函数，但是需要的节点数目可能是指数级的

概述—人工神经网络基础

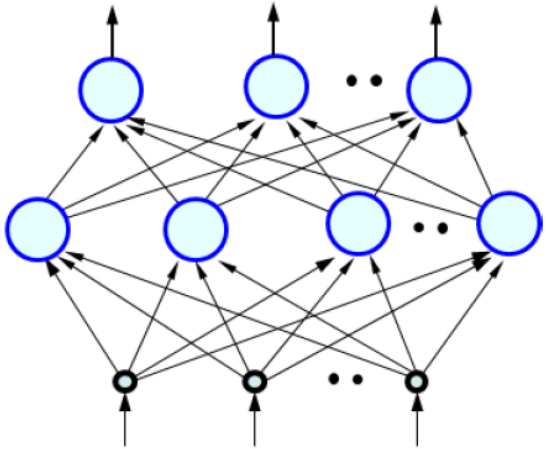
深蓝学院
shenlanxueyuan.com

拓扑结构-前馈网络

没有层内连接，也没有隔层的前馈联接，
各节点前馈连接到下一层节点



单层感知器



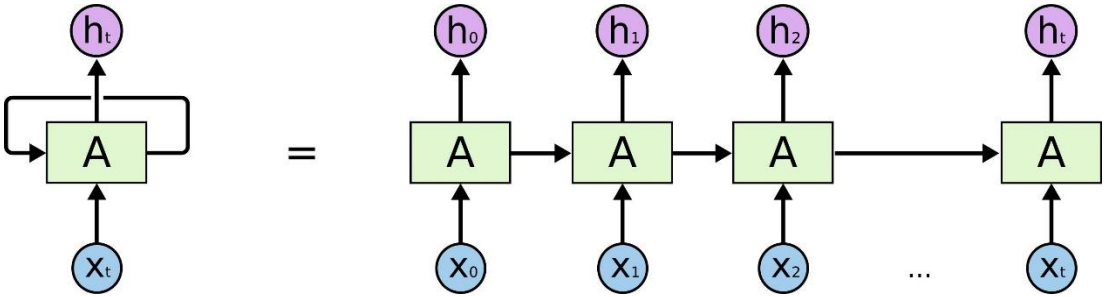
多层感知器

概述—人工神经网络基础

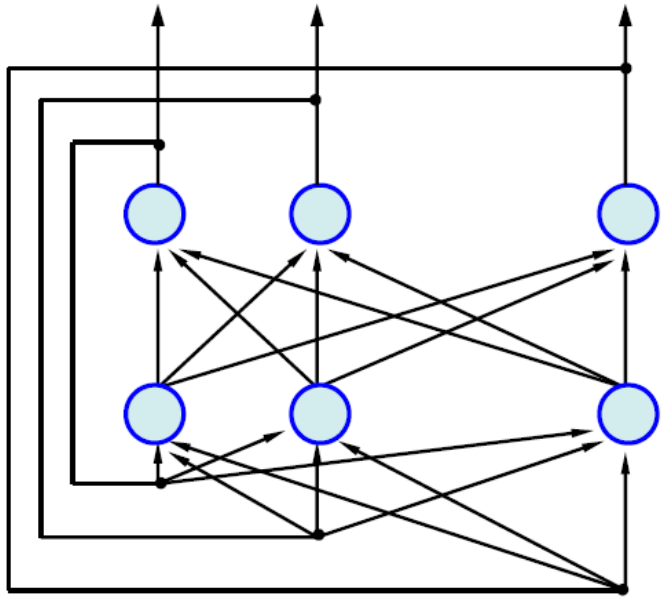
深蓝学院 shenlanxueyuan.com

拓扑结构-反馈网络

网络有回路，信息可以沿回路流动
节点的输出既依赖于当前输入，
也依赖于自己之前的输出



http://colah.github.io/

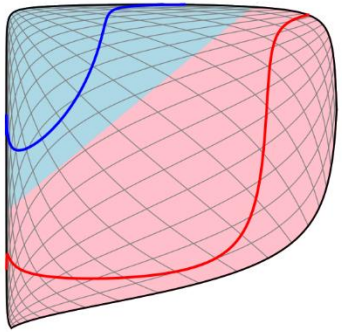


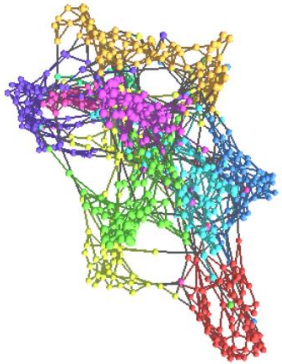
概述—人工神经网络基础

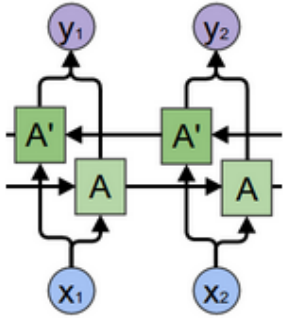
深蓝学院 shenlanxueyuan.com

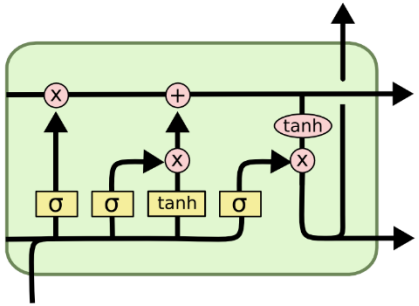
http://colah.github.io/

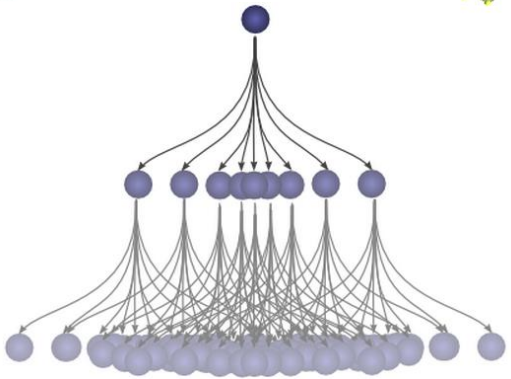
Tips：colah's blog

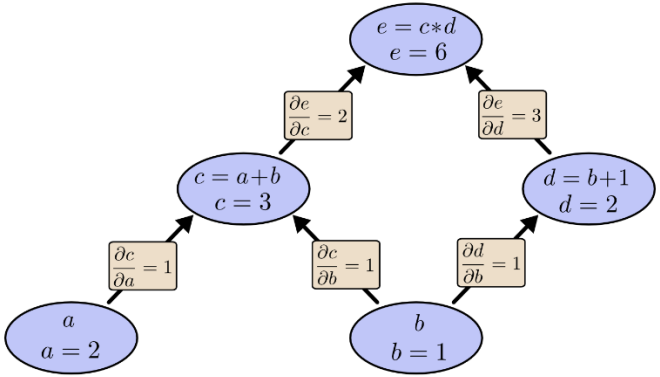












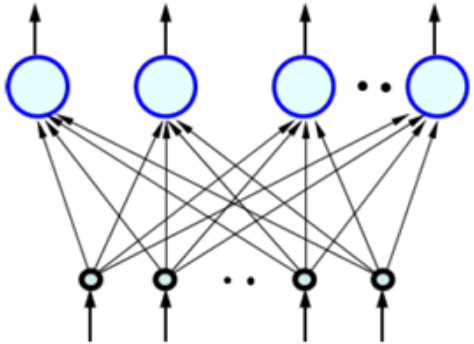
课程大纲

深蓝学院
shenlanxueyuan.com

- ✓ 概述
- ✓ 单层神经网络（单层感知器）
  - 网络描述
  - 模型训练
  - 存在问题
- ✓ 多层神经网络（多层感知器）

单层神经网络—网络描述

网络结构

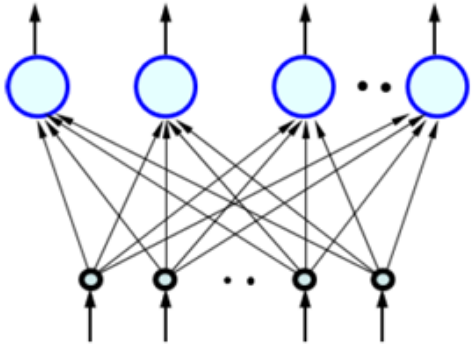


- 输入层：d个输入节点，1个偏置节点
- 输入向量：$x = [1, x_1, x_2, \ldots, x_d]^T$
- 输出层：c个输出节点
- 输出向量：$z = [z_1, z_2, \ldots, z_c]^T$
- 权重集合：$\{w_{ij}\}$
- 没有隐含层

单层神经网络—网络描述

深蓝学院 shenlanxueyuan.com

数学描述



- 对于输出层的节点j，其输入加权和为

$$sum_j = \sum_{i=0}^{d} w_{ij} x_i = \mathbf{w}_j^T \mathbf{x}, \quad x_0 = 1$$

$$\mathbf{w}_j = [w_{0j}, w_{1j}, w_{2j}, \mathrm{K}, w_{dj}]^T$$

- 经过激励函数后的输出值为

$$z_j = f(sum_j) = f(\sum_{i=0}^{d} w_{ij} x_i) = f(\mathbf{w}_j^T \mathbf{x})$$

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

学习任务

- 给定训练样本$\{\mathbf{x}^k\}_{k=1}^n$及相应的目标值$\{\mathbf{t}^k\}_{k=1}^n$，

  确定网络中的所有连接权重$\{w_{ij}\}$
- 当网络完成训练，期望实现 $\mathbf{z}^k = \mathbf{t}^k, k = 1, 2, \mathrm{K}, n$
- 在分类问题中：
  - 两类：输出层只有一个节点，目标值为+1/−1
  - 多类：输出层的节点数目与类别数相同，样本$\mathbf{x}_k$属于第q类时目标值$\mathbf{t}_k$中只有第q个元素非零。例如，ImageNet竞赛的1000类图像识别。

$$\downarrow q$$
$$[0, 0, \mathrm{K}, 0, 1, 0, \mathrm{K}, 0]$$

IMAGENET

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

线性单元定义：激励函数为线性函数，且处处可微

$$\left.\begin{aligned} z_j &= f\left(\sum_{i=0}^{d} w_{ij} x_i\right) = \sum_{i=0}^{d} w_{ij} x_i \\ \mathbf{z}_k &= \mathbf{t}_k, k = 1, 2, \mathrm{K}, n \end{aligned}\right\}$$

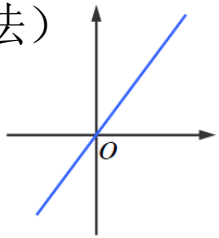
直接求解权重

$$\begin{cases} \sum_{i=0}^{d} w_{ij} x_i^1 = t_j^1 \\ \sum_{i=0}^{d} w_{ij} x_i^2 = t_j^2 \\ \mathrm{K} \\ \sum_{i=0}^{d} w_{ij} x_i^n = t_j^n \end{cases}$$

$j = 1, 2, \mathrm{K}, c$

共计c个线性方程组

（最小二乘法）



但是一旦出现非线性函数……

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

线性单元训练：δ规则（梯度下降法）

在搞数学的人看来，研究深度学习就是研究导数，而且还是一阶导.....

损失函数：

$$E(W)=\frac{1}{2}\sum_{k,j}(t_j^k-z_j^k)^2=\frac{1}{2}\sum_{k,j}(t_j^k-\sum_i w_{ij}x_i^k)^2$$

梯度：

$$\frac{\partial E}{\partial w_{ij}}=-\sum_k(t_j^k-z_j^k)x_i^k$$

二次函数求导

权重修正量：

$$\Delta w_{ij}=-\eta\frac{\partial E}{\partial w_{ij}}=\eta\sum_k\delta_j^k x_i^k,\qquad \delta_j^k=t_j^k-z_j^k$$

梯度下降法：$w^{t+1}=w^t-\eta\frac{\partial E}{\partial w}=w^t+\Delta w$ 直至 $|w^{t+1}-w^t|<\varepsilon$

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

非线性单元定义：激励函数为非线性函数，且处处可微

非线性单元训练：δ规则（梯度下降法）

损失函数：$$E(W)=\frac{1}{2}\sum_{k,j}(t_j^k-z_j^k)^2=\frac{1}{2}\sum_{k,j}(t_j^k-f(sum_j^k))^2,\qquad sum_j^k=\sum_{i=0}^{d}w_{ij}x_i^k$$

链式法则

梯度：$$\frac{\partial E}{\partial w_{ij}}=\sum_k\frac{\partial E}{\partial sum_j^k}\frac{\partial sum_j^k}{\partial w_{ij}}=-\sum_k(t_j^k-z_j^k)f'(sum_j^k)x_i^k$$

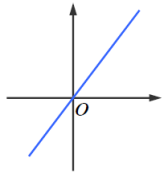
权重修正量：$$\Delta w_{ij}=-\eta\frac{\partial E}{\partial w_{ij}}=\eta\sum_k\delta_j^k x_i^k,\qquad \delta_j^k=f'(sum_j^k)(t_j^k-z_j^k)$$

请尝试自己动笔推导

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

对比：线性&非线性单元各自的权重修正量



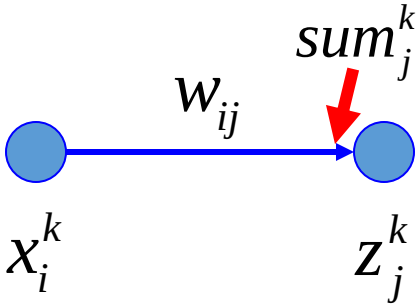
线性单元：$\Delta w_{ij} = \eta \sum_k \delta_j^k x_i^k,\ \delta_j^k = (t_j^k - z_j^k) = f'(sum_j^k)(t_j^k - z_j^k),\ f'(sum_j^k) = 1$

非线性单元：$\Delta w_{ij} = \eta \sum_k \delta_j^k x_i^k,\ \delta_j^k = f'(sum_j^k)(t_j^k - z_j^k)$

输入：x_i^k

误差：δ_i^k（有些文献中也称之为“敏感度”，即sensitivity）

权重修正量 -> 误差×输入



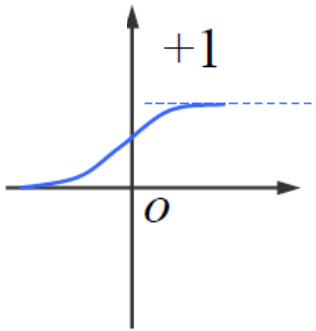
计算输入加权和：$sum_j^k = \sum_{i=0}^{d} w_{ij} x_i^k$

计算权重修正量：$\Delta w_{ij} = \eta \sum_k \delta_j^k x_i^k$

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

非线性激励函数的导数计算



- Sigmoid函数

$$f(s)=\frac{1}{1+e^{-s}}$$

$$f'(s)=\frac{e^{-s}}{(1+e^{-s})^2}=\frac{1}{1+e^{-s}}(1-\frac{1}{1+e^{-s}})=f(s)(1-f(s))$$

$$\delta_j^k=f'(y_j^k)(t_j^k-z_j^k)=z_j^k(1-z_j^k)(t_j^k-z_j^k),\quad z_j^k=f(y_j^k)$$

请尝试自己动笔推导

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

随机梯度下降（单样本）更新算法

Stochastic Updating

1 begin initialize: **w**, η, criterion θ, k=0
2 do $k \leftarrow k+1 \pmod{n}$
3 $\mathbf{x}^k$, randomly chose a sample (pattern)
4 $w_{ij} \leftarrow w_{ij} + \eta \delta_j^k x_i^k$
5 until $\| \nabla J(\mathbf{w}) \| < \theta$
6 return **w**
7 end

注意此处的正负号！

$$\frac{\partial E}{\partial w_{ij}} = -\sum_k (t_j^k - z_j^k) x_i^k$$

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta \sum_k \delta_j^k x_i^k$$

梯度“下降”，即沿着负梯度的方向移动

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

梯度下降（批量）更新算法

Batch Upadating

1 begin initialize: **w**, η, criterion θ, r=0
2 do $r \leftarrow r+1$ (increment epoch)
3 $k=0$, $\Delta w_{ij} \neq 0$
4 do $k \leftarrow k+1 \pmod n$
5 $\mathbf{x}^k$, select a sample (pattern)
6 $\Delta w_{ij} \leftarrow \Delta w_{ij} + \eta \delta_j^k x_i^k$
7 until $k=n$
8 $w_{ij} \leftarrow w_{ij} + \Delta w_{ij}$ // 所有样本完成之后再更新
9 until $\| \nabla J(\mathbf{w}) \| < \theta$
10 return **w**
11 end

思考：

两种方法的适用场合及各自的优缺点？

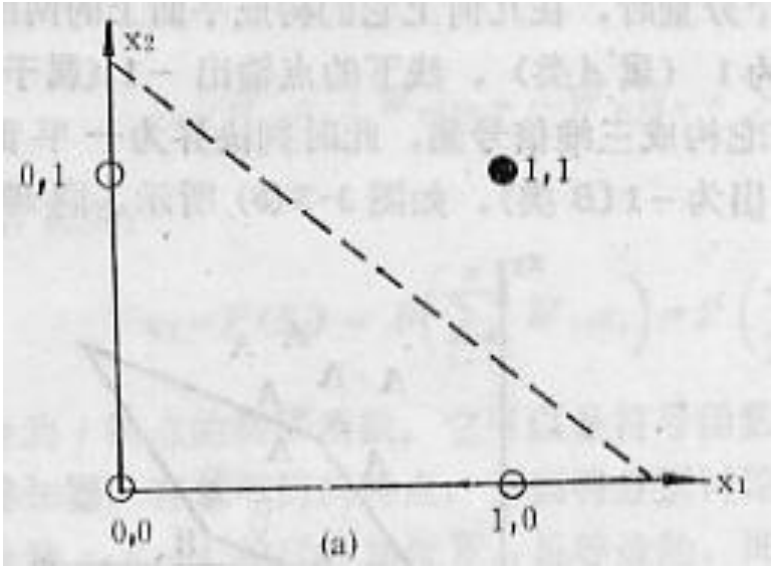
单层神经网络—存在问题

深蓝学院 shenlanxueyuan.com

单层感知器无法解决线性不可分的分类问题

逻辑“与”的真值表

k	x_1	x_2	T
1	0	0	-1(0)
2	0	1	-1(0)
3	1	0	-1(0)
4	1	1	1



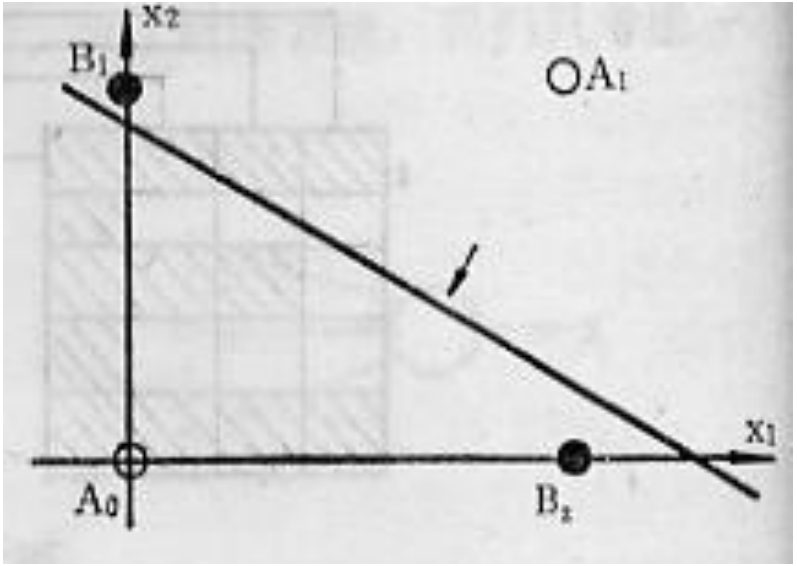
单层神经网络—存在问题

深蓝学院 shenlanxueyuan.com

单层感知器无法解决线性不可分的分类问题

逻辑“异或”的真值表

k	x_1	x_2	T
1	0	0	-1(0)
2	0	1	1
3	1	0	1
4	1	1	-1(0)



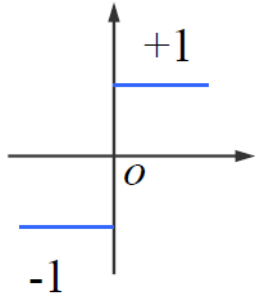
单层神经网络—存在问题

深蓝学院 shenlanxueyuan.com

单层感知器无法解决线性不可分的分类问题

- 激励函数为符号函数时的训练

$$z_j = f(\sum_{i=0}^{d} w_{ij} x_i), \qquad f(s) = \begin{cases} 1 & s \geq 0 \\ -1 & s < 0 \end{cases}$$



单个样本的贡献：

$$\Delta w_{ij} = \begin{cases} 2\eta t_j^k x_i^k & if \ z_j^k \neq t_j^k \\ 0 & if \ z_j^k = t_j^k \end{cases}$$

该公式如何得到？

$$z_j^k \neq t_j^k \Rightarrow 1 - z_j^k t_j^k = 2$$

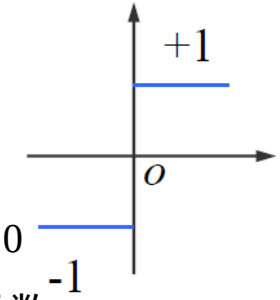
$$z_j^k = t_j^k \Rightarrow 1 - z_j^k t_j^k = 0$$

$$\Rightarrow \Delta w_{ij} = \eta(1 - z_j^k t_j^k) t_j^k x_i^k = \eta(t_j^k - z_j^k t_j^k t_j^k) x_i^k$$

$$= \eta(t_j^k - z_j^k) x_i^k = \eta \delta_j^k x_i^k$$

单层神经网络—存在问题

单层感知器无法解决线性不可分的分类问题



$$f(s)=\begin{cases}1 & s\geq 0\\ -1 & s<0\end{cases}$$

- 补充：前页公式的推导

已知 $\Delta w_{ij}=\eta\sum_k \delta_j^k x_i^k$，其中 $\delta_j^k=t_j^k-z_j^k$

（因为函数$f(s)$不是连续函数，因此δ_j^k直接定义为期望输出与实际输出之差，即不显式定义$f'(s)$）

如果 $z_j^k=t_j^k$ $\quad\Delta w_{ij}=0$

如果 $z_j^k\neq t_j^k$ $\quad\Delta w_{ij}=\eta(t_j^k-(t_j^k))x_i^k=2\eta t_j^k x_i^k$

由于 $z_j^k\neq t_j^k$ ➡ $1-z_j^k t_j^k=2$ 即 $\Delta w_{ij}=\eta(1-z_j^k t_j^k)t_j^k x_i^k$

$z_j^k=t_j^k$ ➡ $1-z_j^k t_j^k=0$

（注意：t,z的取值均只能为1或-1）

所以 $\Delta w_{ij}=\eta(t_j^k-z_j^k t_j^k t_j^k)x_i^k=\eta(t_j^k-z_j^k)x_i^k=\eta\delta_j^k x_i^k$

单层神经网络—存在问题

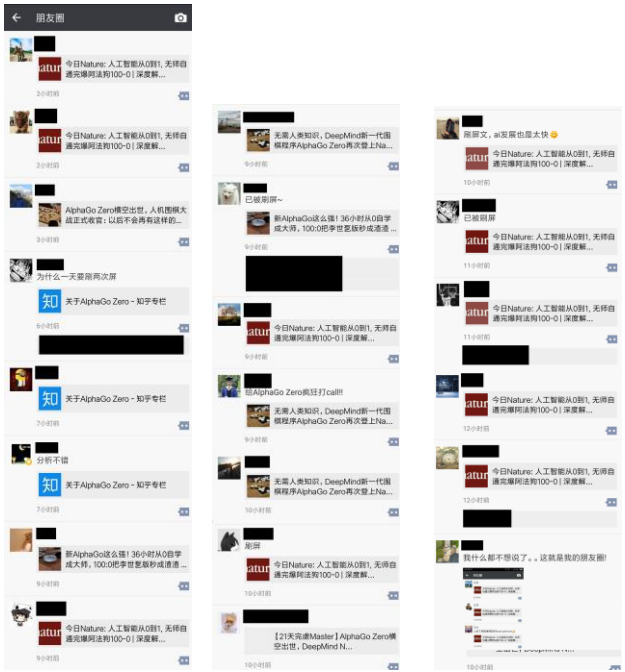
深蓝学院 shenlanxueyuan.com

单层感知器无法解决线性不可分的分类问题

- 无法解决简单的“异或”问题

=> 无法解决各种非线性问题

神经网络的研究
首次陷入低谷



课程大纲

深蓝学院
shenlanxueyuan.com

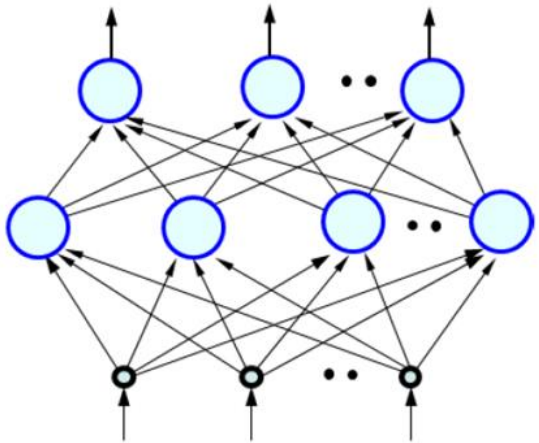
- ✓ 概述
- ✓ 单层神经网络（单层感知器）
- ✓ 多层神经网络（多层感知器）
  - 网络描述
  - 模型训练
  - 误差反向传播算法（BP算法）

多层神经网络—网络描述

深蓝学院 shenlanxueyuan.com

网络结构

- 输入信号：$\{\mathbf{x}^k\}_{k=1}^n$，k表示第k个样本，$k = 1,2,\dots,n$
- 输入层节点数目：$d+1$
- 隐含层节点输出：$\{\mathbf{y}^k\}_{k=1}^n$
- 隐含层节点数目：n_H
- 输出层节点输出：$\{\mathbf{z}^k\}_{k=1}^n$
- 输出层节点数目：c
- 输入层节点i至隐含层节点h的权重：w_{ih}
- 隐含层节点h至输出层节点j的权重：w_{hj}

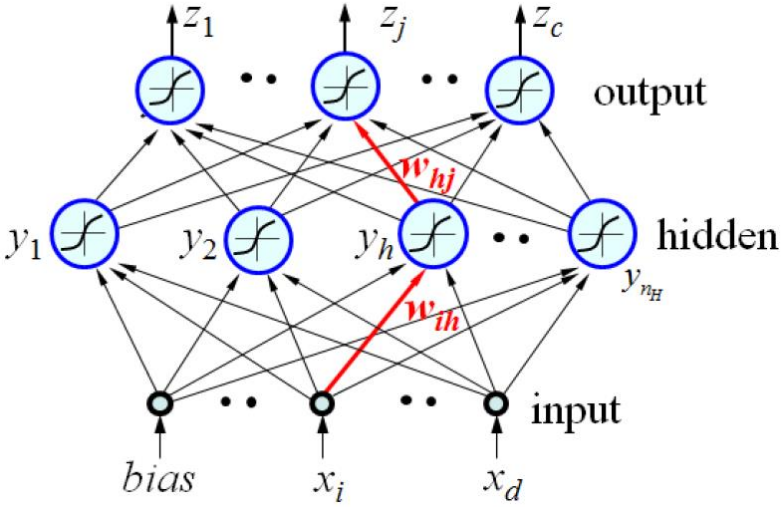


多层神经网络—网络描述

深蓝学院 shenlanxueyuan.com

数学描述

- 对于第k个样本



隐含层输出：$y_h = f(\sum_{i=0}^{d} w_{ih} x_i)$

输出层输出：$z_j = f(\sum_{h=1}^{n_H} w_{hj} y_h) = f(\sum_{h=1}^{n_H} w_{hj} f(\sum_{i=0}^{d} w_{ih} x_i))$

多层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

学习任务

- 给定训练样本$\{\mathbf{x}^k\}_{k=1}^n$及相应的目标值$\{\mathbf{t}^k\}_{k=1}^n$，确定网络中的所有连接权重$\{w_{ih}\}$和$\{w_{hj}\}$
- 当网络完成训练，期望实现 $\mathbf{z}^k = \mathbf{t}^k, k = 1, 2, \mathrm{K}, n$
- 即误差函数

$$E(W) = \frac{1}{2}\sum_{k,j}(t_j^k - z_j^k)^2 \to 0$$

多层神经网络—误差反向传播算法

深蓝学院 shenlanxueyuan.com

简介

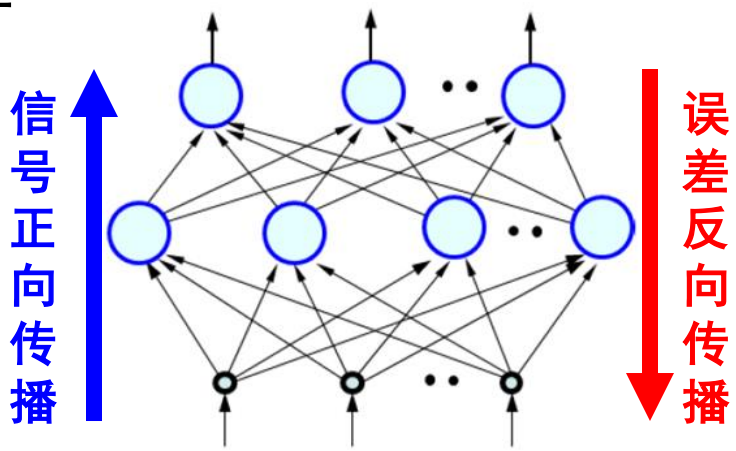
- 误差反向传播算法（Back Propagation）又称BP算法
- 最早由Werbos于1974年提出，1985年Rumelhart等人发展了该理论，**Hinton**等人也在其中做出了贡献
- 神经网络最有效的训练算法
- **目前在深度学习中依然发挥着极其重要的作用**

深度学习巨擘——杰弗里·辛顿，他的家族成就比他现有的成就更让人震惊

多层神经网络—误差反向传播算法

基本原理

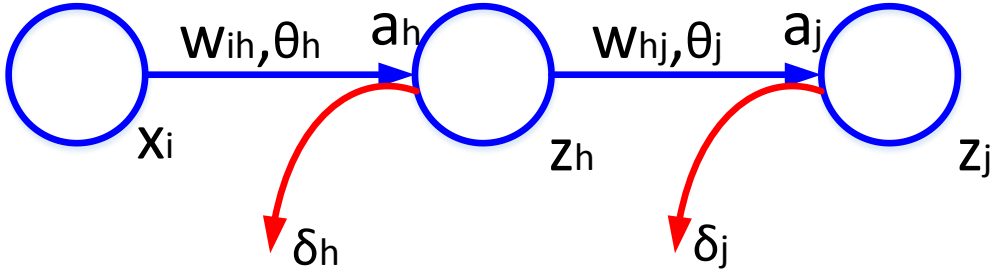
- 利用输出后的误差来估计输出层前一层的误差，再用这个误差估计更前一层的误差，如此一层一层地反传下去，从而获得所有其它各层的误差估计
- 属于有监督学习的方式
- 对网络中的连接权重做动态调整
- 算法核心：**梯度下降法**



多层神经网络—误差反向传播算法

推导过程（相邻两层使用δ规则）

- 模型简化展示



与输入层相关的变量和参数：下标i
与隐含层相关的变量和参数：下标h
与输出层相关的变量和参数：下标j
激励函数的输入：a
激励函数的输出：z
节点误差：δ

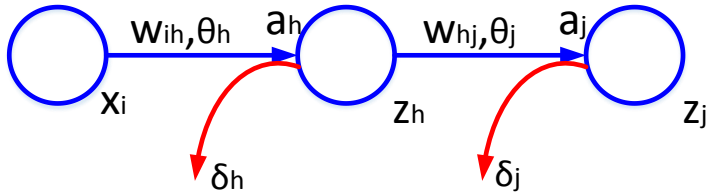
$$a_h = \sum_{i=1}^{d} w_{ih} x_i + \theta_h, \quad z_h = f(a_h)$$

$$a_j = \sum_{h=1}^{n_H} w_{hj} z_h + \theta_j, \quad z_j = f(a_j)$$

损失函数：$E(W) = \frac{1}{2}\sum_j (t_j - z_j)^2$

多层神经网络—误差反向传播算法

推导过程（相邻两层使用δ规则）



- 隐含层-输出层　　链式法则

$$\frac{\partial E}{\partial w_{hj}} = \frac{\partial E}{\partial a_j}\frac{\partial a_j}{\partial w_{hj}} = \delta_j z_h$$

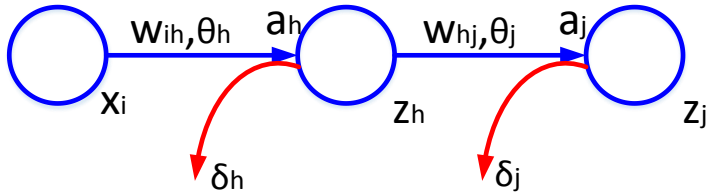
➡ 误差×输入

$$\delta_j = \frac{\partial E}{\partial a_j} = \frac{\partial E}{\partial z_j}\frac{\partial z_j}{\partial a_j} = -(t_j - z_j) f'(a_j)$$

$$\frac{\partial E}{\partial \theta_j} = \frac{\partial E}{\partial a_j}\frac{\partial a_j}{\partial \theta_j} = \delta_j$$

多层神经网络—误差反向传播算法

推导过程（相邻两层使用δ规则）



- 输入层-隐含层　　链式法则

$$\frac{\partial E}{\partial w_{ih}} = \frac{\partial E}{\partial a_h}\frac{\partial a_h}{\partial w_{ih}} = \delta_h x_i$$

➡ 误差×输入

$$\delta_h = \frac{\partial E}{\partial a_h} = \sum_{j=1}^{c}\frac{\partial E}{\partial a_j}\frac{\partial a_j}{\partial z_h}\frac{\partial z_h}{\partial a_h} = \sum_{j=1}^{c}\delta_j w_{hj} f'(a_h) = \sum_{j=1}^{c} w_{hj}(\delta_j f'(a_h))$$

$$\frac{\partial E}{\partial \theta_h} = \frac{\partial E}{\partial a_h}\frac{\partial a_h}{\partial \theta_h} = \delta_h$$

多层神经网络—误差反向传播算法

推导过程

- 权重更新

单样本：

$$\Delta w_* = -\eta \frac{\partial E}{\partial w_*}$$

$$\Delta \theta_* = -\eta \frac{\partial E}{\partial \theta_*}$$

$$w_*^{t+1} = w_*^t + \Delta w_*$$

$$\theta_*^{t+1} = \theta_*^t + \Delta \theta_*$$

批量：

$$\Delta w_* = -\eta \sum_{k=1}^{n} \frac{\partial E}{\partial w_*}$$

$$\Delta \theta_* = -\eta \sum_{k=1}^{n} \frac{\partial E}{\partial \theta_*}$$

$$w_*^{t+1} = w_*^t + \Delta w_*$$

$$\theta_*^{t+1} = \theta_*^t + \Delta \theta_*$$

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

随机梯度下降（单样本）更新算法

Stochastic Backpropagation

1 begin initialize: n_H, **w**, η, criterion θ, $k=0$

2 do $k \leftarrow k+1 \pmod{n}$

3 $\mathbf{x}^k$, randomly chosen a sample (pattern)

4 $w_{hj} \leftarrow w_{hj} + \eta \delta_j^k y_h^k, \quad w_{ih} \leftarrow w_{ih} + \eta \delta_h^k x_i^k$

5 until $\| \nabla J(\mathbf{w}) \| < \theta$

6 return **w**

7 end

单层神经网络—模型训练

深蓝学院 shenlanxueyuan.com

梯度下降（批量）更新算法

Batch Backpropagation

1 begin initialize: n_H, $\mathbf{w}$, η, criterion θ, $r=0$

2 do $r \leftarrow r+1$ (increment epoch)

3 $k=0$, $\Delta w_{ih}=0$, $\Delta w_{hj} \neq 0$

4 do $k \leftarrow k+1 \pmod n$

5 $\mathbf{x}^k$, selected a sample (pattern)

6 $\Delta w_{hj} \leftarrow \Delta w_{hj} + \eta \delta_j^k y_h^k$, $\Delta w_{ih} \leftarrow \Delta w_{ih} + \eta \delta_h^k x_i^k$

7 until $k=n$

8 $w_{hj} \leftarrow w_{hj} + \Delta w_{hj}$, $w_{ih} \leftarrow w_{ih} + \Delta w_{ih}$

9 until $\| \nabla J(\mathbf{w}) \| < \theta$ // 所有样本完成之后再更新

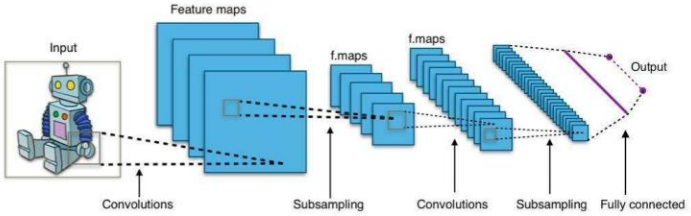
10 return $\mathbf{w}$

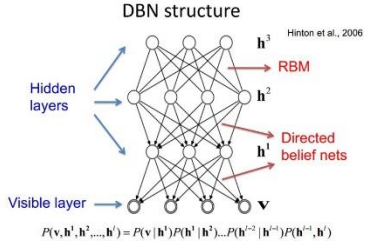
11 end

在实际应用中，也可以将数据分成不同的批，在每一批数据内，按批量更新算法更新网络的权重，在批与批之间则按随机更新算法来更新网络的权重。

多层神经网络—误差反向传播算法

网络设置





- 网络层数
  - 当各结点具有不同的阈值时，具有一个隐含层的网络可以表示任意函数，但由于该条件很难满足，该结论意义不大。
  - 当各结点均采用S型函数时，一个隐含层就足以实现任意判别分类问题，两个隐含层则足以实现输入向量的任意输出函数。
  - 层数太多训练难度会急剧增加，需要特殊的网络结构和训练技巧（如**CNN和DBN**）
  - 通常需要经验性设定

多层神经网络—误差反向传播算法

网络设置

- 各层节点数
  - 输入层：取决于输入数据的维度（图像？音频？文本？）。
  - 隐含层：人为设定，决定了网络的表达能力和复杂程度，同时还需要考虑参数数目和训练难度。对于特定类型的网络，例如卷积神经网络，则重点在于设定卷积核的尺寸和数目。
  - 输出层：取决于输出的类型（例如由分类的类别数决定）。

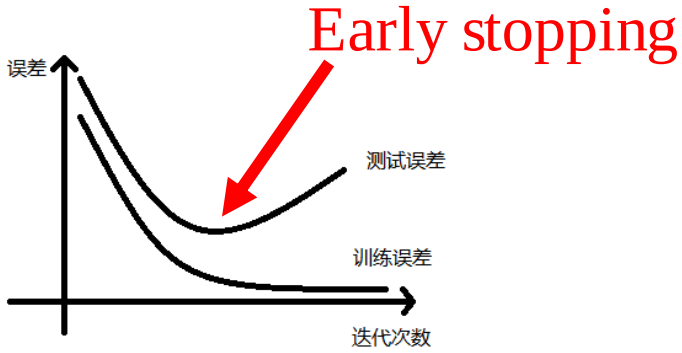
据Prof. Xiaogang Wang的介绍，通常一个在多类别数据上训练好的网络，再在少类别数据上fine-tuning，效果会比直接在少类别数据上训练要好。

多层神经网络—误差反向传播算法

深蓝学院 shenlanxueyuan.com

网络设置

- 输入数据预处理
  - 最好具有零均值、单位标准差：可能影响特别大！
  - 数据增广：裁剪、旋转、加噪声等，在深度学习中极为常用
- 训练停止准则
  - 没有固定准则，但是要防止过拟合
  - “Ctrl + C”



多层神经网络—误差反向传播算法

网络设置

- 参数：
  - 初始权重：通常从一个均匀分布中随机选择初始值，或者用特定的方法：MSRA，xavier等
  - 学习率：太小收敛慢，太大容易震荡，一般从大到小逐渐降低
  - 权重衰减：防止网络出现过拟合 $w^{new} = w^{old}(1-\varepsilon), \quad 0 < \varepsilon < 1$
  - 附加冲量项：轨迹更平滑

$$E_{new}(w) = E(w) + \frac{2\varepsilon}{\eta} w^T w$$

正则

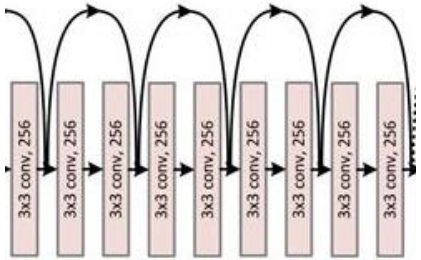
$$w_*^{t+1} = w_*^t + (1-\alpha)\Delta w_* + \alpha(w_*^t - w_*^{t-1})$$

多层神经网络—误差反向传播算法

深蓝学院 shenlanxueyuan.com

存在问题

思考：【面试真题】
ResNet如何避免
梯度爆炸问题？



- 无法训练
  - 网络麻痹：加权和处于Sigmoid饱和区->梯度小->权重调整慢
  - 梯度消失与梯度爆炸
  - 局部最优解：损失函数高度非线性，因此不可避免，只能采用一些策略来尽可能得到更好的局部最优解，例如采用不同的初始化方式
- 训练时间过长
  - 初始值（一开始就陷入局部极小）或者学习率不合适（持续震荡或者步长太小导致损失几乎不变）

参考资料

1. 蔡元龙，模式识别，西北电讯工程学院出版社，1986。

2. 王旭 等，人工神经元网络原理与应用，东北大学出版社，2000。

3. http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial

4. Y. Bengio et al., Deep learning. https://github.com/HFTrader/DeepLearningBook .

5. CVPR/ICCV/ECCV/NIPS等国际顶级会议论文。

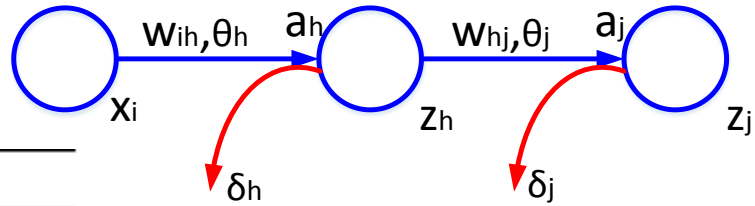
课后作业

1. 在单层神经网络中，采用非线性单元时梯度下降法的公式推导。

2. 根据逻辑“与”功能的真值表，利用单层神经网络（以符号函数作为激励函数）来实现该功能。其中学习率、阈值（即偏置）、初始权重自己设定，“真”用“1”表示，“假”用“0”表示。

3. 与上一题条件类似，实现逻辑“异或”功能。

4. 独立完成右侧网络模型的BP算法推导。



k	x_1	x_2	T
1	0	0	-1(0)
2	0	1	-1(0)
3	1	0	-1(0)
4	1	1	1

逻辑“与”的真值表

k	x_1	x_2	T
1	0	0	-1(0)
2	0	1	1
3	1	0	1
4	1	1	-1(0)

逻辑“异或”的真值表

在线问答

深蓝学院
shenlanxueyuan.com

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening