河北师范大学软件学院
Software College of Hebei Normal University

分类模型

第二部分 非线性分类模型

BP神经网络

张朝晖

2018-2019学年 20181009

1.分类模块

产生式分类模型 A.贝叶斯分类模型

判别式分类模型

线性分类模型
- B. Fisher判别分类
- C. 感知器分类模型
- D. 大间隔分类模型(线性*SVM*)

非线性分类模型
- E. 核*SVM*(非线性*SVM*)
- F. 核Fisher判别分类
- G. 神经网络

其它分类模型
- *H*.KNN分类模型
- *I*.决策树分类模型
- *J*.*Logistic*回归
- *K*.*Softmax*回归

2.聚类模块
- *L*.K-均值聚类
- *M*.高斯混合聚类
- *N*.DBSCAN聚类
- *O*.层次聚类

3.回归模块
- *P*.KNN回归
- *Q*.回归树
- *R*.最小二乘线性回归
- *S*.岭回归
- *T*.LASSO回归

4.集成学习
- *U*.*Bagging*
- *V*.随机森林
- *W*.*Boosting*

5.特征工程
- *X*.主成分分析*(PCA)*
- …

6.评价模块 混淆矩阵(及其相关指标)、ROC曲线、交叉验证

人工神经网络

--前馈神经网络与BP算法

关键词：

人工神经网络

前馈神经网络、人工神经元、激活函数

BP神经网络：梯度下降法、导数链式法则、误差反向传播

自编码器

监督式学习、非监督式学习

分类、回归、表示学习

多层BP神经网络学习

第一阶段：基于逐层表示学习的参数初始化；

第二阶段：基于BP算法的参数精调

主要内容

1. 人工神经网络基本知识、神经元与感知器

   1.1 生物神经网络、生物神经元

   1.2 人工神经网络

   1.3 人工神经网络的基本模型

   1.4 人工神经元

2. 前馈神经网络、多层感知器、及非线性分类
3. BP神经网络
4. 基于前馈神经网络的自编码器(Autoencoders)

机器学习研究方法基本分为两类。

两种研究方法：

- 自顶向下的分析方法(analysis)

  分析待实现功能 → （分解）子功能 → 实现子功能

- 自底向上的综合方法(synthesis)

  将复杂人脑视为由**大量神经元**组成巨大神经网络；
  从基本功能出发，逐步由简单到复杂，组成系统。

  基本单元 → 功能模块 → 系统

→ 人工神经网络

人工神经网络

是生物神经网络的某种模型(数学模型)

是对生物神经网络的模仿

基本处理单元为人工神经元

1.1 生物神经系统与生物神经元

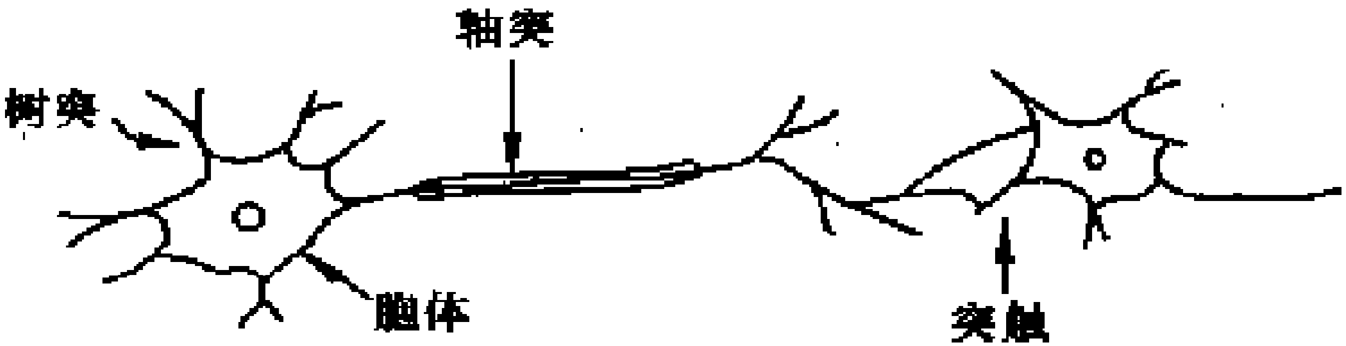
大量生物神经元的广泛、复杂连接，形成生物神经网络 *(Biological Neural Network, BNN)*

实现各种智能活动

生物神经元(neuron)是基本的信息处理单元

(1) 生物神经元(neuron)组成

- 树突(dendrites)，接收来自外接的信息
- 胞体(cell body)，神经细胞主体，信息加工
- 轴突(axon)，细胞的输出装置，将信号向外传递，与多个神经元连接
- 突触 （synapsse)，神经元经突触传递信号给其它神经元（胞体或树突）



(2)生物神经元的基本特征

- 神经元之间彼此连接
- 连接强度决定信号传递的强弱

  连接强度可以随训练而改变

  学习、遗忘、疲劳

  ----神经网络中各神经元之间连接的强弱，按外部的激励信号做自适应变化

- 神经元的兴奋与抑制

  一个神经元接受信号的累积效果（综合大小，代数和）决定该神经元的状态(兴奋、抑制)

  每个神经元可以有一个“阈值”

1.2 人工神经网络

人工神经网络是对**生物神经系统**的模拟，

大量简单的**计算单元（人工神经元）**以某种形式**连接**，形成一个**网络**。

一些因素：

连接强度(连接权值，其大小决定信号传递强弱)，

结点计算特性(激活特性,神经元的输入输出特性)，

网络结构等，

可依规则随外部数据调整，最终实现某种功能。

神经网络的不同网络结构可以体现各种不同的功能；

网络结构的**参数**是通过**学习**逐渐修正的。

1.3 人工神经网络三种基本模型

人工神经网络三要素

网络结构或拓扑（连接形式）
神经元的计算特性(传递函数)
学习规则

上述要素不同组合，形成各种神经网络模型

(1)**前馈型神经网络(*feedforward network*)**

多层感知器

BP网络　　(CNN、自编码器)

RBF网络

(2)**反馈网络(*feedback network*)**

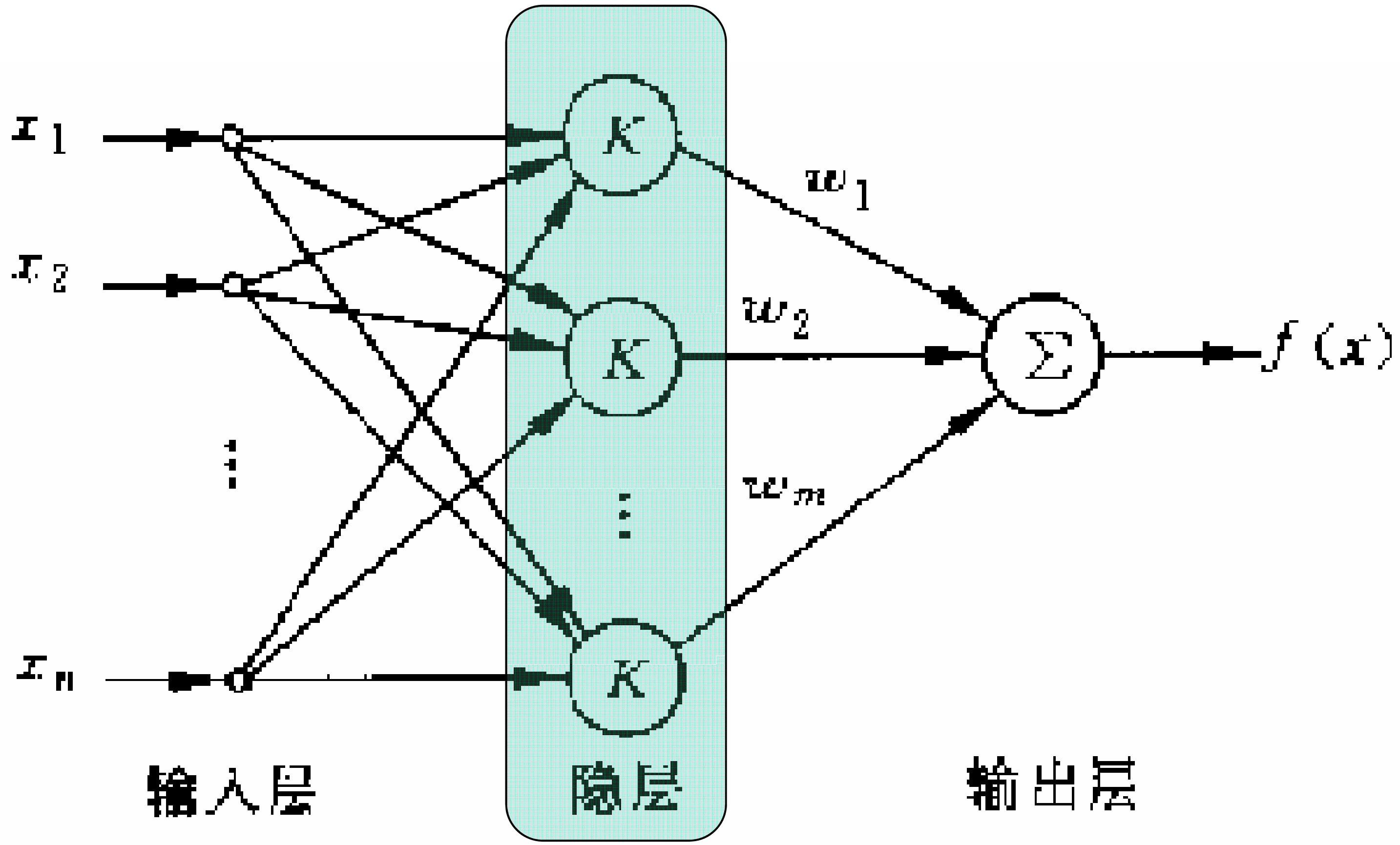
*Hopfield*网络,*RNN*,*LSTM*

(3)**竞争学习网络(*competitive learning network*)**

*SOM*神经网络,*LVQ*神经网络

(1)前馈神经网络

例:RBF神经网络(图中输出节点为1个，可以多个)



径向基函数网络示意图

隐含层节点的**传递函数**

$$\varphi(\boldsymbol{x},\boldsymbol{c}) = K(\|\boldsymbol{x}-\boldsymbol{c}\|) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}\|^2}{r^2}\right)$$

$$\begin{cases} r\text{——窗参数，控制函数径向作用范围} \\ \boldsymbol{c}\text{——核函数中心} \end{cases}$$

RBF**网络输出**

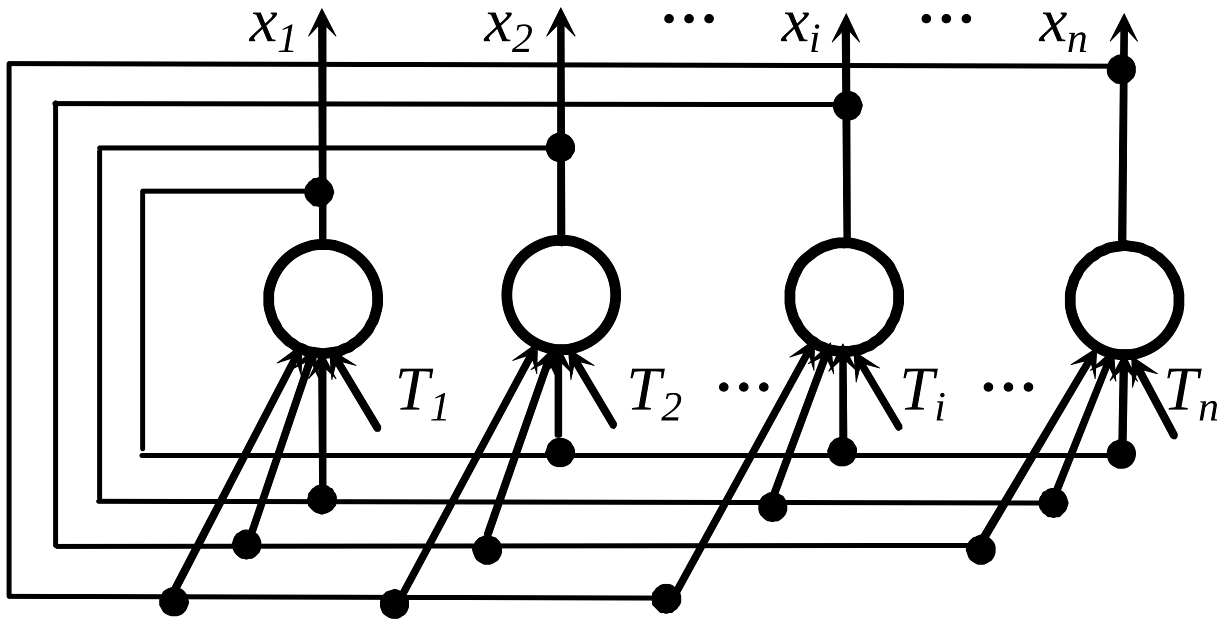
$$y(\boldsymbol{x}) = \sum_{i=1}^{N} \omega_i \varphi(\boldsymbol{x},\boldsymbol{c}_i)$$

(2)反馈型(或递归、回归式)神经网络

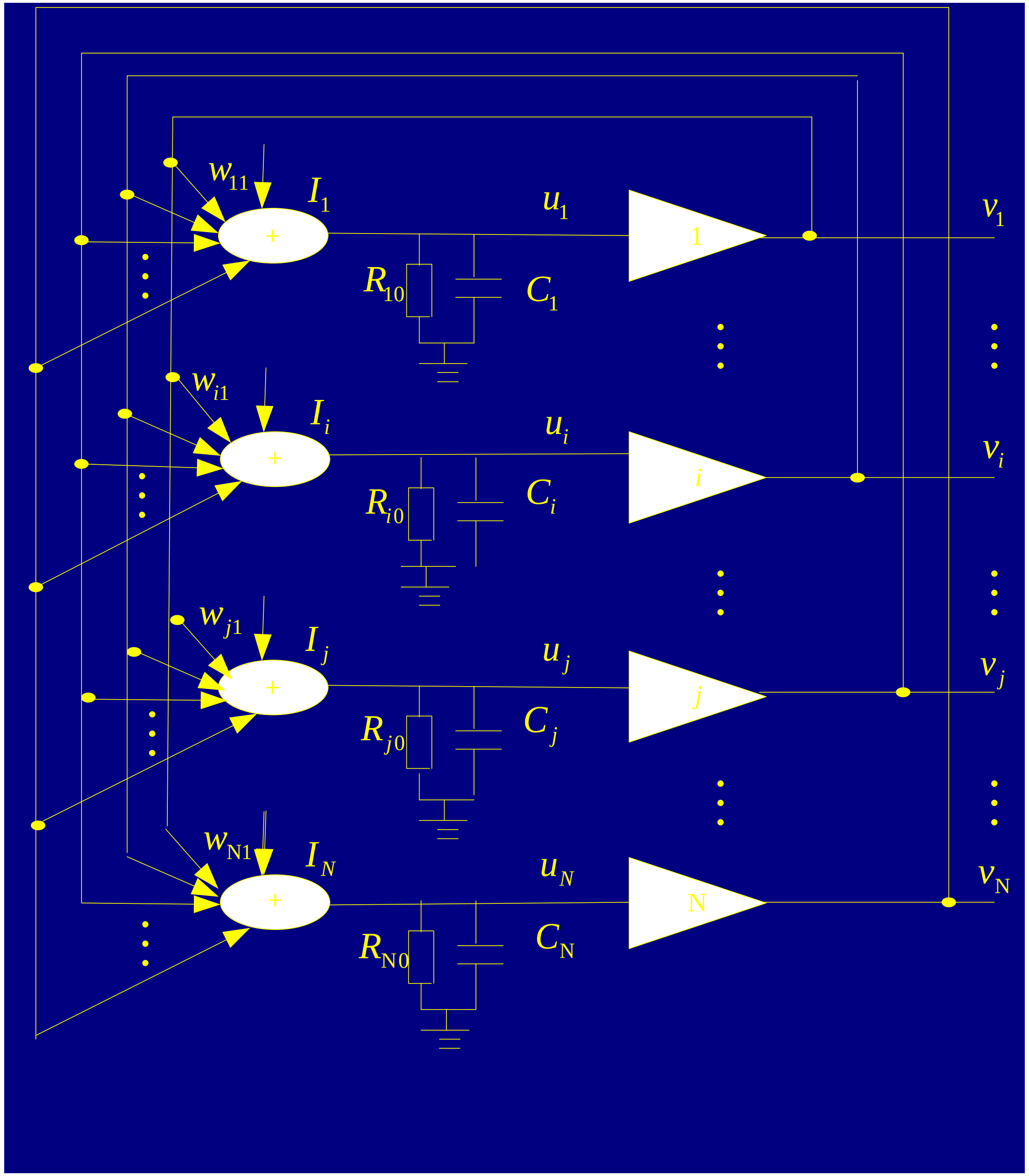
例：Hopfield神经网络

- 离散型、连续型
- 动态网络：初始状态、稳定状态

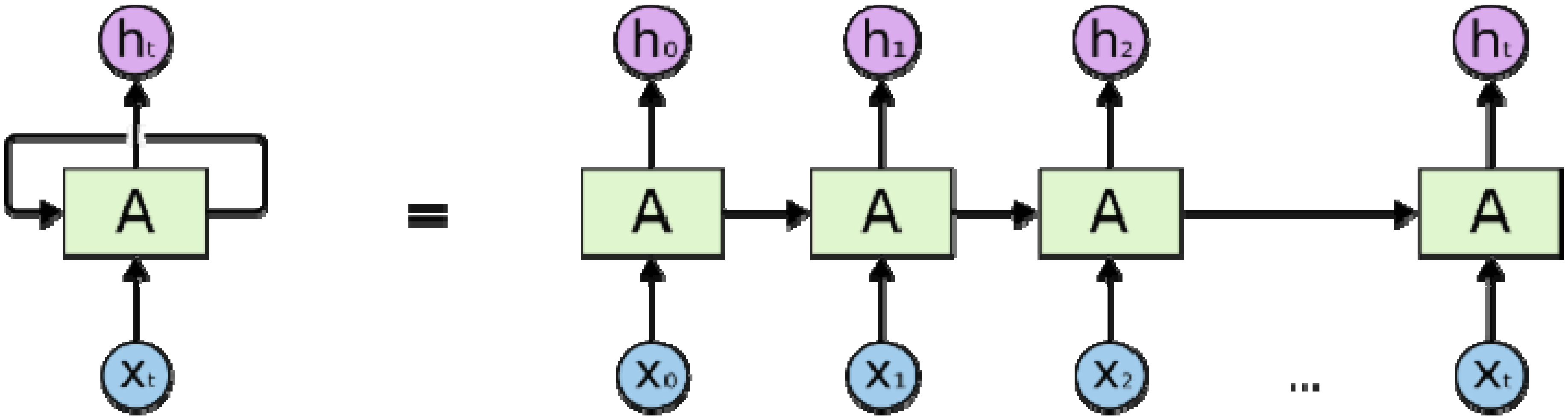
离散形式

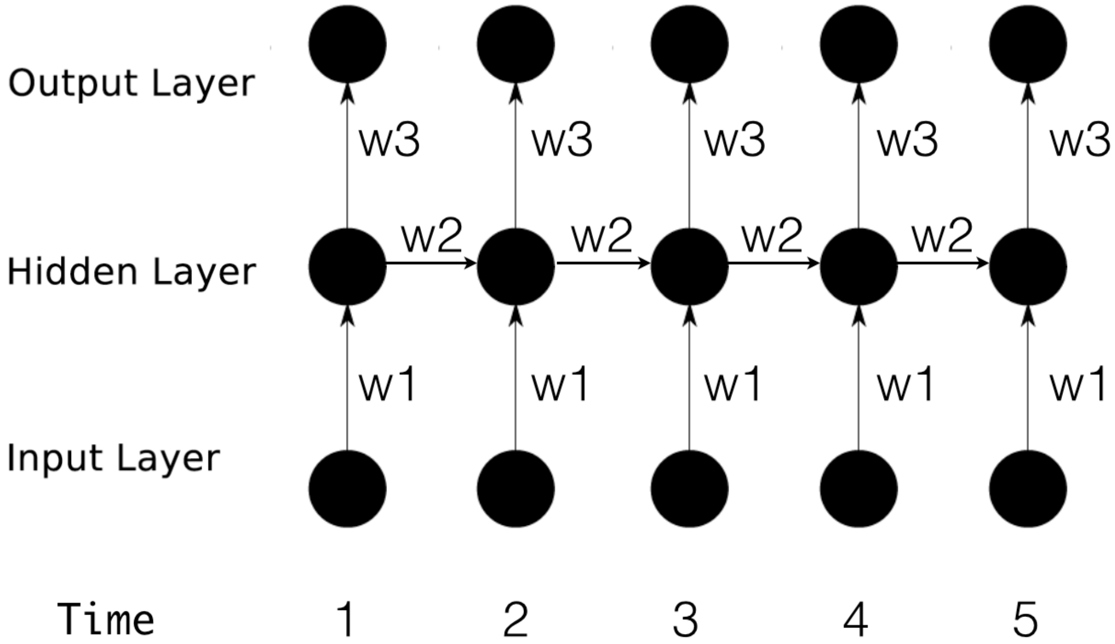


Hopefield 网络的连续形式

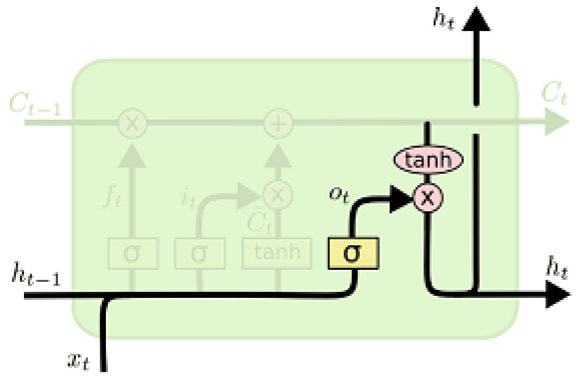


例：循环神经网络(Recurrent Neural Networks, RNN)



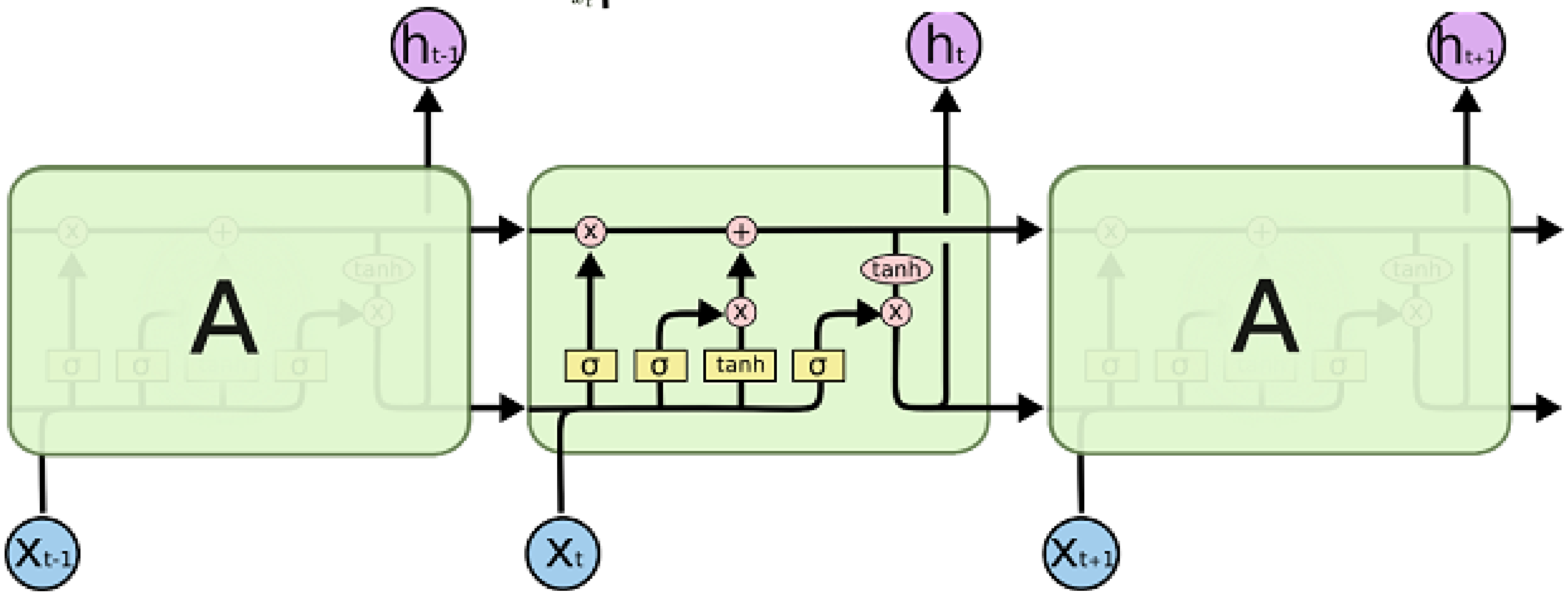


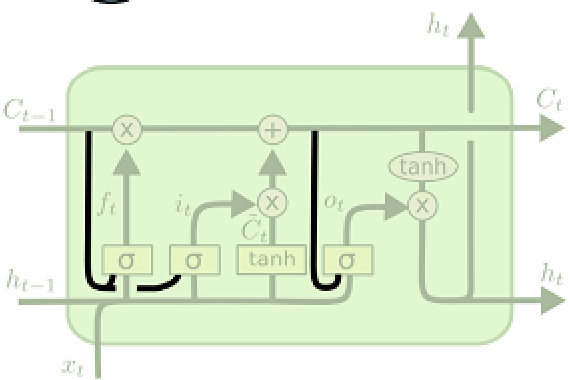
例：LSTM 网络 (Long Short Term Memory networks)



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$





$$f_t = \sigma\left(W_f\cdot[\boldsymbol{C_{t-1}}, h_{t-1}, x_t]\ +\ b_f\right)$$

$$i_t = \sigma\left(W_i\cdot[\boldsymbol{C_{t-1}}, h_{t-1}, x_t]\ +\ b_i\right)$$

$$o_t = \sigma\left(W_o\cdot[\boldsymbol{C_t}, h_{t-1}, x_t]\ +\ b_o\right)$$

(3)竞争型神经网络：SOM神经网络

- **两层网络：**

  1个输入层(信息层)

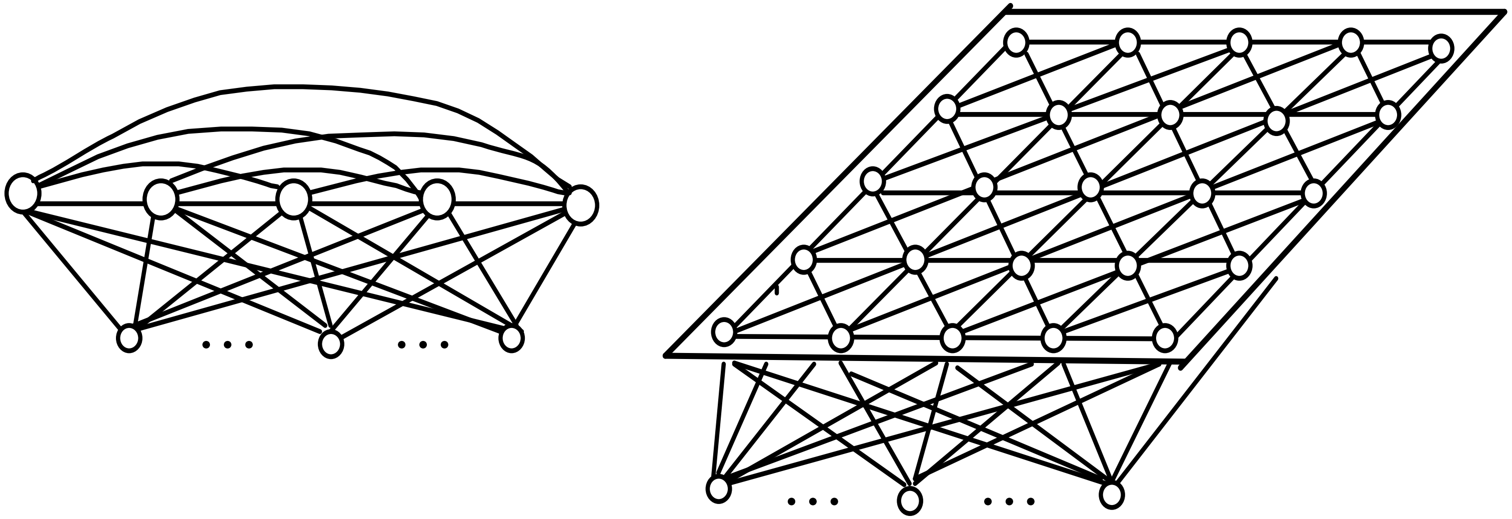
  1个输出层（竞争层、表示层）

  输入和输出之间有前向连接

  输出层各单元之间存在侧向连接，起侧向抑制作用。

- **学习算法：**

  “胜者为王”策略(The Winner Takes All)



(a)一维线阵　　(b)二维平面线阵

例 基于SOM的 不同国家生活质量分析

http://www.ai-junkie.com/ann/som/som5.html

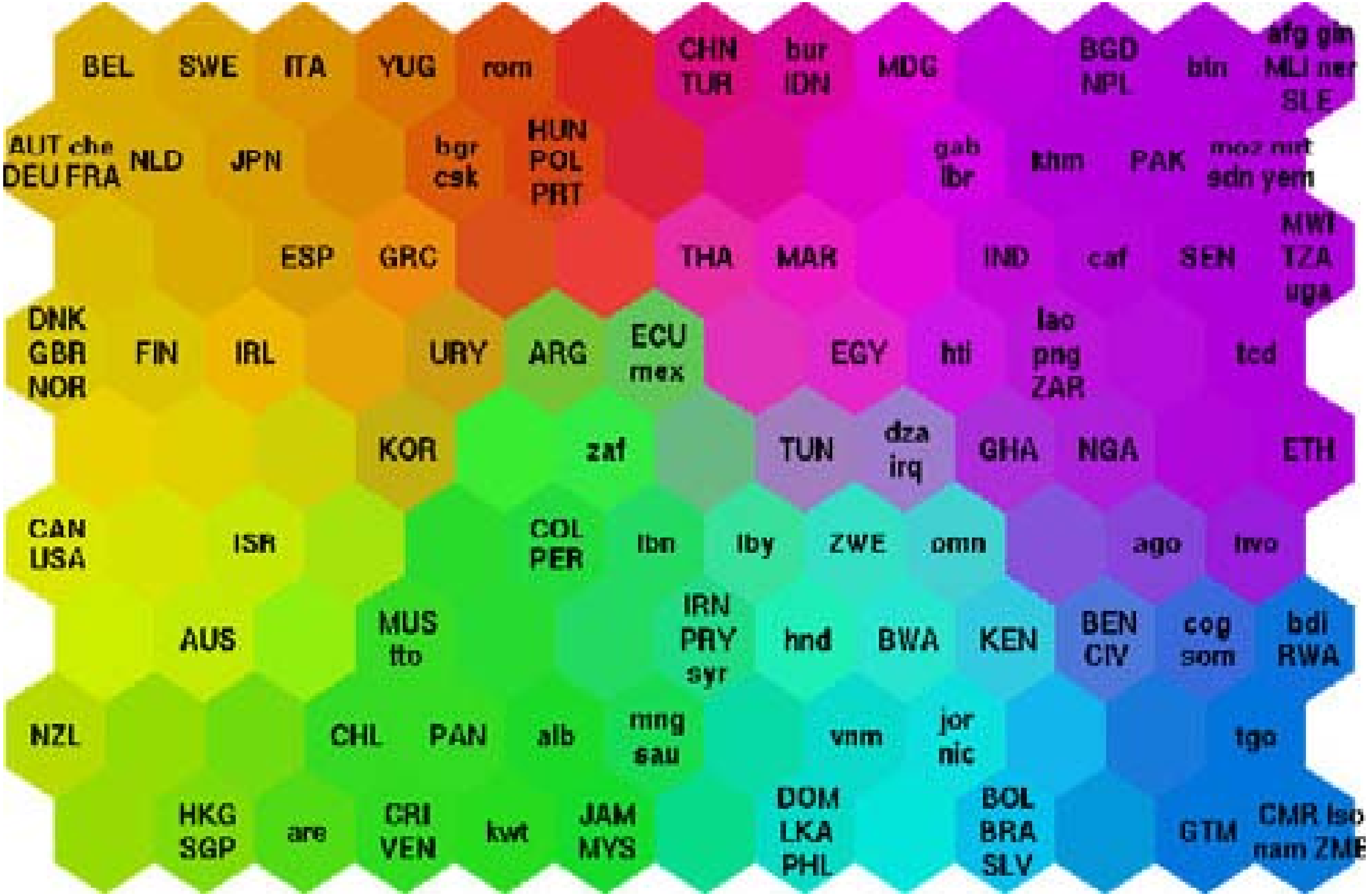
- To classify statistical data describing various quality-of-life factors such as state of health, nutrition, educational services etc.

- Countries with similar quality-of-life factors end up clustered together.

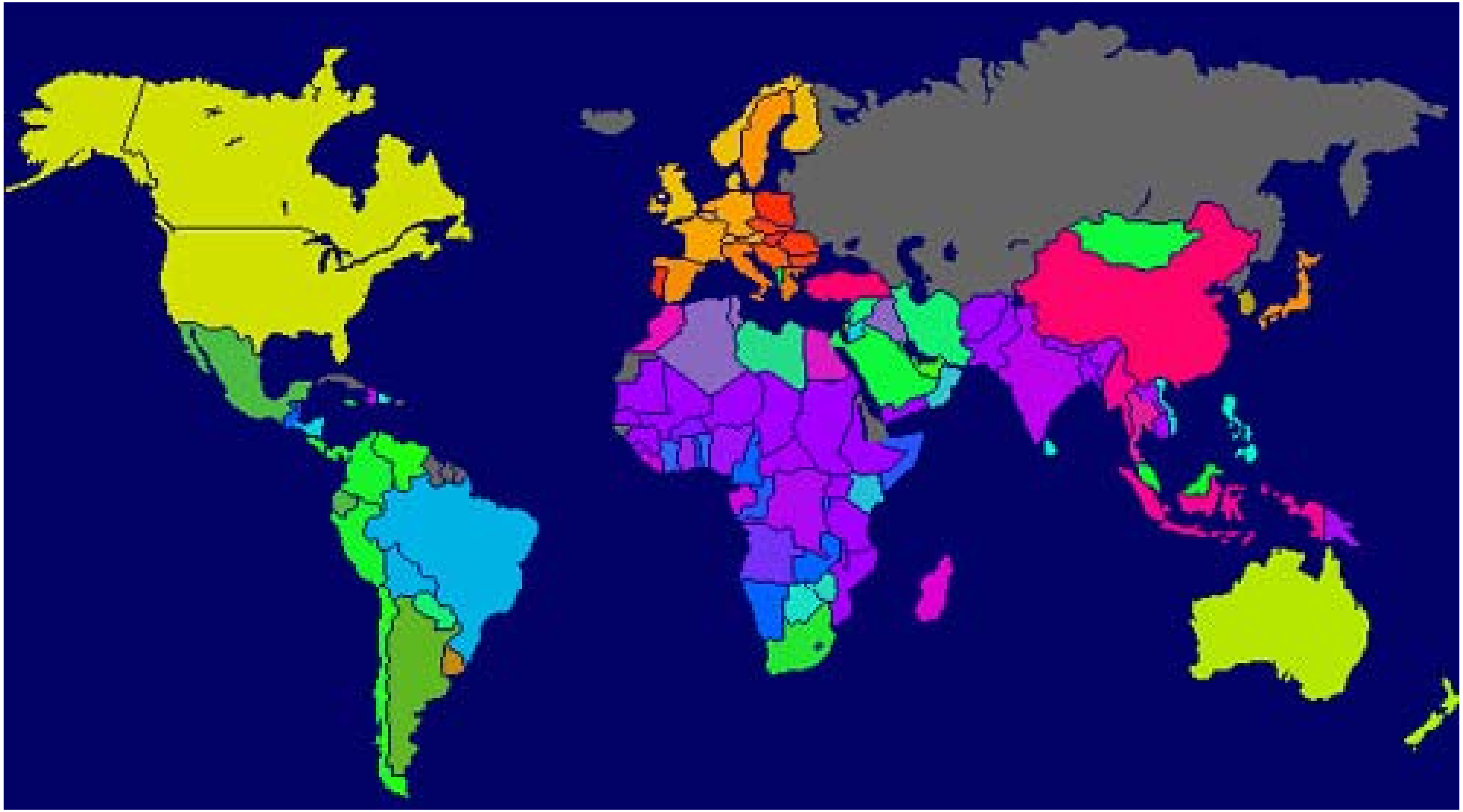
- The countries with better quality-of-life are situated toward *the upper left* and the most poverty stricken countries are toward *the lower right*.

- SOM does not show poverty levels, rather it shows how similar the poverty sets for different countries are to each other. (Similar color = similar data sets).

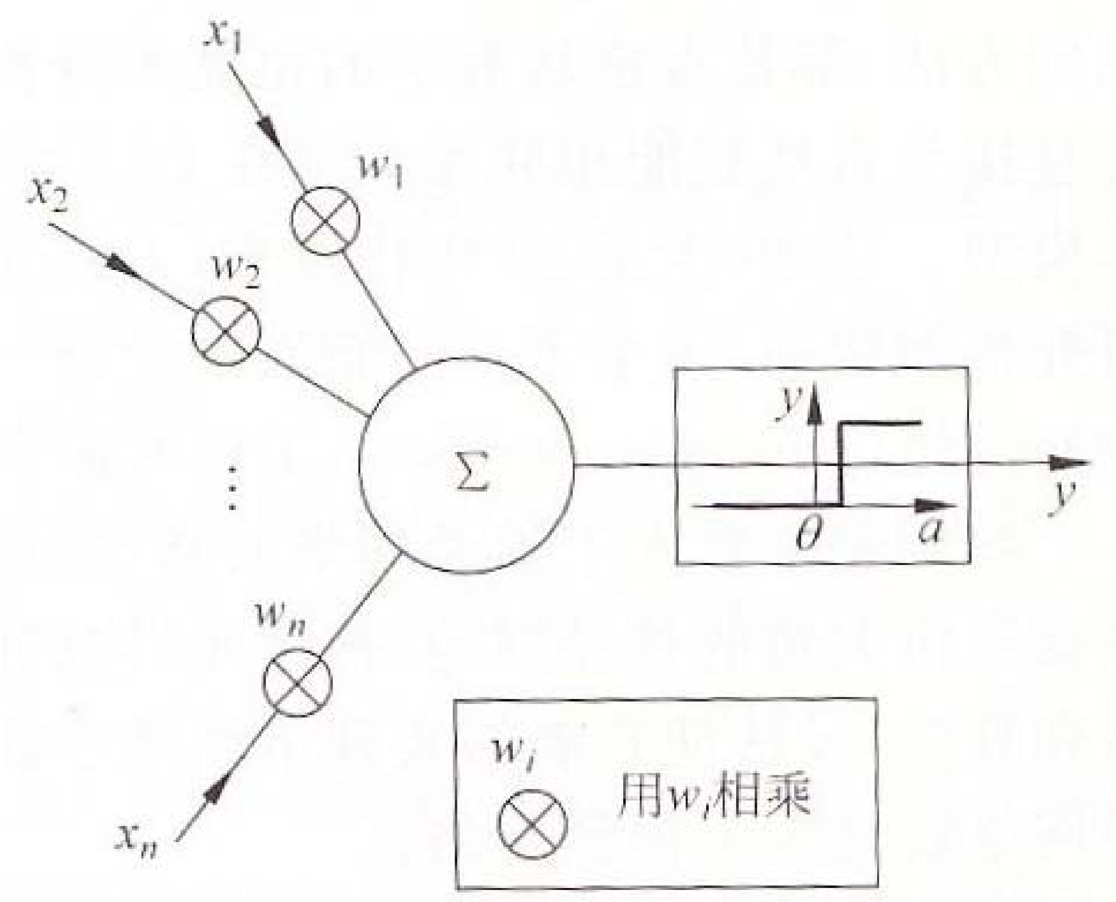
SOM网络


BEL
SWE
ITA
YUG
rom
CHN TUR
bur IDN
MDG
BGD NPL
bin
afg gin MLI ner SLE
AUT che DEU FRA
NLD
JPN
bgr csk
HUN POL PRT
gab lbr
khm
PAK
moz mrt sdn yem
ESP
GRC
THA
MAR
IND
caf
SEN
MWI TZA uga
DNK GBR NOR
FIN
IRL
URY
ARG
ECU mex
EGY
hti
lao png ZAR
tcd
KOR
zaf
TUN
dza irq
GHA
NGA
ETH
CAN USA
ISR
COL PER
lbn
lby
ZWE
omn
ago
mo
AUS
MUS tto
IRN PRY syr
hnd
BWA
KEN
BEN CIV
cog som
bdi RWA
NZL
CHL
PAN
alb
mng sau
vnm
jor nic
tgo
HKG SGP
are
CRI VEN
kwt
JAM MYS
DOM LKA PHL
BOL BRA SLV
GTM
CMR lso nam

映射结果在地图可视化



1.4 人工神经元



McCulloch-Pitts神经元模型(阈值逻辑单元)

输入信号； 连接强度与权向量； 信号累积

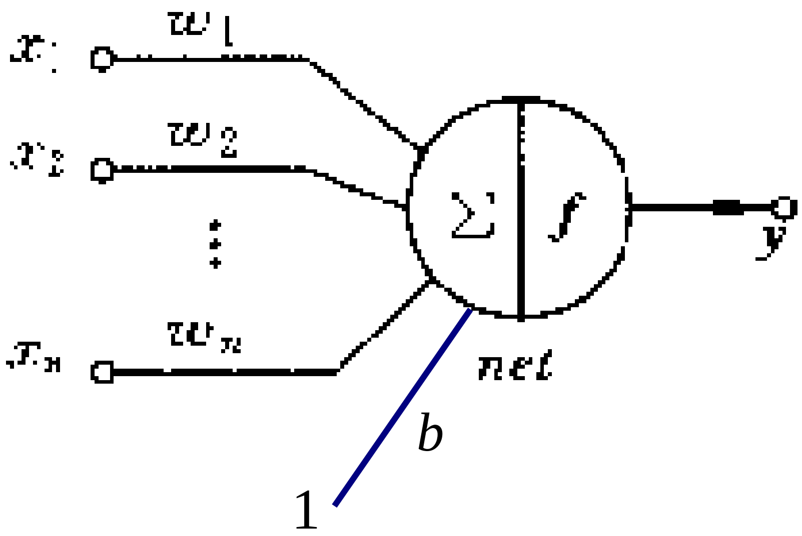
神经元节点的状态：激活/抑制

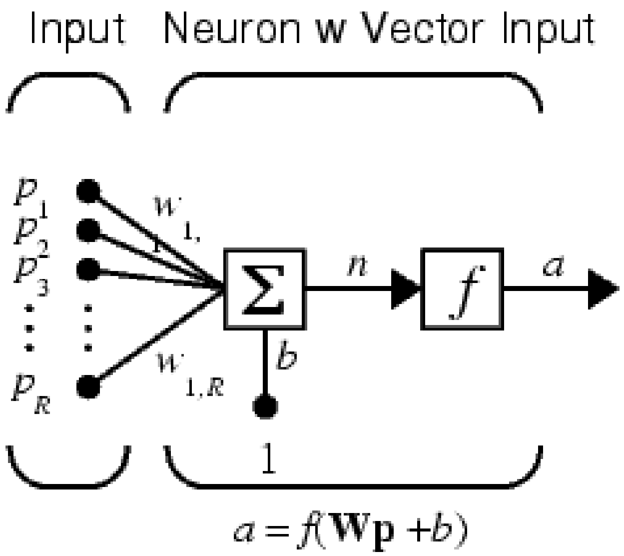
(1) 人工神经元模型的三要素

一组连接　　连接权值，突触连接强度 $\begin{cases} 权值>0，激活 \\ 权值<0，抑制 \end{cases}$

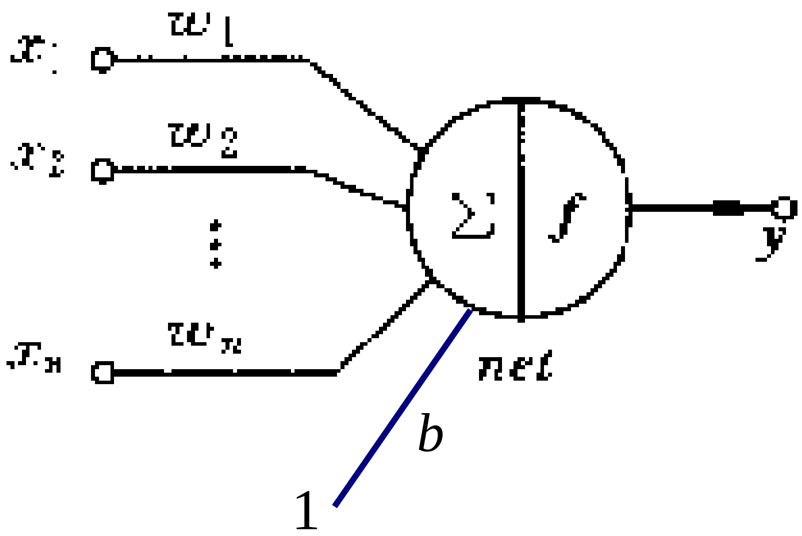
一个加法器　　输入信号关于神经元突触的线性加权

一个激励函数　　将神经元的输出信号限制在有限范围内





(2)基本人工神经元



输入信号 $\boldsymbol{x} = \left[x_1, \ldots, x_n\right]^{\boldsymbol{T}}$

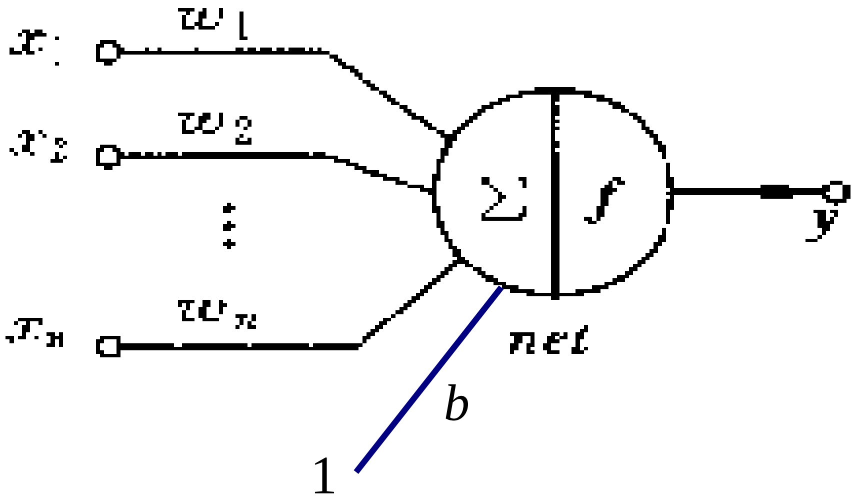
权向量 $\boldsymbol{W} = \left[\omega_1, \ldots, \omega_n\right]^{\boldsymbol{T}}, \omega_i \in \boldsymbol{R}$

偏置量 b

传递函数 $f(\bullet)$

激励(激活)函数，输出

限幅函数(或压挤函数)



将可能的无限域变换到指定的有限范围

单调增函数，通常为"非线性函数"

神经元节点**净输入** $net = \boldsymbol{W} \bullet \boldsymbol{x} + b = \sum_{i=1}^{n} \omega_i x_i + b$

神经元节点**输出** $y = f(net) = f(\sum_{i=1}^{n} \omega_i x_i + b)$

(3)几种常见形式的传递函数(激活函数)

A.线性函数

$$\begin{cases} f(net) = k \cdot net + c \\ f'(net) = k \end{cases} \qquad \begin{cases} f(net) = net \\ f'(net) = 1 \end{cases}$$

B.非线性斜面函数*(Ramp Function)*:

$$f(net) = \begin{cases} b & net \ge \theta \\ k \cdot net & |net| < \theta \\ -b & net \le -\theta \end{cases}$$

$b > 0$为常数，称饱和值，是该神经元节点的最大输出；

输出函数值限制在$[-b, b]$范围内。

C.符号函数

$\left(\begin{array}{l} \textbf{sign}\text{型函数，不可微；对称硬极限函数；} \\ \text{双极函数} \end{array} \right)$

$$f\left(net\right)=sgn\left(net\right)=\begin{cases} 1 & net \geq 0 \\ -1 & net < 0 \end{cases}$$

例：***matlab***函数

$$hardlim\left(net\right)=\begin{cases} 1 & net \geq 0 \\ 0 & net < 0 \end{cases}$$

$$hardlims\left(net\right)=\begin{cases} 1 & net \geq 0 \\ -1 & net < 0 \end{cases}$$

D.阈值函数

$$f(net) = \begin{cases} \beta & net \geq \theta \\ -\gamma & net < \theta \end{cases}$$

其中β,γ,θ非负实数

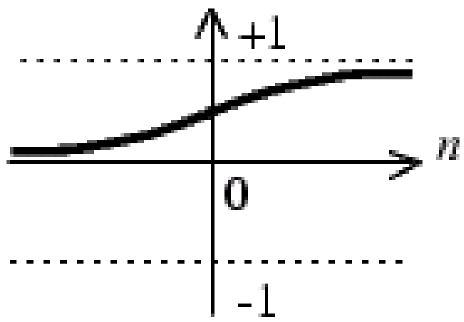
E． sigmoid函数（**S型函数**，连续可微）

一些重要的学习算法要求输出函数可微

对数S型函数（也称"*soft step*"函数）

$$f(net) = \frac{1}{1+e^{-net}} \qquad \text{值域}(0,1)$$

matlab*函数：*logsig



$$f'(net) = f(net)\left(1 - f(net)\right)$$

双曲正切S型函数（TanH函数）

$$f(net) = th(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} = \frac{2}{1 + e^{-2net}} - 1 \quad 值域(-1,1)$$

***matlab*函数：** *tansig*

$$f'(net) = 1 - f^2(net)$$

建议形式：$f(net) = 1.7159 \cdot th(\frac{2}{3} net)$

其它：$f(net) = th(net) + a \cdot net$

E. sigmoid函数（S型函数，连续可微）

非线性，单调；　无限次可微

$|net|$较小时（权值较小），可近似线性函数

--高增益区处理小信号

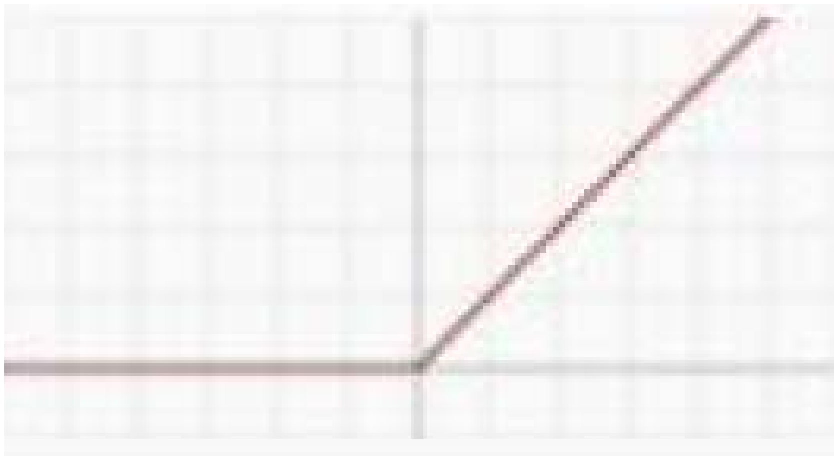
$|net|$较大时（权值较大），可近似阈值函数.

--低增益区处理大信号

F． Reticified Linear Unit 函数(ReLU函数)

$$\boldsymbol{f}(net) = \max\{0, \boldsymbol{net}\} = \begin{cases} 0 & if\ \ net < 0 \\ \boldsymbol{net} & if\ net \geq 0 \end{cases}$$

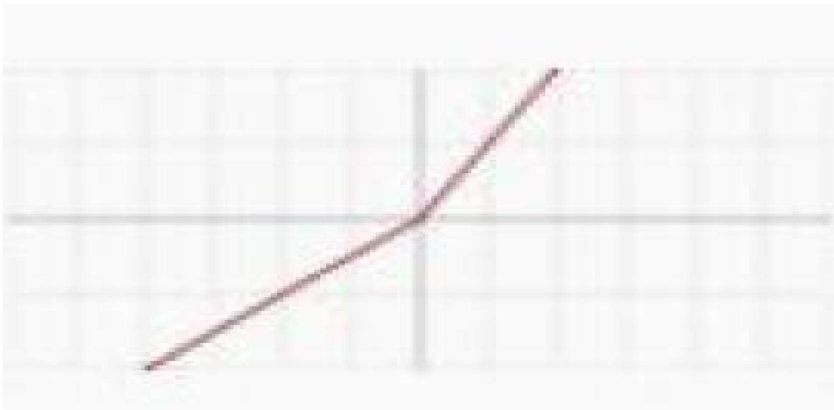
$$\boldsymbol{f}'(net) = \begin{cases} 0 & if\ \ net < 0 \\ 1 & if\ \ net > 0 \end{cases}$$



G. *Parameteric Reticified Linear Unit* 函数(PReLU函数)

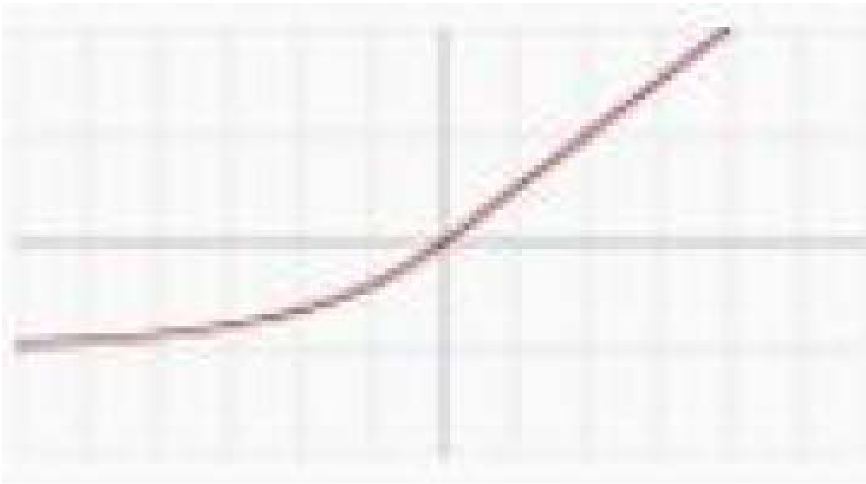
$$f(net)=\begin{cases}\alpha \cdot net & if\ net<0\\ net & if\ net\geq 0\end{cases}$$

$$f'(net)=\begin{cases}\alpha & if\ net<0\\ 1 & if\ net>0\end{cases}$$



H． *Exponential Linear Unit* 函数(ELU函数)

$$
f(net) = \begin{cases} \alpha\left(e^{net} - 1\right) & \text{if } net < 0 \\ net & \text{if } net \geq 0 \end{cases}
$$

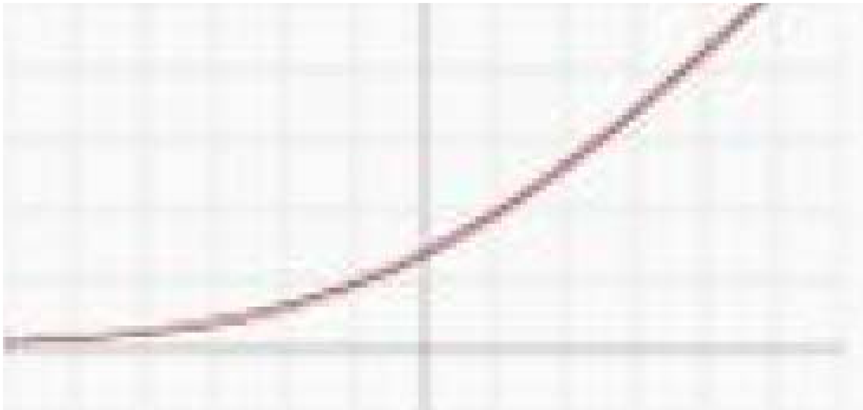


$$
f'(net) = \begin{cases} f(net) + \alpha = \alpha e^{net} & \text{if } net < 0 \\ 1 & \text{if } net > 0 \end{cases}
$$

I. Soft Plus 函数(ReLU函数的平滑版)

$$f(net) = \log_e\left(1 + e^{net}\right)$$

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



J. Bent Identity 函数

$$f(net) = \frac{\sqrt{1 + net^2} - 1}{2} + net$$

$$f'(net) = \frac{net}{2\sqrt{1 + net^2}} + 1$$

关键词：

人工神经网络、人工神经元、激活函数

前馈神经网络

BP神经网络

梯度下降法、导数链式法则、误差反向传播

自编码器

监督式学习、非监督式学习

分类、回归、表示学习

多层BP神经网络学习

第一阶段：基于逐层表示学习的参数初始化；

第二阶段：基于BP算法的参数精调

主要内容

1. 人工神经网络基本知识、神经元与感知器
    - 生物神经网络、生物神经元
    - 人工神经网络、人工神经网络的基本模型、人工神经元

2. 前馈神经网络、多层感知器、及非线性分类

前馈神经网络(单层感知器、多层感知器)

多层感知器的函数逼近

3. BP神经网络

4. 基于前馈神经网络的自编码器(Auto-Encoders)

(1)前馈神经网络

- 网络中的节点分两类：输入节点；计算节点

- 节点按层(layer)组织

  可见层：输入层 (input layer)　含输入节点，无计算能力

  输出层 (output layer) 含输出节点(神经元节点)

  隐含层(hidden layer):　中间层，神经元节点

- 节点连接方式

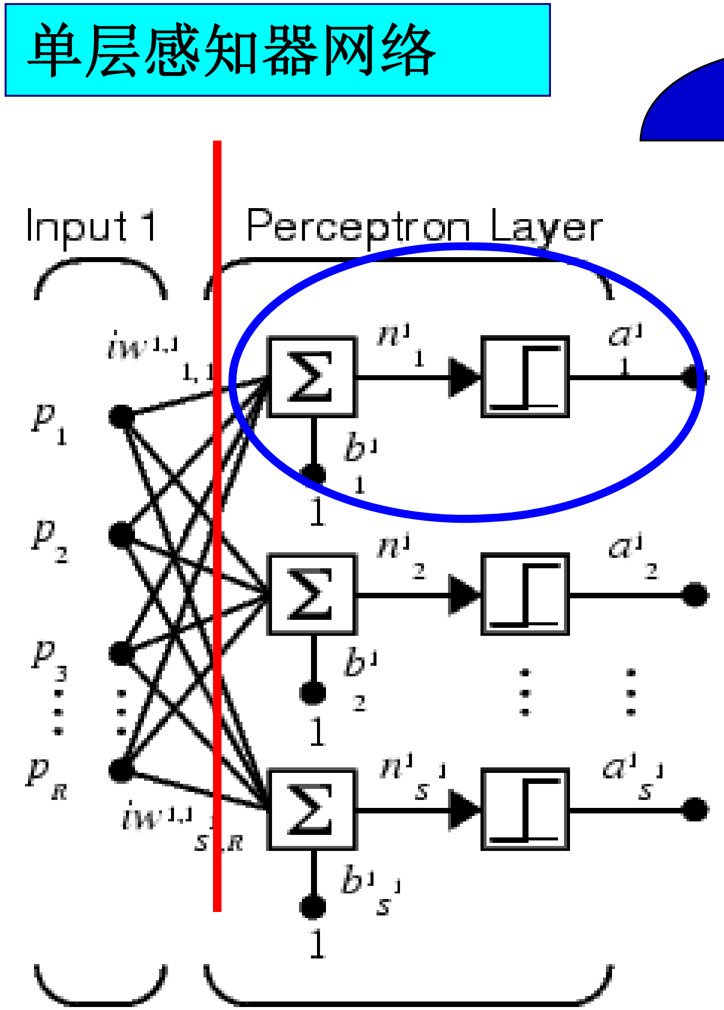
  相邻层的节点间有连接(全连接、局部连接)；

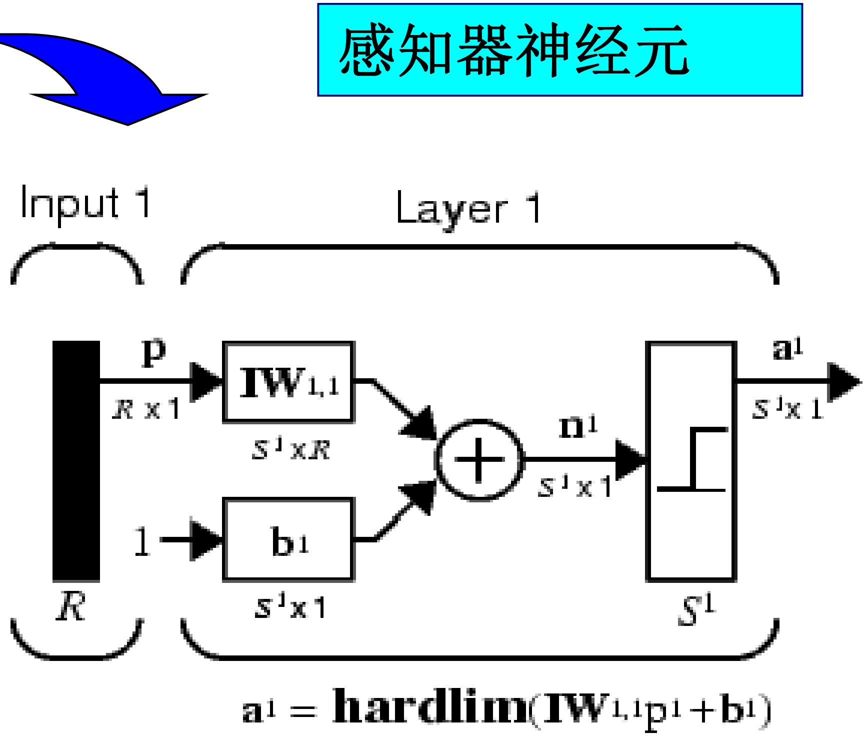
  层内节点间、跨层节点间无连接

- 输入信号由低层向高层单向流动，无反馈

--称为：前馈网络。可用一有向无环图表示。

例：单层感知器神经网络-具有单个计算层的前馈神经网络





Where...

R = number of elements in Input

S^1 = number of neurons in layer 1

感知器神经元的传递函数

$$hardlim(net)=\begin{cases}1 & net \geq 0\\ 0 & net < 0\end{cases}$$

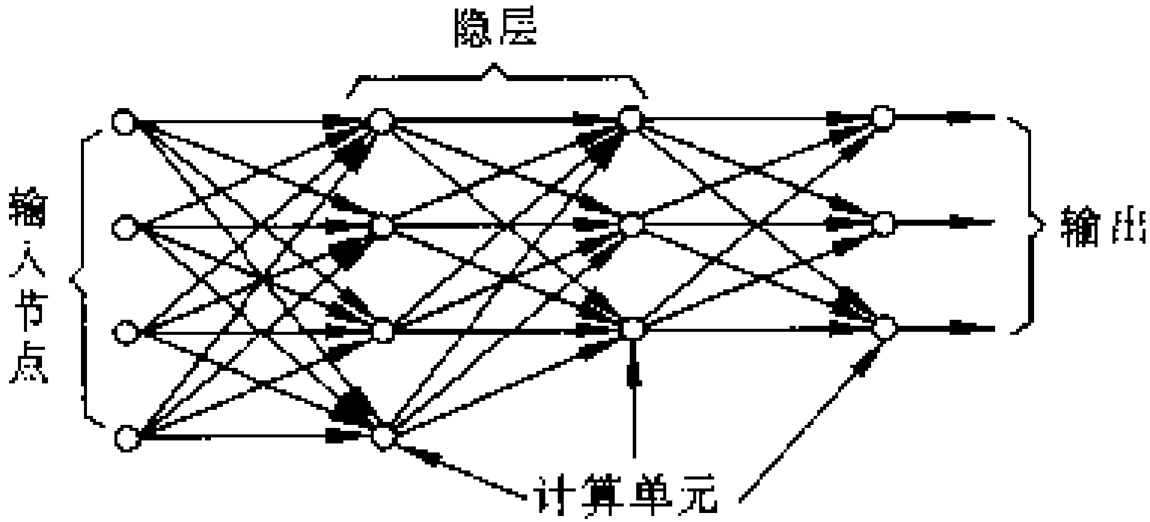
$$hardlims(net)=\begin{cases}1 & net \geq 0\\ -1 & net < 0\end{cases}$$

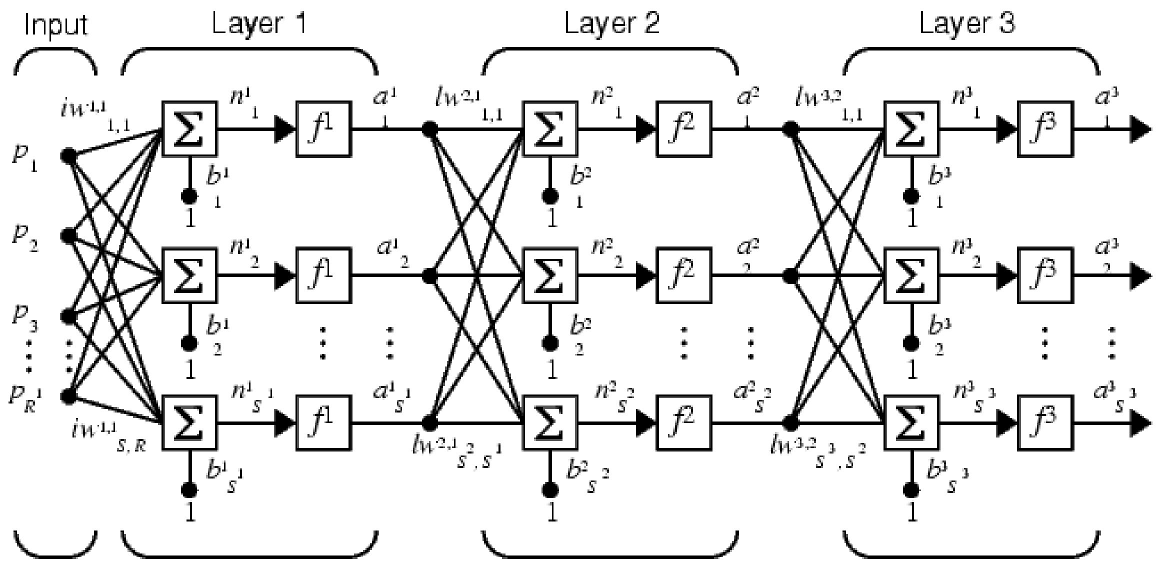
单个感知器神经元实质为线性分类器,可以完成线性可分数据的分类问题

但不能解决非线性问题(例如：“异或(XOR)”问题)

例：多层感知器(计算层数目至少为2)

具有三个计算层的前馈神经网络结构





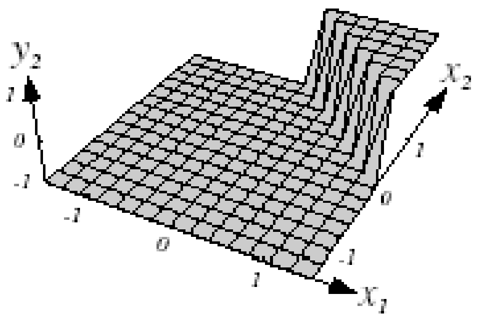
$a^1 = f^1(IW^{1,1}p + b^1)$　　$a^2 = f^2(LW^{2,1}a^1 + b^2)$　　$a^3 = f^3(LW^{3,2}a^2 + b^3)$

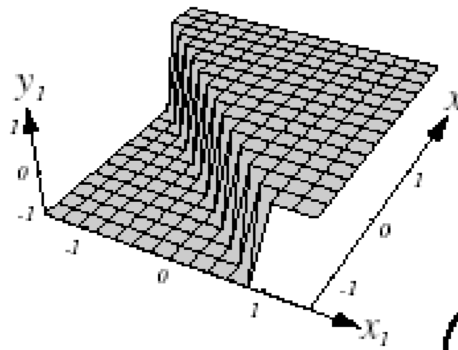
$$a^3 = f^3(LW^{3,2} f^2(LW^{2,1} f^1(IW^{1,1}p + b^1) + b^2) + b^3)$$

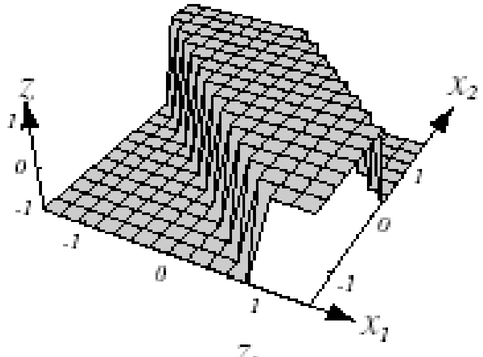
(2) 多层感知器(MLP) 的一致逼近性

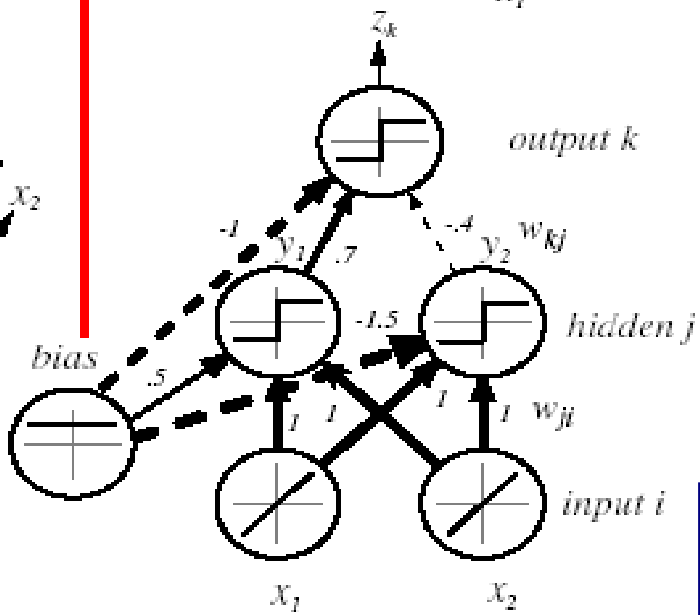
- 三层或三层以上的前馈网络(计算层数目至少为2)通常称为多层感知器
- 多层感知器的适用范围大大超过单层网络。
- 多层神经元组合，可以实现复杂的空间形状分割

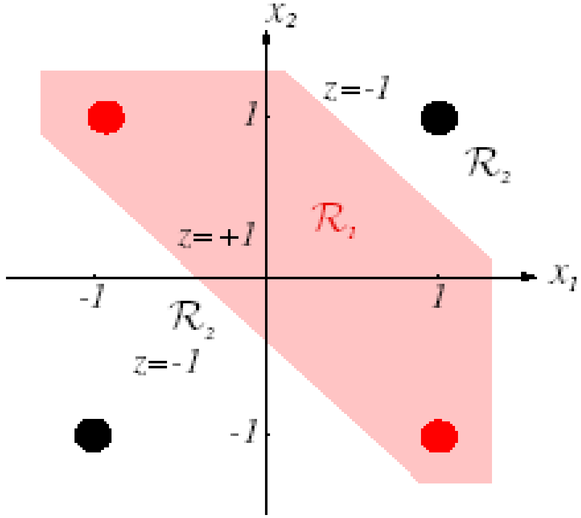
例：三层感知器神经网路实现“XOR 异或”形式的分类











两类问题的非线性分类

ω_1类：$[1,1]^T, [-1,-1]^T$

ω_2类：$[-1,1]^T, [1,-1]^T$

一般的前馈运算

$$\begin{cases} \text{输入层节点数} \boldsymbol{n}_0 \\ \text{隐含层节点数} \boldsymbol{n}_1, \quad \text{激活函数} \theta_1(\bullet) \\ \text{输出层节点数} \boldsymbol{c}, \quad \text{激活函数} \theta_2(\bullet) \end{cases}$$

$$\begin{cases} \text{输入层与隐含层:} \left[\omega_{ij}^{(1)}\right]_{\boldsymbol{n}_0 \times \boldsymbol{n}_1} \quad \left[b_{0j}^{(1)}\right]_{1 \times \boldsymbol{n}_1} \\ \text{隐含层与输出层:} \left[\omega_{\boldsymbol{jk}}^{(2)}\right]_{\boldsymbol{n}_1 \times \boldsymbol{c}} \quad \left[b_{0\boldsymbol{k}}^{(2)}\right]_{1 \times \boldsymbol{c}} \end{cases}$$

$$\boldsymbol{\omega} = \left\{\left[\omega_{ij}^{(1)}\right]_{\boldsymbol{n}_0 \times \boldsymbol{n}_1}, \left[\omega_{\boldsymbol{jk}}^{(2)}\right]_{\boldsymbol{n}_1 \times \boldsymbol{c}}\right\} \qquad \boldsymbol{b} = \left\{\left[b_{0j}^{(1)}\right]_{1 \times \boldsymbol{n}_1}, \left[b_{0\boldsymbol{k}}^{(2)}\right]_{1 \times \boldsymbol{c}}\right\}$$

输入向量 $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{\boldsymbol{n}_0} \end{bmatrix}^{\boldsymbol{T}}$ **输出向量** $\hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 & \hat{y}_2 & \cdots & \hat{y}_{\boldsymbol{c}} \end{bmatrix}^{\boldsymbol{T}}$

$$\boxed{\begin{aligned} &\hat{y}_{\boldsymbol{k}} = g_k\left(\boldsymbol{x}; \boldsymbol{\omega}, \boldsymbol{b}\right) = \theta_2\left(net_{\boldsymbol{k}}^{(2)}\right) \\ &= \theta_2\left\{\sum_{\boldsymbol{j}=1}^{\boldsymbol{n}_1} \omega_{\boldsymbol{jk}}^{(2)} \theta_1\left(net_{\boldsymbol{j}}^{(1)}\right) + b_{0\boldsymbol{k}}^{(2)}\right\} = \theta_2\left\{\sum_{\boldsymbol{j}=1}^{\boldsymbol{n}_1} \omega_{\boldsymbol{jk}}^{(2)} \theta_1\left(\sum_{i=1}^{\boldsymbol{n}_0} \omega_{ij}^{(1)} x_i + \boldsymbol{b}_{0j}^{(1)}\right) + b_{0\boldsymbol{k}}^{(2)}\right\} \\ &\boldsymbol{k} = 1, \ldots, \boldsymbol{c} \end{aligned}}$$

不同计算层的激活函数**通常不同**，同一计算层各节点激活函数也可不同。

但实际应用中：**属于同一计算层的各节点激活函数通常是一致的**

MLP的表达能力

按照***Kolmogorov***定理, 任何一个判决均可用前式所示的三层神经网络实现。

即：　只要给定足够数量的隐含层单元、适当的非线性函数以及权值，任何由输入向输出的连续映射函数均可用一个**三层前馈神经网络**实现。

当神经元的输出函数为sigmoid等函数时，三层前馈网络(含两层计算单元)可以逼近任意的多元非线性函数。

关键词：

人工神经网络、人工神经元、激活函数

前馈神经网络(单层感知器、多层感知器)

基于多层前馈神经网络的函数逼近

BP神经网络 梯度下降法、导数链式法则、误差反向传播

自编码器

监督式学习、非监督式学习

分类、回归、表示学习

多层BP神经网络学习

第一阶段：基于逐层表示学习的参数初始化；

第二阶段：基于BP算法的参数精调

主要内容

1. 人工神经网络基本知识、神经元与感知器

   生物神经网络、生物神经元

   人工神经网络、人工神经网络的基本模型、人工神经元

2. 前馈神经网络、多层感知器、及非线性分类

   单层感知器、多层感知器、基于多层前馈神经网络的函数逼近

3. BP神经网络(可用于分类、回归、监督式特征学习)

   监督式学习+BP算法(梯度下降法+导数链式法则)

   3.1 BP网络的参数学习

   3.2 BP网络的应用

4. 基于前馈神经网络的自编码器(Autoencoders)

   (1)逐层学习、非监督式学习+(2)BP算法