**智能计算第三次实验报告**

**180111226 倪宇琪**

**一、实验目的及要求**

**理解并掌握径向基函数网络（即RBF）的实验原理，并用所学对鸢尾花数据进行分类，并能使用RBF网络逼近函数。**

**二、实验内容**

**1.****基于径向基函数网络对鸢尾花数据进行分类**

**已知150朵莺尾花的4项数据指标：萼片长度sepal length(单位：cm)、萼片宽度sepal width(cm)、花瓣长度petal length(cm)、花瓣宽度petal width(cm)。以及它们3种类别：****Iris Setosa、Iris Versicolour、Iris Virginica**

**试利用RBF神经网络建立合适的分类模型。**

**数据来源****http://archive.ics.uci.edu/ml/datasets/Iris。**

**2.****利用径向基函数网络逼近以下函数**

**三、实验基本原理**

**3.1 径向基函数概述**

**1985年，Powell提出了多变量插值的径向基函数（RBF）方法。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Φ（x）=Φ(‖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Φ（x，c）=Φ(‖x-c‖)。任意一个满足Φ（x）=Φ(‖x‖)特性的函数Φ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），尽管其他距离函数也是可以的。最常用的径向基函数是高斯核函数 ,形式为 k(||x-xc||)=exp{- ||x-xc||^2/(2\*σ)^2) } 其中x\_c为核函数中心,σ为函数的宽度参数 , 控制了函数的径向作用范围。**

**径向基函数（Radialbasis function）是一个取值仅仅依赖于离原点距离的实值函数。 也即**

**或者是到任意一点的距离，点成为中心点，也即：**

**任意一个满足上述两点特性的函数都被称作径向基函数，标准的径向基函数使用欧式距离。最常用的径向基函数是高斯核函数，其形式为：**

**其中为核函数的中心，为函数的宽度参数，控制了函数的径向作用范围。**

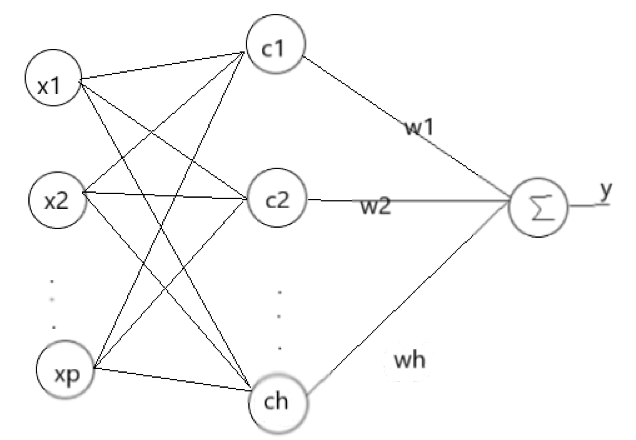
**3.2 RBF网络概述**

**RBF神经网络是一种三层神经网络，其包括输入层，隐层，输出层。与BP神经网络不同的是，其并非完全是线性的。从输入空间到隐层空间的变换是非线性的，而从隐层空间到输出层空间的变换是线性的。网络结构图如下图所示：**

**其中：**

**为输入层**

**为隐层**

****

**RBF网络的基本思想是：用RBF作为隐函数的基，由其构成隐含层空间，这样就可以将输入矢量直接映射到隐含层空间，而不需要通过权连接。当RBF网络的中心点确定后，这种映射关系也就随之确定了。而隐含层空间到输出空间的映射是线性的，即网络的输出是隐含层单元输出的线性加权和，此处的权即为网络可调参数。**

**其中，隐含层的作用是把向量从输出层的p映射到高维度的h，这样低维度线性不可分的情况到高维度就可以变得线性可分了，这就是核函数的思想。**

**由此，网络由输入到输出的映射是非线性的，而网络输出对输入的参数而言又是线性的，网络的权就可以由线性方程组直接解出，从而加快了学习速度，与Hopfield网络相比，避免了局部极小问题。**

**需要说明：**

**径向基神经网络的激活函数可表述为:**

**其中径向基神经网络的结构可得到网络的输出为：**

**采用最小二乘法的损失函数表示为：**

**3.3 RBF网络算法步骤**

**求解的参数有3个，分别是：基函数的中心、方差以及隐函数到输出层的权值。参数的求法比较多样，比如有自组织选取中心学习法，直接计算法，有监督学习法等，这里主要写明自组织选择中心学习法。**

**于无监督学习过程下，求解隐含层基函数的中心和方差。**

**选取个中心做聚类，对于高斯核函数的径向基，方差可由下列公式求解得出：**

**为所选取中心点之间的最大距离。**

**于有监督学习过程下，求解隐含层到输出层之间的权值。**

**对于权值，可使用最小二乘法直接计算得到，即对损失函数求解关于的偏导数，使其等于0，可以化简得到如下计算式子：**

**四、程序简述**

**4.1 基于径向基函数网络对鸢尾花数据进行分类**

**4.1.1数据处理**

[**鸢尾花数据150**](http://archive.ics.uci.edu/ml/datasets/Iris网站，在Data%20Fold里中下载iris.data获取Iris鸢尾花数据文件得到了150)**朵鸢尾花的各项数据指标。内容如下：**

**鸢尾花的四项指标用逗号隔开，最后一列为本行指标对应的鸢尾花的种类。**

**数据集包含3类，分别是Iris Setosa、Iris Versicolour、Iris Virginica，每类有50个实例，四项指标按顺序分别为萼片长度，萼片宽度，花瓣长度，花瓣宽度。**

**由于Iris数据集中有字符串，故在导入MATLAB前需要将其替换成数字，然后，选择该文件选择导入数据，并进入可视化界面并加载数据，由此能够使用该数据集。**

**4.1.2求解步骤**

**读取数据**

**将需要训练的数据（每一大类的前30个样本）矩阵存入****yingweihua\_property\_1、yingweihua\_property\_2、yingweihua\_property\_3、yingweihua\_property\_4中**

**建立RBF网络**

**使用新的数据集测试此网络**

**将待识别的数据（每一大类的后20个样本）存入yingweihua\_property\_5、yingweihua\_property\_6、yingweihua\_property\_7、yingweihua\_property\_8中，然后通过RBF网络进行测试**

**4.2 利用径向基函数网络逼近函数**

**4.2.1求解步骤**

**要逼近的函数为二元函数。**

**已知输入向量P和输出向量T，并知道已知的函数，通过构建径向基函数神经网络来进行曲线拟合，从而找到一个函数能够满足数据的输入与输出关系。**

**应用MATLAB神经网络工具箱中的newrb（）函数从而快速构建一个径向基函数网络，并且根据输入向量和期望值自动调整。从而实现函数逼近。**

**五、实验结果与总结**

**5.1 基于径向基函数网络对鸢尾花数据进行分类**

**用了newrb函数和newrbe函数识别率皆为100%**

**命令窗口输出：**

**NEWRB, neurons = 0, MSE = 0**

**NEWRB, neurons = 50, MSE = 1.02077e-31**

**Rbfoutput**

**GUI图像为：**

****

**5.2 利用径向基函数网络逼近以下函数**

**运行6.2的代码后得出结果：**

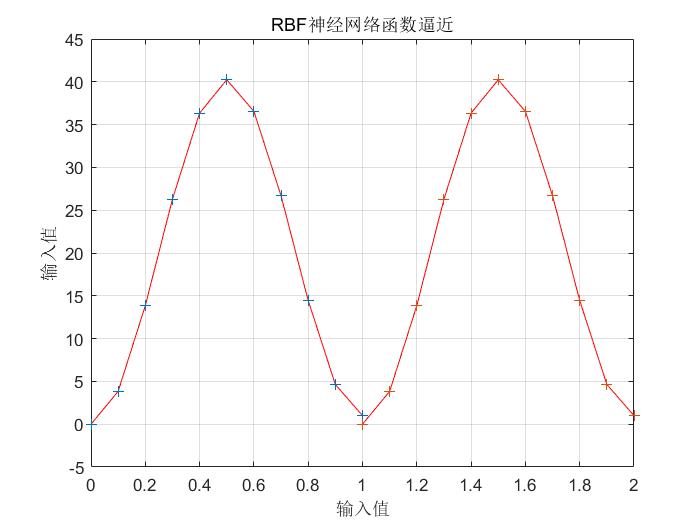
**NEWRB, neurons = 0, MSE = 212.534**

**ans =**

**8.5795e-05**

**%误差**

**输出图像为：**

****

**最终如图所示可以看出RBF网络可以精确的逼近一个单值函数**

**5.3实验总结**

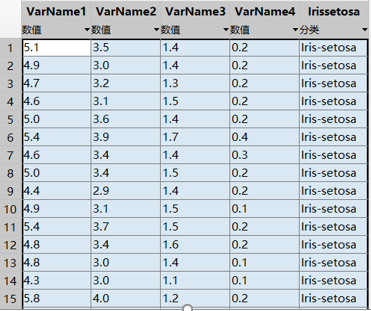
**由于RBF网络与之前做的BP神经网络示意图较为接近，通过本次实验结果与查阅的相关资料后，我做出如下总结：**

|  |  |  |
| --- | --- | --- |
|  | **RBF神经网络** | **BP神经网络** |
| **激活函数的自变量** | **输入模式与权向量的内积** | **输入模式和中心向量的距离** |
| **激活函数** | **Sidmoid函数** | **径向基函数（Guass函数）** |
| **特性** | **局部映射** | **全局逼近** |
| **中间层数** | **只有一个隐含层** | **可以有多个隐含层** |
| **训练速度** | **训练速度较快** | **训练速度较慢** |

**六、实验代码**

**6.1 基于径向基函数网络对鸢尾花数据进行分类**

**6.1.1执行数据处理后的部分截图：**

****

**实验代码如下：**

**>> clear**

**>> filename = 'D:\irisdata.txt';**

**[yingweihua\_property\_\_1, yingweihua\_property\_\_2,yingweihua\_property\_\_3, yingweihua\_property\_\_4, class] = textread(filename,'%f %f %f %f %s','delimiter',',');**

**yingweihua\_property\_1 =[yingweihua\_property\_\_1(1:30,:);yingweihua\_property\_\_1(51:80,:);yingweihua\_property\_\_1(101:130,:)];**

**yingweihua\_property\_2 =[yingweihua\_property\_\_2(1:30,:);yingweihua\_property\_\_2(51:80,:);yingweihua\_property\_\_2(101:130,:)];**

**yingweihua\_property\_3 =[yingweihua\_property\_\_3(1:30,:);yingweihua\_property\_\_3(51:80,:);yingweihua\_property\_\_3(101:130,:)];**

**yingweihua\_property\_4 =[yingweihua\_property\_\_4(1:30,:);yingweihua\_property\_\_4(51:80,:);yingweihua\_property\_\_4(101:130,:)];**

**class1 =[class(1:30,:);class(51:80,:);class(101:130,:)];**

**yingweihua\_property\_5 =[yingweihua\_property\_\_1(31:50,:);yingweihua\_property\_\_1(81:100,:);yingweihua\_property\_\_1(131:150,:)];**

**yingweihua\_property\_6 =[yingweihua\_property\_\_2(31:50,:);yingweihua\_property\_\_2(81:100,:);yingweihua\_property\_\_2(131:150,:)];**

**yingweihua\_property\_7 =[yingweihua\_property\_\_3(31:50,:);yingweihua\_property\_\_3(81:100,:);yingweihua\_property\_\_3(131:150,:)];**

**yingweihua\_property\_8 =[yingweihua\_property\_\_4(31:50,:);yingweihua\_yingweihua\_property\_\_4(81:100,:);yingweihua\_yingweihua\_property\_\_4(131:150,:)];**

**class2 =[class(31:50,:);class(81:100,:);class(131:150,:)];**

**for i = 1:length(class1)**

**switch class1{i};**

**case 'Setosa'**

**Class(i,:)=[1 0 0];**

**case 'Versicolor'**

**Class(i,:)=[0 1 0];**

**otherwise 'Virginica';**

**Class(i,:)=[0 0 1];**

**end**

**end**

**for i = 1:length(class2)**

**switch class2{i};**

**case 'Setosa'**

**yingweihua\_property\_textClass(i,:)=[1 0 0];**

**case 'Versicolor'**

**yingweihua\_property\_textClass(i,:)=[0 1 0];**

**otherwise 'Virginica';**

**yingweihua\_property\_textClass(i,:)=[0 0 1];**

**end**

**end**

**yingweihua\_property\_textClass = yingweihua\_property\_textClass';**

**for i = 1:length(Class)**

**P(i,:) =[yingweihua\_property\_1(i,:),yingweihua\_property\_2(i,:),yingweihua\_property\_3(i,:),yingweihua\_property\_4(i,:)];**

**T(i,:) = Class(i,:);**

**end**

**P = P';**

**T = T';**

**Class = Class';**

**for i = 1:length(yingweihua\_property\_5\*3)**

**textP(i,:) =[yingweihua\_property\_5(i,:),yingweihua\_property\_6(i,:),yingweihua\_property\_7(i,:),yingweihua\_property\_8(i,:)];**

**end**

**textP = textP';**

**net = newrb(P,Class);**

**Rbfoutput = sim (net,textP)**

**[s1,s2] = size(Rbfoutput);**

**count = 0;**

**for i = 1:s2**

**[m ,index] = max(Rbfoutput(:,i));**

**[l,std] = max(yingweihua\_property\_textClass(:,i));**

**if(index==std)**

**count = count + 1;**

**end**

**end**

**sprintf('识别率是%3.3f%%',100\*count/s2)**

**clear,clc;**

**x1=0:0.1:1;**

**x2=1:0.1:2;**

**f=20+x1.\*x1-10.\*cos(2.\*pi.\*x1)-10.\*cos(2.\*pi.\*x2);**

**net=newrb([x1;x2],f,0.001,2); %平方和误差小于0.001**

**y=net([x1;x2]);%神经网络输出值**

**t=[x1;x2];**

**plot(t,y,'r');%仿真的函数图像**