

径向基(RBF)神经网络

机器学习研究室

计算机科学与技术学院
吉林大学

径向基(RBF)神经网络

属于局部逼近网络

插值问题描述:考虑一N维空间到一维空间的映射.

设N维空间有P个输入向量 X_p , $p=1,2,\dots,P$. 它们在输

入空间相应的目标值为 d_p , $p=1,2,\dots,P$. 插值的目的

是寻找一个非线性映射函数 $F(X)$,使得满足下述插

值条件 $F(X_p) = d_p$, $p=1,2,\dots,P$

径向基(RBF)神经网络

径向基函数解决插值问题:

选择P个基函数,对应每个训练数据: $\varphi(\|X - X_p\|)$, $p = 1, 2, \dots, P$

基函数的自变量为X与中心 X^p 的距离,由于距离是

径向同性的,因此称径向基函数.基于径向基函数的

插值函数定义为基函数的线性组合: $F(X) = \sum_{p=1}^P w_p \varphi(\|X - X_p\|)$

代入插值条件 $\sum_{p=1}^P w_p \varphi(\|X_i - X_p\|) = d_i, i = 1, 2, \dots, P$

得到关于 w_p 的P阶线性方程组.令 $\varphi_{ip} = \varphi(\|X_i - X_p\|), i, p = 1, 2, \dots, P$

$$\begin{pmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1P} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2P} \\ \vdots & \vdots & & \vdots \\ \varphi_{P1} & \varphi_{P2} & \cdots & \varphi_{PP} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_P \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_P \end{pmatrix}$$

径向基(RBF)神经网络

写成向量形式为: $\Phi \cdot W = d$

Φ 称为插值矩阵,若其可逆,则可由上式解出 W : $W = \Phi^{-1} \cdot d$

Micchelli定理给出了 Φ 的可逆性条件:对于一大类

函数,如果 X_1, X_2, \dots, X_p 自各不相同,则其可逆. 大量

径向基函数满足Micchelli定理,如

1)高斯函数: $\varphi(r) = \exp(-\frac{r^2}{2\sigma^2})$

2)Reflected Sigmoidal(反演S型)函数

$$\varphi(r) = \frac{1}{1 + \exp(\frac{r^2}{\sigma^2})}$$

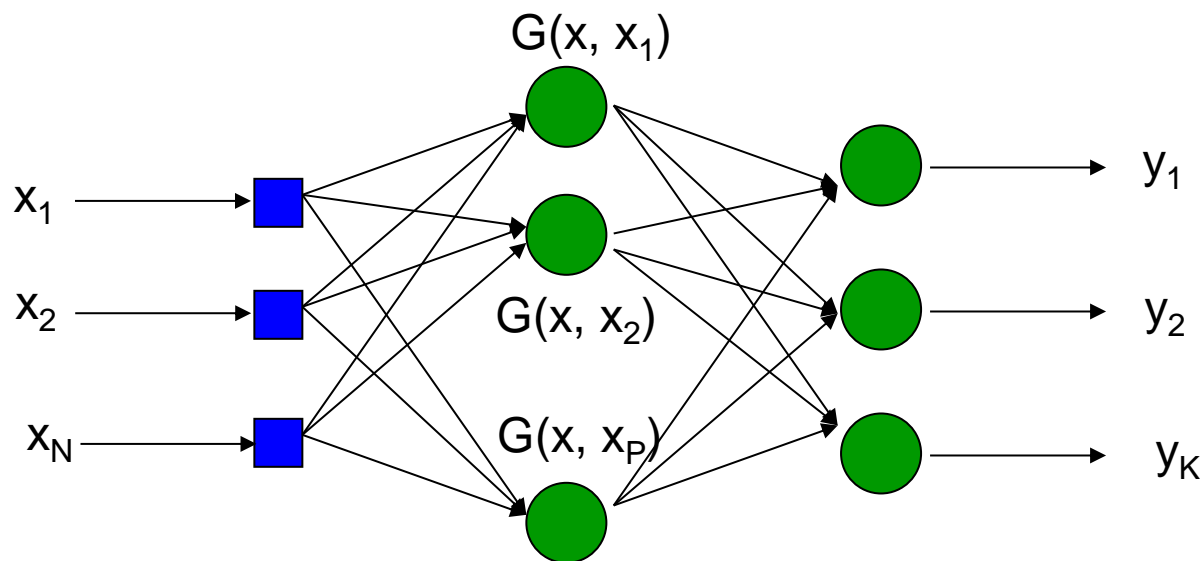
3)Inverse multiquadrics (拟多二次)函数

$$\varphi(r) = \frac{1}{(r^2 + \sigma^2)^{1/2}}$$

径向基(RBF)神经网络

完全内插存在的问题(正则化RBF网络):

- 1) 经过所有训练数据点, 当存在噪声时, 泛化能力差
- 2) 径向基函数数目与训练样本数相同, 当训练样本数远远大于系统的固有自由度时, 问题是超定的, 插值矩阵求逆容易不稳定



径向基(RBF)神经网络

模式可分性:

设 F 为 P 个输入模式 X_p , ($p=1,2,\dots,P$)的集合.每个模式必属于 F_1 和 F_2 的某一类.若存在一个输入空间的超曲面,使得分别属于 F_1 和 F_2 的点分成两部分,就称这些点的二元划分关于该曲面是可分的.若该曲面为线性方程确定的超平面,则称这些点的二元划分关于该平面是线性可分的.

径向基(RBF)神经网络

模式可分性:

设由一组函数构成的向量

将原来N维空间的P个模式点映射到新的M维空间

($M > N$)的相应点上.若在该M维空间上存在M维向量W

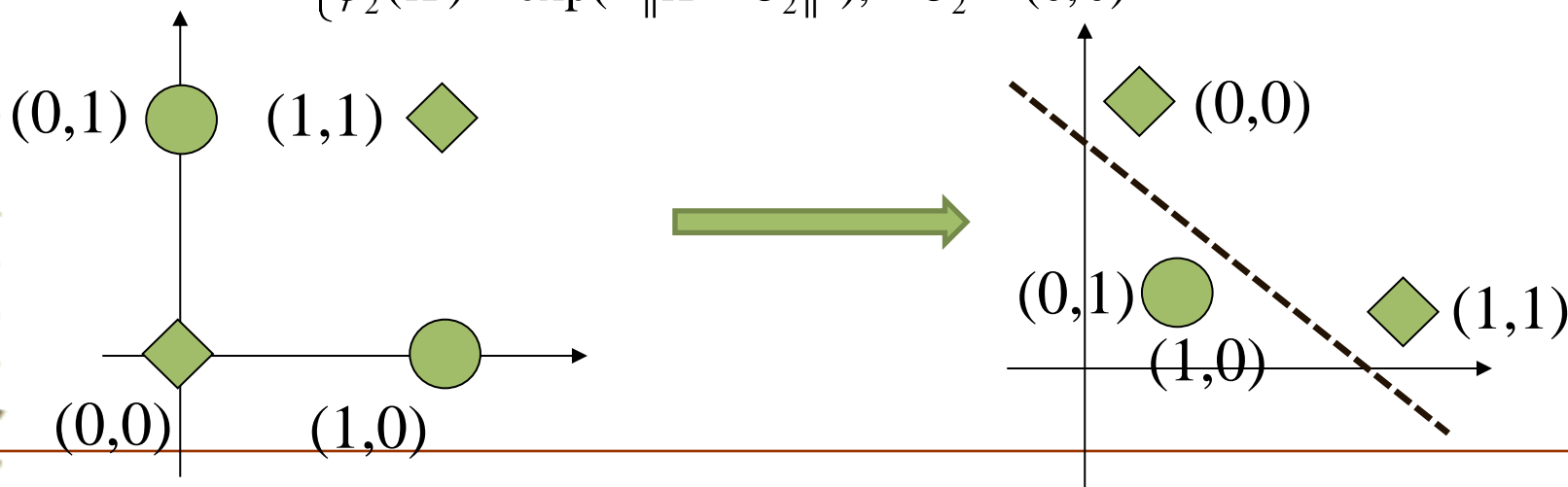
$$\begin{cases} W^T \varphi(X) > 0, & X \in F^1 \\ W^T \varphi(X) < 0, & X \in F^2 \end{cases}$$

则由线性方程 $W^T \varphi(X) = 0$ 确定了M维Φ空间中的一个分界超平面.该超平面使得映射到M维Φ空间的P个点线性可分.而在N维X空间, $W^T \varphi(X) = 0$ 描述的是X空间的一个超曲面,它将原空间的P个模式点分成两类.

径向基(RBF)神经网络

RBF网络将输入空间的模式点非线性映射到一个高维空间的作法是:设置一隐层,令 $\varphi(x)$ 为隐节点的激活函数,并令隐节点数 M 大于输入节点数 N .若 M 足够大,则在隐空间是线性可分的.从隐层到是输出层可采用与感知器类似的解决线性可分问题的算法. 如:

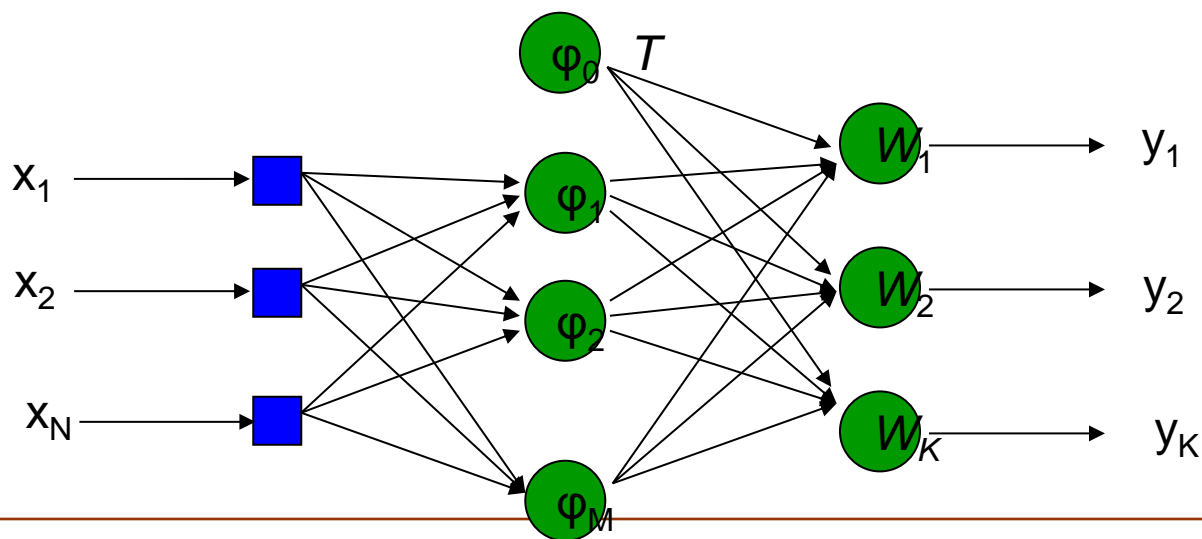
$$\begin{cases} \varphi_1(X) = \exp(-\|X - C_1\|^2), & C_1 = (1,1) \\ \varphi_2(X) = \exp(-\|X - C_2\|^2), & C_2 = (0,0) \end{cases}$$



径向基(RBF)神经网络

广义RBF网络:

- 1) 径向基函数数目 M 与训练样本数 N 不同,且一般 $M \ll N$
- 2) 径向基函数的中心不再限制在数据点上,由训练确定
- 3) 各径向基函数的扩展常数不再统一,由训练确定
- 4) 输出函数的线性中包含阈值参数,用于补偿函数在样本集上的平均值与目标之平均值之间的差别



径向基(RBF)神经网络

广义RBF网络的学习算法:

结构设计(多凭经验)和参数设计(3种参数:各基函数的中心,扩展常数以及输出节点的权值)

输出节点的权值一般由有监督的学习算法确定

各基函数的中心及扩展常数可由下列三种方法确定:

- 1)数据中心从样本中选取:样本密集的地方中心多些,稀疏的地方少些.若数据均匀分布,中心也可均匀分布.总之,选出的数据中心应有代表性.扩展常数由分布确定,如 $\delta = \frac{d_{\max}}{\sqrt{2M}}$, d_{\max} 是数据中心的最大距离, M 是中心数目.
- 2)数据中心的自组织选择(k-means聚类算法)
- 3)数据中心的监督学习算法

径向基(RBF)神经网络

广义RBF网络的k-means聚类学习算法:

首先估计中心的数目M, 设 $C(k)$ 表示第k次迭代时的中心.

1) 初始化中心: $c_1(0), c_2(0), \dots, c_M(0)$

2) 计算各样本点与聚类中心的欧氏距离:

$$\|X_p - c_j(k)\|, p = 1, 2, \dots, P; j = 1, 2, \dots, M$$

3) 相似匹配: 当 $\|X_p - c_{j^*}(k)\| = \min_j \|X_p - c_j(k)\|, p = 1, 2, \dots, P$ 时 X_p 被归为第 j^* 类.

4) 更新各类聚类中心. (i) 均值方法 $c_j(k+1) = \frac{1}{N_j} \sum_{X \in U_j(k)} X$

(ii) 竞争学习算法

$$c_j(k+1) = \begin{cases} c_j(k) + \eta[X_p - c_j(k)] & j = j^* \\ c_j(k) & j \neq j^* \end{cases}$$

径向基(RBF)神经网络

广义RBF网络的k-means聚类学习算法:

5) $k++$,若不满足终止条件($C(k)$ 的改变量小于阈值)转2

扩展常数的确定:设 $d_j = \min_i \|c_j - c_i\|$ 则扩展常数可取为 $\delta_j = \lambda \cdot d_j$

输出层权值的确定: (i) 最小均方算法(类似感知器算法)

(ii) 伪逆法:令 $\phi_{pj} = \phi(\|X_p - c_j\|)$, $p = 1, 2, \dots, P$; $j = 1, 2, \dots, M$

则隐层输出矩阵为 $\hat{\Phi} = (\phi_{pj})_{P \times M}$, 令 $W = (w_1, w_2, \dots, w_M)$

则 $F(X) = \hat{\Phi} \cdot W = d \longrightarrow W = \hat{\Phi}^+ \cdot d$

$$\hat{\Phi}^+ = (\hat{\Phi}^T \hat{\Phi})^{-1} \hat{\Phi}^T$$

径向基(RBF)神经网络

广义RBF网络的数据中心的监督学习算法:

类似BP算法的梯度下降方法(假定单输出):

$$E = \frac{1}{2} \sum_{i=1}^P e_i^2 = \frac{1}{2} \sum_{i=1}^P (d_i - F(x_i))^2 = \frac{1}{2} \sum_{i=1}^P \left(d_i - \sum_{j=1}^M w_j G(\|X_i - c_j\|) \right)^2$$

$$\Delta c_j = -\eta \frac{\partial E}{\partial c_j} \quad \Delta \delta_j = -\eta \frac{\partial E}{\partial \delta_j} \quad \Delta w_j = -\eta \frac{\partial E}{\partial w_j}$$

$$\Delta c_j = \eta \frac{w_j}{\delta_j^2} \sum_{i=1}^P e_i G(\|X_i - c_j\|) (X_i - c_j)$$

$$\Delta \delta_j = \eta \frac{w_j}{\delta_j^3} \sum_{i=1}^P e_i G(\|X_i - c_j\|) \|X_i - c_j\|^2$$

$$\Delta w_j = \eta \sum_{i=1}^P e_i G(\|X_i - c_j\|)$$

径向基(RBF)神经网络

广义RBF网络的数据中心的监督学习算法:
类似BP算法的梯度下降方法(每个数据修正一次):

$$E = \frac{1}{2} e^2$$

$$\Delta c_j = \eta \frac{w_j}{\delta_j^2} e G(\|X - c_j\|) (X - c_j)$$

$$\Delta \delta_j = \eta \frac{w_j}{\delta_j^3} e G(\|X - c_j\|) \|X - c_j\|^2$$

$$\Delta w_j = \eta e G(\|X - c_j\|)$$

径向基(RBF)神经网络

RBF网络与多层感知器的比较:

- RBF: single hidden layer
MLP: hidden layer number ≥ 1
- RBF: hidden layer is nonlinear, output layer is linear
MLP: hidden and output layers are nonlinear for classification, but output layer becomes linear for nonlinear regression.
- RBF: simple for analytical solution
MLP: difficult for analytical solution
- RBF: local approximation for fast learning