

一种矢量倍增算法的神经网络人脸识别方法

张 辰, 潘保昌, 郑胜林, 卢毅涛

(广东工业大学 信息工程学院, 广东 广州 510640)

摘要: 针对神经网络人脸识别中训练速度慢的问题, 深入研究输入矢量长度的变化对神经网络收敛速度的影响, 提出了一种矢量倍增算法对神经网络进行优化. 这种方法是对网络输入矢量的长度倍增, 从而提高神经网络收敛速度. 通过人脸识别实验系统验证了矢量倍增算法的实用性.

关键词: 人脸识别; KL变换; BP神经网络; 矢量倍增算法

中图分类号: TP391.41

文献标识码: A

文章编号: 1007-7162(2006)03-0108-05

人脸识别早在 20 世纪六七十年代就引起了人们的研究兴趣, 近几年随着它在安全验证系统、信用卡验证、医学、档案管理系统、图像检索等领域的巨大应用前景, 它已经成为计算机视觉和模式识别领域的一个研究热点. 国际上形成了以下几类主要的人脸识别方法: 基于几何特征的人脸识别方法, 基于模板匹配的人脸识别方法, 基于 KL 变换和神经网络的人脸识别方法, 隐马尔可夫模型方法, 基于动态链接结构的弹性图匹配方法, 利用运动和颜色信息对动态图像序列进行人脸识别的方法^[1]等.

在基于 KL 变换和神经网络的人脸识别方法中^[2-4], KL 变换对人脸空间进行降维处理, 神经网络对 KL 变换结果进行训练、分类. 研究中发现, 直接将 KL 变换结果输入神经网络训练, 网络收敛速度非常慢. 本文对神经网络的训练方法进行了深入研究, 提出一种矢量倍增算法. 这种方法将 KL 变换得到的矢量的长度倍增, 再输入神经网络进行训练, 达到提高神经网络收敛速度的目的.

1 矢量倍增算法

直接将 KL 变换的数据输入神经网络进行训练, 神经网络收敛速度很慢. 因此本文分析了输入矢量长度的变化对神经网络的收敛速度影响后, 提出一种矢量倍增算法, 这种方法提高了神经网络的收敛速度.

1.1 输入矢量长度的变化对神经网络性能的影响

图 1 所示为一个三层反向传播神经网络, 所有层的传输函数为 $\log\text{sig}$ 函数, n 表示训练矢量的个数, 也代表训练矢量的维数 (训练矢量组为方阵).

设神经网络的一列输入矢量为 $\mathbf{p}$, 如果直接将 $\mathbf{p}$ 输入神经网络, 此时神经网络训练过程为^[5]

$$\text{网络输出: } \mathbf{a}^0 = \mathbf{p}, \mathbf{a}^1 = \log\text{sig}(\mathbf{W}^1 \mathbf{a}^0 + \mathbf{b}^1), \mathbf{a}^2 = \log\text{sig}(\mathbf{W}^2 \mathbf{a}^1 + \mathbf{b}^2),$$

收稿日期: 2005-06-29

作者简介: 张辰 (1982-), 男, 在读研究生, 主要研究方向为信息工程.

网络权值更新: $\mathbf{W}^m = \mathbf{W}^m - \alpha \mathbf{s}^m (\mathbf{a}^{m-1})^T$, $\mathbf{b}^m = \mathbf{b}^m - \alpha \mathbf{s}^m$, $m = 1, 2$

如果将 $\mathbf{p}$ 乘以倍数因子 γ 进行长度延长, 即 $\mathbf{p}' = \gamma \mathbf{p}$, 然后输入神经网络, 此时神经网络训练过程为

网络输出: $\mathbf{a}^0 = \mathbf{p}'$, $\mathbf{a}^1 = \text{logsig}(\mathbf{W}^1 \mathbf{a}^0 + \mathbf{b}^1) = \text{logsig}(\gamma \mathbf{W}^1 \mathbf{p} + \mathbf{b}^1)$, $\mathbf{a}^2 = \text{logsig}(\mathbf{W}^2 \mathbf{a}^1 + \mathbf{b}^2)$,

网络权值更新: $\mathbf{W}^1 = \mathbf{W}^1 - \alpha \mathbf{s}^1 (\mathbf{p}')^T = \mathbf{W}^1 - \gamma \alpha \mathbf{s}^1 (\mathbf{p})^T$, $\mathbf{W}^2 = \mathbf{W}^2 - \alpha \mathbf{s}^2 (\mathbf{a}^1)^T$, $\mathbf{b}^m = \mathbf{b}^m - \alpha \mathbf{s}^m$, $m = 1, 2$

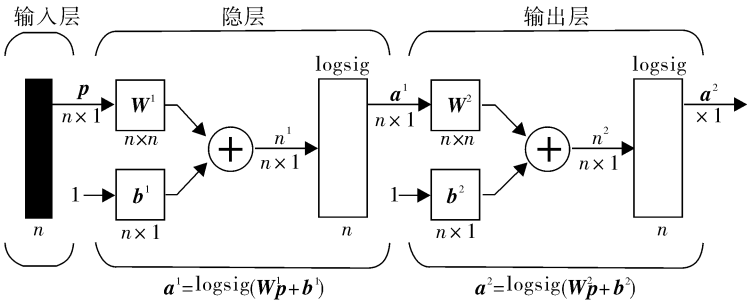


图 1 三层反向传播神经网络

对比两种输入情况可以发现: 首先, 如果神经网络在两种情况下都要求达到相同的输出效果, 则引入倍数因子能使神经网络训练过程中权值 $\mathbf{W}^1$ 的变化范围减少. 其次, 倍数因子可以看作直接作用在权值 $\mathbf{W}^1$ 的学习速度上, 因此引入倍数因子 γ 增加了神经网络首层权值 $\mathbf{W}^1$ 的学习速度 γ 倍, 但对神经网络其他参数的影响较小. 例如分析倍数因子 γ 对敏感性系数 $\mathbf{s}^1$ 的影响: 已知隐层的传输函数为 logsig 因此敏感系数 $\mathbf{s}^1$ 为

$$\mathbf{s}^1 = \begin{bmatrix} a_1^1 (1 - a_1^1) \\ a_2^1 (1 - a_2^1) \\ a_3^1 (1 - a_3^1) \end{bmatrix},$$

logsig 函数的输出在 $(0, 1)$ 之间, 因此 $\mathbf{s}^1$ 中非零元素的取值范围为 $(0, 0.25)$. 可见 γ 对 $\mathbf{s}^1$ 最大的影响范围也仅在 $(0, 0.25)$ 内, 与 γ 对神经网络首层权值 $\mathbf{W}^1$ 学习速度的影响相比非常小.

综上所述, 一般情况下未经优化处理的神经网络输入矢量的长度并不一致, 则神经网络权值 $\mathbf{W}^1$ 训练时的变化范围不一致, 同时权值 $\mathbf{W}^1$ 的学习速度是不等速的, 大量实验证明此时网络收敛很慢, 因此本文提出一种矢量倍增算法, 其先对输入矢量的长度进行归一化处理, 然后增加对矢量长度倍增 γ 倍. 优化后, 神经网络的权值 $\mathbf{W}^1$ 变化范围减少, 同时网络训练时权值 $\mathbf{W}^1$ 的学习速度是等速的, 且增加了 γ 倍. 通过实验验证上述分析, 实验采用图 1 的神经网络, 学习速度为 0.9 单列矢量的训练误差阈值为 0.02 整体训练误差阈值为 0.09 倍数因子 $\gamma = 5$ 其对三类长度为 1 的三维矢量进行训练, 即图 1 中 n 取 3 其中权值矢量 $(w_{21}^1, w_{22}^1, w_{23}^1)$ 训练过程如图 2、图 3

图 2 体现了权值 $(w_{21}^1, w_{22}^1, w_{23}^1)$ 在整个神经网络训练中的调整轨迹曲线, 其说明了引入倍数因子, 能使权值 $(w_{21}^1, w_{22}^1, w_{23}^1)$ 变化范围减小 (图中粗线所示).

图 3 上、下两幅子图分别代表引入、没有引入倍数因子对应的权值在训练中更新量 $\Delta(w_{21}^1, w_{22}^1, w_{23}^1)$ 的分布. 分布在图中的点离坐标原点的距离等于更新量的模, 其能体现每次迭代时更新量的大小, 即模越大更新量越大, 则点分布离坐标原点越远, 相反则越近. 同时, 本文对分布在最

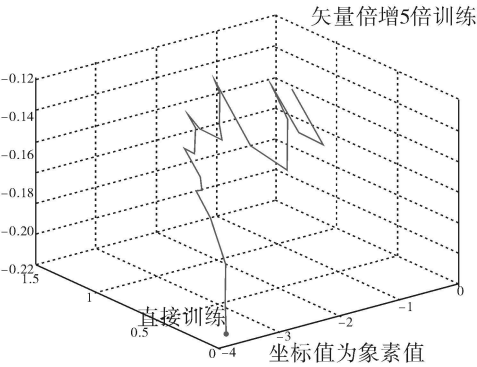


图 2 权值矢量的变化轨迹

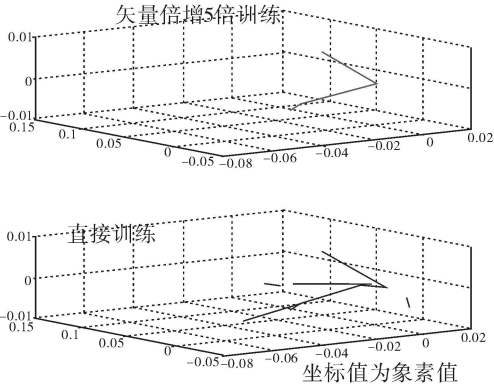


图 3 权值矢量每次迭代的变化量示意图

远的点离坐标原点的距离称为更新量的模的极值, 离坐标原点距离大于极值的 50% 的点的分布称为远场分布, 离坐标原点距离小于极值的 5% 的点的分布称为近场分布, 如图 4 所示. 统计图 3 的分布得到表 1 其说明引入倍数因子使权值 ($w_{21}^1 w_{22}^1 w_{23}^1$) 的更新速度增大, 使网络很快进入了微调整.

表 1 图 3 中更新量分布的统计结果

	迭代次数	更新量变化量极值	远场分布率	近场分布率
引入倍数因子	1 264	0 076 6	0 316 7%	97 228 8%
直接训练	143 2	0 076 4	1 537 4%	82 459 8%

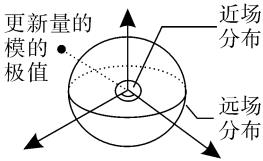


图 4 变化量分布示意图

1 2 矢量倍增算法原理

总结上文, 得到矢量倍增算法的步骤如下: 首先对输入矢量的长度归一化处理, 假设神经网络的 M 个输入矢量为 $\Omega_1, \Omega_2, \dots, \Omega_M$, 根据归一化的公式^[6]

$$\Omega_i' = \frac{\Omega_i}{|\Omega_i|}, \quad i = 1, 2, \dots, M,$$

得到归一化后的输入矢量 $\Omega_1', \Omega_2', \dots, \Omega_M'$. 然后对矢量长度倍增, 即对归一化后的输入矢量组乘以一个倍数因子 γ 得

$$T_i = \gamma \Omega_i', \quad i = 1, 2, \dots, M,$$

得到的矢量 $T_1, T_2, \dots, T_M$, 其长度被延长了 γ 倍. 最后将优化后的矢量输入一个 n 层的神经网络进行运算:

$$a^1 = f^1(W^1 T_i + b^1), \quad i = 1, 2, \dots, M, \quad a^k = f^k(W^k a^{k-1} + b^k), \quad k = 2, \dots, n$$

从形式上观察神经网络的计算过程并没有太大改变, 但是实质上倍数因子 γ 大大提高了神经网络的性能. 本文通过反复的实验得到, 对于图 1 的神经网络, 当学习速度取 0.9 的时候, 倍数因子取 5 比较适合.

2 基于矢量倍增算法的神经网络人脸识别方法

基于矢量倍增算法的神经网络人脸识别方法的系统框图如图 5 所示. 首先对人脸进行 KL 变换得到一组数据, 然后利用矢量倍增算法对数据进行处理并作为神经网络的输入数据, 而神经网络作为分类器, 对人脸进行划分、识别.

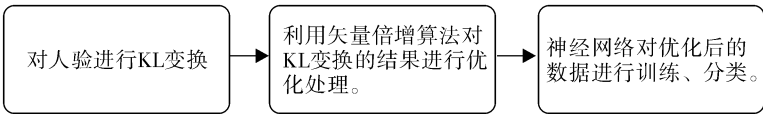


图 5 神经网络人脸识别方法流程图

2 1 KL变换的原理

设有 M 类人脸, 每类 t 张, 共 Mt 张 $I_1 I_2 \dots I_{Mt}$, 其中 $I_i = I_i(x, y)$, $x \in (1, N)$, $y \in (1, N)$. 将其转换为列矢量 $\Gamma_1 \Gamma_2 \dots \Gamma_{Mt}$, 其中 $\Gamma_i = \Gamma_i(t)$, $t \in (1, N^2)$. 计算此列矢量组的均值 Ψ 和每个列矢量对均值的偏移量 Φ :

$$\Psi = \sum_{n=1}^{Mt} \Gamma_n, \Phi_i = \Gamma_i - \Psi, i = 1, 2, \dots, Mt, \Phi = [\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_{Mt}].$$

利用偏移量向量计算人脸库的协方差矩阵, 并计算其特征值、特征向量. 协方差矩阵计算如下:

$$C = \frac{1}{Mt} \times \sum_{n=1}^{Mt} \Phi_n \Phi_n^T = AA^T, A = \frac{1}{\sqrt{Mt}} [\Phi_1, \Phi_2, \Phi_3, \dots, \Phi_{Mt}].$$

C 的特征值、特征向量为 $\lambda_i, \mu_i, i = 1, 2, \dots, N^2$. 对特征值排序, 选取较大的 M 个 (例如当对 10 类人脸, 每类 5 张图, 共 50 张图进行变换时, $M = 10$), 得到对应的特征向量 $\mu_k, k = 1, 2, \dots, M$, 即为特征脸; 人脸库人脸投影到特征脸空间得到一组权值

$$\Omega_i = [\omega_1, \omega_2, \dots, \omega_M], \omega_k = \mu_k^T \times \Phi_i, i = 1, 2, \dots, M.$$

2 2 反向传播神经网络的训练与识别

神经网络训练时, 根据矢量倍增算法对 KL 变换得到的数据 Ω_i 进行归一化处理, 然后对矢量长度倍增, 即

$$T_i = \gamma \cdot \frac{\Omega_i}{|\Omega|}, i = 1, 2, \dots, M,$$

得到 T 再输入神经网络进行训练.

神经网络采用 $n-n-n$ 型三层反向传播神经网络, n 等于输入矢量维数, 也等于待分类人脸类型总数. 神经网络拓扑结构如图 1, 学习速度为 0.9, 单列矢量的训练误差阈值为 0.02, 整体训练误差阈值为 0.09, 倍数因子 $\gamma = 5$, 教师信号为一列矢量, 矢量中元素为 1 的下标代表待识别人脸类型, 其余为 0. 训练过程公式如下:

网络各层输出: $a^0 = T, i = 1, 2, \dots, M, a^1 = \text{logsig}(W^1 a^0 + b^1), a^2 = \text{logsig}(W^2 a^1 + b^2),$

网络参数更新: $W^m = W^m - \alpha s^m (a^{m-1})^T, b^m = b^m - \alpha s^m, m = 1, 2$

训练结束后保存神经网络的参数, 进入人脸识别阶段. 任意选取一张人脸 I' 作为待识别人脸, 将其转化成为列矢量 Γ' , 然后按照下列公式计算, 即得到神经网络输出结果:

$$\Omega' = \begin{bmatrix} \mu_1^T \\ \dots \\ \mu_M^T \end{bmatrix} \cdot \Gamma', T' = \gamma \cdot \frac{\Omega'}{|\Omega'|}$$
$$a^2 = \text{logsig}(W^2 (\text{logsig}(W^1 T' + b^1)) + b^2).$$

取出 a^2 中最大元素的下标, 其为待识别人脸类型.

3 实验结果

人脸库作为数据库. 数据库中一共有 40 类人脸, 每类 10 张不同的人脸图. 实验选取每类人脸中前 5 张作为训练集, 选取全部 10 张作为待识别集. 例如待识别人脸类型 n 等于 10 训练集一共有 50 张人脸图, 待识别集一共有 100 张人脸图. 实验采用的神经网络如上文所述. 为保证实验数据的可比性, 对于相同的人脸库类型总数, 神经网络初始值不变. 实验结果见表 2 表 3 所示.

表 2 迭代次数对比表

人脸种类数	矢量倍增算法	不采用优化技术
10	10 518	20 248
15	17 253	25 671
20	26 078	35 431
25	38 499	53 842
30	48 289	无法收敛
35	62 832	无法收敛

表 3 误差对比表

人脸种类数	矢量倍增算法	不采用优化技术
10	0 06	0 19
15	0 106 67	0 193 33
20	0 135	0 325
25	0 164	0 336
30	0 183 33	无法收敛
35	0 182 86	无法收敛

经过矢量倍增算法优化后的人脸识别系统中, 神经网络的收敛速度大大提高. 特别对于识别 30 类人脸以上的情况, 没有优化的人脸识别系统, 神经网络出现振荡, 无法收敛; 而经过优化的, 神经网络处理数据游刃有余.

4 结论

本文采用的基于矢量倍增算法的神经网络人脸识别方法首先对人脸进行 KL 变换得到一组数据, 然后利用了矢量倍增算法对数据进行处理并作为神经网络的输入数据, 而神经网络作为分类器, 对人脸进行分类、识别.

本文对 BP 神经网络识别方法的算法进行了深入研究, 论述了如何通过优化神经网络的输入矢量来提高神经网络的收敛速度的问题. 在此基础上提出了一种矢量倍增算法, 其对网络输入矢量的长度归一化处理, 然后使用倍数因子对其长度倍增, 达到提高神经网络收敛速度的目的. 通过大量的实验, 本文证实了矢量倍增算法的实用性, 使神经网络的收敛速度大大提高.

参考文献:

[1]苏剑波, 徐波. 应用模式识别技术导论——人脸识别与语音识别[M]. 上海: 上海交通大学出版社, 2001.

[2]Matthew Turk, Alex Pentland. Eigenfaces for recognition[J]. Journal of Cognitive Neuroscience, 1991, 3(1): 71-86

[3]M. Iby, Sirovich. 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

[4]甘俊英, 张有为. 基于 BP 神经网络的人脸识别[J]. 系统工程与电子技术, 2003, 25(1): 113-115

[5]Martin T Hagan, Howard B Demuth, Mark H Beale. 神经网络设计[M]. 戴葵译. 北京: 机械工业出版社, 2002

[6]同济大学数学教研室. 工程数学: 线性代数(第 3 版)[M]. 北京: 高等教育出版社, 1999

(下转第 117 页)

Writing Numbers Identification by Projection in Seven Segments Frame “8”

MA Bi-yan, ZHENG Sheng-lin, ZHAO Quan-you

(Faculty of Information Engineering Guangdong University of Technology, Guangzhou 510640, China)

Abstract According to the fact that the electronic screen just needs a frame “8” to denote the 10 numbers from 0 to 9, it is attempted to take the numbers as a whole to be put to the frame “8” and then recognized. This text is put forward according to the traditional recognition method, the baseline is used to project while bias is used in traditional recognition method. Projection is made to each frame of the frame “8” according to the shell character of the digit, and hand-writing numbers are changed to standard seven segments frame numbers, which can predigest the distilling of the feature of the number. Compare to the traditional recognition method, this one can be carried out much more easily. Its validity is proved in our experiment.

Key words digit recognition, frame “8”; projection

(上接第 112 页)

Neural Network Face Recognition Based on Length Extension Technique

ZHANG Chen, PAN Bao-chang, ZHENG Sheng-lin, LU Yi-tao

(Faculty of Information Engineering Guangdong University of Technology, Guangzhou 510640, China)

Abstract After a great deal of investigations on the principle of BP model and the structure of the neural network, this paper manages to demonstrate how to increase the convergent speed of the BP neural network by optimizing the input of the network. Furthermore, a creative optimization technique—length extension technique, is presented here to improve the performance of neural network. After lots of experiments, this paper successfully increases the convergent speed of the BP neural network.

Key words face recognition, K-L expansion, BP neural network, length extension technique (LET)