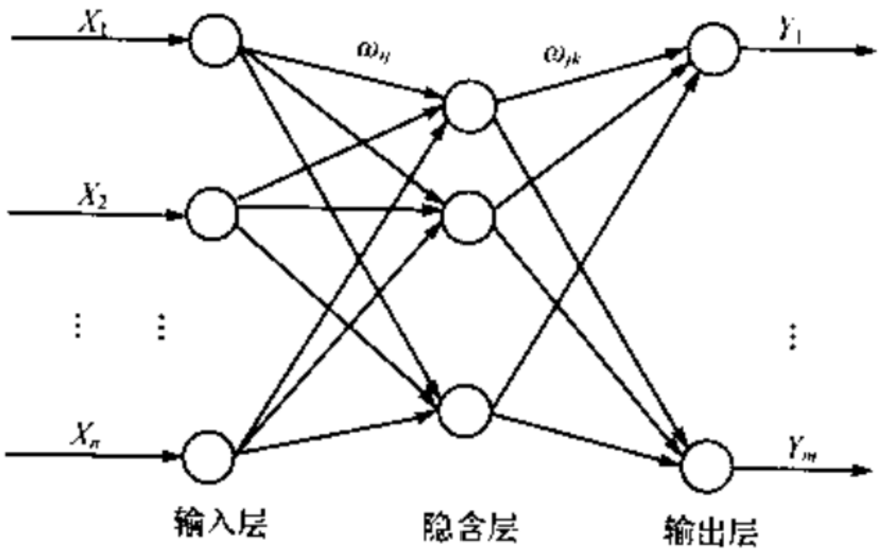
神经网络

BP神经网络/反向传播神经网络

概述

BP神经网络是一种多层前馈神经网络，特点为信号向前传递，误差反向传递。在向前传递中，出入信号冲输入层经隐含层逐层处理，直至输出层。每层的神经元状态只影响下一层神经元状态。如果输出层得不到期望输出，则转入反向传播，根据预测误差调整网络权重和阈值，从而使BP神经网络预测输出不断逼近期望输出。

BP神



经网络可以看作一个**非线性函数**，网络输入值和预测值分别为该函数的自变量和因变量。当输入节点数为n，输出节点数为m时，BP神经网络就表达了从n个自变量到m个因变量的函数映射关系。

训练BP神经网络步骤

1. 初始化网络，根据系统输入输出序列(X,Y)确定网络输入层节点数n，隐藏层节点数l，输出层节点数m，初始化输入层，隐含层和输出层神经元之间的连接权值ω_{ij}，ω_{jk}，初始化隐含层阈值a，输出层阈值b，给定学习速率和神经元激励函数。
2. 隐含层输出计算。根据输入向量X，输入层和隐含层间的连接权值ω_{ij}以及隐含层阈值a，计算出隐含层输出H。

$$H_i = f(\sum_{i=1}^{n} \omega_{ij} x_i - a_j) \qquad j = 1, 2, \cdots, l$$

其中，f为隐藏层激励函数
3. 输出层输出计算。根据隐含层输出H，链接权值ω_{jk}和阈值b，计算BP神经网络预测输出O。

$$O_k = \sum_{j=1}^{i} H_j \omega_{jk} - b_k \qquad k = 1, 2, \cdots, m$$

4.误差计算。根据网络预测输出O和期望输出Y，计算网络预测误差e，

$$e_k = Y_k - O_k \qquad k = 1, 2, \cdots, m$$

5.权值更新。根据网络预测误差e更新网络连接权值ω_{ij}，ω_{jk}.

$$\omega_{ij} = \omega_{ij} + \eta H_j (1 - H_j) x(i) \sum_{k=1}^{m} \omega_{jk} e_k \qquad i = 1, 2, \cdots, n; j = 1, 2, \cdots, l;$$
$$\omega_{jk} = \omega_{jk} + \eta H_j e_k \qquad J = 1, 2, \cdots, l; k = 1, 2, \cdots, m;$$

式中η为学习速率
6.阈值更新。根据网络预测误差e更新网络节点阈值a，b。

$$a_j = a_j + \eta H_j(1-H_j)\sum_{k=1}^{m}\omega_{jk}e_k \qquad j=1,2,\cdots,l$$
$$b_k = b_k + e_k \qquad k=1,2,\cdots,m$$

7.判断算法迭代是否结束，若未结束，返回步骤二。

隐含层节点数选择

最佳隐藏节点数选择可参考以下公式：

$$l < n - l$$
$$l < \sqrt{(m-n)} + a$$
$$l = log_2 n$$

式中，n为输入层节点数；l为隐含层节点数；a为$0-10$间的数字。

附加动量法

BP神经网络的采用梯度修正法作为权值和阈值的学习算法，从网络预测误差的负梯度方向修正权值和阈值，没有考虑以前的经验积累，学习过程收敛缓慢，可以采用附加动量的权值学习公式：

$$\omega(k) = \omega(k-1) + \Delta\omega(k) + a[\omega(k-1) - \omega(k-2)]$$

式中$\omega(k)$，$\omega(k-1)$，$\omega(k-2)$分别为k,$k-1$,$k-2$时刻的权值，a为动量学习率。

变学习率学习算法

BP神经网络学习率η的取值在$[0-1]$之间，学习率η越大，对权值的修改越大，网络学习速度越快。但过大的学习率η会使权值学习过程中产生震荡，过小的学习率使网络收敛过慢，权值难以趋于稳定。变学习率方法是指学习概率η在BP神经网络进化初期较大，网络收敛迅速，随着学习过程的进行，学习率不断减小，网络趋于稳定。变学习率计算公式为：

$$\eta(t) = \eta_{max} - t(\eta_{max} - \eta_{min}) - t_{max}$$

式中，η_{max}为最大学习率；η_{mia}为最小学习率；t_{max}为最大迭代次数；t为当前迭代次数

BP神经网络工具箱

`newff` **BP神经网络参数设置函数**

```
net = newff(P,T,S,TF,BTF,BLF,PF,IPF,OPF,DDF)
```

该函数构建了一个BP神经网络，其中：
`P` :输入数据矩阵
`T` :输出数据矩阵
`S` :隐含层节点数

`TF` :节点传递函数，包括硬限幅传递函数 `hardlim` ，对称硬限幅传递函数 `hardlims` ，线性传递函数 `purelin` ，正切S型传递函数 `tansig` ，对数S型传递函数 `logsig` 。

`BTF` :训练函数，包括梯度下降BP算法训练函数 `traingd` ；动量反传的梯度下降BP算法训练函数 `traingdm` ；动态自适应学习率的梯度下降BP算法训练函数 `traingda` ；动量反传和动态自适应学习率的梯度下降BP算法训练函数 `traingdx` ； `Levenberg_Marquardt` 的BP算法训练函数 `trainlm` 。

`BTF` :网络学习函数，包括BP学习规则 `learngd` ，带动量项的BP学习规则 `learngdm` 。

其他参数一般采用系统默认参数。

`train` **BP神经网络训练函数**

```
[net,tr] = train(NET,X,T,Pi,Ai)
```

该函数用训练数据训练BP神经网络，其中：
`NET`：待训练网络。
`X`：输入数据矩阵。
`T`：输出数据矩阵
其他参数一般采用系统默认参数。

`sim` **BP神经网络预测函数**

```
y = sim(net,x)
```

该函数用训练好的BP神经网络预测函数输出，其中：
`net`：训练好的网络。
`x`：输入数据。
`y`：网络预测数据。

多隐含层BP神经网络

多隐含层泛化能力强，预测精度高，但训练时间长。可以通过 `newff` 函数中的第三个参数构建多隐含层的BP神经网络：

```
net = newff(inputn, outputn, [5,5])
```

该语句构架了双隐含层的BP神经网络，每个隐含层的节点数为5。

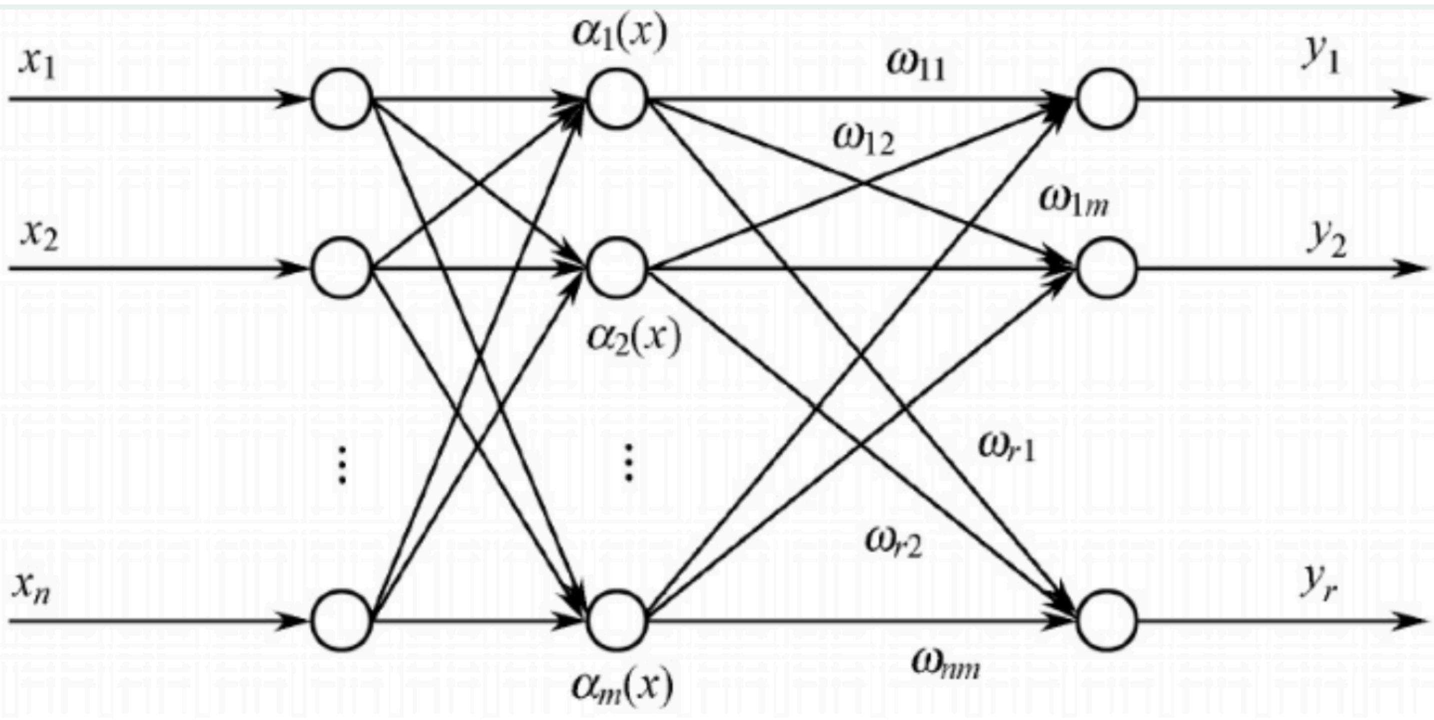
节点传递函数

隐含层和输出层函数的选择对BP神经网络预测精度有较大影响。一般隐含层节点传递函数选用 `logsig` 或者 `tansig` 函数，输出层传递函数选择 `tansig` 或 `purelin` 函数

RBF神经网络/径向基神经网络

概述

径向基函数神经网络（Radial Basis Function Neural Network，RBF神经网络）是一类常用的三层前馈网络，既可用于函数逼近，也可用于模式分类。与其他类型的人工神经网络相比，RBF网络有生理学基础，结构简单，学习速度快，优良的逼近性能和泛化能力等特点。



径向基函数的网络结构

RBF

径向基函数是一个取值仅仅依赖于离原点距离的实值函数，也就是$\phi(x) = \phi(||x||)$,或者还可以是到任意一点c的距离，c点称为中心点，也就是$\phi(x, c) = \phi(||x - c||)$。任意一个满足$\phi(x) = \phi(||x||)$特性的函数$\Phi$都叫做径向基函数，标准的一般使用欧氏距离(也叫做欧式径向基函数)。
常用的径向基函数包含：
Gaussian 函数

$$\phi(x) = e^{-\frac{r^2}{2\sigma^2}}$$

反常S型函数

$$\phi(x) = \frac{1}{1+e^{\frac{r^2}{\sigma^2}}}$$

拟多二次函数

$$\phi(x) = \frac{1}{(r^2+c^2)^{\frac{1}{2}}}$$

变种高斯函数

$$\phi(x) = e^{-\frac{||x-\mu_t||^2}{\sigma_t^2}}$$

其中μ_t为中心点，σ_t为径基宽度。径基宽度决定了径向基函数下降的快慢.

RBF神经网络

RBF神经网络的拓扑结构是一种三层前向网络:

1. 输入层由信号源结点构成,仅起到数据信息的传递作用，对输入信息不进行任何变换;
2. 第二层为隐含层，结点数视需要而定，隐含层神经元的核函数(作用函数)为高斯函数，对输入信息进行空间映射变换;
3. 第三层为输出层，它对输入模式做出响应，输出层神经元的作用函数为线性函数，对隐含层神经元输出的信息进行线性加权后输出，作为整个神经网络的输出结果。
   由此可知，RBF求解的参数有3个：基函数的中心μ_t、方差σ_t以及隐含层到输出层的权值ω_{ij}。

RBF网络训练流程

1.选择核函数

$$\phi(x_i, c_i) = e^{-\frac{||x-\mu_t||^2}{\sigma_t^2}}$$

其中c_j为第i个神经元的中心; σ为高斯核的宽度，$||x_i - c_i||$为样本x_i到中心点c_j的欧氏距离。
RBF网络定义为：

$$f(x) = \sum_{j=1}^{q} \omega_j \psi(x, c_j)$$

其中ω_j为第j个神经元的权重。
2. 定义误差函数为均方误差，目标是为了最小化误差函数：

$$E = \frac{1}{2m}\sum_{i=1}^{m} e_i^2 = \frac{1}{2m}\sum_{i=1}^{m}(f(x)-y)^2 = \frac{1}{2m}\sum_{i=1}^{m}(\sum_{j=1}^{q}\omega_j\psi(x,c_j)-y)^2$$

利用BP算法反向传播误差，并利用梯度下降法分别求得RBF网络参数优化的方向。
3. 输出层的神经元线性权重迭代公式

$$\Delta\omega = \frac{\delta E}{\delta\omega} = \frac{1}{m}\sum_{i=1}^{m}(f(x)-y)\psi(x,c) = \frac{1}{m}\sum_{i=1}^{m} e_i\psi(x,c)$$
$$\omega_{k+1} = \omega_k - \eta\Delta\omega$$

4.隐含层的神经元中心点迭代公式

$$\Delta c_j = \frac{\delta E}{c_j} = \frac{1}{m\sigma_j^2}\sum_{i=1}^{m}(f(x)-y)\omega\psi(x,c_j)(x-c_j)$$
$$c_{k+1} = c_k - \eta\Delta c$$

5.隐含层的高斯核宽度迭代公式

$$\Delta \delta_j = \frac{\delta E}{\delta_j} = \frac{1}{m\delta_j^3}\sum_{i=1}^{m}(f(x)-y)\omega\psi(x,c_j)||x_i-c_j||^2$$
$$\delta_{k+1} = \delta_k - \eta\Delta \delta$$

6. 对RBF中不同的参数分别设置不同的学习率，经过多轮迭代直至误差函数收敛，结束训练