中山大学学报（自然科学版）
第 39 卷 第 3 期 ACTA SCIENTIARUM NATURALIUM Vol. 39 No. 3
2000 年 5 月 UNIVERSITATIS SUNYATSENI May 2000

文章编号：0529-6579（2000）03-0025-05

立体神经视觉系统中零件识别的学习方法*

熊 银 根

（中山大学 无线电电子学系，广东 广州 510275）

摘 要：提出了一种立体神经视觉系统中零件识别的学习方法，与标准的 BP 算法对比有两点改进：①用变尺度方向代替负梯度方向作为搜索方向；②用可变的最优学习率来代替不变的学习率．采用上述 2 个改进后，训练速度和收敛性都有较大的改善．实际应用表明，所提出的学习方法的训练速度、收敛性和稳定性都比标准 BP 算法有较大的提高．

关键词：立体神经视觉；学习方法；神经网络；变尺度方向；最优学习率

中图分类号：TP205 文献标识码：A

在智能装配环境下，要求立体神经视觉系统对零件的识别不仅要快，而且要稳健[1,2]，在具有噪声的环境下必须满足连续操作要求．与其它领域一样，神经网络在三维零件识别中能提供智能，其学习和并行处理能力[3~5]能提供快的识别速度和连续稳定的操作．然而，神经网络本身还存在着问题，在训练过程中，需要花很长时间进行学习，而且收敛性不能得到保证，缺乏稳健性．为了能在智能装配系统中充分利用神经网络的优点，必须研究出更有效的学习方法．

在标准 BP 算法（SBP）的基础上，本文提出了一种用于零件识别的学习方法（LM-PR），在该方法中，把理想输出与实际输出之间的误差构造成目标函数，通过调整神经网络上神经元的权值使目标函数极小化，使用最优学习率代替不变的学习率，采用变尺度方向代替负梯度方向，以达到改进学习速度和收敛性的目的．为了验证该方法的稳定性和实用性，给出了一些实例，应用结果表明，与 SBP 算法相对比，本文所提出的学习方法能在较短的训练时间内获得较高的精度，提高了学习速度和效率．

1 学习算法的基本构成

对一般的三层神经网络，如图 1 所示，其输入为 $x_k(k = 1, 2, \cdots, n)$，实际输出为 $y_j(j = 1, 2, \cdots, N)$，理想输出为 $d_j(j = 1, 2, \cdots, N)$，理想输出与实际输出之间的误差定义为

$$E = \frac{1}{2}\sum_{j=1}^{N}(y_j - d_j)^2 \tag{1}$$

* 基金项目：广东省博士后基金资助项目

收稿日期：1999-09-23；作者简介：熊银根（1962～），男，副教授．

其中 y_j 是权(w_{ij}, w_{ik}) 的函数, 调整权来使误差函数达到极小. 节点作用函数为

$$y = f(x) = 1/(1+e^{-x}) \tag{2}$$

输出层和隐含层上每个神经元的输入分别为

$$A_j(j) = \sum_{i=1}^{m} WJ_{ji}X_i;\ A_i(i) = \sum_{k=1}^{n} WI_{ik}X_k \tag{3}$$

其中, $WJ_{ji}(j = 1, 2, \cdots, N, i = 1, 2, \cdots, m)$ 和 $WI_{ik}(i = 1, 2, \cdots, m, k = 1, 2, \cdots, n)$ 分别为输出层与隐含层之间, 隐含层与输入层之间的权值.

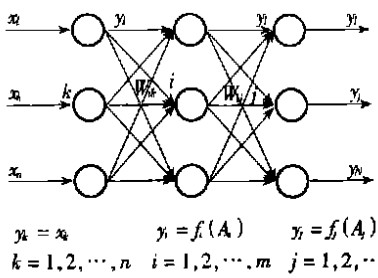

图 1 三层神经网络的结构

Fig. 1 Diagram of a three – layer neural network

对输出层与隐含层有

$$\frac{\partial E}{\partial \boldsymbol{WJ}_{ji}} = \frac{\partial E}{\partial Y_j}\frac{\partial y_j}{\partial \boldsymbol{WJ}_{ji}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial A_j}\frac{\partial A_j}{\partial \boldsymbol{WJ}_{ji}} = (y_j - d_j)y_j(1-y_j)y_j \tag{4}$$

对隐含层与输入层之间有

$$\frac{\partial E}{\partial \boldsymbol{WI}_{ik}} = \frac{\partial E}{\partial Y_i}\frac{\partial y_i}{\partial A_i}\frac{\partial A_i}{\partial \boldsymbol{WI}_{ik}} = \frac{\partial E}{\partial yy_j}\frac{\partial y_j}{\partial A_j}\frac{\partial A_j}{\partial y_i}(1-y_i)y_iy_k =$$

$$(y_j - d_j)y_j(1-y_j)\boldsymbol{WJ}_{ji}(1-y_i)y_iy_k = \frac{\partial E}{\partial \boldsymbol{WJ}_{ji}}\boldsymbol{WJ}_{ji}(1-y_i)y_k \tag{5}$$

根据上述方程, 标记
$$\begin{cases} \boldsymbol{WJ}_j = [\boldsymbol{WJ}_{ji}] \\ SJ_j = [SJ_{ji}] \end{cases} \quad i = 1, 2, \cdots, m; j = 1, 2, \cdots, N \tag{6}$$

其中, $\boldsymbol{WJ}_j(j = 1, 2, \cdots, N)$ 为输出层与隐含层之间的权矢量; $\boldsymbol{SJ}_j(j = 1, 2, \cdots, N)$ 为修正权矢量的方向矢量; $[\boldsymbol{SJ}_{ji}] = -\left[\frac{\partial E}{\partial \boldsymbol{WJ}_{ji}}\right]$ $(i = 1, 2, \cdots, m; j = 1, 2, \cdots, N)$ 为负梯度方向.

$$\begin{cases} \boldsymbol{WI}_i = [\boldsymbol{WI}_{ik}] \\ SI_i = [SI_{ik}] \end{cases} \quad k = 1, 2, \cdots, n; i = 1, 2, \cdots, m \tag{7}$$

其中, $\boldsymbol{WI}_i(i = 1, 2, \cdots, m)$ 为隐含层与输入层之间的权矢量; $\boldsymbol{SI}_i(i = 1, 2, \cdots, m)$ 为修正权矢量的方向矢量; $[\boldsymbol{SI}_{ik}] = -\left[\frac{\partial E}{\partial \boldsymbol{WI}_{ik}}\right]$ $(i = 1, 2, \cdots, m; k = 1, 2, \cdots, n)$ 为负梯度方向.

可以把两个权矢量结合成一个新矢量.

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{WJ}_j \\ \boldsymbol{WI}_i \end{bmatrix} \quad (j = 1, 2, \cdots N; i = 1, 2, \cdots, m) \tag{8}$$

$\boldsymbol{W}$ 为整个网络的权矢量.

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{SJ}_j \\ \boldsymbol{SI}_i \end{bmatrix} \quad (j = 1, 2, \cdots N; i = 1, 2, \cdots, m) \tag{9}$$

S 为修正权矢量的方向矢量. 因此, 网络的权矢量可以用下列方式进行修正.

$$\boldsymbol{W}^{l+1} = \boldsymbol{W}^l + \alpha^l S^l \tag{10}$$

其中, α^l 为学习率, 并且为常数; l 为迭代次数. 在输入层有

$$y_k = x_k \quad k = 1, 2, \cdots, n \tag{11}$$

这就是SBP 算法的学习过程.

2 学习算法的改进

在SBP算法中, 为了实现学习过程, 采用了最速下降法极小化误差函数 E, 存在的问

题是要在很长时间进行学习，而且收敛性不能得到保证，缺乏稳健性．在 SBP 算法中，使用负梯度方向和固定步长是导致收敛速度慢和收敛性得不到保证的主要原因，为了克服该算法所固有的问题，这里提出了一种在迭代过程中寻找最优方向和最优步长的方法．

2.1　构造新方向

这里构造一个变尺度方向来代替式（9）所表示的方向，

$$\nabla E\boldsymbol{J}^l=\left[\frac{\partial E^l}{\partial \boldsymbol{WJ}_{ji}}\right],\ \nabla E\boldsymbol{I}^l=\left[\frac{\partial E^l}{\partial \boldsymbol{WI}_{ik}}\right],\ i=1,2,\cdots,m;\ k=1,2,\cdots,n;\ j=1,2,\cdots,N \tag{12}$$

$$\nabla E^l=\begin{bmatrix}\nabla E\boldsymbol{J}^l\\ \nabla E\boldsymbol{I}^l\end{bmatrix} \tag{13}$$

式中，l 为迭代次数．可以把方向定义为

$$\boldsymbol{S}^l=\begin{bmatrix}\boldsymbol{SJ}^l\\ \boldsymbol{SI}^l\end{bmatrix}=-\boldsymbol{H}^l\nabla E^l \tag{14}$$

其中，$\boldsymbol{H}^l$ 为变尺度矩阵，将在后面描述，矩阵 $\boldsymbol{H}^l$ 计算后，可以用下列方法修正权系数

$$\boldsymbol{W}^{l+1}=\boldsymbol{W}^l+\alpha^l\boldsymbol{S}^l \tag{15}$$

其中，α^l 为最优学习率；l 为迭代次数．

可以使用下列方法修改变尺度矩阵 $\boldsymbol{H}$

即　　$$\boldsymbol{H}^{l+1}=\boldsymbol{H}^l+\boldsymbol{M}^l+\boldsymbol{N}^l \tag{16}$$

其中，$\boldsymbol{M}^l=\alpha^l\dfrac{\boldsymbol{S}^l[\boldsymbol{S}^l]^{\mathrm{T}}}{[\boldsymbol{S}^l]\boldsymbol{Q}^l}$，$\boldsymbol{N}^l=-\dfrac{[\boldsymbol{H}^l\boldsymbol{Q}^l][\boldsymbol{H}^l\boldsymbol{Q}^l]^{\mathrm{T}}}{[\boldsymbol{Q}^i]^{\mathrm{T}}\boldsymbol{H}^l\boldsymbol{Q}^l}$，$\boldsymbol{Q}^l=\begin{bmatrix}\boldsymbol{QJ}^l\\ \boldsymbol{QI}^l\end{bmatrix}$，$\begin{cases}\boldsymbol{QJ}^l=[\boldsymbol{QJ}_{ji}^l]\\ \boldsymbol{QI}^l=[\boldsymbol{QI}_{ik}^l]\end{cases}$

$$\begin{cases}\boldsymbol{QJ}_{ji}^l=\nabla E(\boldsymbol{WJ}_{ji}^{l+1})-\nabla E(\boldsymbol{WJ}_{ji}^l)=\nabla E\boldsymbol{J}^{l+1}-\nabla E\boldsymbol{J}^l\\ \boldsymbol{QI}_{ik}^l=\nabla E(\boldsymbol{WI}_{ik}^{l+1})-\nabla E(\boldsymbol{WI}_{ik}^l)=\nabla E\boldsymbol{I}^{l+1}-\nabla E\boldsymbol{I}^l\end{cases}$$

2.2　构造最优学习率

从式(10)可以看出，学习率 α 是一个常数，用这种方法很难找到最优值．从优化的角度来看，α 是一维搜索的步长，在优化过程中，我们必须在方向 $\boldsymbol{S}^l$ 上找到最优步长．

为了寻找最优步长 α^*，可以使用解析法，也可以使用数值法，首先来构造目标函数，根据式(10)，可以修正输出层与隐含层之间的权值．

$$\boldsymbol{WJ}_{ji}^{l+1}=\boldsymbol{WJ}_{ji}^l+\alpha^l\boldsymbol{SJ}_{ji}^l\qquad i=1,2,\cdots,M;\ j=1,2,\cdots,N \tag{17}$$

从输出层反向传播到隐含层，输出层上每个神经元的输入为

$$A_j^{l+1}(j)=\sum_{i=1}^{m}(\boldsymbol{WJ}_{ji}^{l+1}x_i)\quad j=1,2,\cdots,N \tag{18}$$

使用式(17)得　　$$A_j^{l+1}=\sum_{i=1}^{m}(\boldsymbol{WJ}_{ji}^l+\alpha^l\boldsymbol{SJ}_{ji}^l)x_i\quad j=1,2,\cdots,N \tag{19}$$

输出层神经元的输出为

$$y_j^{l+1}=f(A_j^{l+1})=1/[1+\exp(-A_j^{l+1})]\quad j=1,2,\cdots,N \tag{20}$$

利用式(19)得　$$y_j^{l+1}=1\Big\backslash\Big[1+\exp\Big(-\sum_{i}^{m}(\boldsymbol{WJ}_{ji}^l+\alpha^l\boldsymbol{SJ}_{ji}^l)x_i\Big)\Big]\quad j=1,2,\cdots,N \tag{21}$$

即　　$$y_j^{l+1}=1\Big\backslash\Big[1+\exp\Big(-\sum_{i}^{m}(\boldsymbol{WJ}_{ji}^l+\alpha^l\boldsymbol{SJ}_{ji}^l)y_i^{l+1}\Big)\Big]\quad j=1,2,\cdots,N \tag{22}$$

用同样的方法，可以求得隐含层上神经元的输出为

$$y_i^{l+1}=1\Big\backslash\Big[1+\exp\Big(-\sum_{k=1}^{n}(\boldsymbol{WI}_{ik}^{l}+\alpha^{l}\boldsymbol{SI}_{ik}^{l})x_k\Big)\Big]\quad k=1,2,\cdots,n;\ i=1,2,\cdots,m \tag{23}$$

从式(22)和式(23)可以得

$$y_j^{l+1}(\alpha^{l})=1\Bigg\backslash\Bigg[1+\exp\Bigg(-\sum_{i=1}^{m}\frac{\boldsymbol{WJ}_{ji}^{l}+\alpha^{l}\boldsymbol{SJ}_{ji}^{l}}{1+\exp\Big(-\sum_{k=1}^{n}(\boldsymbol{WI}_{ik}^{l}+\alpha^{l}\boldsymbol{SI}_{ik}^{l})x_k\Big)}\Bigg)\Bigg]\quad j=1,2,\cdots,N \tag{24}$$

输出层上节 j 个神经元的误差函数为

$$E_j^{l+1}(\alpha^{l})=(y_j^{l+1}(\alpha^{l})-d_j)^2=$$
$$\Bigg[1\Bigg\backslash\Bigg(1+\exp\Bigg(-\sum_{i=1}^{m}\frac{\boldsymbol{WJ}_{ji}^{l}+\alpha^{l}\boldsymbol{SJ}_{ji}^{l}}{1+\exp\Big(-\sum_{k=1}^{n}(\boldsymbol{WI}_{ik}^{l}+\alpha^{l}\boldsymbol{SI}_{ik}^{l})x_k\Big)}\Bigg)\Bigg)-d_j\Bigg]^2\quad j=1,2,\cdots,N \tag{25}$$

因此，整个误差函数可以表示为

$$E^{l+1}=\frac{1}{2}\sum_{j=1}^{N}E_j^{l+1}(\alpha^{l})=\frac{1}{2}\sum_{j-1}^{N}(y_j^{l+1}(\alpha^{l})-d_j)^2 \tag{26}$$

为了获得最优步长，对目标函数 E 进行极小化

$$\min\frac{1}{2}\sum_{j=1}^{N}\Bigg[1\Bigg\backslash\Bigg(1+\exp\Bigg(-\sum_{i=1}^{m}\frac{\boldsymbol{WJ}_{ji}^{l}+\alpha^{l}\boldsymbol{SJ}_{ji}^{l}}{1+\exp\Big(-\sum_{k=1}^{n}(\boldsymbol{WI}_{ik}^{l}+\alpha^{l}\boldsymbol{SI}_{ik}^{l})x_k\Big)}\Bigg)\Bigg)-d_j\Bigg]^2 \tag{27}$$

从式(27)可以看出，这里要进行极小化的目标函数很复杂，很难用解析法进行求解，为了求出最优步长，可以用数值法进行求解.

3 实验及其结果分析

为了检验所提出的 LMPR 算法的实用性和可靠性，这里把该算法应用到对一个非线性系统的学习中，并且把所得结果与 SBP 的进行对比. 该非线性系统的数学模型、结构和其它特性都不知道，可以认为是一个“黑盒子”，唯一知道的是它的输入和理想输出，其两个输入为：$x_1(S)=1$，$x_2(S)=1.0S$，理想输出为：$y_d(S)=1+\exp(-0.05S)$ 对这个非线性系统，构造一个四层的 BP 神经网络，输入层具有两个神经元，用于两个输入 x_1，x_2，每个隐含层具有 10 个神经元，输出层具有一个神经元，用于一个输出，利用上述输入和理想输出表达式，当 $S=1,2,\cdots,100$，获得 100 个模式作为训练矢量，分别采用 SBP 算法和本文提出的 LMPR 算法训练这个神经网络，训练 5 min 后的结果如图 2 所示，图中 y_d 表示非线性系统的理想输出，y_l 和 y_s 分别为 LMPR 算法和 SBP 算法所得的结果. 从图 2 可以看出，y_l 非常接近 y_d（几乎分不出来）. 图 3 给出了 2 种方法的误差曲线.

正如所预料的，随着训练时间的增加，两种方法的误差都下降，然而，本文所提出的 LMPR 算法无论是在速度还是在精度方面都超过了 SBP 算法. 例如，当训练 90 min 时，LMPR 算法所得的误差为 0.022 893，SBP 算法所得的误差为 0.063 101.

此外，本文还把所提出的学习方法应用到核线匹配的学习和二维点到三维点转化过程中的学习. 核线匹配的学习所用的神经网络为 3-20-20-1 的 BP，二维点到三维点转化过程中的学习所用的神经网络为 4-50-50-3 的 BP，实验结果都表明，本文所提出的方法能在相同的训练时间内提供比 SBP 算法更高的精度，能在相同的误差下，花更短的时间.

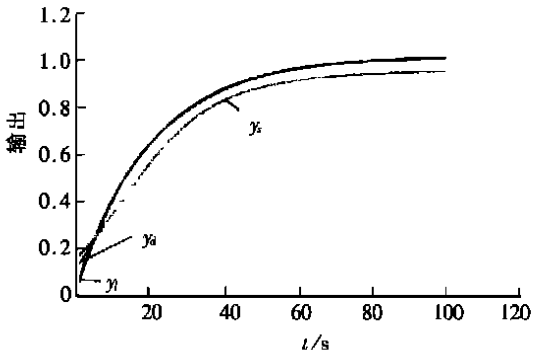

图 2　对神经网络训练的结果对比

Fig. 2　The output of the neural network trained by LM or SBP respectively

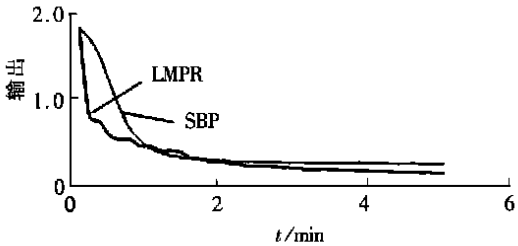

图 3　2 种算法对网络训练的误差曲线

Fig. 3　The errors versus the training time by using LMPR and SBP

参考文献：

[1] BUURMAN J, BIERHUIZEN D J W. A two stage object identification system in the deft intelligent assembly cell [J]. SPIE, 1990, 1386: 185～196.

[2] BESL P J, JAN R C. Three-dimensional object recognition[J]. Computer Surveys, 1995, 17(1): 75～145.

[3] HUANG S H, ZHANG H C. Neural-expert hybrid approach for intelligent manufacturing: A survey[J]. Computer in Industry, 1995, 26: 107～126.

[4] XU H Y, BRIRD C R. Synergism of neural network and expert systems for system identification[J]. Expert System with Applications, 1992, 5: 25～33.

[5] ROTH M W. Survey of neural network technology for automatic target recognition[J]. IEEE Transactions on neural networks, 1990, 1(1): 28～43.

Learning Mechanism for Parts Recognition in Stereoscopic Neuro-vision System

XIONG Yin-gen *

Abstract: A learning mechanism for parts recognition (LMPR) in a stereoscopic nuro-vision system is presented. It differs from the mechanism used in the standard back propagation (SBP) neural network in two ways. First, the searching direction is changed from the negative gradient direction to the variable metric direction. Secondly, the constant learning rate is changed to a variable optimal learning rate. With the combination of variable metric direction and variable optimal learning rate, the speed of the training process is greatly improved and the convergence is assured. Application examples are presented. The results have indicated that the proposed LMPR is superior in comparison with the SBP in the areas of learning speed, convergence and stability.

Keywords: stereoscopic neuro-vision; learning mechanism; neural networks; variable metric direction; variable optimal learning rate

* Department of Radio and Electronics, Zhongshan University, Guangzhou 510275, China