常用机器学习算法及其MATLAB实现

郁 磊[1] 陈媛媛[2]

1 山西大学复杂系统研究所

2 中北大学信息与通信工程学院

E-mail: yulei@sxu.edu.cn

Mobile: 153 8851 9576

说在前面的话

……

目录

目录

第一讲 MATLAB编程基础与进阶

MATLAB简介

- MathWorks：http://www.mathworks.com
- MATLAB是什么?
  - The language of Technical Computing
  - MATLAB® is the high-level language and interactive environment used by millions of **engineers and scientists** worldwide. It lets you explore and visualize ideas and collaborate across disciplines including signal and image processing, communications, control systems, and computational finance.
- MATLAB能做什么?
  - 【在探索中前行】“新视野”号使用MATLAB图像分析优化飞行轨迹
  - 【为了举起大力神杯】开战！2015机器人世界杯大赛

Can you speak MATLAB?

MATLAB版本与选择

- 每半年会出一个新的版本，一年两个版本，譬如：MATLAB R2017a与MATLAB R2017b
- Release Notes: https://cn.mathworks.com/help/matlab/release-notes.html
- 若只是利用MATLAB进行一些算法设计、模拟仿真、数值运算，那么各个版本间差别不大
- 若涉及到MATLAB、Simulink与硬件交互，那么建议关注新版本的功能

本讲义的MATLAB版本为R2012a，R2010b以后均可以支持。

MATLAB的安装

- 下载地址：http://pan.baidu.com/s/1c2w5qnu 提取密码：vld0
- 安装教程：http://jingyan.baidu.com/article/676629974101de54d51b8430.html
  - 不使用Internet安装
  - 安装秘钥：crack文件夹→install.txt→standalone
  - 不使用Internet手动激活
  - Crack文件夹→lic_standalone.dat

MATLAB开发环境介绍

MATLAB变量命名规则

- 变量名区分大小写
- 变量名长度不超过63位 （MATLAB R2012a 32bit和64bit计算机上测试结果）
- 变量名以字母开头，可以由字母、数字和下划线组成，但不能使用标点
- **变量名应简洁明了，通过变量名可以直观看出变量所表示的物理意义**

MATLAB数据类型

- 数字
- 字符与字符串
- 矩阵
- 元胞数组
- 结构体

Demo Time

MATLAB矩阵操作

- 矩阵的定义与构造
- 矩阵的四则运算
- 矩阵的下标

Demo Time

MATLAB逻辑与流程控制

- if … else … end
- for … end
- while … end
- switch … case … end

Demo Time

MATLAB脚本与函数文件

- 脚本文件
- 函数文件

Demo Time

MATLAB基本绘图操作

◆ 绘图命令

- **plot** 2-D line plot
- **line** Create line object
- **plotyy** 2-D line plots with y-axes on both left and right side
- **plot3** 3-D line plot

◆ 设置坐标轴和网络线属性

- **axis** Axis scaling and appearance
- **xlim, ylim, zlim** Set or query axis limits
- **grid** Grid lines for 2-D and 3-D plot
- **box** Axes border
- **xlabel, ylabel, zlabel** Label *x*-, *y*-, and *z*-axis

◆ 标注图形

- **title** Add title to current axes
- **text** Create text object in current axes
- **gtext** Mouse placement of text in 2-D view
- **legend** Graph legend for lines and patches

◆ 子图绘制及多曲线绘制

- **subplot** Create axes in tiled positions
- **hold** Retain current graph in figure

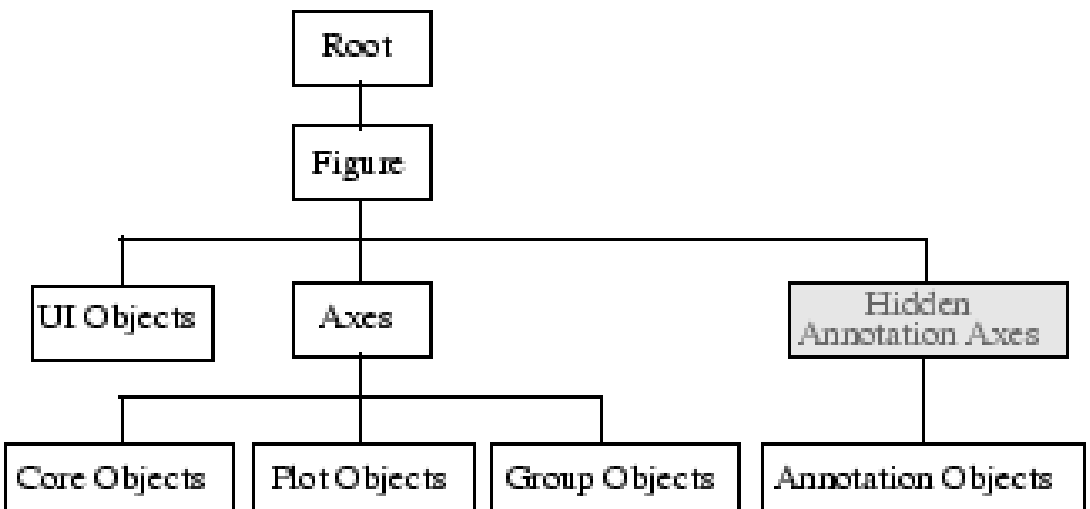
MATLAB图形保存与导出

- Edit→Copy Figure
- File→Export Setup
- print函数

Demo Time

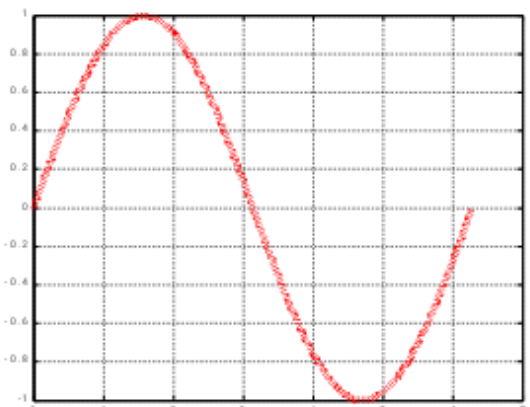
图形句柄系统

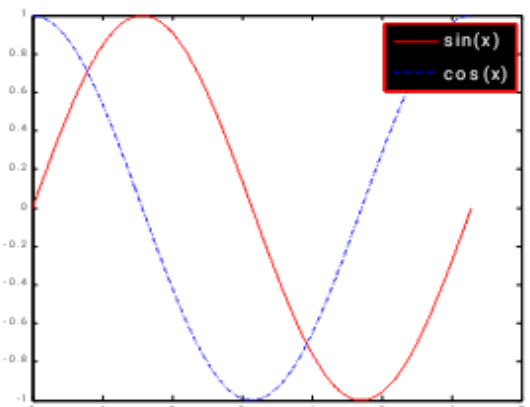
- 图形对象：用于界面制作和数据可视的基本绘图元素。
- 图形对象是图形系统中最基本、最底层的单元。
- 图形对象的属性由属性名和属性值两部分组成。
- 句柄是图形对象的标识代码，句柄含有图形对象的各种必要的属性信息。
- 根屏幕的句柄为0，图形窗口的句柄为整数，其他对象的句柄为浮点数。

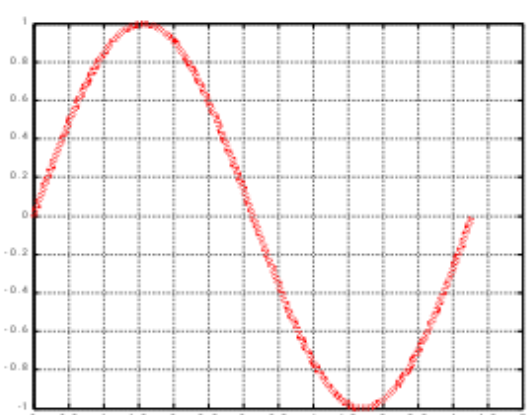


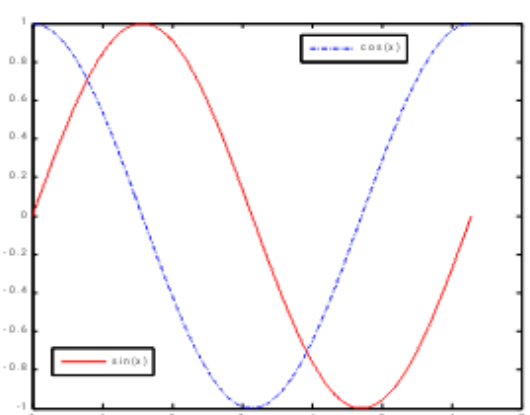
图形句柄系统

- 如何设置线条的属性呢？
- 如何修改网格的间隔呢？
- 如何设置图例的字体及大小呢？
- ……









MATLAB文件导入

- mat格式
- txt格式
- xls格式
- csv格式
- ……

Demo Time

MATLAB编程习惯与风格

- 变量的命名规则
- Cell Mode
- 程序发布（Publish）

Demo Time

闻道有先后，术业有专攻

所谓高手，就是他们比你更早经历了那些bug，比你经历了更多bug，仅此而已。

MATLAB程序调试

- 错误信息的阅读
  - Index must be a positive integer or logical.
  - Undefined function or variable “B”.
  - Inner matrix dimensions must agree.
  - Function definitions are not permitted at the prompt or in scripts.
  - Index out of bounds because numel(A)=5.
  - In an assignment A(I) = B, the number of elements in B and I must be the same.
  - Expression or statement is incorrect--possibly unbalanced (, {, or [.
  - Too many input arguments.
  - ……

Demo Time

MATLAB程序调试

- 断点调试
  - 设置/清除断点
  - 进入/退出调试模式
  - 循环体的调试

Demo Time

MATLAB程序调试

- 如何充分地利用网络资源？
  - www.mathworks.com/matlabcentral/fileexchange/
  - www.matlabsky.com
  - www.ilovematlab.cn
  - Some Tips:
    - 再现问题(完整的错误信息、源程序文件、数据、所使用的MATLAB版本、计算机操作系统版本等)
    - 帖子标题(应简要说明问题的本质)

Demo Time

MATLAB程序调试

- 如何查看、编辑MATLAB自带的工具箱函数？
  - edit
  - varargin/varargout/nargin/nargout
    - **varargin:** Variable length input argument list.
    - **varargout**: Variable length output argument list.
    - **nargin:** Number of input arguments specified for a function.
    - **nargout:** Number of output arguments specified for a function.

Demo Time

向量化编程

- 及时清除不用的变量
- 使用变量前，预分配内存空间
- 选择恰当的数据类型
- 循环与向量化
  - 按列优先循环
  - 循环次数多的变量安排在内层
- 给一些函数“瘦身”
- ……

Demo Time

目录

第二讲 常用的机器学习算法及其MATLAB实现

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

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

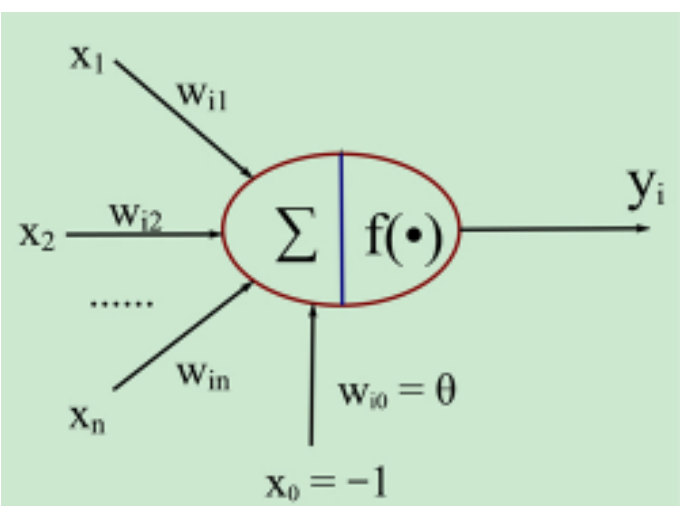
$$y_i = \mathrm{f}(\mathrm{net}_i)$$

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

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

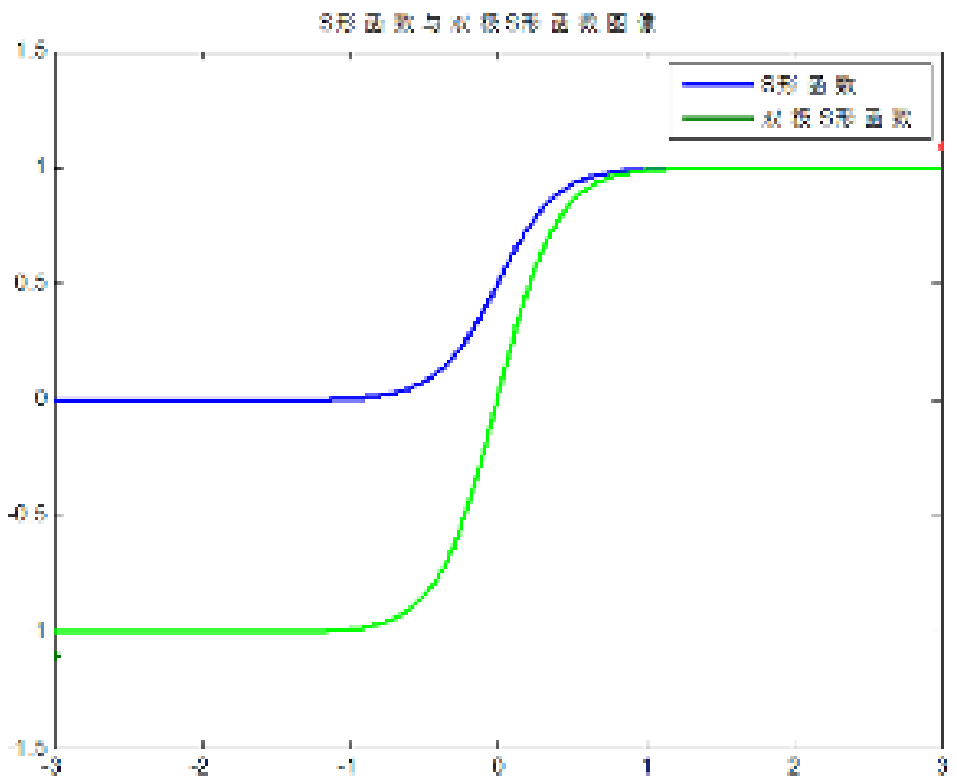
$$\mathrm{net}_i = XW$$

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



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
  - 阈值函数
  - S型函数（Sigmoid）
  - 双极S型函数



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

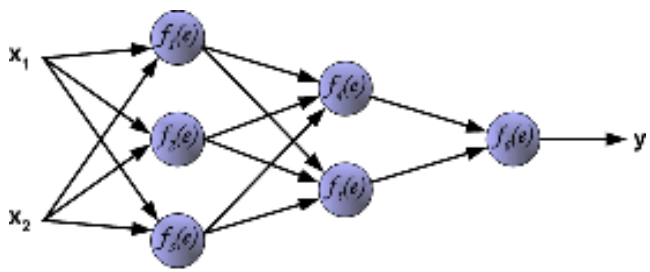
BP神经网络概述

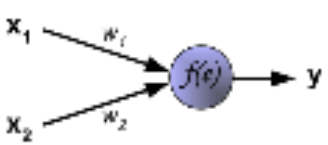
- **Backpropagation** is a common method of teaching artificial neural networks how to perform a given task.
- It is a **supervised learning** method, and is a generalization of **the delta rule**. It requires a teacher that knows, or can calculate, the desired output for any input in the training set.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

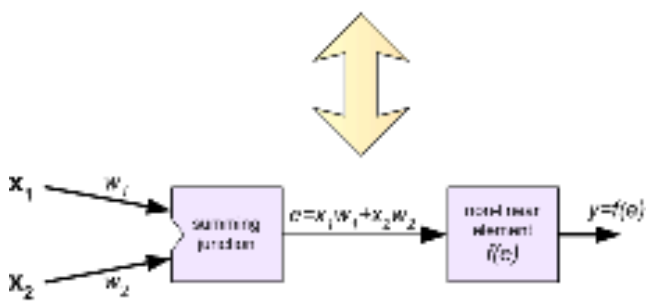
BP神经网络概述

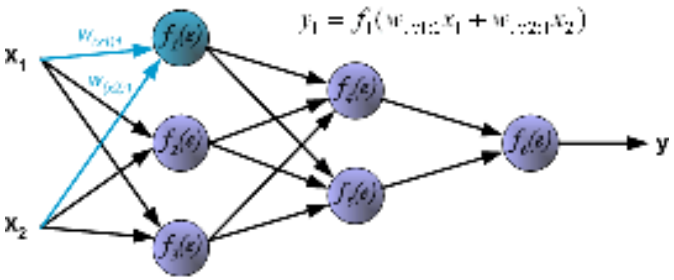
- 学习算法
  - **Phase 1: Propagation**
    1. Forward propagation of a **training pattern's input** through the neural network in order to generate the propagation's output activations.
    2. Back propagation of the **propagation's output activations** through the neural network using the **training pattern's 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.

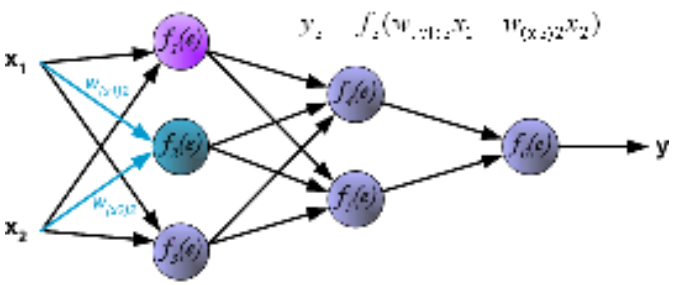
BP神经网络概述

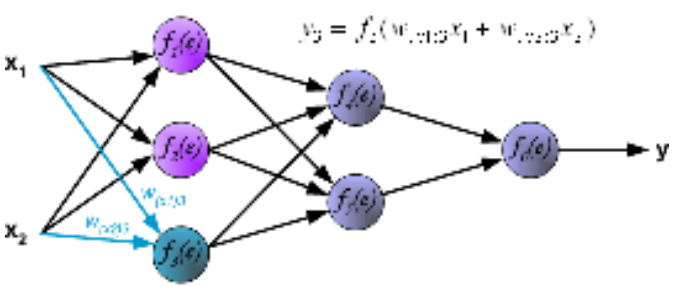






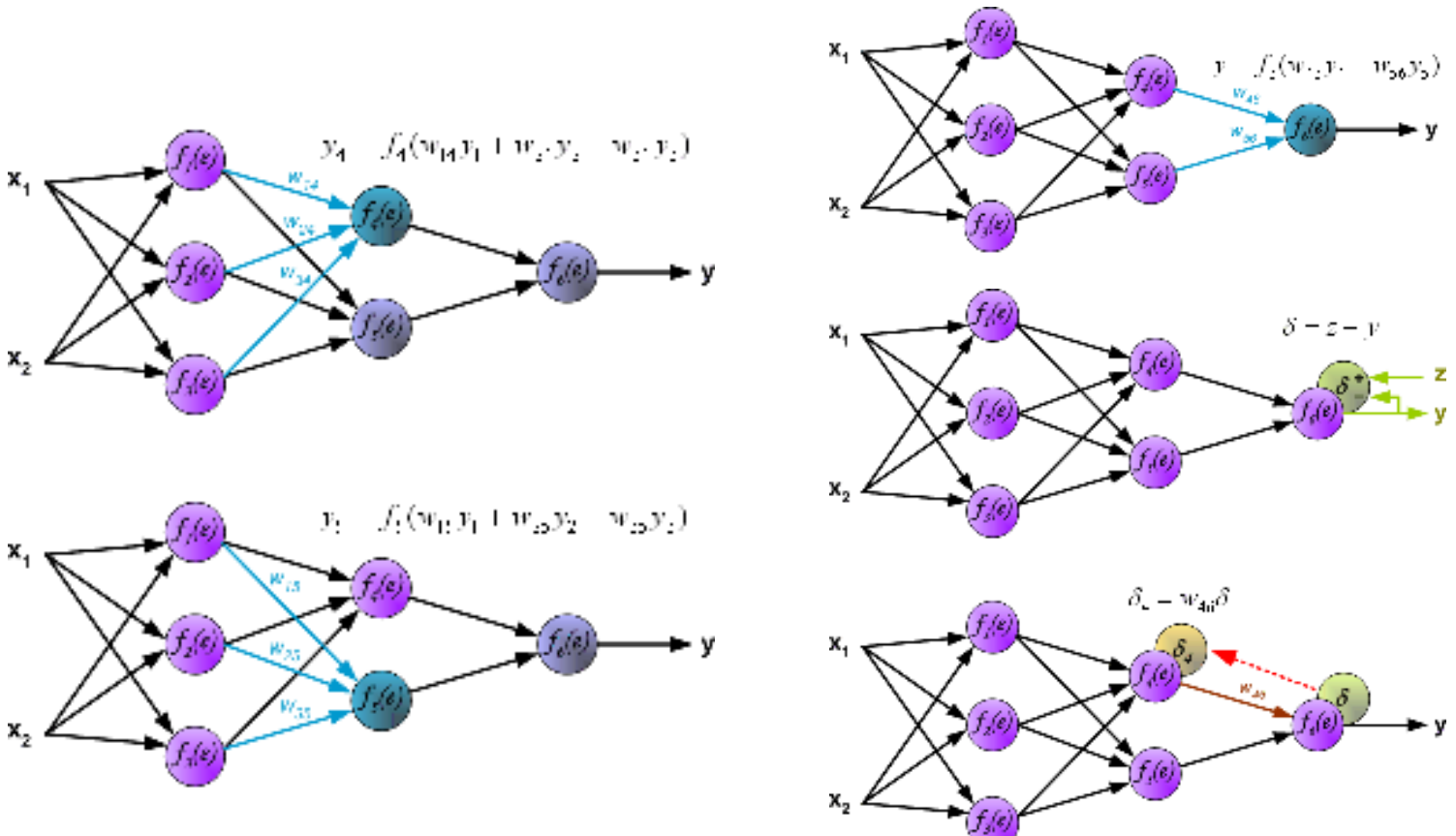






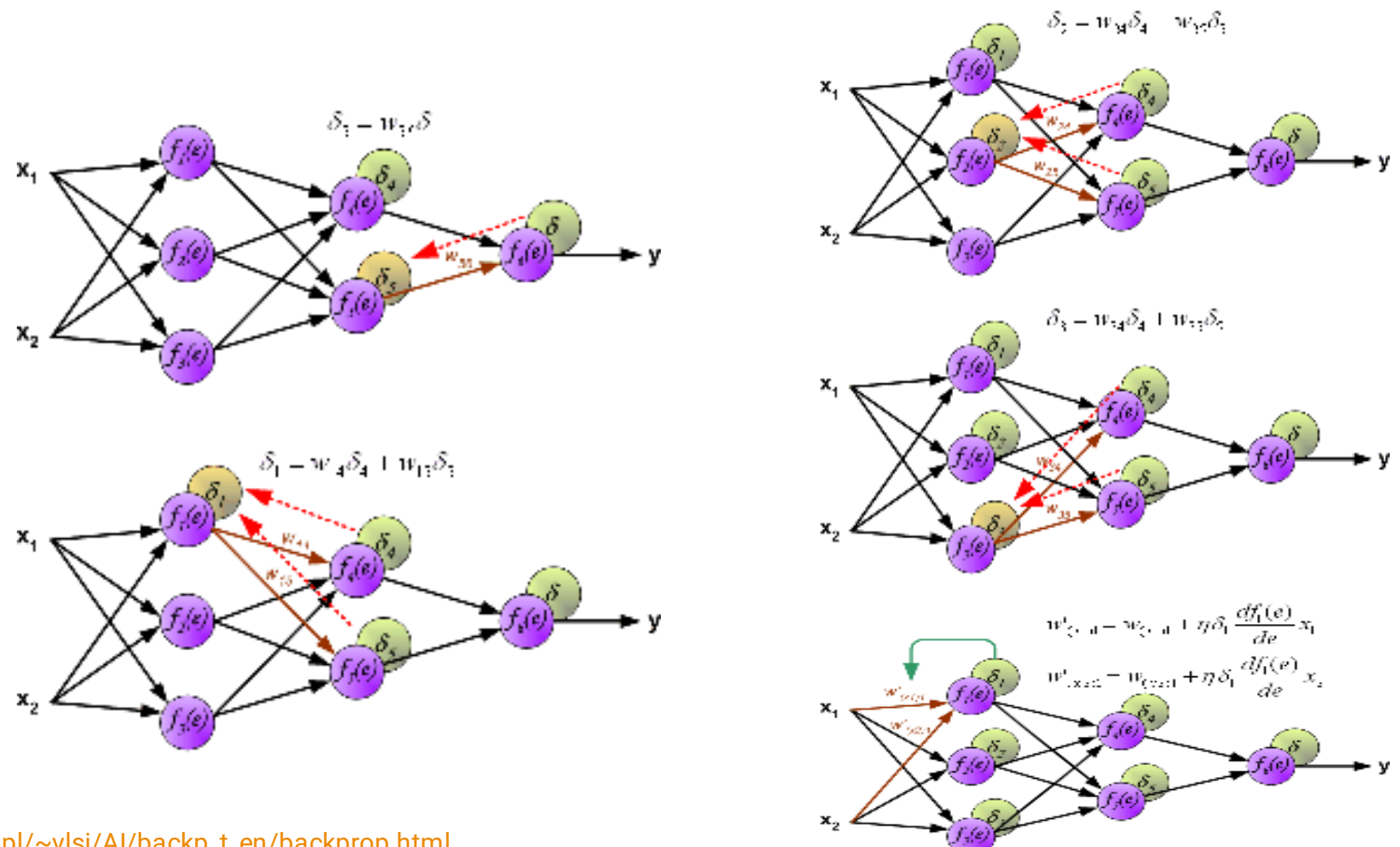
Source: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

BP神经网络概述



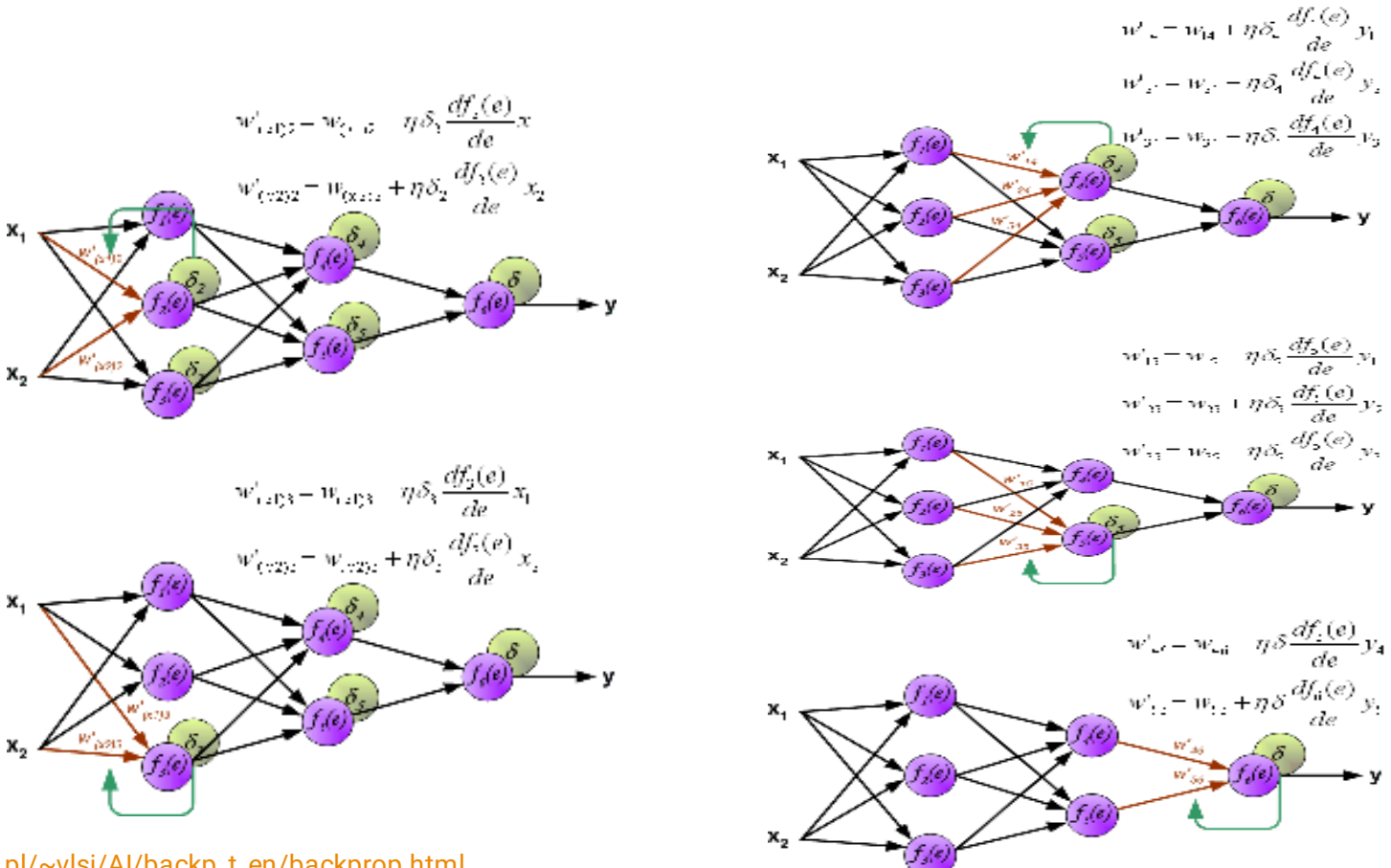
Source: http://galaxy.agh.edu.pl/~vlsi/AI/backp_t_en/backprop.html

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



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

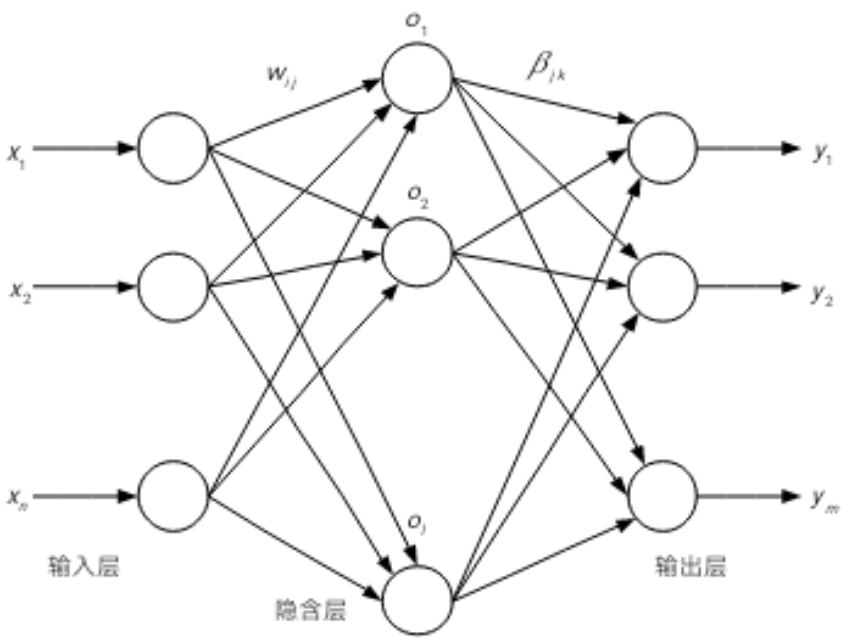
- **mapminmax**
  - Process matrices by mapping row minimum and maximum values to [-1 1]
  - [Y, PS] = mapminmax(X, YMIN, YMAX)
  - Y = mapminmax('apply', X, PS)
  - X = mapminmax('reverse', Y, PS)
- **newff**
  - Create feed-forward backpropagation network
  - net = newff(P, T, [S1 S2...S(N-l)], {TF1 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）

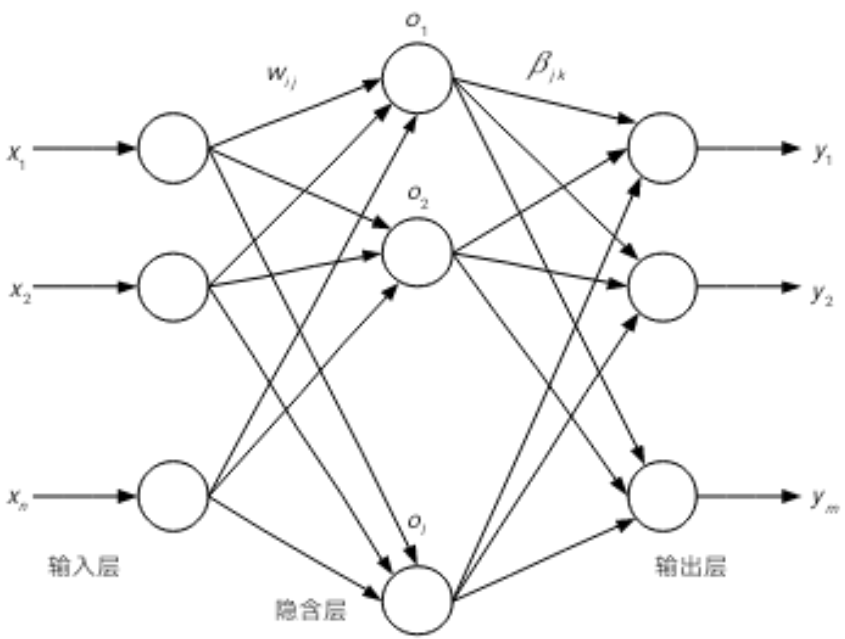
极限学习机原理概述

- The learning speed of feedforward neural networks is in general **far slower than required** and it has been a **major bottleneck** in their applications for past decades.
- Two key reasons behind may be:
  1) the slow **gradient-based** learning algorithms are extensively used to train neural networks.
  2) all the parameters of the networks are **tuned iteratively** by using such learning algorithms.



极限学习机原理概述

- Unlike these traditional implementations, **Extreme Learning Machine (ELM)** which **randomly** all the hidden nodes parameters of generalized **S**ingle-hidden **L**ayer **F**eedforward **N**etworks (**SLFN**s) and **analytically** determines the output weights of SLFNs.
- All the hidden node parameters are **independent** from the target functions or the training datasets. **All the parameters of ELMs can be analytically determined instead of being tuned.**
- In theory, this algorithm tends to provide the **good generalization performance** at extremely **fast learning speed**.

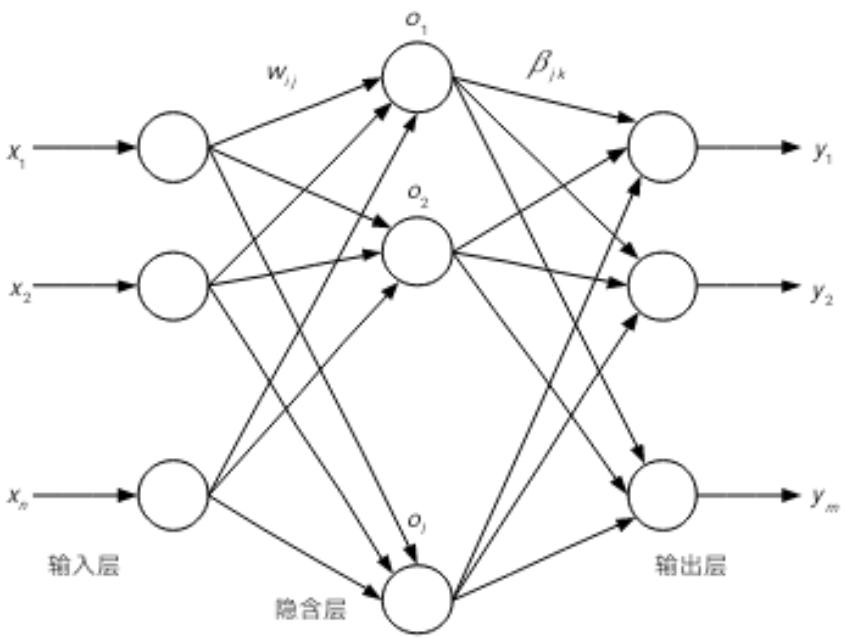


极限学习机原理概述

$$w=\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ & \cdots\cdots & & \\ w_{l1} & w_{l2} & \cdots & w_{ln} \end{bmatrix}_{l\times n} \qquad \beta=\begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \beta_{21} & \beta_{22} & \cdots & \beta_{2m} \\ & \cdots\cdots & & \\ \beta_{l1} & \beta_{l2} & \cdots & \beta_{lm} \end{bmatrix}_{l\times m} \qquad b=\begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_l \end{bmatrix}_{l\times 1}$$

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1Q} \\ x_{21} & x_{22} & \cdots & x_{2Q} \\ & \cdots\cdots & & \\ x_{n1} & x_{n2} & \cdots & x_{nQ} \end{bmatrix}_{n\times Q} \qquad Y=\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1Q} \\ y_{21} & y_{22} & \cdots & y_{2Q} \\ & \cdots\cdots & & \\ y_{m1} & y_{m2} & \cdots & y_{mQ} \end{bmatrix}_{m\times Q}$$

$$T=\begin{bmatrix} t_1, t_2, \ldots, t_Q \end{bmatrix}_{m\times Q}, t_j=\begin{bmatrix} t_{1j} \\ t_{2j} \\ \cdots \\ t_{mj} \end{bmatrix}_{m\times 1}=\begin{bmatrix} \sum_{i=1}^{l}\beta_{i1}g(w_i x_j+b_i) \\ \sum_{i=1}^{l}\beta_{i2}g(w_i x_j+b_i) \\ \cdots\cdots \\ \sum_{i=1}^{l}\beta_{im}g(w_i x_j+b_i) \end{bmatrix}_{m\times 1} \quad (j=1,2,\ldots,Q)$$

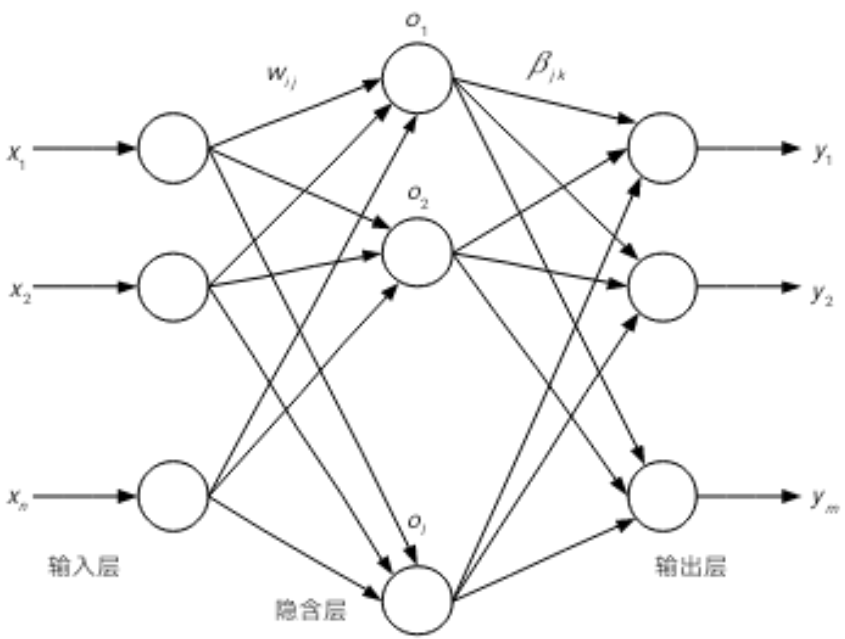


极限学习机原理概述

$$T = \left[t_1, t_2, \ldots, t_Q \right]_{m \times Q}, t_j = \begin{bmatrix} t_{1j} \\ t_{2j} \\ \cdots \\ t_{mj} \end{bmatrix}_{m \times 1} = \begin{bmatrix} \sum_{i=1}^{l} \beta_{i1} g(w_i x_j + b_i) \\ \sum_{i=1}^{l} \beta_{i2} g(w_i x_j + b_i) \\ \cdots\cdots \\ \sum_{i=1}^{l} \beta_{im} g(w_i x_j + b_i) \end{bmatrix}_{m \times 1} \quad (j = 1, 2, \ldots, Q)$$

$$H\beta = T'$$

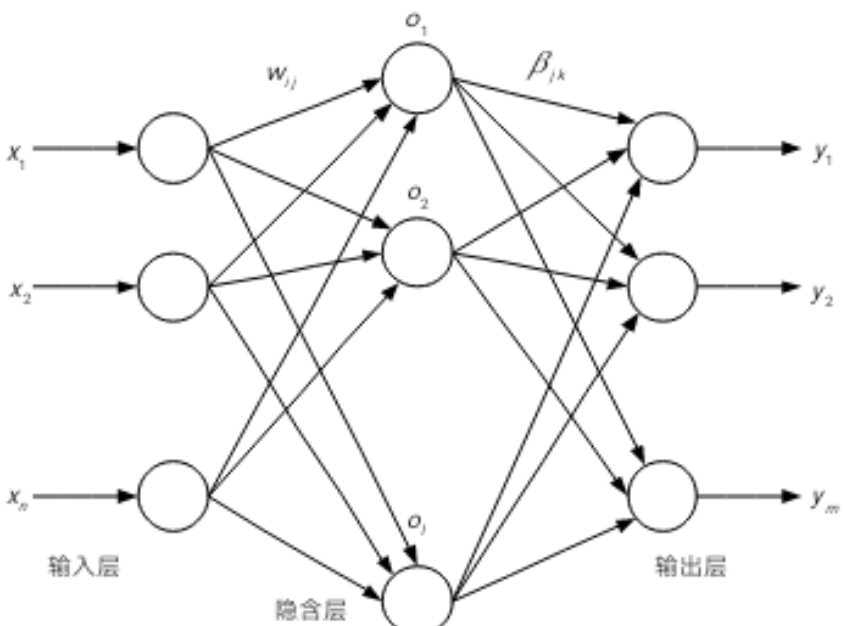
$$H(w_1, w_2, \ldots, w_l, b_1, b_2, \ldots, b_l, x_1, x_2, \ldots, x_Q) = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & g(w_2 \cdot x_1 + b_2) & \mathrm{g}(w_l \cdot x_1 + b_l) \\ g(w_1 \cdot x_2 + b_1) & g(w_2 \cdot x_2 + b_2) & g(w_l \cdot x_2 + b_l) \\ \cdots\cdots & & \\ g(w_1 \cdot x_Q + b_1) & g(w_2 \cdot x_Q + b_2) & g(w_l \cdot x_Q + b_l) \end{bmatrix}_{Q \times l}$$



极限学习机原理概述

Theorem 2.1. *Given a standard SLFN with N hidden nodes and activation function* $g: R \to R$ *which is infinitely differentiable in any interval, for N arbitrary distinct samples* $(\mathbf{x}_i, \mathbf{t}_i)$, *where* $\mathbf{x}_i \in \mathbf{R}^n$ *and* $\mathbf{t}_i \in \mathbf{R}^m$, *for any* $\mathbf{w}_i$ *and* b_i *randomly chosen from any intervals of* $\mathbf{R}^n$ *and* $\mathbf{R}$, *respectively, according to any continuous probability distribution, then with probability one, the hidden layer output matrix* $\mathbf{H}$ *of the SLFN is invertible and* $\|\mathbf{H}\beta - \mathbf{T}\| = 0$.

Theorem 2.2. *Given any small positive value* $\varepsilon > 0$ *and activation function* $g: R \to R$ *which is infinitely differentiable in any interval, there exists* $\tilde{N} \leqslant N$ *such that for N arbitrary distinct samples* $(\mathbf{x}_i, \mathbf{t}_i)$, *where* $\mathbf{x}_i \in \mathbf{R}^n$ *and* $\mathbf{t}_i \in \mathbf{R}^m$, *for any* $\mathbf{w}_i$ *and* b_i *randomly chosen from any intervals of* $\mathbf{R}^n$ *and* $\mathbf{R}$, *respectively, according to any continuous probability distribution, then with probability one,* $\|\mathbf{H}_{N\times\tilde{N}}\beta_{\tilde{N}\times m} - \mathbf{T}_{N\times m}\| < \varepsilon$.



极限学习机原理概述

Algorithm ELM: Given a training set $\aleph = \{(\mathbf{x}_i, \mathbf{t}_i) | \mathbf{x}_i \in \mathbf{R}^n, \mathbf{t}_i \in \mathbf{R}^m, i = 1, \ldots, N\}$, activation function $g(x)$, and hidden node number $\tilde{N}$,

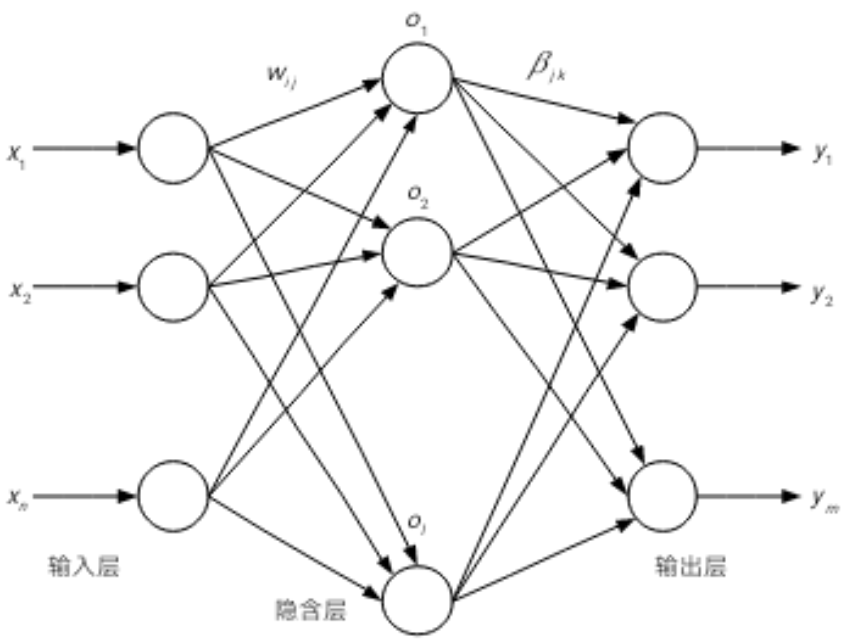
Step 1: Randomly assign input weight $\mathbf{w}_i$ and bias b_i, $i = 1, \ldots, \tilde{N}$.

Step 2: Calculate the hidden layer output matrix $\mathbf{H}$.

Step 3: Calculate the output weight β

$$\beta = \mathbf{H}^{\dagger}\mathbf{T}, \tag{16}$$

where $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_N]^{\mathrm{T}}$.



极限学习机原理概述

- Compared BP Algorithm and SVM , ELM has several salient features:
  - **Ease of use.** No parameters need to be manually tuned except predefined network architecture.
  - **Faster learning speed.** Most training can be completed in *milliseconds, seconds,* and *minutes*.
  - **Higher generalization performance.** It could obtain better generalization performance than BP in most cases, and reach generalization performance similar to or better than SVM.
  - **Suitable for almost all nonlinear activation functions.**. Almost all piecewise continuous (including discontinuous, differential, non-differential functions) can be used as activation functions.
  - **Suitable for fully complex activation functions**. Fully complex functions can also be used as activation functions in ELM.

支持向量机分类原理概述

- **Support vector machines** (**SVMs**) are a set of related **supervised learning** methods that analyze data and recognize patterns, used for classification and regression analysis.
- The original SVM algorithm was invented by **Vladimir Vapnik** and the current standard incarnation (soft margin) was proposed by Corinna Cortes and Vladimir Vapnik.
- A support vector machine constructs a **hyperplane** or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression, or other tasks.
- a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class, since in general the larger the margin the lower the generalization error of the classifier.

支持向量机分类原理概述

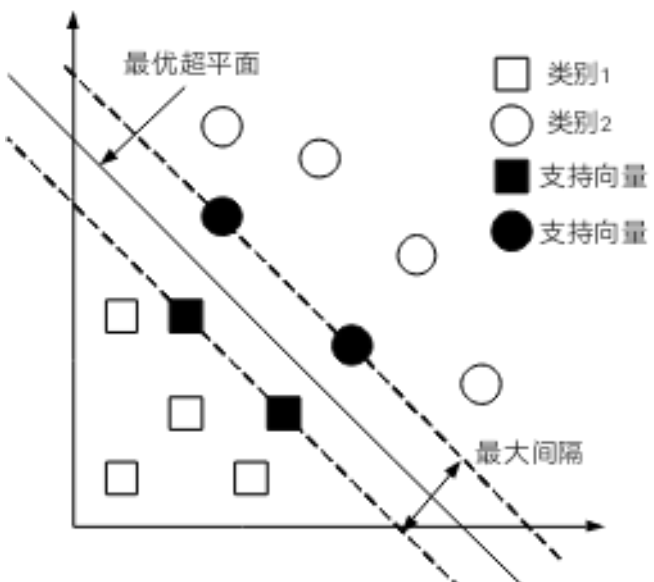
- We want to find the **maximum-margin hyperplane** that divides the points having $y_i = 1$ from those having $y_i = -1$. Any hyperplane can be written as the set of points satisfying

$$w \cdot x + b = 0$$

- We want to choose the w and b to maximize the margin, or distance between the parallel hyperplanes that are as far apart as possible **while still separating the data.**

$$\min \quad \frac{\|w\|^2}{2}$$

$$s.t \quad y_i (w \cdot x_i + b) \geq 1, i = 1, 2, \ldots, I$$



支持向量机分类原理概述

- **Primal form**

$$\min \quad \frac{\|w\|^2}{2}$$

$$s.t \quad y_i(w \cdot x_i + b) \geq 1, i = 1, 2, \ldots, l$$

- **Dual form**

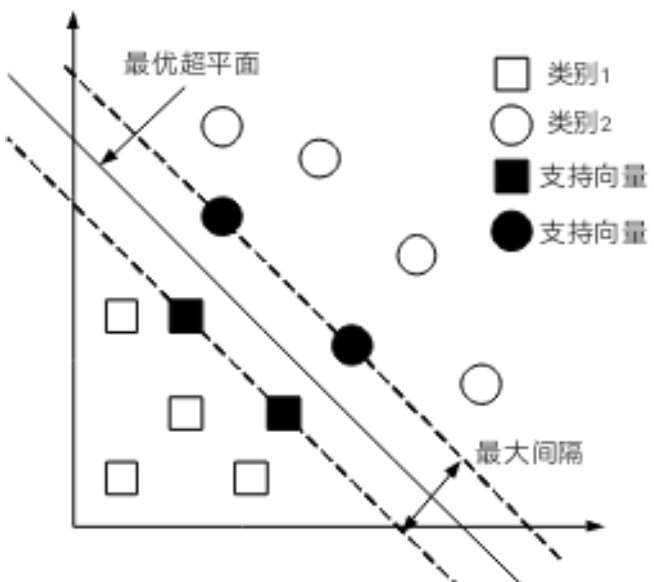
$$\max \quad Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

$$s.t \quad \sum_{i=1}^{l} \alpha_i y_i = 0, \alpha_i \geq 0$$

- **Decision function**

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} \alpha_i^* y_i (x \cdot x_i) + b^*\right)$$

$$w^* = \sum_{i=1}^{l} \alpha_i^* x_i y_i \qquad b^* = -\frac{1}{2} w^* \cdot (x_r + x_s)$$



支持向量机分类原理概述

- **Primal form**

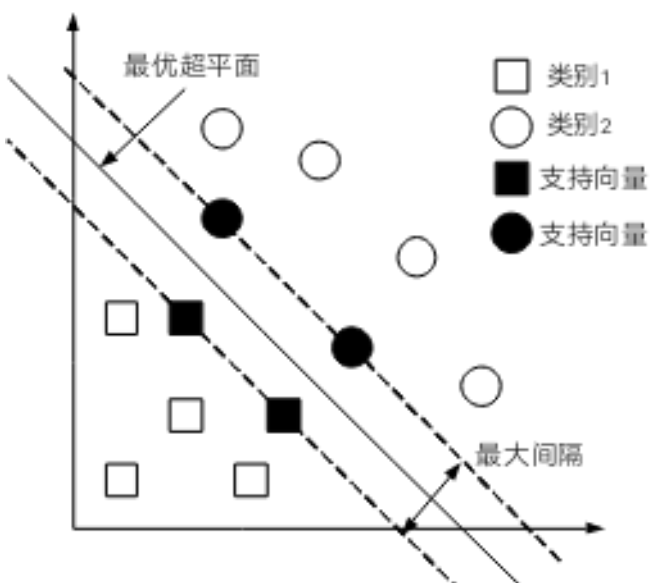
$$\min \quad \frac{\|w\|^2}{2} + C\sum_{i=1}^{l}\xi_i$$

$$s.t \quad \begin{cases} y_i(w\cdot x_i + b) \geq 1-\xi_i \\ \xi_i > 0 \end{cases} \quad i = 1, 2, \ldots, l$$

- **Dual form**

$$\max \quad Q(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j)$$

$$s.t \quad \begin{cases} \sum_{i=1}^{l}\alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases} \quad i = 1, 2, \ldots, l$$



支持向量机分类原理概述

- **Dual form**

$$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$$

$$\max \ Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$s.t \begin{cases} \sum_{i=1}^{l} \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C \end{cases}, i = 1, 2, \ldots, l$$

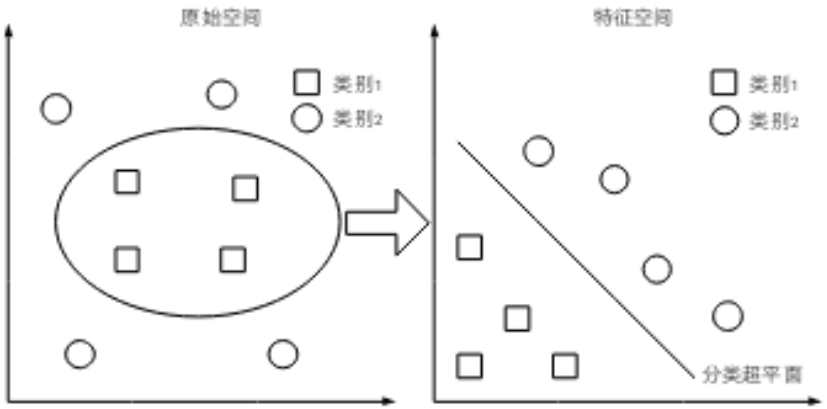
- **Decision function**

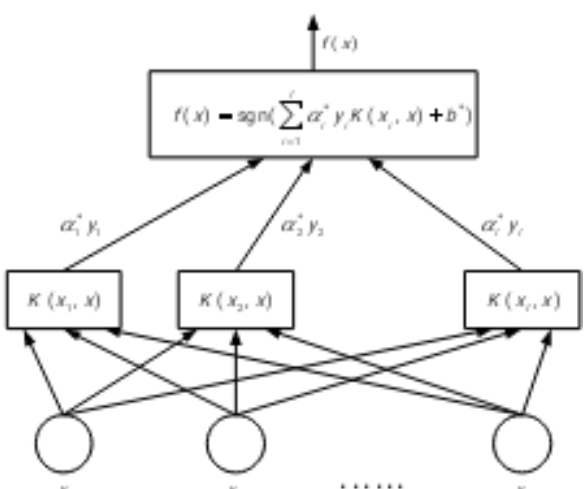
$$f(x) = \mathrm{sgn}(w^* \cdot \Phi(x) + b^*)$$

$$= \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i^* y_i \Phi(x_i) \cdot \Phi(x) + b^*\right)$$

$$= \mathrm{sgn}\left(\sum_{i=1}^{l} \alpha_i^* y_i K(x_i, x) + b^*\right)$$

$$w^* = \sum_{i=1}^{l} \alpha_i^* y_i \Phi(x_i)$$



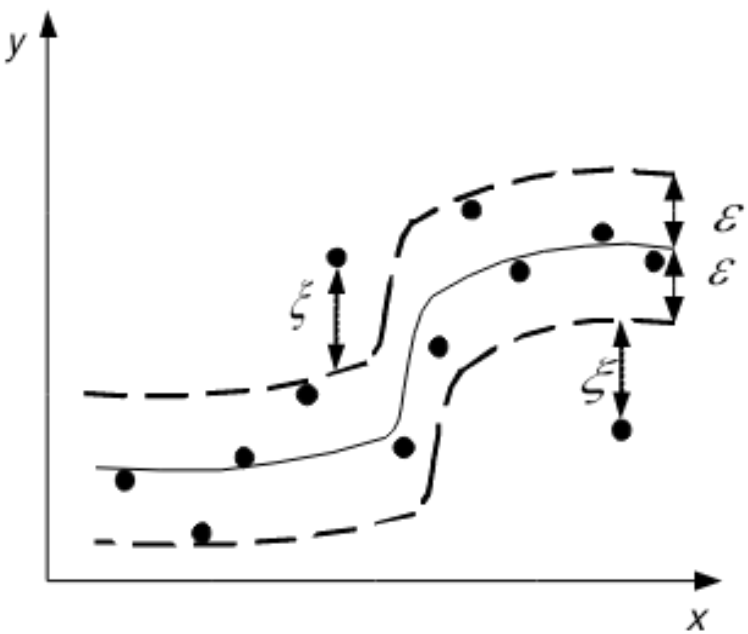


支持向量机回归原理概述

- 为了利用SVM解决回归拟合方面的问题，Vapnik等人在SVM分类的基础上引入了**不敏感损失函数**，从而得到了回归型支持向量机（Support Vector Machine for Regression，SVR）。
- SVM应用于回归拟合分析时，其基本思想不再是寻找一个最优分类面使得两类样本分开，而是寻找一个最优分类面**使得所有训练样本离该最优分类面的误差最小**。

$$\min \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$s.t \ \begin{cases} y_i - w\cdot\Phi(x_i) - b \le \varepsilon + \xi_i \\ -y_i + w\cdot\Phi(x_i) + b \le \varepsilon + \xi_i^* \quad i = 1, 2, \ldots, l \\ \xi_i \ge 0, \xi_i^* \ge 0 \end{cases}$$

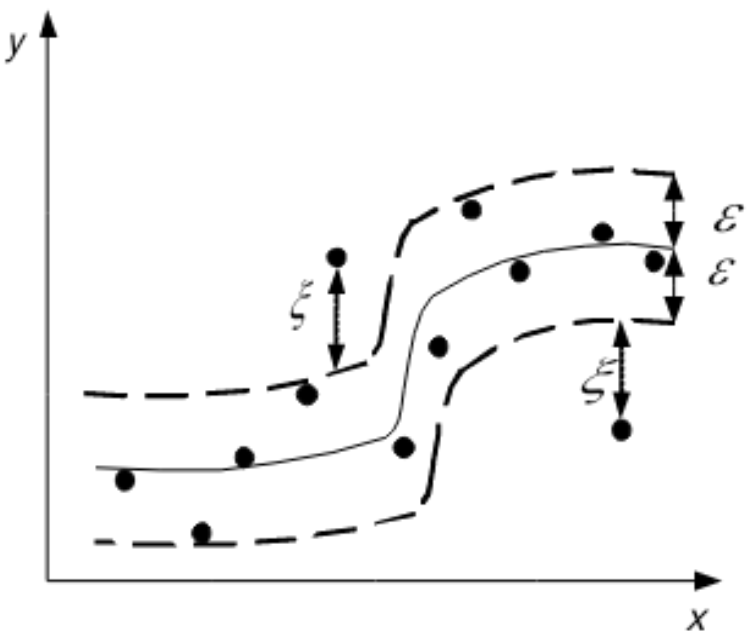


支持向量机回归原理概述

- **Primal form**

$$\min \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)$$

$$s.t \ \begin{cases} y_i - w\cdot\Phi(x_i) - b \le \varepsilon + \xi_i \\ -y_i + w\cdot\Phi(x_i) + b \le \varepsilon + \xi_i^* \quad i = 1, 2, \ldots, l \\ \xi_i \ge 0, \xi_i^* \ge 0 \end{cases}$$



- **Dual form**

$$\max_{\alpha,\alpha^*}\left[-\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i, x_j) - \sum_{i=1}^{l}(\alpha_i + \alpha_i^*)\varepsilon + \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)y_i\right]$$

$$s.t \ \begin{cases} \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \\ 0 \le \alpha_i \le C \\ 0 \le \alpha_i^* \le C \end{cases}$$

支持向量机回归原理概述

- **Decision function**

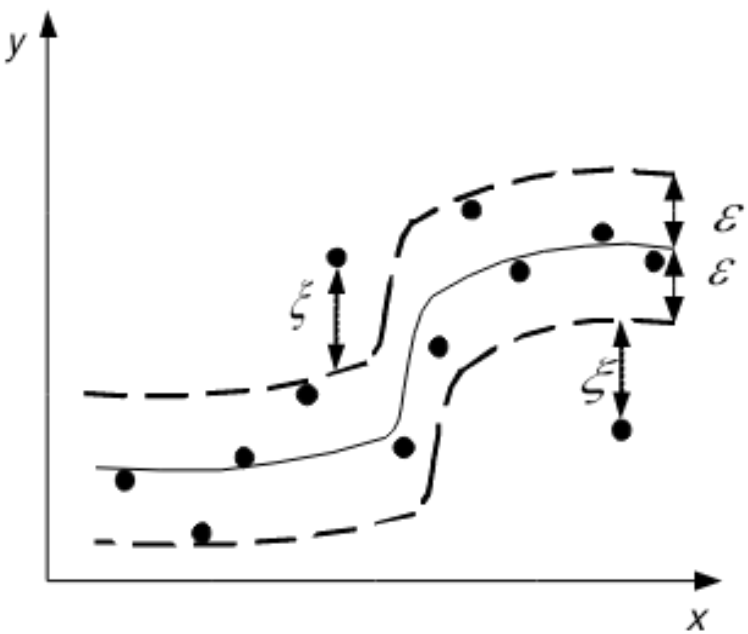
$$f(x) = w^* \cdot \Phi(x) + b^* = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \Phi(x_i) \cdot \Phi(x) + b^*$$

$$= \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(x_i, x) + b^*$$

$$w^* = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \Phi(x_i)$$

$$b^* = \frac{1}{N_{nsv}} \sum_{0<\alpha_i<C} \left[y_i - \sum_{x_i \in SV} (\alpha_i - \alpha_i^*) K(x_i, x_j) - \varepsilon \right]$$

$$+ \frac{1}{N_{nsv}} \sum_{0<\alpha_i<C} \left[y_i - \sum_{x_j \in SV} (\alpha_j - \alpha_j^*) K(x_i, x_j) + \varepsilon \right]$$

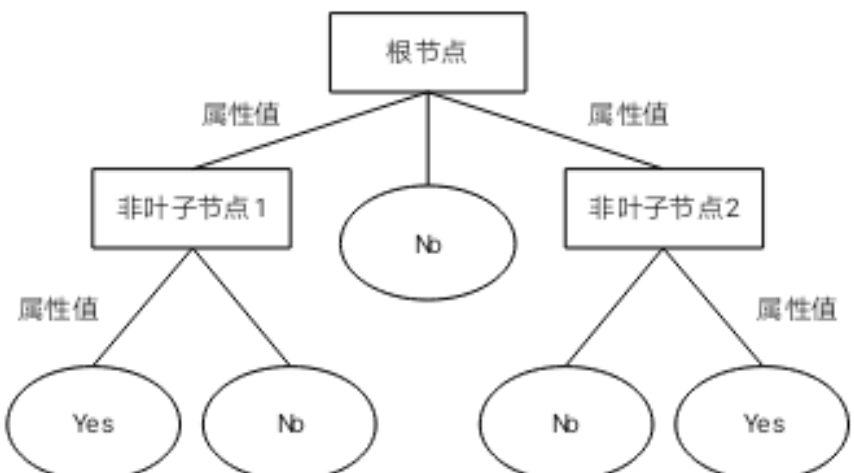


多分类SVM

- Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements.
- The dominating approach for doing so is to **reduce** the single multiclass problem into multiple **binary classification** problems.
- **one-against-all**
  - Classification of new instances for one-against-all case is done by a winner-takes-all strategy, in which the classifier with the highest output function assigns the class.
- **one-against-one**
  - For the one-against-one approach, classification is done by a max-wins voting strategy, in which every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally the class with most votes determines the instance classification.

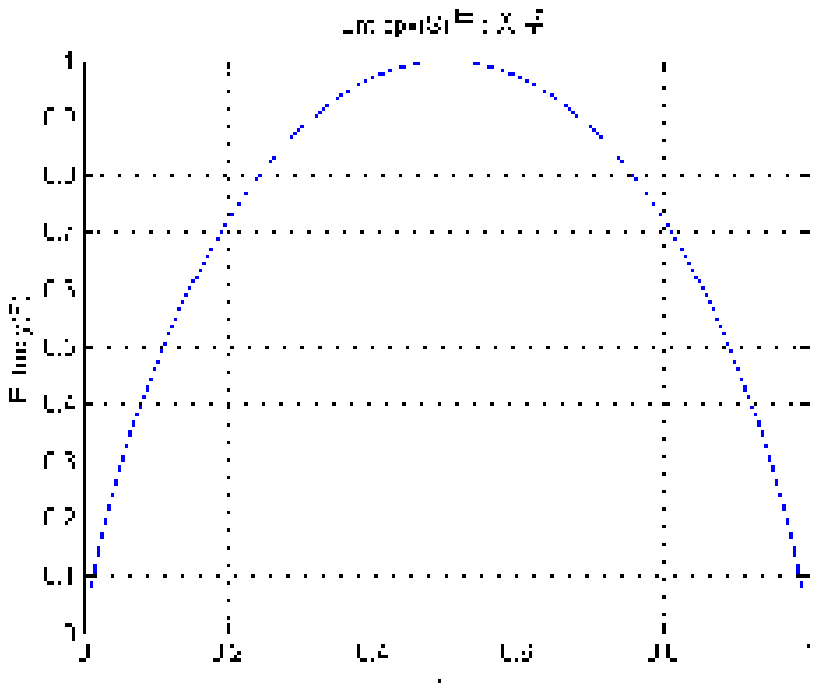
决策树原理概述

- 决策树通过把样本实例从根节点排列到某个叶子节点来对其进行分类。树上的**每个非叶子节点代表对一个属性取值的测试**，其分支就代表测试的每个结果；而**树上的每个叶子节点均代表一个分类的类别**，树的最高层节点是根节点。
- 简单地说，决策树就是一个类似流程图的树形结构，采用**自顶向下的递归方式**，从树的根节点开始，在它的内部节点上进行属性值的测试比较，然后按照给定实例的属性值确定对应的分支，最后在决策树的叶子节点得到结论。这个过程在**以新的节点为根的子树上重复**。



决策树原理概述

- ID3算法



决策树原理概述

- ID3算法

[illegible] $Gain(S, A)$ [illegible]

$$Gain(S, A) = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

[illegible] $V(A)$ [illegible] S_v [illegible]

[illegible] ID3 [illegible] Examples [illegible] Attribute list [illegible]

（1） [illegible] N；

（2） [illegible]

（3） [illegible] Attribute list [illegible] N [illegible]

（4） [illegible] Attribute list [illegible] Attribute [illegible] N；

（5） [illegible] Attribute [illegible] S_v [illegible] Examples [illegible] Attribute [illegible]

（6） [illegible] S_v [illegible] Examples [illegible] Attribute [illegible] Attribute list [illegible]

决策树原理概述

- **C4.5算法**

针对 ID3 算法存在的一些缺点，许多学者包括 Quinlan 都做了大量的研究。C4.5 算法便是 ID3 算法的改进算法，其相比于 ID3 改进的地方主要有：

（1） 用信息增益率（gain ratio）来选择属性。

信息增益率是用信息增益和分裂信息量（split information）共同定义的，如下所示：

$$GainRatio(S,A)=\frac{Gain(S,A)}{SplitInformation(S,A)}$$

其中，分裂信息量的定义为：

$$SplitInformation(S,A)=-\sum_{i=1}^{c}\frac{|S_i|}{|S|}\log_2\frac{|S_i|}{|S|}$$

采用信息增益率作为选择分支属性的标准，克服了 ID3 算法中信息增益选择属性时偏向选择取值多的属性的不足。

（2） 树的剪枝。

剪枝方法是用来处理过拟合问题而提出的。剪枝一般分两种方法：先剪枝和后剪枝。

先剪枝方法通过提前停止树的构造，比如决定在某个节点不再分裂，而对树进行剪枝。一旦停止，该节点就变为叶子节点，该叶子节点可以取它所包含的子集中类别最多的类作为节点的类别。

后剪枝的基本思路是对完全成长的树进行剪枝，通过删除节点的分支，并用叶子节点进行替换，叶子节点一般用子集中最频繁的类别进行标记。

C4.5 算法采用悲观剪枝法（Pessimistic Pruning）是 Quinlan 在 1987 年提出的，属于后剪枝方法的一种。它使用训练集生成决策树，并用训练集进行剪枝，不需要独立的剪枝集。悲观剪枝法的基本思路是：若使用叶子节点代替原来的子树后，误差率能够下降，则就用该叶子节点代替原来的子树。

决策树原理概述

优点

- 决策树易于理解和实现。人们在通过解释后都有能力去理解决策树所表达的意义。
- 对于决策树，数据的准备往往是简单或者是不必要的。其他的技术往往要求先把数据归一化，比如去掉多余的或者空白的属性。
- 能够同时处理数据型和常规型属性。其他的技术往往要求数据属性的单一。
- 是一个白盒模型。如果给定一个观察的模型，那么根据所产生的决策树很容易推出相应的逻辑表达式。

缺点

- 对于各类别样本数量不一致的数据，在决策树当中信息增益的结果偏向于那些具有更多数值的特征。

随机森林概述

- **Bootstrap抽样**

设集合 S 中含有 n 个不同的样本 $\{x_1, x_2, ..., x_n\}$，若每次有放回地从集合 S 中抽取一个样本，一共抽取 n 次，形成新的集合 S^*，则集合 S^* 中不包含某个样本 $x_i\,(i=1,2,...,n)$ 的概率为

$$p = \left(1 - \frac{1}{n}\right)^n$$

当 $n \to \infty$ 时，有

$$\lim_{n\to\infty} p = \lim_{n\to\infty}\left(1 - \frac{1}{n}\right)^n = e^{-1} \approx 0.368$$

因此，虽然新集合 S^* 的样本总数与原集合 S 的样本总数相等（都为 n），但是新集合 S^* 中可能包含了重复的样本（有放回抽取）。若除去重复的样本，新集合 S^* 中仅包含了原集合 S 中约 1 - 0.368 = 63.2%的样本。

随机森林概述

- **Bagging算法**

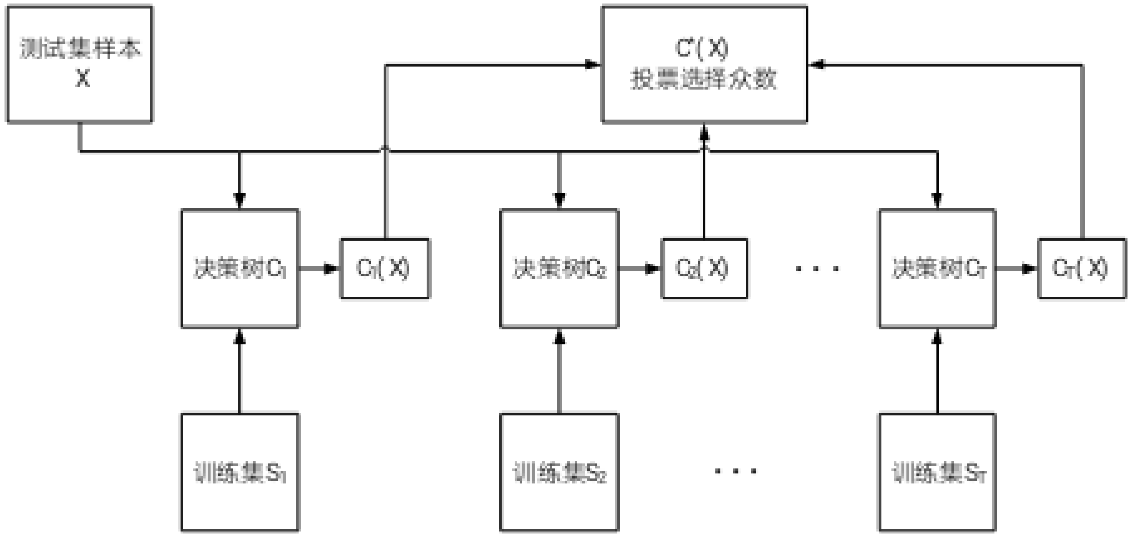
Bagging（Bootstrap aggregating）[illegible]

（1） [illegible]

（2） [illegible]

（3） [illegible]

（4） [illegible]



随机森林概述

- **随机森林算法**

随机森林算法与 Bagging 算法类似，均是基于 Bootstrap 方法重采样，产生多个训练集。不同的是，随机森林算法在构建决策树的时候，采用了随机选取分裂属性集的方法。详细的随机森林算法流程如下所示：

不妨设样本的属性个数为 M ， m 为大于零且小于 M 的整数。

（1） 利用 Bootstrap 方法重采样，随机产生 T 个训练集 $S_1, S_2, \ldots, S_T$ ；

（2） 利用每个训练集，生成对应的决策树 $C_1, C_2, \ldots, C_T$ ；在每个非叶子节点（内部节点）上选择属性前，从 M 个属性中随机抽取 m 个属性作为当前节点的分裂属性集，并以这 m 个属性中最好的分裂方式对该节点进行分裂（一般而言，在整个森林的生长过程中， m 的值维持不变）。

（3） 每棵树都完整成长，而不进行剪枝。

（4） 对于测试集样本 X ，利用每个决策树进行测试，得到对应的类别 $C_1(X), C_2(X), \ldots, C_T(X)$ ；

（5） 采用投票的方法，将 T 个决策树中输出最多的类别作为测试集样本 X 所属的类别。

遗传算法概述

- 遗传算法（Genetic Algorithm，GA）是一种进化算法，其基本原理是仿效生物界中的“物竞天择、适者生存”的演化法则，它最初由美国Michigan大学的J. Holland教授于1967年提出。
- 遗传算法是从代表问题可能潜在的解集的一个**种群（population）**开始的，而一个种群则由经过基因（gene）编码的一定数目的**个体(individual)**组成。因此，第一步需要实现从表现型到基因型的映射即编码工作。初代种群产生之后，按照适者生存和优胜劣汰的原理，逐代（generation）演化产生出越来越好的近似解，在每一代，根据问题域中个体的**适应度（fitness）**大小选择个体，并借助于自然遗传学的遗传算子（genetic operators）进行组合交叉和变异，产生出代表新的解集的种群。这个过程将导致种群像自然进化一样，后生代种群比前代更加适应于环境，末代种群中的最优个体经过解码（decoding），可以作为问题近似最优解。

遗传算法概述

- 遗传算法有三个基本操作：**选择（Selection）、交叉（Crossover）和变异（Mutation）。**
- （1）**选择**。选择的目的是为了从当前群体中选出优良的个体，使它们有机会作为父代为下一代繁衍子孙。根据各个个体的适应度值，按照一定的规则或方法从上一代群体中选择出一些优良的个体遗传到下一代种群中。选择的依据是适应性强的个体为下一代贡献一个或多个后代的概率大。
- （2）**交叉**。通过交叉操作可以得到新一代个体，新个体组合了父辈个体的特性。将群体中的各个个体随机搭配成对，对每一个个体，以交叉概率交换它们之间的部分染色体。
- （3）**变异**。对种群中的每一个个体，以变异概率改变某一个或多个基因座上的基因值为其他的等位基因。同生物界中一样，变异发生的概率很低，变异为新个体的产生提供了机会。

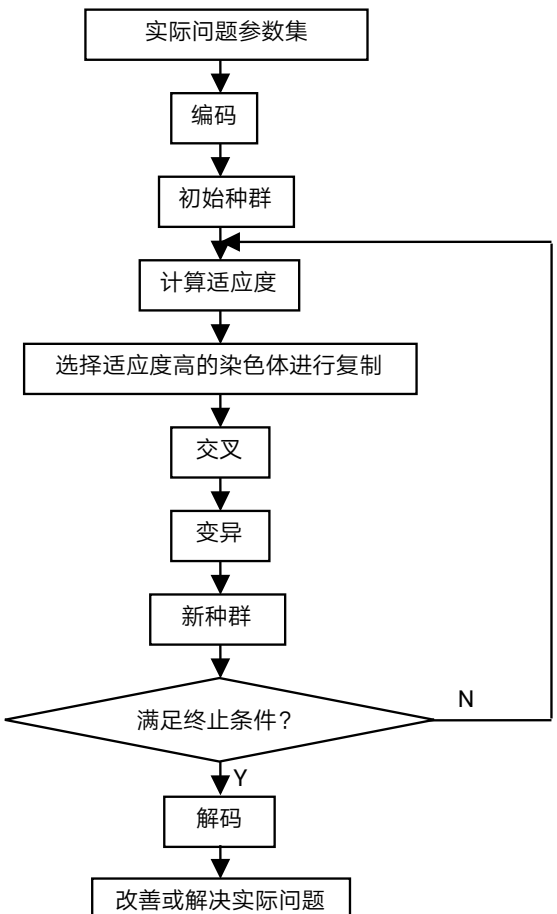
遗传算法概述

•遗传算法的基本步骤：

1)**编码**：GA在进行搜索之前先将解空间的解数据表示成遗传空间的基因型串结构数据，这些串结构数据的不同组合便构成了不同的点。

2)**初始群体的生成**：随机产生N个初始串结构数据，每个串结构数据称为一个个体，N个个体构成了一个群体。GA以这N个串结构数据作为初始点开始进化。

3)**适应度评估**：适应度表明个体或解的优劣性。不同的问题，适应性函数的定义方式也不同。

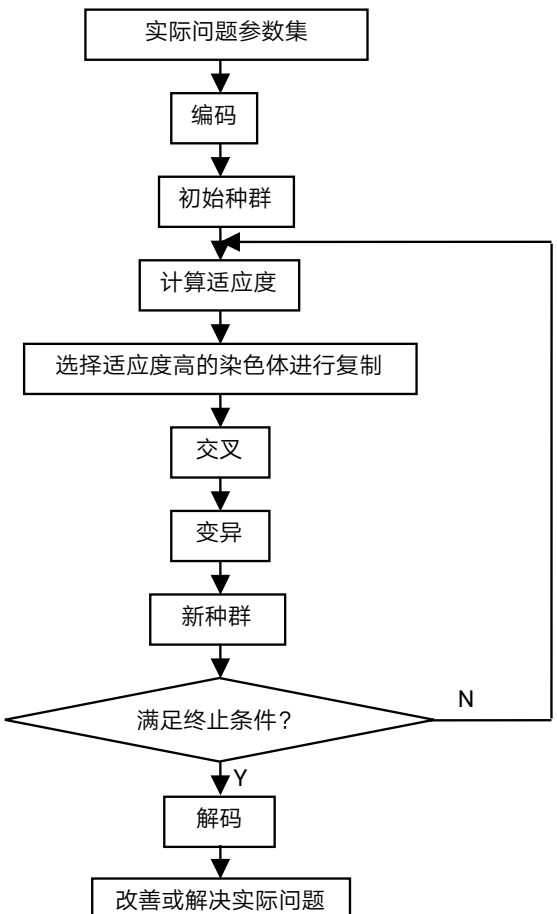


遗传算法概述

4)**选择**：选择的目的是为了从当前群体中选出优良的个体，使它们有机会作为父代为下一代繁殖子孙。遗传算法通过选择过程体现这一思想，进行选择的原则是适应性强的个体为下一代贡献一个或多个后代的概率大。选择体现了达尔文的适者生存原则。

5)**交叉**：交叉操作是遗传算法中最主要的遗传操作。通过交叉操作可以得到新一代个体，新个体组合了其父辈个体的特性。交叉体现了信息交换的思想。

6)**变异**：变异首先在群体中随机选择一个个体，对于选中的个体以一定的概率随机地改变串结构数据中某个串的值。同生物界一样， GA中变异发生的概率很低，通常取值很小。



遗传算法工具箱

- MATLAB内嵌遗传算法工具箱: gadst
- Sheffield大学遗传算法工具箱: gatbx
- **北卡罗来纳大学遗传算法工具箱: gaot**

主成分分析概述

- 主成分分析（Principle Component Analysis, PCA）
- 用少数的若干新变量（原变量的线性组合）替代原变量，新变量要尽可能多地反映原变量的数据信息，同时，新变量之间相互正交，可以消除原变量中相互重叠的信息。
- 设样本的标准化输入变量矩阵为：

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots\cdots & x_{1k} \\ x_{21} & x_{22} & \cdots\cdots & x_{2k} \\ & & \cdots\cdots & \\ x_{n1} & x_{n2} & \cdots\cdots & x_{nk} \end{bmatrix}$$

- 要求构造一个变量 P_1 满足：$P_1 = X t_1$ ， $\|t_1\| = 1$
- 同时，使得变量 P_1 能携带标准化输入变量矩阵 $X_{n\times k}$ 的信息。

主成分分析概述

• 从概率统计观点可知，变量的方差越大，该变量包含的信息越多。因此，上述问题可以转化为要求变量 P_1 的方差最大。P_1 的方差为

$$Var(P_1)=\frac{1}{n}\|P_1\|^2=\frac{1}{n}t_1'X'Xt_1=t_1'Vt_1 \qquad V=\frac{1}{n}X'X$$

• 构造拉格朗日函数

$$L=t_1'Vt_1-\lambda_1(t_1't_1-1)$$

其中，λ_1 为拉格朗日系数。分别计算 L 对 λ_1 和 t_1 的偏导数，并令其为零，则有：

$$\begin{cases}\dfrac{\partial L}{\partial t_1}=2Vt_1-2\lambda_1t_1=0\\ \dfrac{\partial L}{\partial \lambda_1}=-(t_1't_1-1)=0\end{cases} \qquad Vt_1=\lambda_1t_1$$

由此可知，t_1 是 V 的一个标准化特征向量，λ_1 为其对应的特征值。此时，$Var(P_1)=t_1'Vt_1=t_1'\lambda_1t_1=\lambda_1t_1't_1=\lambda_1$

主成分分析概述

- 也就是说，所要求的 t_1是矩阵 V的最大特征值 λ_1所对应的标准化特征向量。此时所对应的构造变量 $P_1 = Xt_1$称为第一主成分。
- 以此类推，可以求出 X的第 m个主成分 $P_m = Xt_m$
- 由上面的分析可知，第一主成分携带的信息最多，第二主成分次之，……。
- 前 m个主成分携带的信息总和为

$$\sum_{i=1}^{m} Var(P_i) = \sum_{i=1}^{m} \lambda_i$$

主成分回归分析 = 主成分分析 + 多元线性回归分析

偏最小二乘法概述

- 偏最小二乘法（Partial Least Squares, PLS）
- PCA方法提取出的前若干个主成分携带了原输入变量矩阵的大部分信息，消除了相互重叠部分的信息。并没有考虑主成分对输出变量的解释能力，**方差贡献率很小但对输出变量有很强解释能力的主成分将会被忽略掉**，这无疑会对校正模型的性能产生一定的影响。偏最小二乘法（PLS）可以很好地解决这个问题。
- PLS的基本思路是逐步回归，逐步分解输入变量矩阵和输出变量矩阵，并综合考虑提取的主成分对输入变量矩阵和输出变量矩阵的解释能力，直到满足性能要求为止。
- 设标准化的输入变量矩阵和输出变量矩阵分别为

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots\cdots & x_{1k} \\ x_{21} & x_{22} & \cdots\cdots & x_{2k} \\ & & \cdots\cdots & \\ x_{n1} & x_{n2} & \cdots\cdots & x_{nk} \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_n \end{bmatrix}$$

偏最小二乘法概述

- 要求构造变量 t_1和 u_1满足：

$$\begin{cases} t_1 = Xw_1 \ , \ \|w_1\| = 1 \\ u_1 = Yc_1 \ , \ \|c_1\| = 1 \end{cases}$$

- 同时，应满足以下三个条件：

  ① t_1 应尽可能大地携带输入变量矩阵 X 的信息；

  ② u_1 应尽可能大地携带输出变量矩阵 Y 的信息；

  ③ t_1 和 u_1 应具有尽可能大的相关程度。

- 由主成分分析原理可知，条件（1）和条件（2）等价于要求 t_1和 u_1的方差尽可能地大，即

$$\begin{cases} Var(t_1) \rightarrow \max \\ Var(u_1) \rightarrow \max \end{cases}$$

偏最小二乘法概述

- 根据典型相关分析，条件（3）可以转换为使得 t_1和 u_1的相关系数尽可能地大，即

$$r(t_1, u_1) \rightarrow \max$$

- 上述问题可以转化为计算 t_1和 u_1 的协方差的最大值，即

$$Cov(t_1, u_1) = t_1' u_1 = w_1' X' Y c_1 \rightarrow \max$$

$$st: \|w_1\| = 1, \|c_1\| = 1$$

- 构造拉格朗日函数 $L = w_1' X' Y c_1 - \lambda_1 (w_1' w_1 - 1) - \lambda_2 (c_1' c_1 - 1)$

$$\begin{cases} \frac{\partial L}{\partial w_1} = X' Y c_1 - 2\lambda_1 w_1 = 0 \\ \frac{\partial L}{\partial c_1} = w_1' X' Y - 2\lambda_2 c_1 = 0 \\ \frac{\partial L}{\partial \lambda_1} = -(w_1' w_1 - 1) = 0 \\ \frac{\partial L}{\partial \lambda_2} = -(c_1' c_1 - 1) = 0 \end{cases}$$

$$\theta = 2\lambda_1 = 2\lambda_2 = w_1' X' Y c_1 = Cov(t_1, u_1)$$

$$\begin{cases} X' Y c_1 = \theta w_1 \\ Y' X w_1 = \theta c_1 \end{cases} \qquad \begin{cases} X' Y Y' X w_1 = \theta^2 w_1 \\ Y' X X' Y c_1 = \theta^2 c_1 \end{cases}$$

偏最小二乘法概述

- 可见，w_1 是矩阵 $X^{'}YY^{'}X$ 的特征向量，对应的特征值为 θ^2。
- 使 t_1 和 u_1 的协方差最大的 w 是对应于矩阵 $X^{'}YY^{'}$ 最大特征值的单位特征向量。
- 同理，c_1 是矩阵 $Y^{'}XX^{'}Y$ 的特征向量，对应的特征值为 θ^2。
- 使 t_1 和 u_1 的协方差最大的 c 是对应于矩阵 $Y^{'}XX^{'}$ 最大特征值的单位特征向量。

$$\begin{cases} X = t_1 p_1^{'} + X^{*} \\ Y = t_1 r_1^{'} + Y^{*} \end{cases}$$

- 若回归方程的精度已经满足要求，则停止；否则，利用残差矩阵 X^{*} 和 Y^{*}，计算第二主成分，并重新建立回归方程。以此类推，直到回归方程的精度满足要求为止。

特征选择方法概述

- **Filter vs. Wrapper**
  - ✓ Filter无需利用学习模型的性能，即可进行特征选择，主要依赖一些评价准则，如：相关系数、互信息、信息熵等
  - ✓ Wrapper需要建立学习模型，通过模型的性能进行评价特征的优劣。
- **搜索法**
  - ✓ 随机搜索
  - ✓ 启发式搜索
- **正则化方法（L1范数、LASSO）**
- **…**

前向选择法vs.后向选择法概述

- **前向选择法**
  - ✓ 自下而上的选择方法，又称集合增加法
  - ✓ 特征集合初始化为一个空集
  - ✓ 每次向特征集合中添加一个输入变量，当新加入的变量致使模型性能更优时，则保留该输入变量，否则不保留。
- **后向选择法**
  - ✓ 自上而下的选择方法，又称集合缩减法
  - ✓ 特征集合初始化为全部的输入变量
  - ✓ 每次从特征集合中剔除一个输入变量，当剔除后致使模型性能更优时，则剔除该输入变量，否则保留该输入变量。
- **广义方法**
  - ✓ 一次增加或剔除多个变量
  - ✓ 区间法

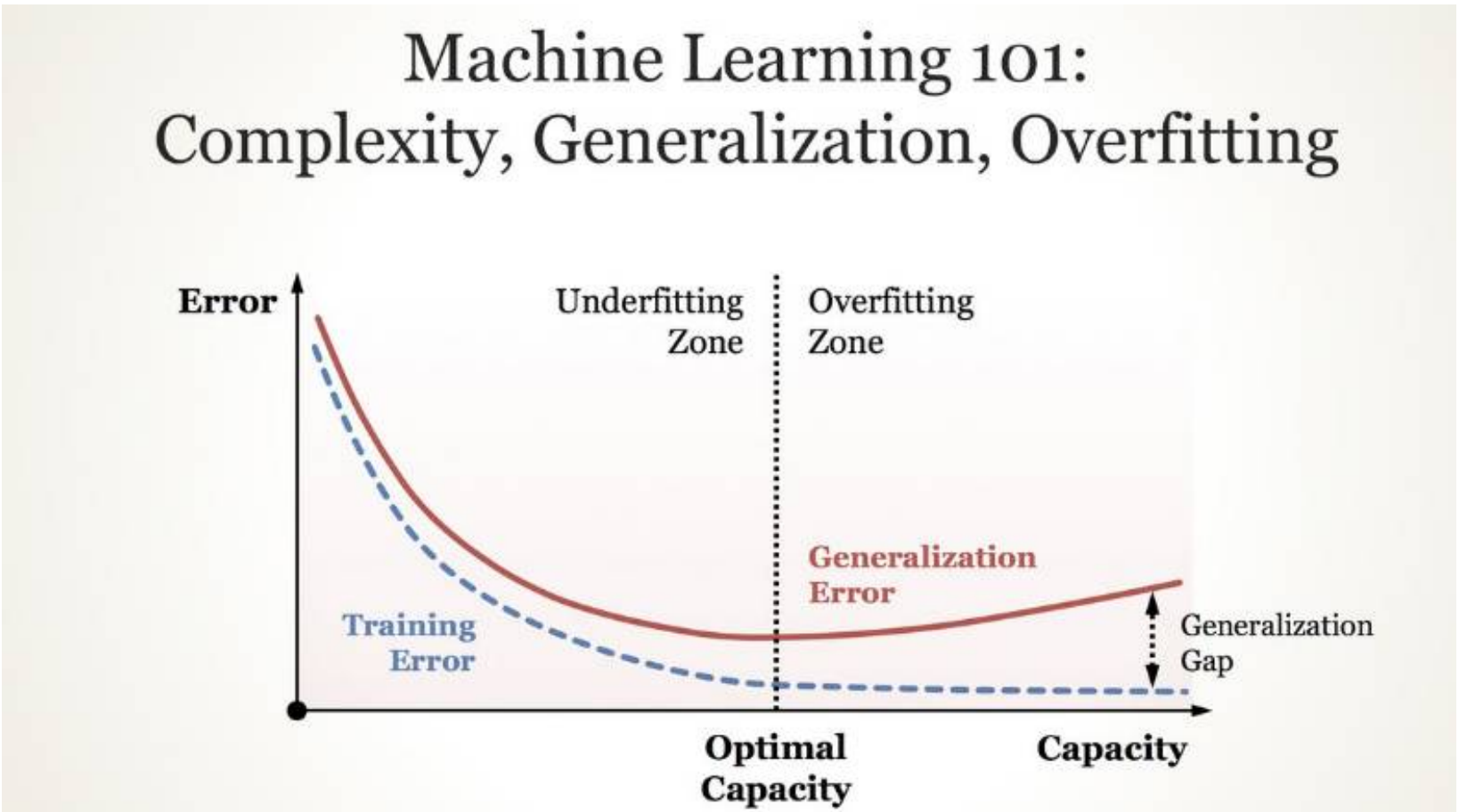
遗传算法特征选择概述

- 将N个输入变量用一个长度为N的染色体表示，染色体的每一位代表一个输入变量。
- 每一位的基因取值只能是“1”和“0”两种情况。
- 如果染色体的某一位值为“1”，表示该位对应的输入变量被选中，参与模型建立。
- 反之，如果染色体的某一位值为“0”，则表示对应的输入变量未被选中，不参与模型建立。

目录

第三讲 模型 vs. 数据，谁更重要？

模型复杂度 vs 误差


Machine Learning 101:
Complexity, Generalization, Overfitting
Error
Underfitting Zone
Overfitting Zone
Generalization Error
Training Error
Generalization Gap
Optimal Capacity
Capacity

ImageNet & ILSVRC2010-2017

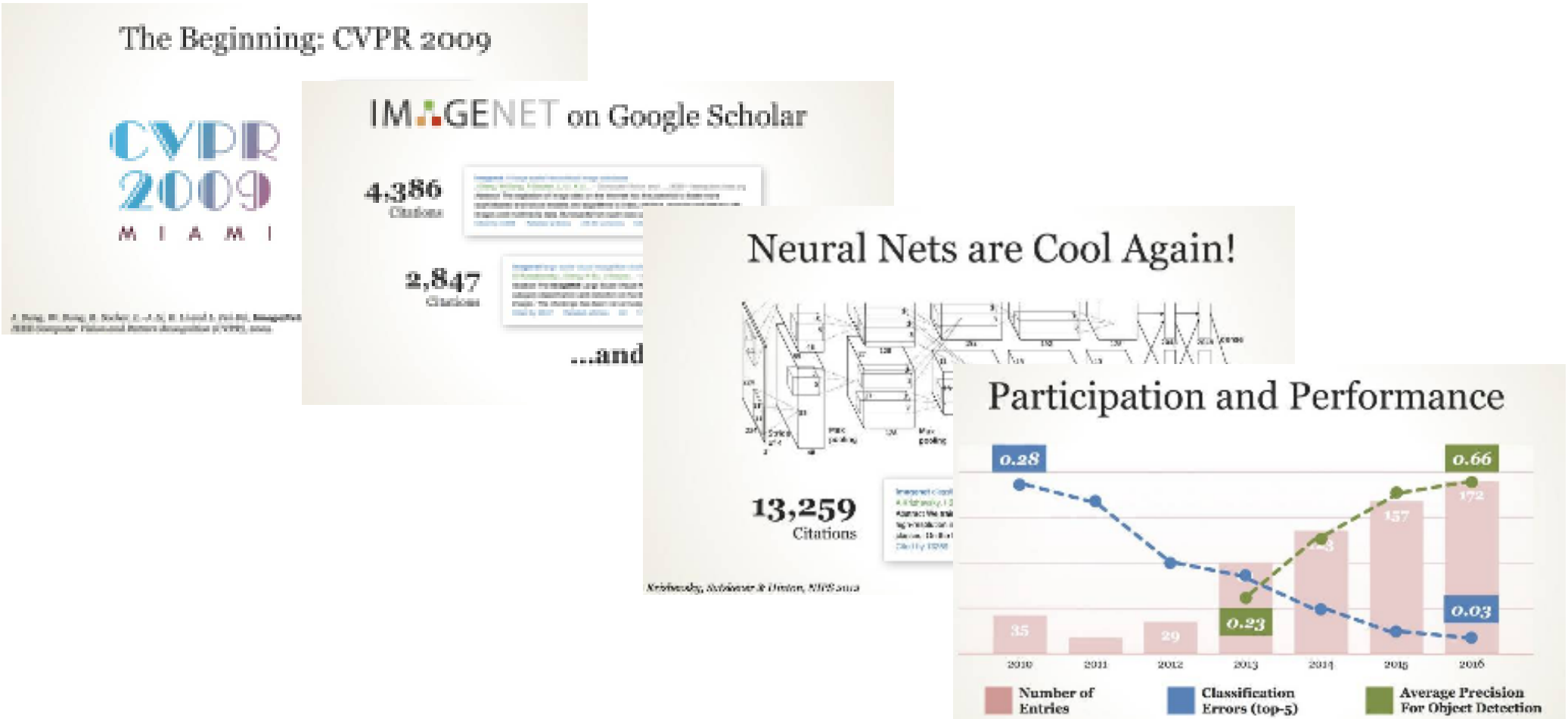
- **What is ImageNet?**

ImageNet is an image dataset organized according to the WordNet hierarchy. Each meaningful concept in WordNet is called a "synonym set" or "synset". There are **more than 100,000 synsets in WordNet**, majority of them are nouns (80,000+). **In ImageNet, we aim to provide on average 1000 images to illustrate each synset.**

- **Why ImageNet?**

The ImageNet project is inspired by a growing sentiment in the image and vision research field – the need for more data. Ever since the birth of the digital era and the availability of web-scale data exchanges, **researchers in these fields have been working hard to design more and more sophisticated algorithms to index, retrieve, organize and annotate multimedia data**. But good research needs good resource. To tackle these problem in, it would be tremendously helpful to researchers if there exists a large-scale image database. This is the motivation for us to put together ImageNet.

ImageNet & ILSVRC2010-2017



近红外光谱大数据？

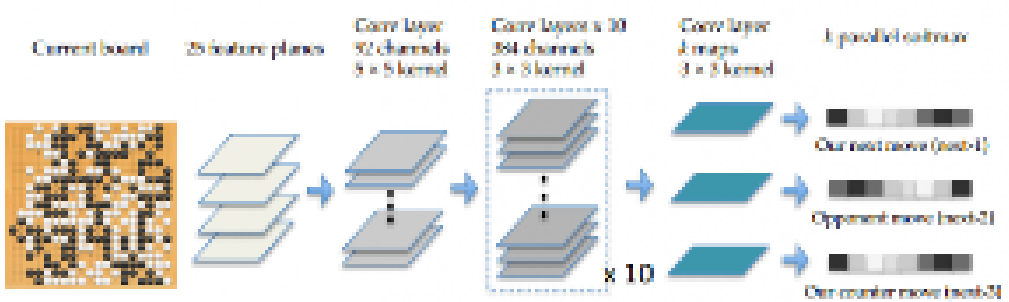

Chemical industry
agriculture
NIR
Medicine industry
Big Data

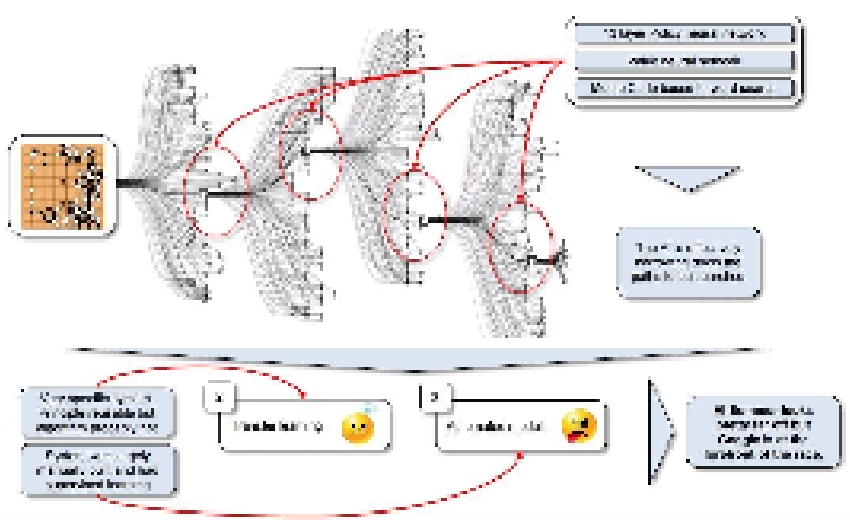
目录

第四讲 深度学习框架概述

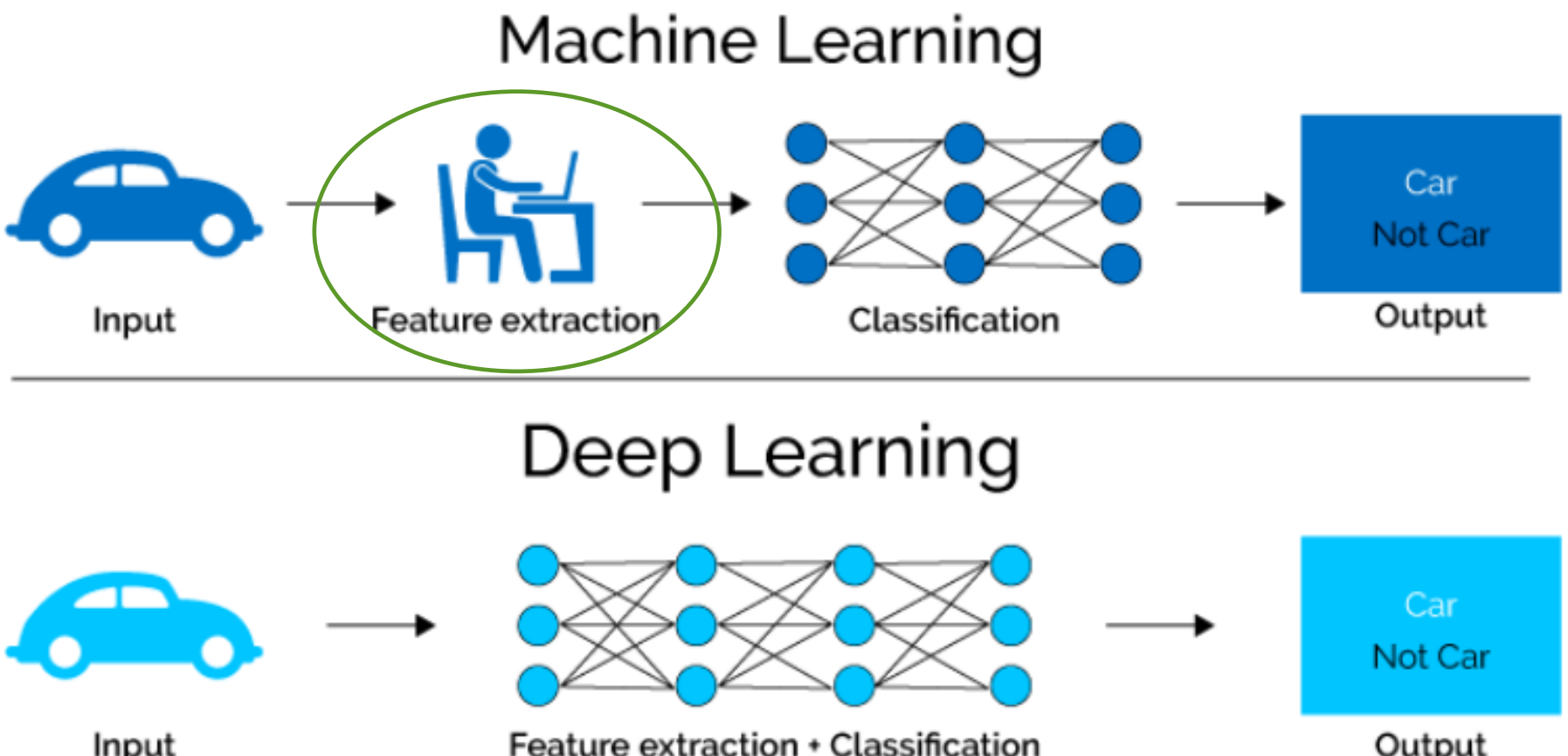
AlphaGo

12 vs. 40

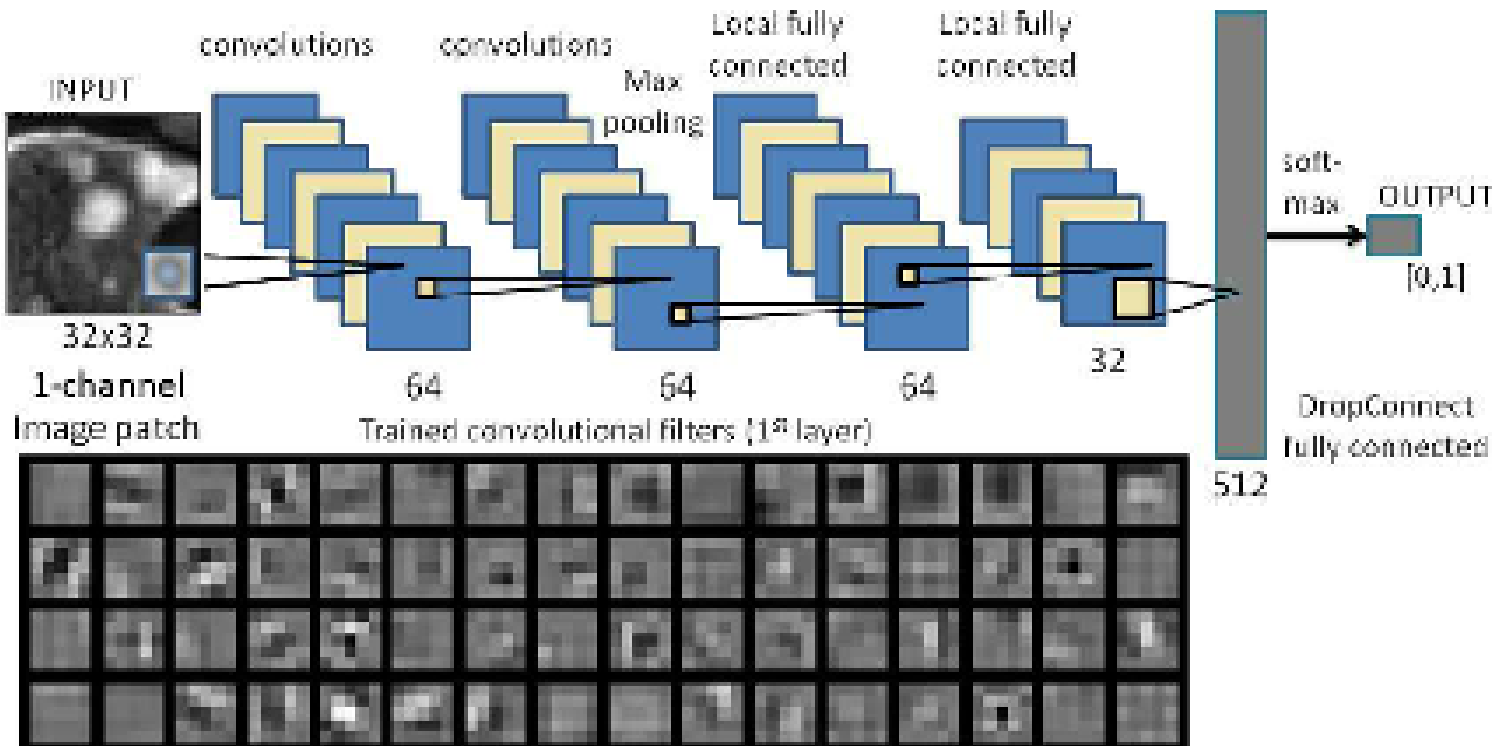


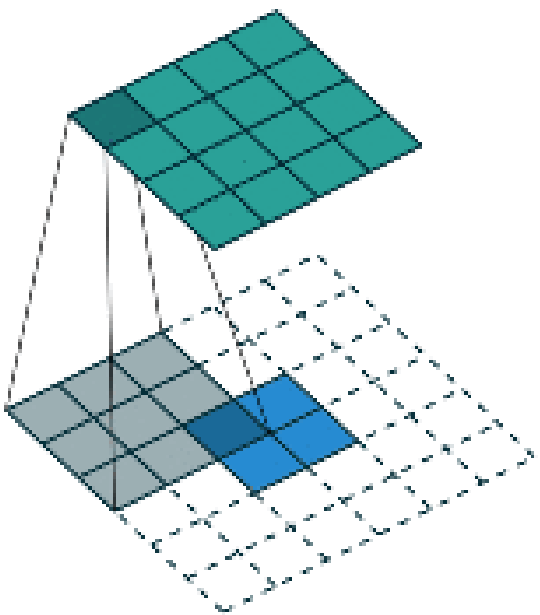


深度学习vs传统的机器学习

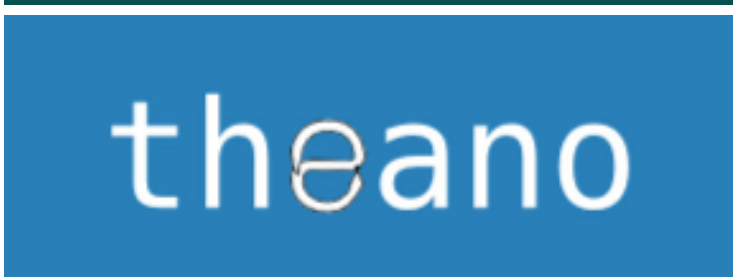


卷积神经网络（Convolutional Neural Network, CNN）



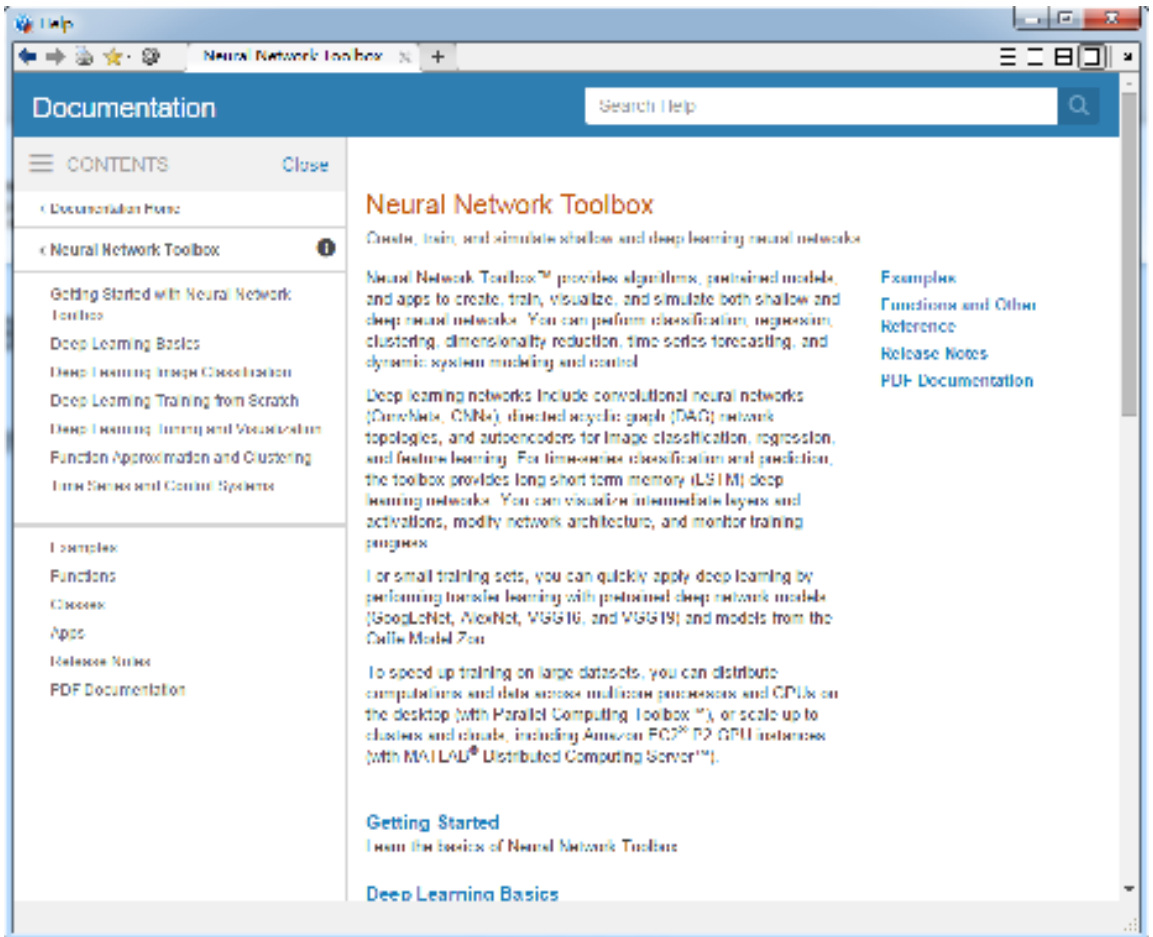


深度学习框架 – Python



深度学习框架 – MATLAB

- MATLAB R2017b
- MatConvNet
- ……

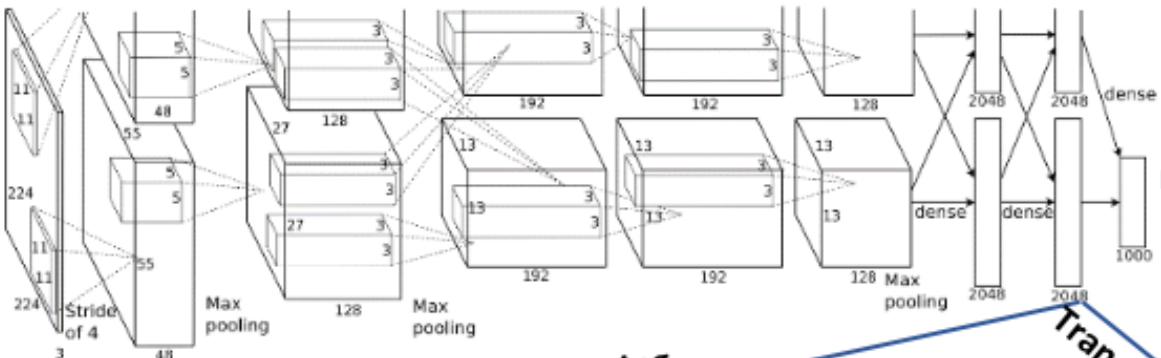


目录

第五讲 迁移学习理论概述

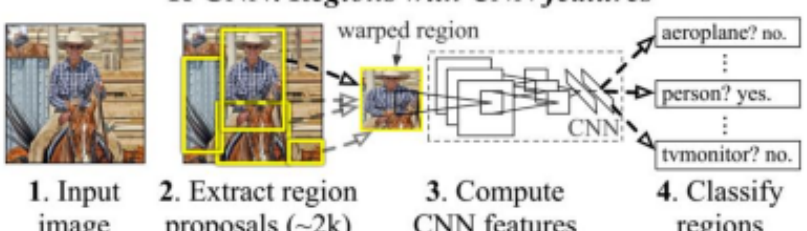
迁移学习（Transfer Learning）

迁移学习（Transfer Learning）



Transfer learning

Transfer learning



迁移学习在近红外光谱分析的应用

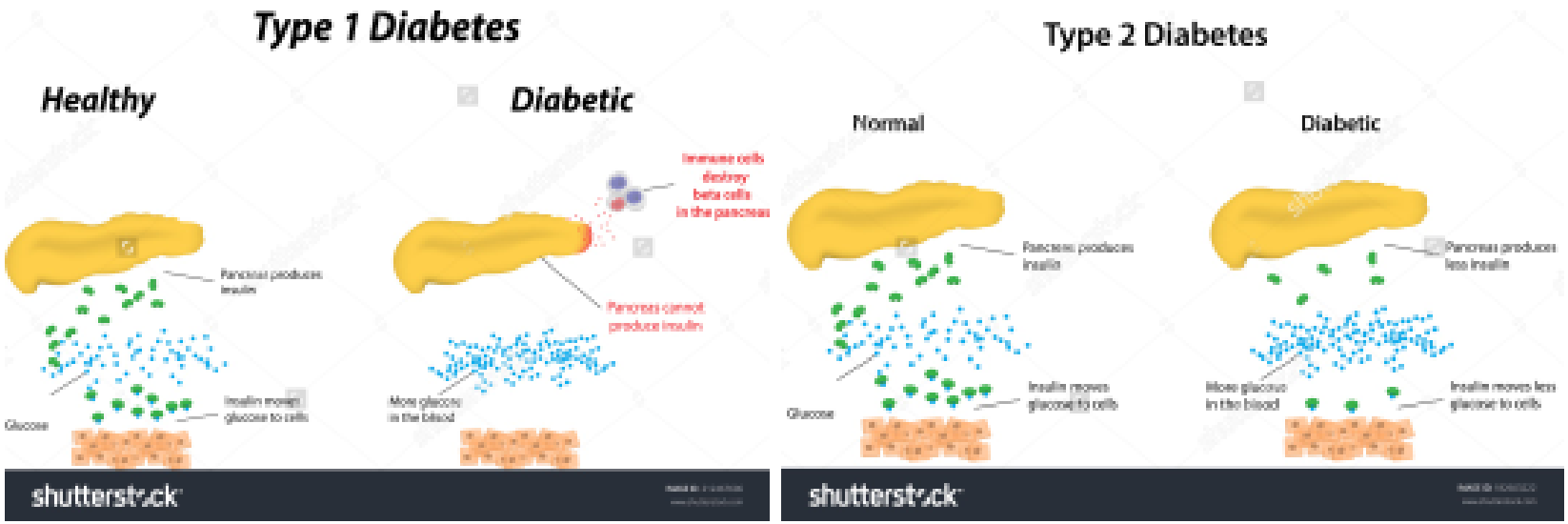
不同仪器设备之间的模型传递

不同研究对象之间的模型迁移

目录

第六讲 近红外光谱分析技术在生物医学工程领域中的应用讨论

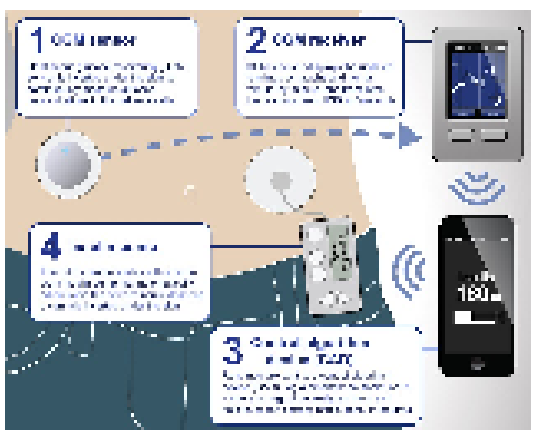
什么是糖尿病？


Type 1 Diabetes
Healthy
Diabetic
Pancreas produces insulin
Glucose
Insulin moves glucose to cells
Immune cells destroy beta cells in the pancreas
Pancreas cannot produce insulin
More glucose in the blood
shutterstock
Type 2 Diabetes
Normal
Diabetic
Pancreas produces insulin
Glucose
Insulin moves glucose to cells
Pancreas produces less insulin
More glucose in the blood
Insulin moves less glucose to cells
shutterstock

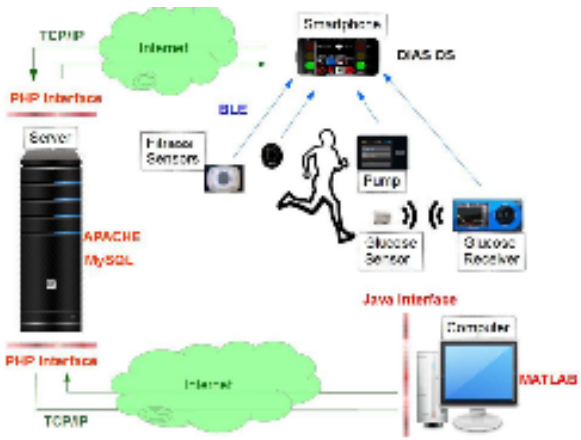

DIABETES
90%
10%
TYPE2
TYPE1

“人工胰脏”系统（Artificial Pancreas）

Open loop insulin pump
1990 ~ 2000

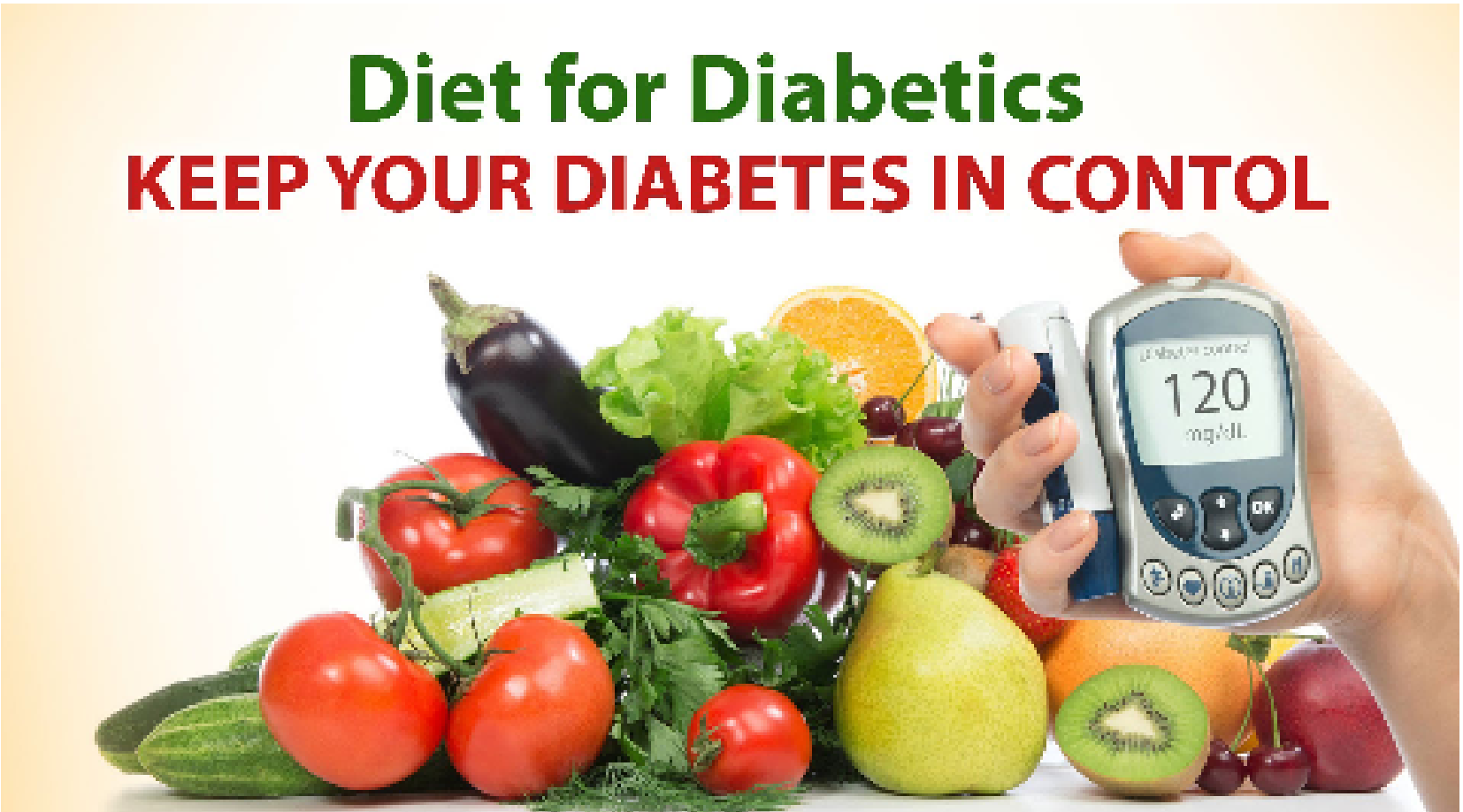


Closed loop insulin pump
2000 ~ Now



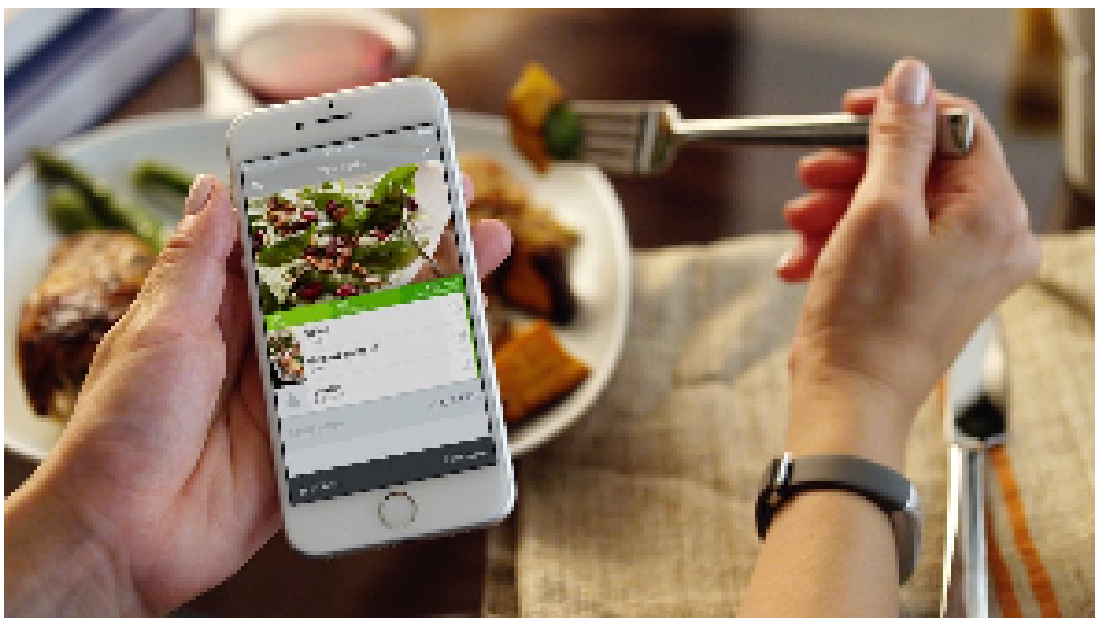
Enhanced artificial pancreas
2010 ~

面向糖尿病患者的膳食营养成分检测



面向糖尿病患者的膳食营养成分检测

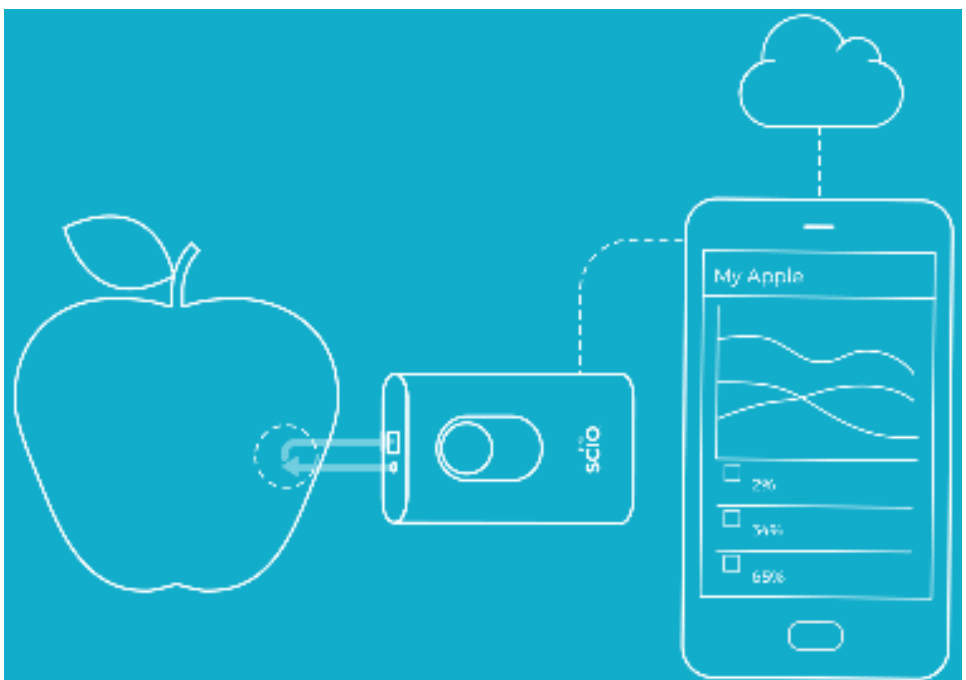
Image Recognition-based



NIRS-based

面向糖尿病患者的膳食营养成分检测

SCIO Scanner (700-1100nm)



Dimensions Handheld Device	Cover: 67.7 x 40.2 x 18.8mm Without cover: 54 x 36.4 x 15.4mm
Weight Handheld Device	35 gr
Standalone Sensor Module	27.5 x 9.5 x 3.15mm
Operational distance	Contact to 2 cm (can be customized, contact us for more details)
Typical scan time	2-5 seconds
Connectivity	Online and offline scanning supported
Temperature range - operation	4-35°C
Temperature range - storage	-40-85°C
SCiO App compatibility	iPhone 5 and above iOS 9 or higher Android 4.3 or higher Requires a Bluetooth Low Energy device

Thanks

FAQ时间