机器学习及其MATLAB实现—从基础到实践 第3课

法律声明

【声明】本视频和幻灯片为炼数成金网络课程的教学资料，所有资料只能在课程内使用，不得在课程以外范围散播，违者将可能被追究法律和经济责任。

课程详情访问炼数成金培训网站

http://edu.dataguru.cn

课程目录

人工神经网络概述

- 什么是人工神经网络？
  - In machine learning and cognitive science, **artificial neural networks** (**ANNs**) are a family of statistical learning models inspired by **biological neural networks** (the central nervous systems of animals, in particular the brain) and are used to **estimate or approximate functions** that can depend on a large number of inputs and are generally unknown.
- 人工神经元模型

$$\text{net}_i = \sum_{j=1}^{n} w_{ij} x_j - \theta$$

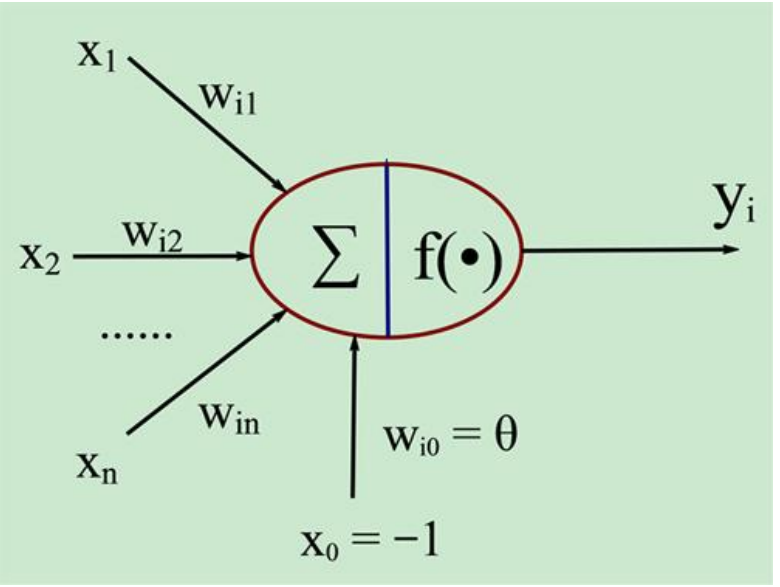
$$y_i = \text{f}(\text{net}_i)$$

$$\text{X} = [\ \text{x0}\ ,\ \text{x1}\ ,\ \text{x2}\ ,\ \ldots\ldots\ ,\ \text{xn}\ ]$$

$$\text{W} = \begin{bmatrix} w_{i0} \\ w_{i1} \\ w_{i2} \\ \vdots \\ w_{in} \end{bmatrix}$$

$$\text{net}_i = \text{XW}$$

$$y_i = \text{f}(\text{net}_i) = \text{f}(\text{XW})$$

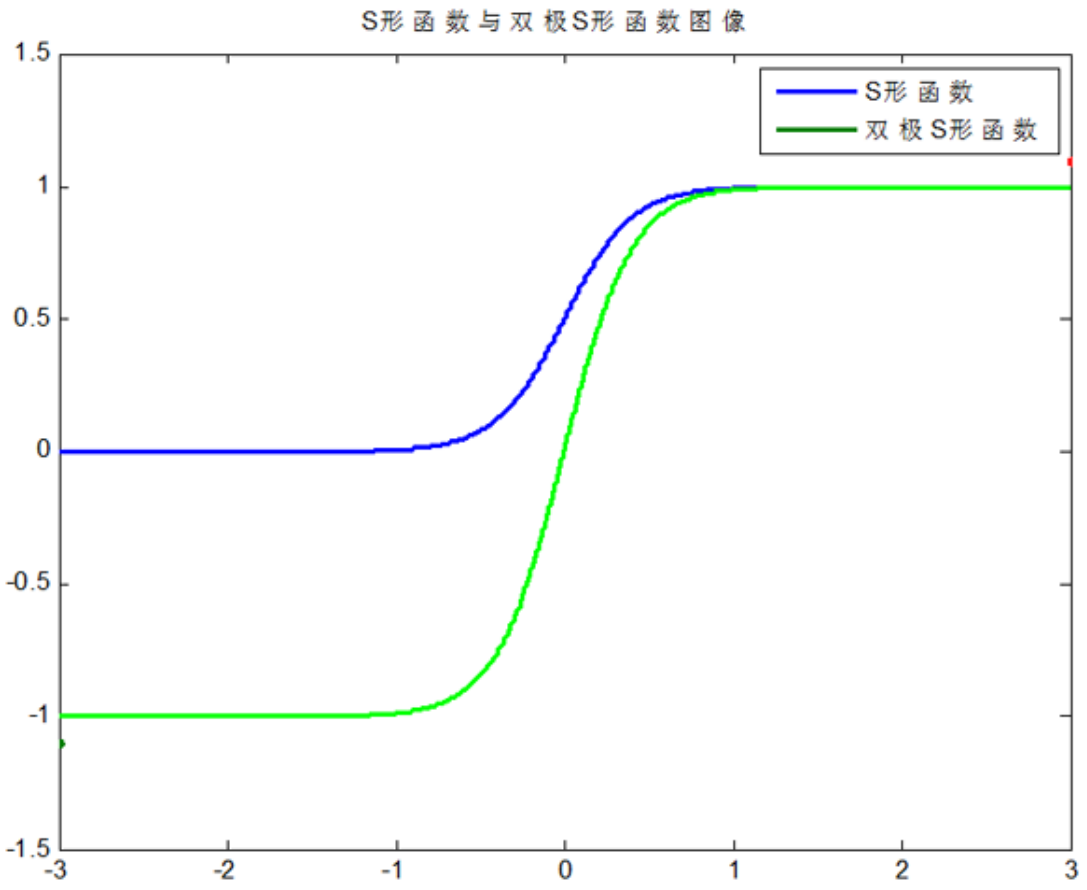


人工神经元模型

- 常用的激活函数 y = f(x)
  - 线性函数 $f(x) = k * x + c$
  - 斜坡函数 $f(x) = \begin{cases} T, & x > c \\ k * x, & |x| \le c \\ -T, & x < -c \end{cases}$
  - 阈值函数 $f(x) = \begin{cases} 1, x \ge c \\ 0, x < c \end{cases}$
  - S型函数（Sigmoid） $f(x) = \frac{1}{1 + e^{-\alpha x}} \ (0 < f(x) < 1)$

    $f'(x) = \frac{\alpha e^{-\alpha x}}{(1 + e^{-\alpha x})^2} = \alpha f(x)[1 - f(x)]$
  - 双极S型函数 $f(x) = \frac{2}{1 + e^{-\alpha x}} - 1 \ (-1 < f(x) < 1)$

    $f'(x) = \frac{2\alpha e^{-\alpha x}}{(1 + e^{-\alpha x})^2} = \frac{\alpha [1 - f(x)^2]}{2}$



神经网络概述

- 神经网络可以分为哪些？
  - 按照连接方式，可以分为：前向神经网络 vs. 反馈（递归）神经网络
  - 按照学习方式，可以分为：有导师学习神经网络 vs. 无导师学习神经网络
  - 按照实现功能，可以分为：拟合（回归）神经网络 vs. 分类神经网络

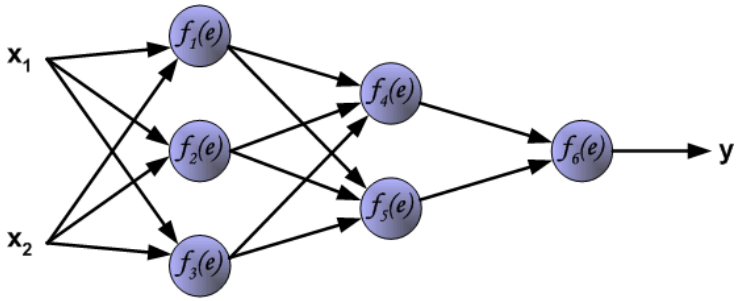
BP神经网络概述

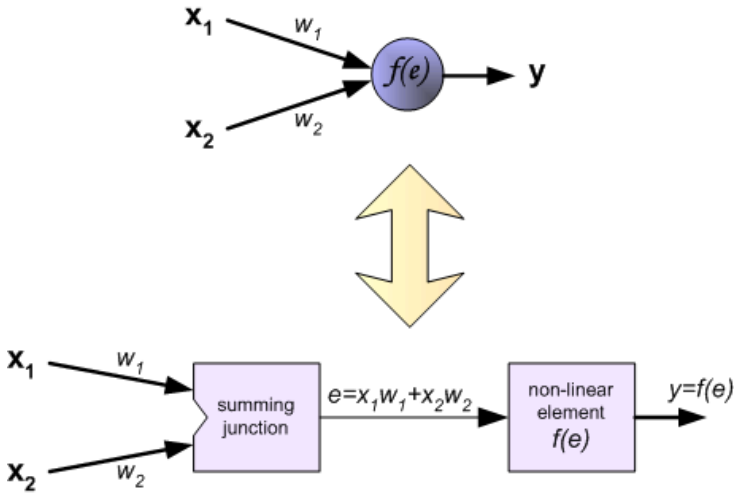
- **Backpropagation** is a common method of teaching artificial neural networks how to perform a given task.
- It is a **supervised learning** method, and is a generalization of **the delta rule**. It requires a teacher that knows, or can calculate, the desired output for any input in the training set.
- Backpropagation requires that the **activation function** used by the artificial neurons (or "nodes") be **differentiable**.

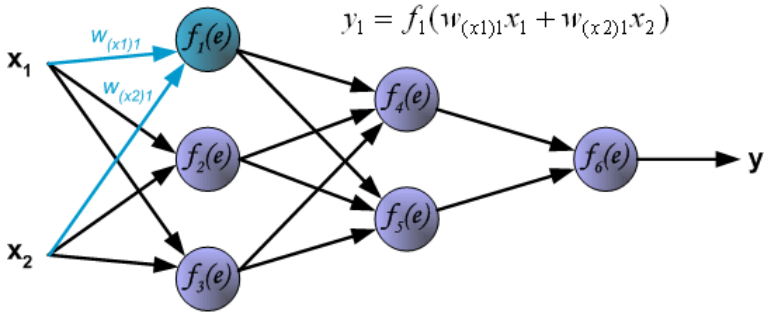
BP神经网络概述

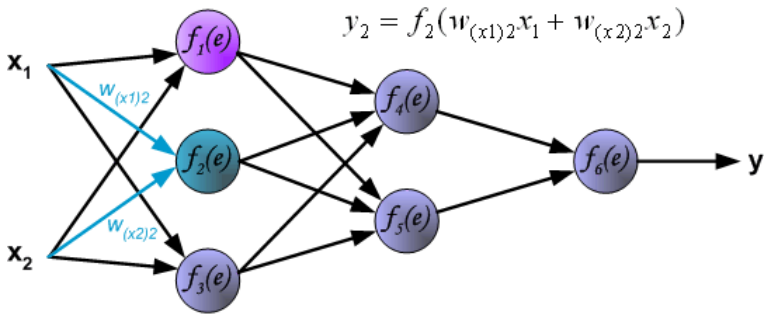
- 学习算法
  - **Phase 1: Propagation**
    1. Forward propagation of a **training pattern’s input** through the neural network in order to generate the propagation’s output activations.
    2. Back propagation of the **propagation’s output activations** through the neural network using the **training pattern’s target** in order to **generate the deltas** of all output and hidden neurons.
  - **Phase 2: Weight Update**
    1. Multiply its output delta and input activation to **get the gradient** of the weight.
    2. Bring the weight in **the opposite direction** of the gradient by subtracting a ration of it from the weight.

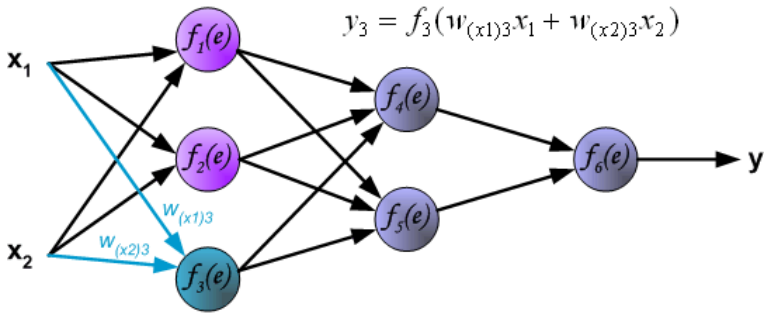
BP神经网络概述





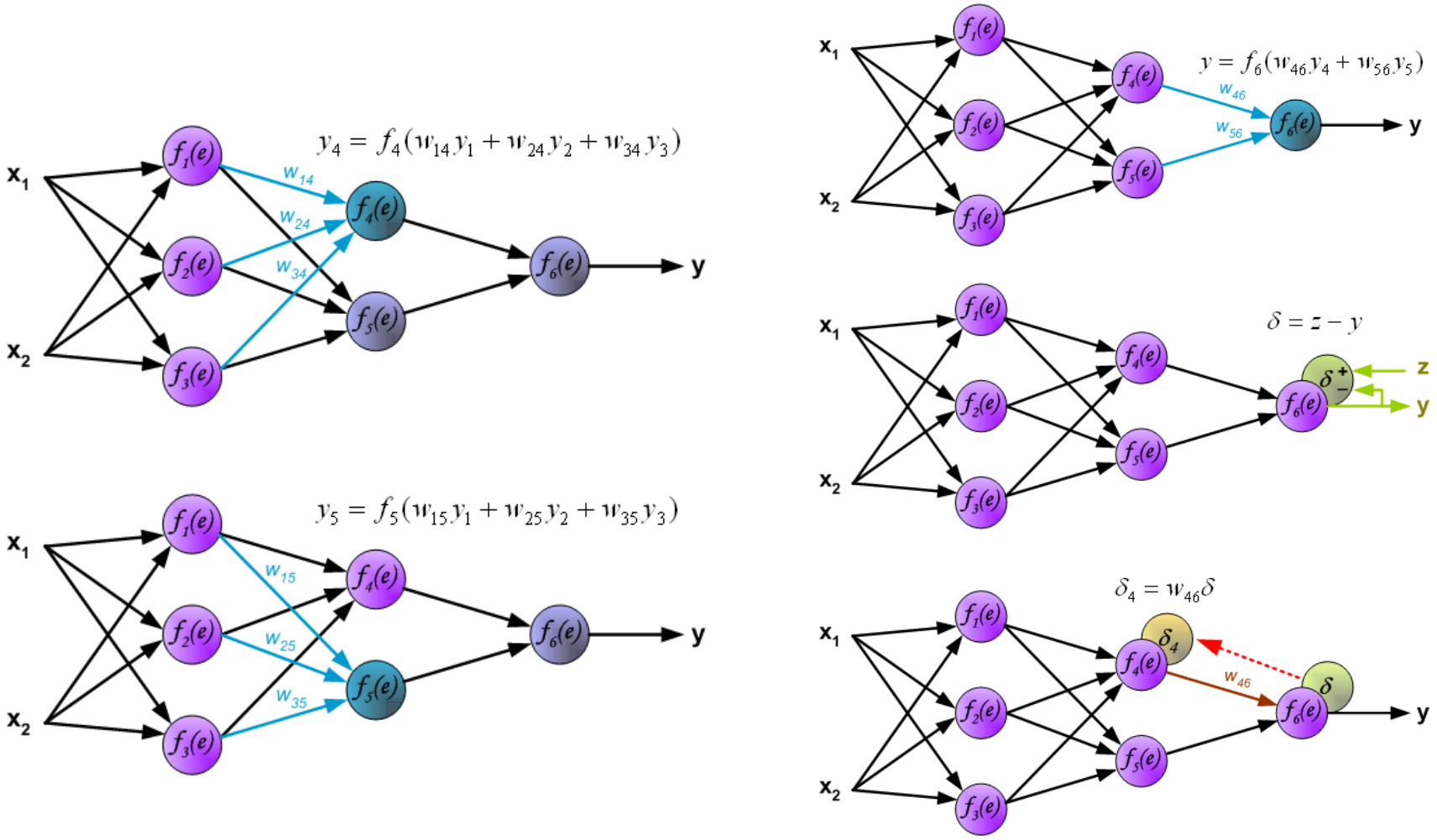






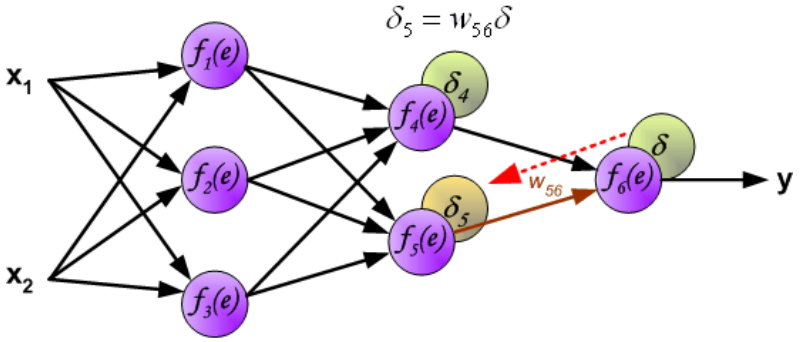
Source: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

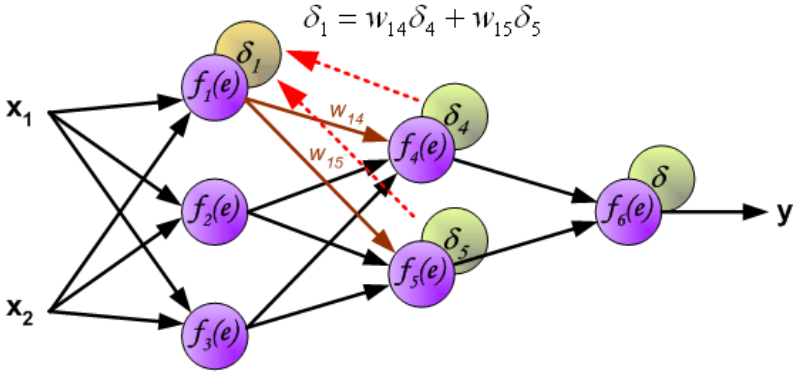
BP神经网络概述

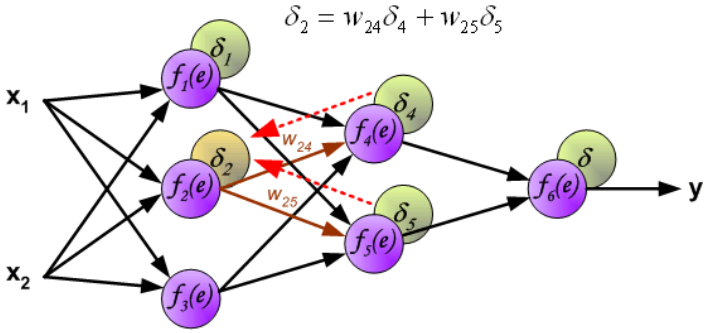


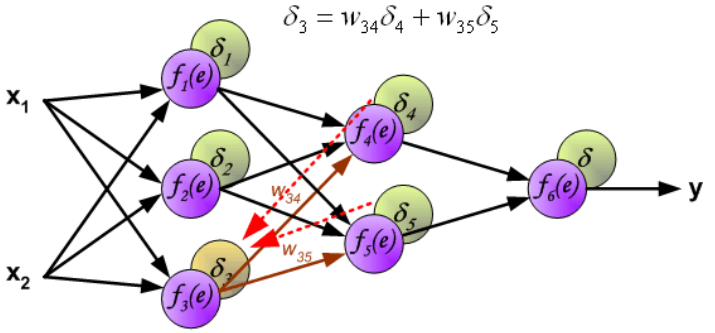
Source: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

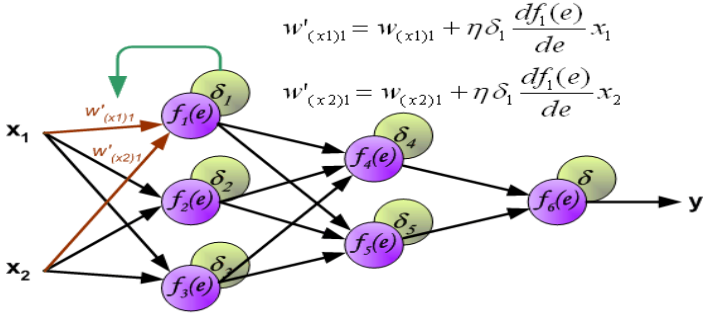
BP神经网络概述









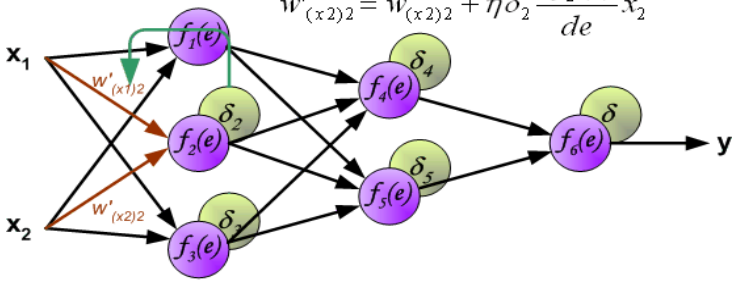


Source: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

BP神经网络概述

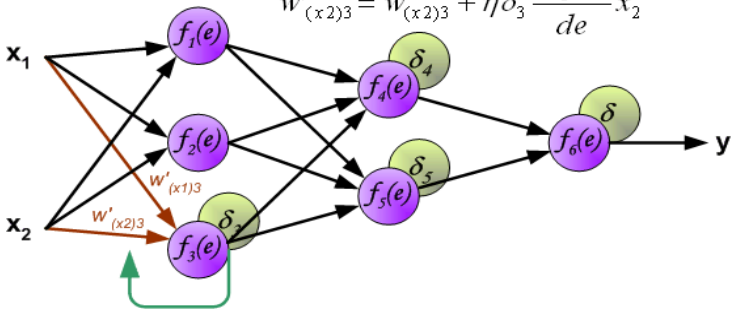
$$w'_{(x1)2} = w_{(x1)2} + \eta\delta_2 \frac{df_2(e)}{de} x_1$$

$$w'_{(x2)2} = w_{(x2)2} + \eta\delta_2 \frac{df_2(e)}{de} x_2$$



$$w'_{(x1)3} = w_{(x1)3} + \eta\delta_3 \frac{df_3(e)}{de} x_1$$

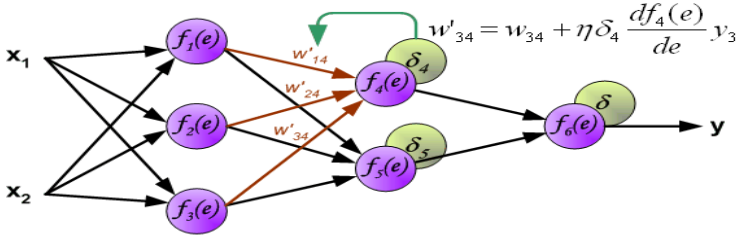
$$w'_{(x2)3} = w_{(x2)3} + \eta\delta_3 \frac{df_3(e)}{de} x_2$$



$$w'_{14} = w_{14} + \eta\delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 \frac{df_4(e)}{de} y_2$$

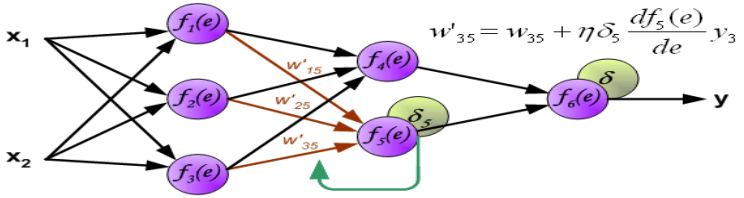
$$w'_{34} = w_{34} + \eta\delta_4 \frac{df_4(e)}{de} y_3$$



$$w'_{15} = w_{15} + \eta\delta_5 \frac{df_5(e)}{de} y_1$$

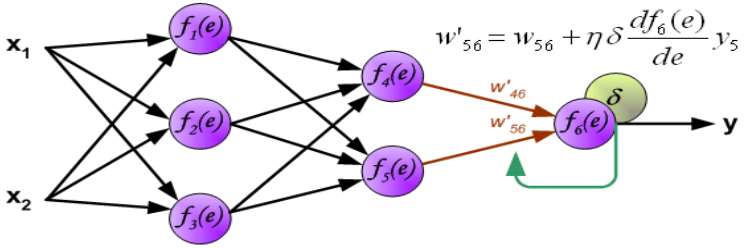
$$w'_{25} = w_{25} + \eta\delta_5 \frac{df_5(e)}{de} y_2$$

$$w'_{35} = w_{35} + \eta\delta_5 \frac{df_5(e)}{de} y_3$$



$$w'_{46} = w_{46} + \eta\delta \frac{df_6(e)}{de} y_4$$

$$w'_{56} = w_{56} + \eta\delta \frac{df_6(e)}{de} y_5$$



Source: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

数据归一化

什么是归一化？

- 将数据映射到[0, 1]或[-1, 1]区间或其他的区间。

为什么要归一化？

- 输入数据的单位不一样，有些数据的范围可能特别大，导致的结果是神经网络收敛慢、训练时间长。
- 数据范围大的输入在模式分类中的作用可能会偏大，而数据范围小的输入作用就可能会偏小。
- 由于神经网络输出层的激活函数的值域是有限制的，因此需要将网络训练的目标数据映射到激活函数的值域。例如神经网络的输出层若采用S形激活函数，由于S形函数的值域限制在(0,1)，也就是说神经网络的输出只能限制在(0,1)，所以训练数据的输出就要归一化到[0,1]区间。
- S形激活函数在(0,1)区间以外区域很平缓，区分度太小。例如S形函数f(X)在参数a=1时，f(100)与f(5)只相差0.0067。

归一化算法

- y = (x - min)/(max - min)
- y = 2 * (x - min) / (max - min) - 1

重点函数解读

- **mapminmax**
  - Process matrices by mapping row minimum and maximum values to [-1 1]
  - [Y, PS] = mapminmax(X, YMIN, YMAX)
  - Y = mapminmax('apply', X, PS)
  - X = mapminmax('reverse', Y, PS)
- **newff**
  - Create feed-forward backpropagation network
  - net = newff(P, T, [S1 S2...S(N-l)], {TF1 TF2...TFNl}, BTF, BLF, PF, IPF, OPF, DDF)
- **train**
  - Train neural network
  - [net, tr, Y, E, Pf, Af] = train(net, P, T, Pi, Ai)
- **sim**
  - Simulate neural network
  - [Y, Pf, Af, E, perf] = sim(net, P, Pi, Ai, T)

参数对BP神经网络性能的影响

- 隐含层神经元节点个数
- 激活函数类型的选择
- 学习率
- 初始权值与阈值
- ……
- 交叉验证（cross validation）
- 训练集（training set）
- 验证集（validation set）
- 测试集（testing set）
- 留一法（Leave one out, LOO）

炼数成金逆向收费式网络课程

- Dataguru（炼数成金）是专业数据分析网站，提供教育，媒体，内容，社区，出版，数据分析业务等服务。我们的课程采用新兴的互联网教育形式，独创地发展了逆向收费式网络培训课程模式。既继承传统教育重学习氛围，重竞争压力的特点，同时又发挥互联网的威力打破时空限制，把天南地北志同道合的朋友组织在一起交流学习，使到原先孤立的学习个体组合成有组织的探索力量。并且把原先动辄成千上万的学习成本，直线下降至百元范围，造福大众。我们的目标是：低成本传播高价值知识，构架中国第一的网上知识流转阵地。
- 关于逆向收费式网络的详情，请看我们的培训网站 http://edu.dataguru.cn

Thanks

FAQ时间