doi: 10.7690/bgzdh.2015.11.022

基于人工神经网络的核素识别分析方法

刘议聪[1]，王 伟[2]，牛德青[1]
（1. 中国兵器工业第五八研究所智能检测技术部，四川 绵阳 621000；
2. 中国兵器工业第五八研究所军品部，四川 绵阳 621000）

摘要：针对传统的方法解析γ谱数据实现核素识别存在步骤多、精度低、专业知识要求高、识别速度慢等缺点，提出了一种基于人工神经网络的核素识别分析方法。在对全谱γ数据进行主成分分析的基础上，提取出γ谱的主要特征，将此特征信息输入人工神经网络，利用BP网络算法和RBF网络算法可快速地完成γ谱解析。分析结果表明：该方法降低了对探测器能量分辨率的要求，同时避免了寻峰、能量刻度与效率刻度等问题，简化了核素识别的过程，有效地提高了放射性核素的快速识别能力。

关键词：神经网络；快速识别；γ谱；主成分分析
中图分类号：TP183 **文献标志码：**A

Nuclide Identification and Analysis Using Artificial Neural Network

Liu Yicong[1], Wang Wei[2], Niu Deqing[1]
(1. *Department of Intelligent Detection Technology*, *No*. 58 *Research Institute of China Ordnance Industry*, *Mianyang* 621000, *China*; 2. *Department of Military Products*, *No*. 58 *Research Institute of China Ordnance Industry*, *Mianyang* 621000, *China*)

Abstract: The traditional method of nuclide identification by analyzing the data in γ spectrum takes various steps and suffers low accuracy and identification speed. In addition, it requires much more professional knowledge and work. This paper proposed a novel method of nuclide identification and analysis using neural network. Based on the principal component analysis (PCA) of all the data in γ spectrum, it extracts the main features of γ spectrum and then imports the extracted information into the neural network. It can implement fast analysis of the data in γ spectrum with high accuracy by using BP and RBF network algorithms. The experimental results show that our proposed method does not require high energy resolution of detector as well as solves the problems of peak searching, energy calibration and efficiency calibration. Therefore, it simplifies the process of nuclide identification and significantly improves the ability of fast identification of radioactive nuclide.

Keywords: neural network; fast identification; γ spectrum; PCA

0 引言

人工神经网络的信息处理能力有3个显著的特点：一是非线性特性；二是具有大量的并行分布结构；三是学习和归纳能力。因此，它在建模、时间序列分析、模式识别、信号处理以及控制等方面得到了广泛的应用[1]。

传统的γ能谱识别方法是：通过数字多道分析仪得到原始的γ能谱数据，经平滑、数字滤波，扣除本底、寻峰等分析，再进行能量刻度，得到道址与能量的函数关系，通过某个峰位的道址，获取其对应的能量，再经反检索核素库后，即可得到该处峰位对应的核素种类。而利用人工神经网络只需要将数字多道分析仪测得的γ能谱数据直接输入非线性网络，通过非线性运算，就可以很快地得到核素识别结果。相比于传统的核素识别方法，利用人工神经网络进行核素识别效率更高，对核技术专业的“门槛”要求更低。从20世纪80年代到现在，国内外已有很多科研工作者将人工神经网络应用于各种能谱分析，也拓宽了人工神经网络的应用领域。基于此，笔者通过对测得的γ谱预先进行主成分分析(principal component analysis，PCA)、提取特征、降低输入维度后，将数据输入人工神经网络，通过BP(back propagation)网络与径向基函数(radial-basis function，RBF)网络2种算法对核素识别进行研究。

1 理论基础

1.1 BP网络原理

BP网络是采用误差逆向传播学习算法的多层前馈神经网络[2]，对非线性系统具有很强的模拟能力，已成为目前应用最为广泛的人工神经网络算法[3]。BP神经网络模型由输入层、隐含层和输出层组成[4]。BP网络神经元常用的传递函数包括log-sigmoid型函数logsig，tan-sigmoid型函数tansig以及线性函数purelin，其中sigmoid型传递函数将网络输出限制在[–1,+1]内，线性函数purelin可以使

收稿日期：2015-06-26；修回日期：2015-08-09
作者简介：刘议聪(1991—)，男，四川人，硕士，助理工程师，从事智能信号处理与控制研究。

网络输出取到任意值[5]。因此，在隐层中通常采用 sigmoid 函数进行中间结果的传递，而在最后输出层用线性函数对输出进行值域扩展。一个典型 2 层神经元 BP 神经网络如图 1 所示，其隐藏层传递函数为 tansig，输出层传递函数为 purelin 函数。

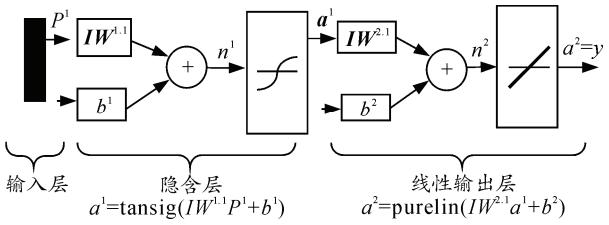

图 1　由 2 层神经元构成的 BP 网络结构

BP 网络的学习属于有监督学习，训练过程需要提供输入向量 $\boldsymbol{P}$ 和期望输出 $\boldsymbol{T}$，训练过程中的权值和偏差根据网络误差性能进行调整，最终实现期望的功能。前向型神经网络采用均方误差作为默认的网络性能函数，网络学习过程是使均方误差最小化的过程。BP 网络的学习过程可描述为[6]：工作信号正向传播和误差信号反向传播 2 个阶段。第 1 阶段输入信号从输入层经隐含层传至输出层，在输出端产生输出信号，此为正向传播；第 2 阶段误差信号由输出层开始逐层向前传播，同时网络权值 $\boldsymbol{w}$ 和阈值 b 由误差反馈进行调节以使网络的实际输出更加接近期望输出，让网络性能函数值达到预定的目标。

BP 网络的学习算法步骤可以归纳为：

1) 设置变量和参量：$\boldsymbol{X}_K=[x_{k1},x_{k2},\cdots,x_{kM}]$ (k=1,2,⋯,N)为输入向量，即训练样本，N 为训练样本个数；$\boldsymbol{w}_{MI}(n)$为第 n 次迭代时输入层与隐层 I 之间的权值向量，$\boldsymbol{w}_{IJ}(n)$为第 n 次迭代时隐层 I 与隐层 J 之间的权值向量，$\boldsymbol{w}_{JP}(n)$为第 n 次迭代时隐层 J 与输出层之间的权值向量。$\boldsymbol{Y}_K=[y_{k1},y_{k2},\cdots,y_{kP}]$，($k$=1,2,⋯,$N$)为第 n 次迭代时网络的实际输出。$\boldsymbol{T}_K=[t_{k1},t_{k2},\cdots,t_{kP}]$($k$=1,2,⋯,$N$)为期望输出，$\eta$ 为学习速率，n 为迭代次数。

2) 初始化赋值。

3) 对随机输入样本 $\boldsymbol{X}_K$，n=0。前项计算 BP 网络每层神经元的输入信号 u 和输出信号 v，其中

$$v_p^P(n)=y_{kp}(n),\ p=1,2,\cdots,P。\tag{1}$$

4) 由期望输出 $\boldsymbol{T}_K$ 和上一步求得的实际输出 $\boldsymbol{Y}_K(n)$ 计算误差 $E(n)$，判断其是否满足要求，若满足，转至 7)；不满足，则转至 5)。

5) 判断 n+1 是否大于最大迭代次数，若大于转至 7)，否则，对于输入样本 $\boldsymbol{X}_K$，反向计算每层神经元的局部梯度 δ。其中：

$$\delta_p^P(n)=y_p(n)(1-y_p(n))(T_p(n)-y_p(n)),p=1,2,\cdots,P;\tag{2}$$

$$\delta_j^J(n)=f'(n)(u_j^J(n))(\sum_{p=1}^{P}\delta_p^P(n)w_{jp}(n),j=1,2,\cdots,J;\tag{3}$$

$$\delta_i^I(n)=f'(n)(u_i^I(n))(\sum_{j=1}^{J}\delta_j^J(n)w_{ij}(n),i=1,2,\cdots,I。\tag{4}$$

6) 按下式计算权值修正量 Δw，并修正权值；n=n+1，转至 3)：

$$\Delta w_{jp}(n)=\eta\delta_p^P(n)v_j^J(n);\tag{5}$$

$$w_{jp}(n+1)=w_{jp}(n)+\Delta w_{jp}(n);\tag{6}$$

$$\Delta w_{ij}(n)=\eta\delta_j^J(n)v_i^I(n);\tag{7}$$

$$w_{ij}(n+1)=w_{ij}(n)+\Delta w_{ij}(n);\tag{8}$$

$$\Delta w_{mi}(n)=\eta\delta_i^I(n)x_k^m(n);\tag{9}$$

$$w_{mi}(n+1)=w_{mi}(n)+\Delta w_{mi}(n)。\tag{10}$$

7) 判断是否学完所有训练样本，是，则结束，否，则转至 3)。

至此，BP 网络完成了学习，并得到了合理的网络权值 $\boldsymbol{w}$ 和阈值 b。

1.2　RBF 神经网络原理

RBF 神经网络属于多层前向神经网络。它是一种 3 层前向网络，输入层由信号源节点组成；第 2 层为隐含层，隐层单元的变换函数是对中心点径向对称且衰减的非负非线性函数；第 3 层为输出层，对输入模式作出响应。一个典型的 RBF 神经网络结构如图 2 所示。

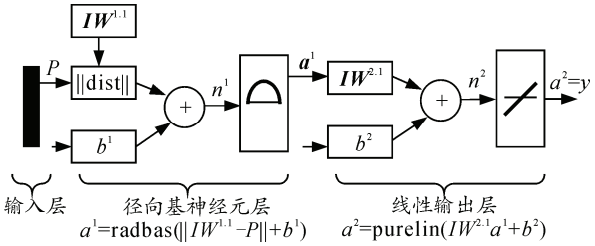

图 2　典型的 RBF 神经网络结构

RBF 神经网络的基本思想是利用径向基函数作为隐层的“基”，构成隐层空间，隐层对输入向量进行变换，将低维的模式数据变换到高维空间内，使得在低维线性不可分问题在高维空间内线性可分[1]。RBF 神经网络结构简单、训练简洁、学习收敛速度快，能够逼近任意非线性函数。

与其他类型的神经元有所不同的是，径向基神

经元传递函数的输入是权值向量和输入向量之间的向量距离与偏置 b 的乘积，图中 a_i^1 是向量 $\boldsymbol{a}^1$ 的第 i 个元素，$\boldsymbol{IW}^{1,1}{}_i$ 是权值矩阵 $\boldsymbol{IW}^{1,1}$ 的第 i 个行向量。||dist||表示权值与输入向量的点积产生的输出向量，其中各个元素是输入向量 $\boldsymbol{P}$ 与矩阵 $\boldsymbol{IW}^{1,1}$ 各行之间的向量距离。偏置向量 $\boldsymbol{b}1$ 与||dist||相乘的结果可以由矩阵点乘得到[5-6]。

径向基函数网络的隐含层采用径向基函数作为激励函数。隐含层每个神经元与输入层相连的权值向量 $\boldsymbol{w}1_i$ 和输入向量 $\boldsymbol{X}^q$（表示第 q 个输入向量）之间的距离乘上阈值 $b1_i$ 作为本身的输入，如图 3。

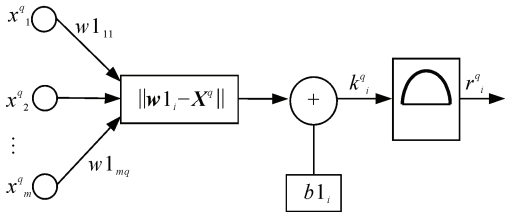

图 3　RBF 网络隐含层神经元的输入与输出

由图中结构得隐含层的第 i 个神经元的输入为

$$k_i^q=\sqrt{\sum_j(w1_{ji}-x_j^p)^2}\times b1_i。\tag{11}$$

输出为

$$r_i^q=\mathrm{e}^{-(k_i^q)^2}=\mathrm{e}^{-(\|w1_{ji}-x^q\|\times b1_i)^2}。\tag{12}$$

径向基函数的阈值 $b1_i$ 可以调节函数的灵敏度，但实际工作中更常用另一参数 $\boldsymbol{C}$（称为扩展函数）。$b1_i$ 与 $\boldsymbol{C}$ 的关系有多种确定方法，在 Matlab 神经网工具箱中，$b1_i$ 和 $\boldsymbol{C}$ 的关系为

$$b1_i=\frac{0.832\,6}{C_i}。\tag{13}$$

由此可见，C 值的大小反映了输出对输入的响应宽度。输出层的输入为各隐含层神经元输出的加权求和。由于激励函数为 purelin 函数，因此输出为

$$y^q=\sum_{i=1}^{n}r_i\times w2_i。\tag{14}$$

RBF 网络的训练过程分为 2 步：第 1 步为无监督学习，确定训练输入层与隐含层间的权值 $\boldsymbol{w}1$；第 2 步为有监督学习，确定训练隐含层与输出层间的权值 $\boldsymbol{w}2$。在训练以前，需要提供输入向量 $\boldsymbol{X}$、对应的目标向量 $\boldsymbol{T}$ 与径向基函数的扩展常数 $\boldsymbol{C}$。训练的目标是求取 2 层的最终权值 $\boldsymbol{w}1$、$\boldsymbol{w}2$ 和阈值 $\boldsymbol{b}1$、$\boldsymbol{b}2$。

1.3　特征提取原理

对数据的模式作特征提取，常用的方法是 PCA[7-8]。PCA 方法由 Turk 和 Pentlad 提出[9-10]，以 K-L（karhunen-loeve）正交变换为基础，将数据从高维空间投影到低维空间中，有效地找出数据中最“主要”的元素和结构，去除噪音和冗余，将原有的复杂数据降维，揭示隐藏在复杂数据背后的简单结构。对新求出的“主元”向量的重要性进行排序，根据需要取前面最重要的部分，将后面的维数省去，可以达到压缩数据的目的[11]，同时最大程度地保持了原有数据的信息。

2　利用神经网络进行核素识别

利用 NaI 探测器，在相同的测试环境下，分别对 ^{241}Am、^{137}Cs、^{60}Co、^{109}Cd、^{238}Pu、^{133}Ba、^{226}Rn 等 7 种放射性核素源在单核素及多核素混合源情况下测量多组数据，从谱仪中得到其对应的 γ 能谱数据（道址-计数值）。然后将得到的 γ 能谱数据按核素进行分组，选择样本总数的 2/3 作为训练样本，用以训练搭建好的神经网络；剩下样本总数的 1/3 作为测试样本用以测试训练好的神经网络的性能。

2.1　γ 谱数据的预处理

谱仪中得到的原始 γ 能谱数据为 1 024 道，从模式识别的角度来看，每一道址的计数值均为样本的一个属性，即每个核素样本的特征为 1 024 维，如果利用全谱信息，则输入节点就有 1 024 个，这样的网络显然有一些庞大，并且信息复杂，难以处理。所以，笔者先采用 PCA 将所得到的 1 024 维数据通过非线性变换投射到另一空间进行降维，在投影的主成分空间内只利用其主成分信息输入神经网络进行训练。笔者借助 Matlab 平台实现 PCA，具体实现方法为：

1) 所有的训练样本组成矩阵 $\boldsymbol{P}$，其中每一行为一个样本，每一列为样本的一个属性；

2) 将 $\boldsymbol{P}$ 的每一列进行零均值化，即减去这一列的均值；

3) 求出协方差矩阵 $\boldsymbol{C}$ 以及其特征值和对应的特征向量；

4) 将特征向量按对应特征值大小从大到小按行排列成矩阵，取前 k 行组成矩阵 $\boldsymbol{P}_1$；

5) $\boldsymbol{Y}=\boldsymbol{P}_1\times\boldsymbol{P}$ 即为降维到 k 维（$k<1\,024$）后的数据。其中 $\boldsymbol{Y}$ 为 PCA 变换系数矩阵，它保存着原始信号中的最大能量并且截断后所得的均方误差最小。将矩阵 $\boldsymbol{Y}$ 按属性进行归一化后，保存为.mat 格式的数据包。

2.2　利用 BP 学习算法进行核素识别

由于 Robert Hecht-Nielson 证明了具有一个隐

含层的 3 层 BP 神经网络可以有效地逼近任意连续函数[12]，故笔者采用一个 3 层的人工神经网络进行实验。首先读入上一步已经建立好的训练样本数据包，然后利用 Matlab 自带函数 newff 建立一个神经网络，语句格式如下：

net=newff (minmax(Y),[20 10 7],{'tansig', 'tansig', 'purelin'},'traincgb');

其中：minmax(Y)代表输入样本 *Y* 中输入元素的典型值；[20 10 7]代表用 3 层网络，有 2 层隐含层，1 层输入层，隐含层共 30 个神经元，输出层用 7 个节点表示。{'tansig', 'tansig', 'purelin'}代表 2 个隐含层用 tan−sigmoid 型传递函数，输出层采用线性传递函数；'traincgb'代表采用 Powell−Beale 共轭梯度算法的训练函数。

BP 网络中经常用到的训练函数还有 traingd、traingdm、trainrp、trainlm、trainscg 等。traingd 是标准的梯度下降训练算法，其训练速度很慢。trainlm 是采用 Levenberg−Marquardt 算法的训练函数。对于中小规模的神经网络，最好采用 L−M 算法，它具有最快的收敛速度与最小的存储量，是梯度下降法与高斯牛顿法的结合。对于大型神经网络，最好选择 scaled 共轭梯度法训练函数 trainscg 或者弹性梯度算法训练函数 trainrp。

设置 BP 网络训练的相关参数，部分设置语句：

```
net.trainParam.epochs = 500;       % 最大训练次数
net.trainParam.goal = 1e-4;        % 最小均方误差
net.trainParam.min_grad = 1e-20; % 最小梯度
net.trainParam.show = 200;         % 训练显示间隔
net.trainParam.time = inf;         % 最大训练时间
```

设置好参数后，开始对神经网络进行训练。由于输入神经网络的样本中共有 7 种核素，故输出共用 7 个节点，输出量用 7×1 的列向量表示。本文中约定核素排列的顺序为 ^{241}Am、^{109}Cd、^{238}Pu、^{226}Rn、^{133}Ba、^{60}Co、^{137}Cs。在样本输出中，若某个样本中存在某种核素，则在输出矩阵中该核素对应的位置上元素为 1。例如，若某个样本中存在 ^{137}Cs，则其输出矩阵为$[0\ 0\ 0\ 0\ 0\ 0\ 1]^{T}$；若某个样本中存在 ^{241}Am 与 ^{60}Co，则样本输出矩阵为$[1\ 0\ 0\ 0\ 0\ 1\ 0]^{T}$，以此类推。

2.3　利用 RBF 学习算法进行核素识别

在 Matlab 平台读入预处理好的.mat 格式数据包，然后利用 newrbe 函数创建一个严格的径向基网络或者利用 newrb 函数创建一个径向基网络。利用 newrbe 创建神经网络时，其神经元数量等于输入样本数量，并可以一次性得到一个零误差的径向基网络；而用 newrb 创建网络时，最初是没有 RBF 神经元的，它先从输入数据中最大误差的样本入手，增加 RBF 神经元并得到输出，然后再重新设计网络线性层用以逐步减小误差，继续重复根据下一个最大误差样本，再增加一个 RBF 神经元……依此下去，直到误差达到要求的性能或者 RBF 神经元数达到上限时，建立网络结束。所以，从建立网络速度的角度来看，newrb 远不如 newrbe 快，但是后者能获得更小规模的神经网络，使复杂度减低。笔者选择 newrb 函数建立 RBF 神经网络，语句格式如下：

net = newrb(P,T,goal,spread,MN,DF);

其中：***P*** 为输入矩阵；***T*** 为期望输出矩阵；goal 是标量，为规定的均方误差，笔者设定为 1e−9；spread 也是标量，表示径向基函数的扩散速度，缺省值为 1，此处设定为 6；MN 为最大神经元个数，此处设定为训练样本个数；DF 为两次显示之间所添加神经元的个数，是一个控制显示级别的参数，此处设定为 1。

由于 RBF 神经网络不需要专门的训练算法对其进行训练。所以，笔者可以直接利用创建好的 RBF 神经网络进行预测分类。

3　核素识别结果

3.1　原始数据预处理结果

利用文中第 2.1 节方法，将所有训练样本组成矩阵后进行 PCA，通过 Matlab 平台下运算，得到分析结果如图 4 所示。图中横坐标表示主成分个数，纵坐标表示方差解释程度，图中的线表示累积方差解释程度，即某一维度下可以完整表达原始数据特征信息的百分比。

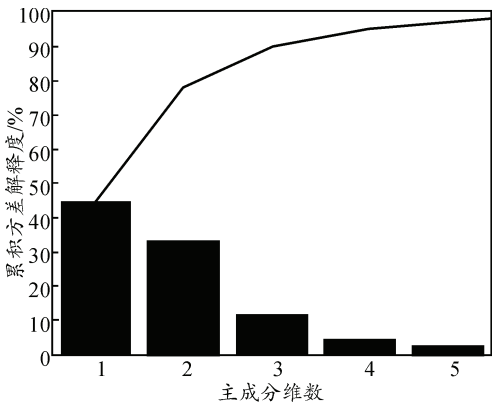

图 4　累积方差解释程度趋势

从图 4 和表 1 中可以看出：5 个主成分就可以表达原始数据特征信息的 96%以上，再结合 Matlab 定量计算出的解释百分比，选择 8 个主成分可以表达原始特征信息的 99.36%，为了获取更多的特征信息以及保证实验识别的准确性，笔者设定累积方差解释度的阈值为 99%，即选定 8 个或 8 个以上主成分组成变换矩阵，使原始训练样本矩阵投影到主成分空间中提取出主成分信息，再经数据归一化后，即可将其输入神经网络进行训练。

表 1 方差解释程度 %

主成分个数	方差解释	累积方差解释度
1	44.78	44.78
2	32.91	77.69
3	11.84	89.53
4	4.63	94.16
5	2.66	96.83

3.2 利用 BP 网络的识别结果

将经预处理后的数据输入 BP 神经网络进行训练，得到的结果如图 5 和图 6 所示。

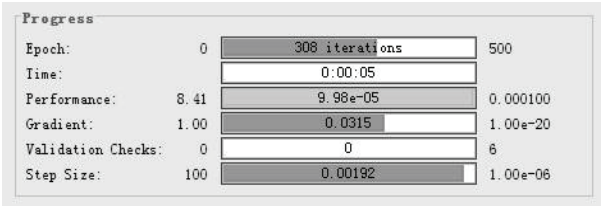

图 5 BP 网络训练结果

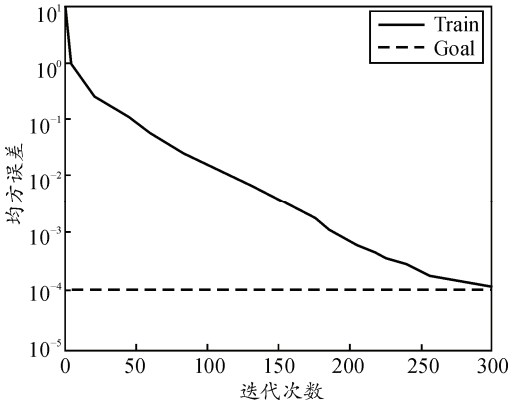

图 6 性能函数收敛过程

表 2 BP 网络样本输出值

待识别核素	训练核素						
	^{241}Am	^{109}Cd	^{238}Pu	^{226}Rn	^{133}Ba	^{60}Co	^{137}Cs
^{241}Am	1.003 1	−0.000 4	−0.002 6	−0.001 4	0.000 5	0.001 6	0.001 4
^{241}Am	1.000 1	−0.000 6	0.000 4	−0.000 2	0.000 1	−0.000 2	−0.000 1
^{241}Am	0.753 1	−0.100 4	−0.026 7	−0.014 7	−0.091 4	0.360 3	−0.083 6
^{241}Am	0.999 4	0.000 7	−0.000 3	−0.000 8	−0.000 7	0.000 6	0.000 5
^{60}Co	−0.202 4	−0.065 5	−0.028 3	0.00 41	−0.070 5	1.318 2	−0.185 6
^{60}Co	−0.028 2	0.000 5	−0.017 2	−0.016 6	0.023 2	1.057 7	−0.091 6
^{60}Co	0.096 8	−0.010 9	−0.014 2	−0.029 8	−0.003 9	1.115 7	−0.221 8
^{60}Co	0.251 6	−0.094 5	−0.064 7	0.023 8	−0.123 9	0.967 4	−0.202 8
^{60}Co	−0.051 4	−0.018 3	0.036 3	0.022 7	−0.028 7	0.653 9	0.364 2
^{60}Co	0.137 7	−0.046 3	0.041 0	0.042 8	−0.010 6	0.643 4	0.258 2
^{137}Cs	0.021 3	−0.041 0	−0.005 6	0.011 9	−0.072 2	0.305 1	0.896 5
^{137}Cs	0.193 2	0.040 1	0.011 9	0.025 8	0.018 7	0.357 8	0.489 8
^{137}Cs	−0.030 8	−0.018 5	−0.098 6	−0.035 8	−0.079 6	1.070 8	−0.436 3
^{109}Cd	−0.000 4	1.000 1	−0.000 3	−0.000 1	0.000 3	0.000 1	−0.000 1
^{238}Pu	0.000 3	0.000 5	0.999 5	−0.000 3	−0.000 1	0.000 3	−0.000 1
^{226}Rn	0.000 1	0.000 3	0.000 2	0.999 5	0.000 1	0	−0.000 3
^{133}Ba	−0.000 3	0.000 2	−0.001 1	0	0.999 5	−0.000 6	0
^{241}Am+^{109}Cd	1.000 4	1.000 1	0.000 1	−0.000 1	0.000 2	−0.000 1	−0.000 5
^{109}Cd+^{238}Pu	−0.000 2	1.000 4	1.000 1	0.000 1	0	−0.000 2	−0.000 3
^{238}Pu+^{133}Ba+^{109}Cd	0	0.999 8	0.999 9	−0.000 1	1.000 3	−0.000 1	−0.000 5

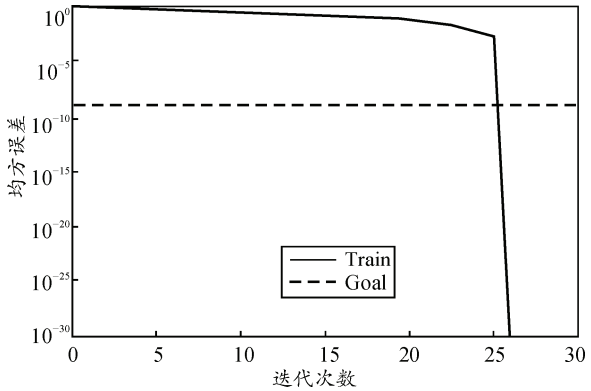

图 7 RBF 网络的均方误差表现

从图 5 中可看出：输入的数据在训练的第 308 步时，性能函数即达到设定的目标(1×10^{-4})，在预置的训练步数以内达到收敛，完成了对网络的训练。

然后进行网络测试，测试样本同样先通过 PCA 投影至与训练样本相同的主成分空间内，经提取特征、数据归一化处理后输入神经网络，测试该神经网络的分类性能，识别的结果如表 2 所示，表格中数据已经进行过反归一化处理，每一行数据代表一个样本输出。

从识别的结果来看，不论是单核素样本还是多核素样本，BP 网络的识别效果较为理想。20 组测试样本中只有 1 组样本分类错误，分类正确率为 95%。如果将上述输出矩阵设定阈值进行圆整，则可以得到相应的二值矩阵，通过二值矩阵可以很直观的看出某种核素是否存在其样本中。另外，分类

错误样本的原因是 ^{137}Cs 本身谱特征信息较弱有关，容易形成测量与识别干扰，同时训练样本数不充分，也是造成 BP 网络错误识别的重要原因。

3.3　利用 RBF 网络的识别结果

同样利用文中第 2.1 节中经过预处理后的数据，将其输入 RBF 神经网络进行测试，RBF 网络的均方误差表现如图 7 所示，其样本输出结果如表 3 所示，表中的数据已经过反归一化处理，其中每一行为一个样本输出。

从 RBF 神经网络分类的结果来看，在第 26 步建立 RBF 神经元时，RBF 网络的均方误差就达到了规定的值，为 $7.319\ 92\times10^{-31}$，网络收敛迅速，分类精度很高，不论是单核素样本还是多核素样本，RBF 网络均能将测试样本完全正确分类，其正确率高达 100%，识别效果非常理想。

表 3　RBF 网络样本输出值

待识别核素	训练核素						
	^{241}Am	^{109}Cd	^{238}Pu	^{226}Rn	^{133}Ba	^{60}Co	^{137}Cs
^{241}Am	1	2.22e−16	−8.88e−16	0	1.61e−16	2.22e−16	−3.33e−16
^{241}Am	1	−4.44e−16	−2.22e−16	0	2.22e−16	2.22e−16	5.55e−16
^{241}Am	0.381 5	0.150 2	0.136 4	0	0.024 9	0.294 4	0.163 4
^{241}Am	1	0	5.0e−16	0	−1.11e−16	3.33e−16	1.11e−16
^{60}Co	0.023 1	−0.020 3	−0.022 2	0	1.0e−06	0.939 3	0.061 1
^{60}Co	0.134 2	0.020 8	0.009 0	0	−0.005 5	0.502 6	0.345 3
^{60}Co	0.385 1	0.079 6	0.074 6	0	0.008 8	0.415 9	0.112 4
^{60}Co	0.313 8	0.143 3	0.127 5	0	0.051 7	0.685 6	−0.152 8
^{60}Co	0.026 8	−0.009 5	−0.014 5	0	0.007 2	0.858 9	0.118 5
^{60}Co	0.336 0	0.143 9	0.149 3	0	0.062 9	0.572 4	−0.085 6
^{137}Cs	0.490 5	0.037 5	0.058 5	0	0.057 2	0.310 9	0.138 0
^{137}Cs	0.129 8	0.056 6	0.045 0	0	0.004 8	0.211 6	0.597 7
^{137}Cs	0.199 7	0.038 8	0.031 8	0	−0.013 1	0.145 1	0.622 3
^{109}Cd	0	1	−1.11e−16	0	5.55e−17	0	0
^{238}Pu	5.55e−17	0	1	0	−1.11e−16	0	−1.11e−16
^{226}Rn	0	0	0	1	0	0	0
^{133}Ba	−1.11e−16	5.55e−17	5.55e−17	0	1	0	0
^{241}Am+^{109}Cd	1	1	5.55e−17	0	0	0	1.11e−16
^{109}Cd+^{238}Pu	−2.22e−16	1	1	0	1.11e−16	3.33e−16	2.22e−15
^{238}Pu+^{133}Ba+^{109}Cd	5.55e−17	1	1	0	1	1.11e−16	0

4　结束语

根据上述实验结果，对比 2 种神经网络，不难得出，RBF 神经网络不论从识别精度、识别准确度上，还是性能函数收敛速度上，其性能均优于 BP 网络。究其原因，是因为 BP 网络存在一定的局限性，由于 BP 网络需要的参数较多，且参数的选择目前并没有一个有效的方法，只能通过经验给出隐层层数，神经元个数等参数。此外，BP 网络对于样本具有很强的依赖性，BP 网络模型的逼近能力和推广学习能力与学习样本的典型性密切相关，若样本代表性差、矛盾样本多，则网络很难达到预期的性能。再者，BP 算法容易陷入局部最优，BP 网络算法理论上可以实现任意非线性映射，但实际应用中，很可能经常陷入到局部极小值中，难以跳出。

BP 神经网络与 RBF 神经网络是模式识别领域中广泛用于分类与逼近的方法，在实际应用中具体选择哪一种算法，还需视具体情况而定。笔者提出的 2 种神经网络均可以实现 γ 谱的快速解析，符合核素快速识别的要求。

参考文献：

[1] 高隽. 人工神经网络原理及仿真实例[M]. 北京: 机械工业出版社, 2003: 49−51.

[2] 葛君伟, 沙静, 方义秋. 具有混沌学习率的 BP 算法[J]. 计算机工程, 2010, 36(23): 168−170.

[3] 李志清, 傅秀芬. 基于 PCA 的 3 种改进 BP 算法性能研究[J]. 计算机工程, 2011, 37(21): 108−110.

[4] 赵建辉, 赵建平, 孙永江, 等. BP 神经网络在雷达故障诊断中的应用[J]. 兵工自动化, 2013, 32(2): 37−38.

[5] 朱凯, 王正林. 精通 Matlab 神经网络[M]. 北京: 电子工业出版社, 2010: 214−215.

[6] 张德丰. Matlab 神经网络应用设计[M]. 2 版. 北京: 机械工业出版社, 2012: 171−176.

[7] 边肇祺, 张学工. 模式识别[M]. 2 版. 北京: 清华大学出版社, 2010: 224−228.

[8] 俞利强, 马道钧. 基于 PCA 技术的神经网络说话人识别研究[J]. 计算机工程与应用, 2010, 46(19): 211−213.

[9] Turk M, Pentland A. Eigenfaces for recognition [J]. Journal of Cognitive Neuroscience, 1991, 3(1): 71−86.

[10] Kirby M, Sirovich L. Application of the Karhunen-Loeve procedure for the characterization of human faces[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(1): 103−108.

[11] 李康顺, 李凯, 张文生. 一种基于改进 BP 神经网络的 PCA 人脸识别算法[J]. 计算机应用与软件, 2014, 31(1): 158−159.

[12] Robert Hecht-Nielsen. Theory of the Back Propagation Neural Network[C]. Proceedings of the International Joint Conference on Neural Networks, 1989: 593−605.