北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Data Analytics with R

R语言数据分析

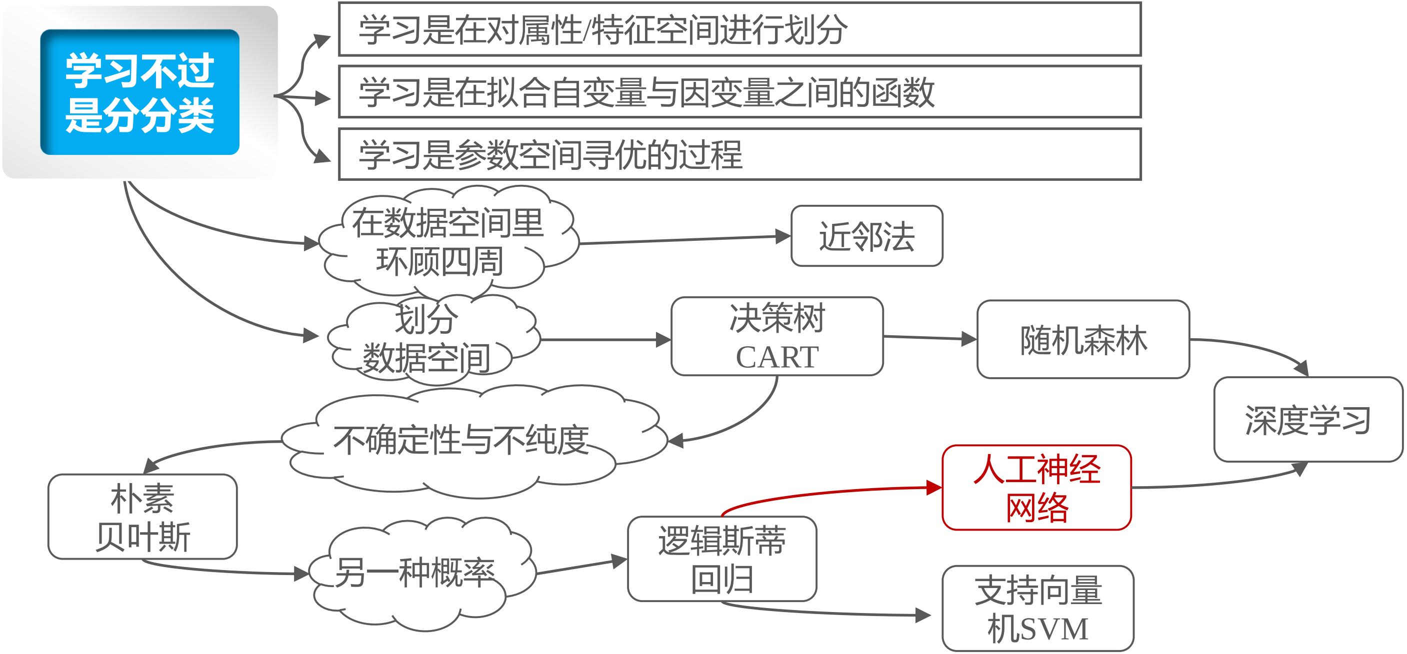
既是世间法、自当有分别

艾新波 / 2018 • 北京

课程体系

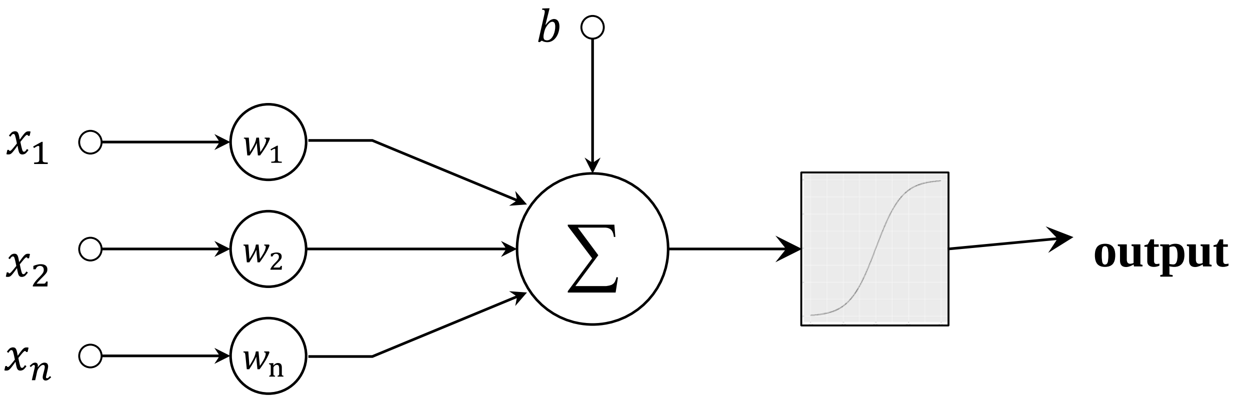
- R语言数据分析
  - 上部：论道 
    - 第1章 气象万千、数以等观
    - 第2章 所谓学习、归类而已
    - 第3章 格言联璧话学习
    - 第4章 源于数学、归于工程
  - 中部：执具 
    - 第5章 工欲善其事必先利其器
    - 第6章 基础编程
    - 第7章 数据对象
    - 第8章 人人都爱tidyverse
    - 第9章 最美不过数据框
  - 下部 博术 
    - 第10章 观数以形
    - 第11章 相随相伴、谓之关联
    - 第12章 既是世间法、自当有分别
    - 第13章 方以类聚、物以群分
    - 第14章 庐山烟雨浙江潮

算法模型



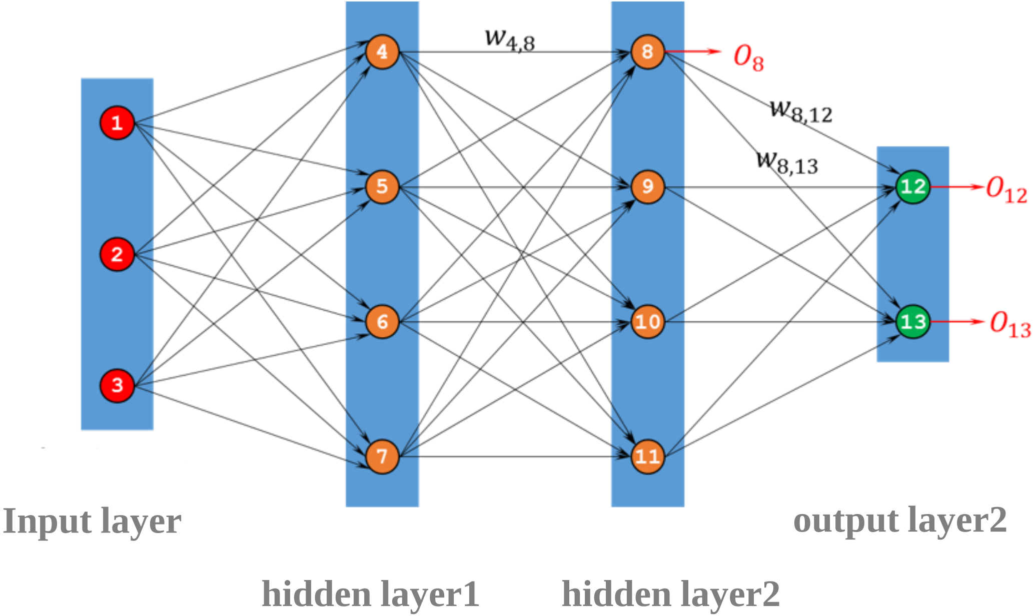
学习不过是分分类
学习是在对属性/特征空间进行划分
学习是在拟合自变量与因变量之间的函数
学习是参数空间寻优的过程
在数据空间里环顾四周
近邻法
划分数据空间
决策树 CART
随机森林
深度学习
不确定性与不纯度
朴素贝叶斯
人工神经网络
另一种概率
逻辑斯蒂回归
支持向量机SVM

扩充计算单元



$$O_j = \frac{1}{1+e^{-I_j}} = \frac{1}{1+e^{-(w^T x+b)}}$$

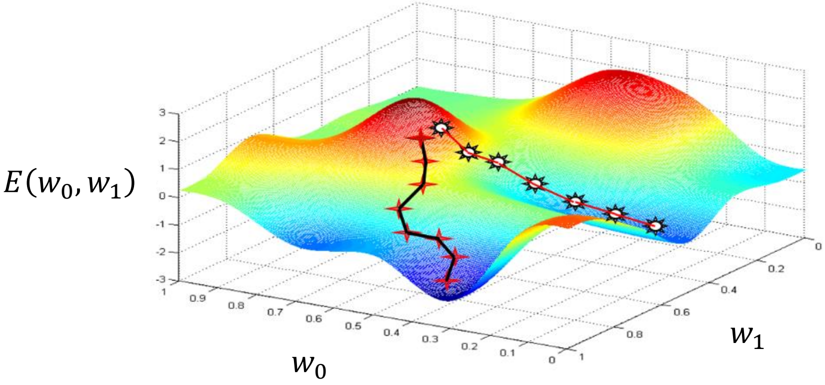
人工神经网络



1
2
3
4
5
6
7
8
9
10
11
12
13
$w_{4,8}$
$w_{8,12}$
$w_{8,13}$
o_8
o_{12}
o_{13}
Input layer
hidden layer1
hidden layer2
output layer2

人工神经网络

学习策略——最小化误差平方和：$E(\vec{w}) \equiv \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2$



学习过程犹如瞎子爬山，只清楚脚下的情况——根据梯度来最速下降

图片引自Andrew Ng《Machine Learning》公开课，作了修改

人工神经网络

学习策略——最小化误差平方和： $E(\vec{w}) \equiv \frac{1}{2}\sum_{d\in D}(t_d - o_d)^2$

误差曲面最陡的方向为梯度： $\Delta E(\vec{w}) \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots, \frac{\partial E}{\partial w_n}\right]$

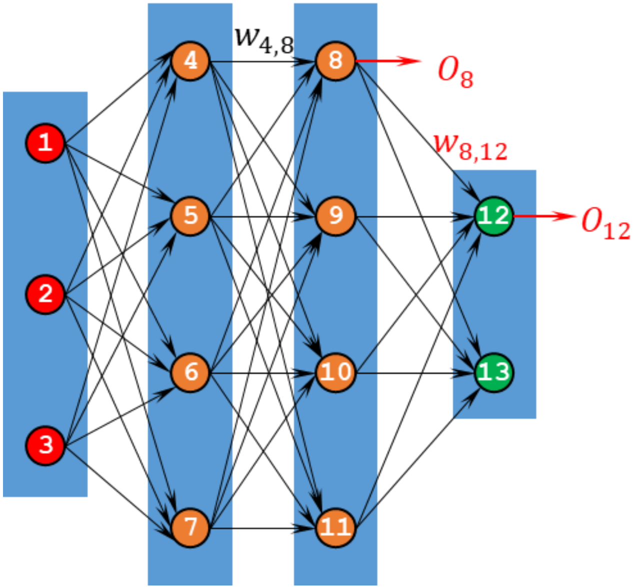
既然梯度确定了E最陡峭的上升方向，那么梯度下降的训练法则为：

$$\vec{w} \leftarrow \vec{w} + \Delta w = \vec{w} - \lambda \Delta E(\vec{w})$$

$$w_i \leftarrow w_i + \Delta w_i = w_i - \lambda \frac{\partial E}{\partial w_i}$$

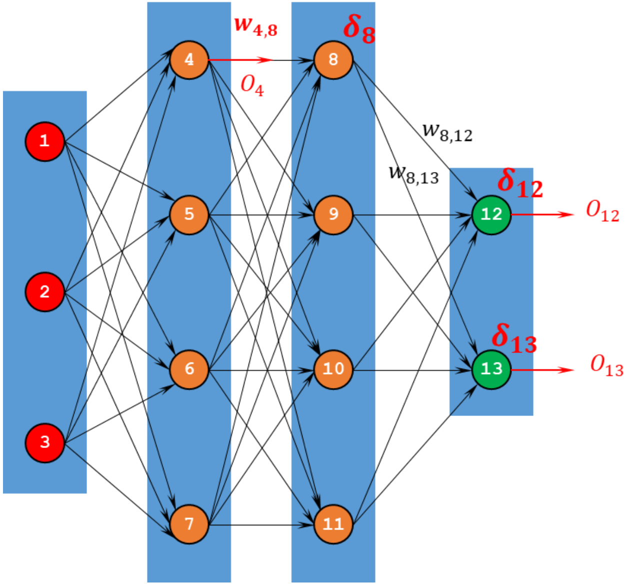
误差反向传播：**链式法则**

人工神经网络



$$
\begin{aligned}
\Delta w_{8,12} &= -\lambda \times \frac{\partial E}{\partial w_{8,12}} \\
&= -\lambda \times \frac{\partial}{\partial w_{8,12}}\left(\frac{1}{2}\left(\left(T_{12}-O_{12}\right)^2+\left(T_{13}-O_{13}\right)^2\right)\right) \\
&= -\lambda \times \frac{1}{2} \times 2 \times \left(T_{12}-O_{12}\right) \times \left(-\frac{\partial O_{12}}{\partial w_{8,12}}\right) \\
&= \lambda \times \left(T_{12}-O_{12}\right) \times \frac{\partial O_{12}}{\partial I_{12}} \times \frac{\partial I_{12}}{\partial w_{8,12}} \\
&= \lambda \times \left(T_{12}-O_{12}\right) \times \frac{\partial}{\partial I_{12}}\left(\frac{1}{1+e^{-I_{12}}}\right) \times O_8 \\
&= \lambda \times \left(T_{12}-O_{12}\right) \times O_{12} \times \left(1-O_{12}\right) \times O_8 \\
&= \lambda \times O_8 \times \delta_{12}
\end{aligned}
$$

人工神经网络



$$\Delta w_{4,8} = -\lambda \times \frac{\partial E}{\partial w_{4,8}}$$

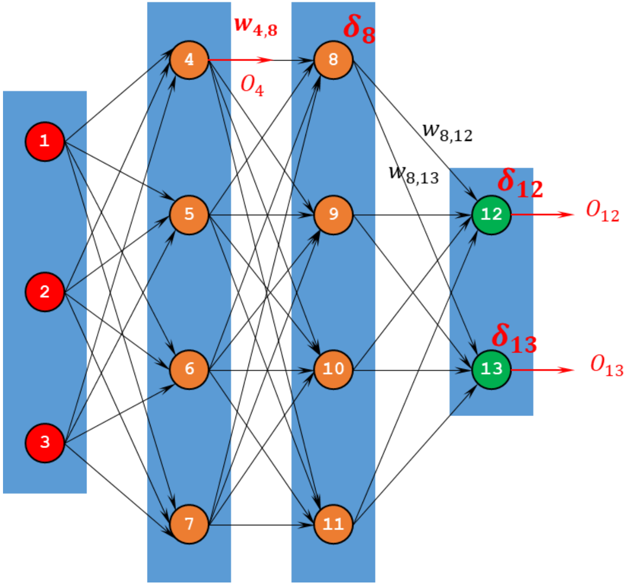
$$= -\lambda \times \frac{\partial}{\partial w_{4,8}}\left(\frac{1}{2}\left(\left(T_{12}-O_{12}\right)^2 + \left(T_{13}-O_{13}\right)^2\right)\right)$$

$$= -\lambda \times \frac{1}{2} \times \{2 \times \left(T_{12}-O_{12}\right) \times \left(-\frac{\partial O_{12}}{\partial w_{4,8}}\right)$$

$$+ 2 \times \left(T_{13}-O_{13}\right) \times \left(-\frac{\partial O_{13}}{\partial w_{4,8}}\right)\}$$

$$= \lambda \times \left(\left(T_{12}-O_{12}\right) \times \frac{\partial O_{12}}{\partial w_{4,8}} + \left(T_{13}-O_{13}\right) \times \frac{\partial O_{13}}{\partial w_{4,8}}\right)$$

人工神经网络



$$\Delta w_{4,8} = -\lambda \times \frac{\partial E}{\partial w_{4,8}}$$

$$= \lambda \times \left((T_{12} - O_{12}) \times \frac{\partial O_{12}}{\partial w_{4,8}} + (T_{13} - O_{13}) \times \frac{\partial O_{13}}{\partial w_{4,8}} \right)$$

$$= \lambda \times \{ (T_{12} - O_{12}) \times \frac{\partial O_{12}}{\partial I_{12}} \times \frac{\partial I_{12}}{\partial O_8} \times \frac{\partial O_8}{\partial I_8} \times \frac{\partial I_8}{\partial w_{4,8}}$$

$$+ (T_{13} - O_{13}) \times \frac{\partial O_{13}}{\partial I_{13}} \times \frac{\partial I_{13}}{\partial O_8} \times \frac{\partial O_8}{\partial I_8} \times \frac{\partial I_8}{\partial w_{4,8}} \}$$

$$= \lambda \times \{ \delta_{12} \times w_{8,12} \times O_8 \times (1 - O_8) \times O_4$$

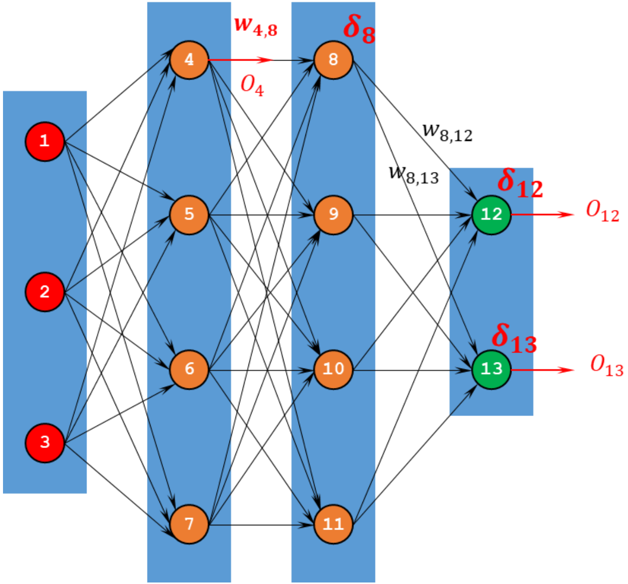
$$+ \delta_{13} \times w_{8,13} \times O_8 \times (1 - O_8) \times O_4 \}$$

$$= \lambda \times O_8 \times (1 - O_8) (\delta_{12} \times w_{8,12} + \delta_{13} \times w_{8,13}) \times O_4$$

$$= \lambda \times O_8 \times (1 - O_8) (\delta_{12} \times w_{8,12} + \delta_{13} \times w_{8,13}) \times O_4$$

$$= \lambda \times O_4 \times \delta_8$$

人工神经网络



权值更新公式：

$$w_{i,j} \leftarrow w_{i,j} + \Delta w_{i,j}$$

其中$\Delta w_{i,j} = \lambda \times O_i \times \delta_j$

误差传播公式：

$$\delta_j = O_j(1 - O_j)\sum_k \delta_k w_{j,k}$$

人工神经网络

输入：

D由训练元组和目标值组成的数据集

λ学习率

$network$多层前馈神经网络

输出：训练后的神经网络

初始化 $network$的所有权值和偏置

repeat

 for 训练集 D中的每个训练元组 X **do**

  / /向前传播输入

  for 每个输入层单元 j **do**

   $O_j = I_j$

  end for

  for 隐藏或输出层的每个单元 j **do**

   $I_j = \sum_i w_{ij} O_i + b_j$

   $O_j = \dfrac{1}{1 + e^{-I_j}}$

  end for

  / /向后传播误差

  for 输出层的每个单元 j **do**

   $\delta_j = O_j\left(1 - O_j\right)\left(T_j - O_j\right)$

  end for

人工神经网络

for 由最后一个到第一个隐藏层，对于隐藏层的每个单元 j **do**

$\delta_j = O_j(1 - O_j)\sum_k \delta_k w_{jk}$

end for

for $network$中的每个权值 w_{ij} **do**

$\Delta w_{ij} = \lambda O_i \delta_j$ //权重增量

$w_{ij} = w_{ij} + \Delta w_{ij}$ //权重更新

end for

for $network$中的每个偏置 b_j **do**

$\Delta b_j = \lambda \delta_j$ //偏置增量

$b_j = b_j + \Delta b_j$ //偏置更新

end for

end for

until 满足终止条件

谢谢聆听

Thank you

教师个人联系方式

艾新波

手机：13641159546

QQ：23127789

微信：13641159546

E-mail: 13641159546@126.com

axb@bupt.edu.cn

地址：北京邮电大学科研楼917室

课程网址：https://github.com/byaxb/RDataAnalytics