

一种基于 BP 神经网络的数字识别新方法

刘 炆, 汤传玲, 王 静, 石文莹, 赵小兰
(合肥工业大学, 安徽 合肥 230009)

摘 要: 针对标准 BP 神经网络收敛速度慢、易陷入局部极小点的缺点, 提出了一种新的 BP 神经网络改进算法。该算法通过变步长法和牛顿法来改进 BP 算法, 加快了网络的收敛速度, 且收敛速度快于其他的改进算法。在此基础上将 BP 神经网络应用于数字识别中, 为其网络建立识别模型。利用仿真实验观察 BP 网络的泛化能力以及识别准确性, 比较 BP 算法及其改进方案, 提出改进方案中分别需要注意的地方。

关键词: BP 神经网络; 数字识别; 变步长法; 牛顿法

中图分类号: TP183

文献标识码: A

文章编号: 1674-7720(2012)07-0036-04

A new method of numerical recognition based on improved BP neural network

Liu Yang, Tang Chuanling, Wang Jing, Shi Wenying, Zhao Xiaolan
(Hefei University of Technology, Hefei 230009, China)

Abstract: In this paper, a new improved BP algorithm is presented to solve the problem that BP neural network can easily fall into slow convergence and minimum. This algorithm integrates the variable step method with the Newton method. The algorithm can speed up the convergence rate of BP neural network, and the convergence rate is faster than other algorithm. The BP neural network is used in the digital recognition and the identification model for the network is established. Using simulation experiments to observe generalization ability and identification accuracy of BP neural network, compare BP neural network with its improvement scheme, and put forward the improvement plan in note respectively place.

Key words: BP neural network; numerical recognition; variable step method; Newton method

神经网络是一种由简单单元组成的具有适应性的广泛并行互连的网络, 其组织能够对真实世界物体做模拟生物神经系统的交互反应。神经网络已经成为一门比较重要的交叉性学科, 被广泛应用于智能控制、模式识别、辅助决策、信号处理和计算机视觉等领域。BP (Back Propagation) 神经网络在各门学科领域中都有非常重要的实用价值, 也是在模式识别中应用最广泛的一种人工神经网络, 它具有很强的自学习、自适应能力以及非线性逼近能力。由于基于误差梯度下降的 BP 算法易于陷入局部极小点而不收敛, 目前, 已经有很多学者对 BP 算法进行了研究, 并针对它的这些缺陷提出了一些改进方法。

1 基本 BP 算法

1.1 BP 算法原理

BP 网络是采用误差反向传播算法的多层前馈网

络, 网络的输入和输出是一种非线性映射关系, 其中, 神经元的传递函数为 Sigmoid 函数。

BP 网络的学习规则是采用梯度下降算法。在网络学习的过程中, 将输出层节点的期望输出 (目标输出) 与实际输出 (计算输出) 的均方误差——逐层向输入层进行反向传播, 再分配给各连接节点, 并且计算出各连接节点的参考误差。在这个基础上可以调整各连接权值, 使所获得的网络的期望输出与实际输出的均方误差达到最小。

将第 j 个样本输入到网络时, 得到均方误差为:

$$E_j = \sum_{i=1}^{n_0} \phi(e_{i,j}) = \frac{1}{2} \sum_{i=1}^{n_0} e_{i,j}^2 = \frac{1}{2} \sum_{i=1}^{n_0} (y_{i,j} - \hat{y}_{i,j})^2 \quad (1)$$

其中, $e_{i,j}$ 表示输入第 j 个样本时, 输出层第 i 个节点的

期望输出值 $y_{i,j}$ 与实际输出值 $\hat{y}_{i,j}$ 间的差值, 即 $e_{i,j}=y_{i,j}-\hat{y}_{i,j}$; n_0 为输出层的节点数。

连接权的调整主要有成批处理和逐个处理两种方法。成批处理是指一次性将所有训练样本输入, 并计算总误差, 然后调整连接权。逐个处理是指每输入一个样本就调整一次连接权值。本文采用逐个处理的方法, 并根据误差的负梯度来修改连接权值。

BP 网络的学习规则为:

$$W_{p,k+1}^{(r)} = W_{p,k}^{(r)} + \Delta W_{p,k}^{(r)} \quad (2)$$

$$\Delta W_{p,k}^{(r)} = -\eta \frac{\partial E_k}{\partial W_{p,k}^{(r)}} \quad (3)$$

其中, $W_{p,k}^{(r)}$ 代表第 r 层 (从第一个隐含层开始 $r=1$, 输入层 $r=0$) 的连接权阵 $W^{(r)}$ 的第 p 行, 即 $W_{p,k}^{(r)}$ 是由第 $r-1$ 层各节点到第 r 层的第 p 个节点所有连接权值组成的一个行向量; k 代表迭代次数; E_k 是第 k 次迭代的均方误差; η 是学习步长, $0 < \eta < 1$ 。

设 BP 网络有 1 个隐含层, 由神经元的输入与输出关系得:

$$\hat{y}_{i,k} = f_0(\bar{y}_{i,k}) = f_0\left(\sum_{j=1}^{n_l} w_{j,i} h_{j,k}\right) \quad (4)$$

其中, $\bar{y}_{i,k}$ 代表最后一个隐含层节点到输出层第 i 个节点的加权和; $f_0(\bar{y}_{i,k})$ 是输出层的传递函数; $w_{j,i}$ 是最后一个隐含层的第 j 个节点和输出层的第 i 个节点之间的权值; n_l 是最后一个隐含层的节点数; $h_{j,k}$ 是第 l 隐含层的第 j 个节点的输出。

BP 网络可以存储和学习大量的输入—输出模式映射关系, 且无需事先揭示和描述这种映射关系的数学方程。在人工神经网络的实际应用中, 很多神经网络模型就是采用 BP 网络及其变化形式的。

BP 神经网络模型拓扑结构包括输入层、隐层和输出层 3 层, 如图 1 所示。

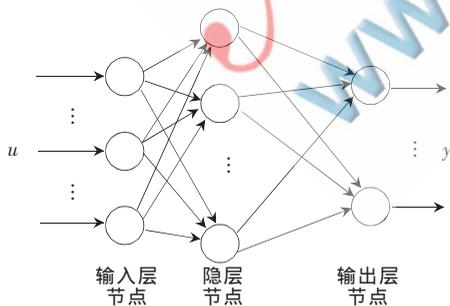


图 1 BP 神经网络拓扑结构

1.2 BP 算法的程序实现

BP 算法的程序实现的具体步骤如下:

(1) 网络初始化。设计网络连接结构, 规定输入变量和输出变量个数、隐含的层数和各层神经元的个数, 对

W (位于隐层到输出层之间)、权值矩阵 V (位于输入层到隐层之间) 赋随机数, 将训练次数计数器 q 和样本模式计数 p 置为 1, 学习率 η 设为 0~1 内的小数, 误差 E 置 0, 网络训练后达到的精度 E_{\max} 设为一个正的小数。

(2) 输入训练样本对, 利用现有的权值来计算网络中各神经元的实际输出。用当前样本 x_p 、 d_p 对 d (期望输出向量)、向量数组 X (输入向量) 赋值, 用式 (5) 和式 (6) 计算 O (输出层输出向量) 和 Y (隐层输出向量) 中各分量。

$$O_k = f\left(\sum_{j=0}^m w_{jk} y_j\right), k=1, 2, \dots, l \quad (5)$$

$$y_i = f\left(\sum_{i=0}^n v_{ij} x_i\right), j=1, 2, \dots, m \quad (6)$$

(3) 网络的输出误差。假设共有 P 对网络样本, 且网络对不同的样本将使用不同的误差 $E^p = \sqrt{\sum_{k=1}^l (d_k^p - O_k^p)^2}$, 可以把全部样本输出误差的平方 (E^p)² 相加再开方, 用来作为总的输出误差, 也可将最大误差 E_{\max} 代表网络的总输出误差, 实际中的更多将采用均方根误差 $E_{\text{RME}} =$

$\sqrt{\frac{1}{P} \sum_{p=1}^P (E^p)^2}$ 来作为网络的总误差。

(4) 各层误差信号的计算。应用式 (7) 和式 (8) 计算 δ_j^y 和 δ_k^o

$$\delta_j^y = \left(\sum_{k=1}^l \delta_k^o W_{jk}\right) y_j (1 - y_j) \quad (7)$$

$$\delta_k^o = (d_k - O_k) O_k (1 - O_k) \quad (8)$$

(5) 各层权值间的调整。应用式 (9) 和式 (10) 计算 V 、 W 中的各个分量。

$$\Delta V_{ij} = \eta \delta_j^y x_j = \eta \left(\sum_{k=1}^l \delta_k^o W_{jk}\right) y_j (1 - y_j) x_i \quad (9)$$

$$\Delta W_{jk} = \eta \delta_k^o y_j = \eta (d_k - O_k) O_k (1 - O_k) y_j \quad (10)$$

(6) 检验是否对所有的样本完成了一次轮训。若 $p < P$, 计数器 p 、 q 加 1, 返回步骤 (2); 否则, 进行步骤 (7)。

(7) 检验网络总误差是否达到精度要求。如当用 E_{RME} 作为网络的总误差时, 若满足 $E_{\text{RME}} < E_{\min}$, 则训练结束; 否则, E 置 0, p 置 1, 返回步骤 (2)。

2 BP 算法的缺陷及其改进

2.1 BP 算法的缺陷

3 层前馈网络已经得到越来越广泛的应用, 因为它有一个非凡优势, 即将 BP 算法用于具有非线性转移函数的 3 层前馈网络, 可以以任意精度逼近任何非线性函数。但是标准的 BP 算法在应用中还有一些内在的缺陷:

(1) 误差下降缓慢、迭代次数多、调整时间长, 影响收敛的速度, 会造成收敛速度慢, 如果加快收敛速度则易于产生振荡。

(2) 存在局部极小问题, 但是却得不到全局最优。训练时会因陷入某个局部极小点而不能自拔, 使得训练很

难收敛在给定的误差内。

2.2 BP 算法的改进

尽管标准的 BP 算法原理简单、实现方便,但由于训练过程中使用的是一个较小的常数,因此存在局部极小的点收敛速度慢问题。对于复杂的问题,训练过程则需要迭代几千甚至上万次才能将结果收敛到期望的精度。并且训练中的稳定性又要求学习率很小,因此梯度下降法会使得训练很慢。因此,标准的 BP 网络在很多方面表现出了不实用性,特别是对实时性很强的系统。鉴于传统的 BP 算法存在着以上的问题,国内外学者已经提出不少有效的改进方法,主要有两类:一类是基于标准梯度下降(如附加动量法、自适应学习率法等)进行改进;另一类是基于标准数值优化(如变步长法、牛顿法以及 Levenberg-Marquardt 算法等)进行改进。

2.2.1 基于变步长法改进

BP 算法之所以收敛速度慢,一个主要原因是学习速率 η 选取不当。 η 值选取过大,在算法收敛接近目标时易产生震荡,无法收敛到正确解;而 η 值选得过小,又会使整体学习速率变慢。因此,可以结合滤波器的思想设计一种自适应学习速率 η 。

假如第 $t-1$ 次计算的误差为 ε_{t-1} ,第 t 次计算的学习速率为 η_t ,第 $t+1$ 次计算的学习速率为 η_{t+1} ,误差为 ε_t 。可以定义 $\beta = \text{sgn}(\varepsilon_t - \varepsilon_{t-1})$,则 $\eta_{t+1} = \eta e^{-\lambda \beta}$ 。其中, λ 为自适应系数,取值为 0.01~0.2;而 β 实质上起到了一个滤波器的作用,导致误差减小的学习速率得以加强,反之则会被抑制。

2.2.2 基于牛顿法改进

由于梯度法在搜索过程中收敛速度较慢,牛顿法在这方面有所改进,它不但利用了准则函数在搜索点的梯度,还利用了它的二次导数。牛顿法的原理是求 ε 的二次近似的驻点,它用一个二次函数逼近 ε ,然后再求其驻点。若该函数为有强极小点的二次函数,它就能实现一步极小化。如果 ε 不是二次函数,则牛顿法一般就不能在一步内收敛。实际上根本无法确定它是否收敛,因为这取决于具体的函数和初始点。其主要改进是改变更新规则为:

$$\Delta W_{ij} = -\eta \left[\frac{\frac{\partial e_j}{\partial W_{ij}}}{\left(\frac{\partial^2 e_j}{\partial^2 W_{ij}} + \varepsilon_0 \right)} \right] \quad (11)$$

2.2.3 基于 Levenberg-Marquardt 算法改进

BP 网络的训练实质是非线性目标函数的优化问题,基于数值优化的 LM 算法不但利用了目标函数的一阶导数信息,还利用了目标函数的二阶导数信息。LM 算法的迭代公式为:

$$X(k+1) = X(k) - (J^T(k)J(k) + \mu I)^{-1} J^T(k) F(X(k)) \quad (12)$$

其中, I 为单位矩阵, μ 为阻尼因子, J 是包含网络误差对权值和阈值一阶导数的 Jacobin 矩阵。

2.3 已提出的较好的改进方法

参考文献[1]中提出了将附加动量法和自适应学习率调整法两种策略结合起来的一种方法。这两种方法相结合既可有效地抑制网络陷入局部极小,又有利于缩短学习时间。但上述两种改进方法都是在标准梯度下降法基础上发展起来的,它们只利用了目标函数对网络权值和阈值的一阶导数信息。因此,当用于较复杂的问题时,仍存在收敛速度不够快的问题。

基于数值优化的 LM 算法是根据迭代的结果动态地调整阻尼因子,即动态地调整迭代的收敛方向,可使每次的迭代误差函数值都有所下降。它是梯度下降法和牛顿法的结合,收敛速度较快。但 LM 算法在计算过程中会产生大量的中间结果矩阵,因此,需要较大的内存空间,比较适宜中型网络使用。

可以考虑设计一个新的数字识别系统,以变步长法和牛顿法来改进 BP 算法,使其能够适应对数字样本组成的样本集的训练。

3 基于 BP 网络对数字识别系统的设计

本文采用逐像素特征提取方法提取数字样本的特征向量。设定归一化后的图像形成一个 8×16 的像素矩阵,依次取每列的元素转化为 128×1 的列矩阵,即数字字符的特征向量。这样把最后提取到的样本特征向量送到数字识别模块中,经过神经网络样本训练并记录下网络参数,最后对识别数据进行识别并给出结果。BP 神经网络的数字识别系统数据处理流程如图 2 所示。

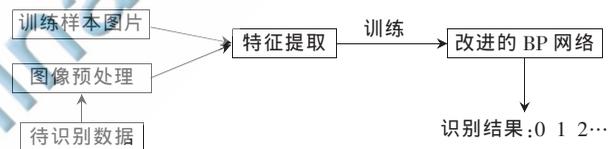


图2 BP神经网络的数字识别系统数据处理流程图

3.1 确定输入层神经元节点数

BP 神经网络的输入层神经元节点数就是图像预处理后所输出的特征的个数。采用 8×16 的归一化处理以及逐像素特征提取方法,利用每个点的像素值作为特征,设定每个字符的标准归一化的宽度为 8,高度为 16,那么每个字符就有 $8 \times 16 = 128$ 个特征像素点,这样对于每个输入的样本,就有 128 个特征,由此可以确定输入层的节点数为 128。

3.2 确定隐含层神经元节点数

相对而言,隐含层神经元节点数没有特别的要求,隐含层神经元的数目越多, BP 网络也就越精确,训练时间也就会加长。然而,需要注意的是,隐含层节点不宜选取太多,否则会造成识别率的急剧下降,同时还会降低网络的抗噪声能力。到目前为止,隐含层的神经元的设定要不断地试验才有可能得到最佳的数目,在这里可以设定隐含层神经元的个数为 10。

3.3 确定输出层神经元节点数

对数字字符, 不管它的训练样本中有多少种字体, 有多少个训练样本, 也只有 10 个数需要编码, 只需设定输出神经元个数为 4 就可满足 $2^m \geq N$ 的条件。其中, m 为输出层神经元个数, N 是需要编码的训练样本个数。

由于采用 S 函数做激活函数, 它的输出不可能达到 0 或 1, 只能是无限接近。可以在编码时做一个简单处理, 将 0 改为 0.1, 1 改为 0.9。

根据要求采用 10 个节点确定输入层的节点数, 编码方案为:

- 0 的编码为: (0.1, 0.1, 0.1, 0.1)
- 1 的编码为: (0.1, 0.1, 0.1, 0.9)
- 2 的编码为: (0.1, 0.1, 0.9, 0.1)
- 3 的编码为: (0.1, 0.1, 0.9, 0.9)
- 4 的编码为: (0.1, 0.9, 0.1, 0.1)
- 5 的编码为: (0.1, 0.9, 0.1, 0.9)
- 6 的编码为: (0.1, 0.9, 0.9, 0.1)
- 7 的编码为: (0.1, 0.9, 0.9, 0.9)
- 8 的编码为: (0.9, 0.1, 0.1, 0.1)
- 9 的编码为: (0.9, 0.1, 0.1, 0.9)

4 实验结果与分析

在隐节点数一定的情况下, 为了获得较好的泛化能力, 存在一个最佳次数 t_0 , 这时会使得测试数据的均方误差最小。因此, 可随意设置最小均方误差, 本文设为 0.000 1, 设最大训练次数为 16 000, 这样就不会出现训练不足的情况, 使其可以达到最小的均方误差。

4.1 改进前后的 BP 神经网络的性能比较

将参考文献[1]中提出的方法、基本 BP 算法与本文提出的方法进行实验数据比较, 收敛速度比较如表 1 所示。

表 1 3 种算法比较

	基本 BP 算法	参考文献[1]算法	本文算法
最少迭代次数	1 642	104	63
最大迭代次数	5 704	658	447
平均迭代次数	3 256	186	97

从表 1 可以看出, 标准 BP 算法的收敛速度很慢, 使用参考文献[1]中的方法可以加快网络收敛速度, 而使用本文提出的新方法, 速度比这两种方法更快。但由于本文提出的方法是两种基于数值优化方法的结合, 它比参考文献[1]算法需要更大的内存空间, 但需要的内存空间少于 Levenberg-Marquardt 算法。因此, 对于一个给定的问题, 采用何种 BP 算法, 取决于问题的复杂程度、训练样本数、权值以及阈值数目等各种因素。

参考文献

- [1] 王婷, 江文辉, 肖南峰. 基于改进 BP 神经网络的数字识别[J]. 电子设计工程, 2011, 19(3): 108-112.
 - [2] 万红, 冯向荣, 吕明相. 改进 BP 算法在识别含噪声字符中的应用[J]. 微计算机信息, 2008, 24(10): 207-208, 190.
 - [3] 赵长青, 李欧迅, 黄书童. BP 神经网络数字识别系统的设计方法[J]. 桂林航天工业高等专科学校学报, 2010(3): 292-294.
 - [4] 施少敏, 马彦恒, 陈建泗. 基于 BP 神经网络的数字识别方法[J]. 网络与信息技术, 2006, 25(10): 40-41.
- (收稿日期: 2011-12-26)

作者简介:

刘扬, 男, 1966 年生, 硕士, 副教授, 主要研究方向: 机械设计及理论, CAD/CAE/CAM。