

多层前向人工神经网络图像分类算法

顾哲彬 曹飞龙

(中国计量大学理学院 杭州 310018)

摘要 传统人工神经网络的输入均为向量形式,而图像由矩阵形式表示,因此,在用人工神经网络进行图像处理时,图像将以向量形式输入至神经网络,这破坏了图像的结构信息,从而影响了图像处理的效果。为了提高网络对图像的处理能力,文中借鉴了深度学习的思想与方法,引进了具有矩阵输入的多层前向神经网络。同时,采用传统的反向传播训练算法(BP)训练该网络,给出了训练过程与训练算法,并在 USPS 手写数字数据集上进行了数值实验。实验结果表明,相对于单隐层矩阵输入前向神经网络(2D-BP),所提多层网络具有较好的分类效果。此外,对于彩色图片分类问题,利用所提出的 2D-BP 网络,给出了一个有效的可行方法。

关键词 多层前向神经网络,学习算法,图像分类,梯度优化

中图分类号 TN911.71,TP183 **文献标识码** A

Algorithm of Multi-layer Forward Artificial Neural Network for Image Classification

GU Zhe-bin CAO Fei-long

(College of Sciences, China Jiliang University, Hangzhou 310018, China)

Abstract The input of traditional artificial neural network is in vector form, but the image is represented by matrix. Therefore, in the process of image processing, the image will be inputted into the neural network in vector form, which will destroy the structure information of image, and thus affect the effect of image processing. In order to improve the ability of network on image processing, the multilayer feedforward neural networks with matrix inputs are introduced based on the idea and method of deep learning. At the same time, the traditional back-propagation algorithm (BP) is used to train the network, and the training process and training algorithm are given. After a lot of experiments, the network structure with good performance were determined, and the numerical experiments were carried out on the USPS handwritten digital data set. The experimental results show that the proposed multilayer network has better classification results than the single hidden layer feed forward neural network with matrix input (2D-BP). In addition, to deal with the problem of color image classification, this paper provided an effective and feasible method, the new 2D-BP network, to deal with it.

Keywords Multilayer feedforward neural networks, Learning algorithm, Image classification, Gradient optimization

1 引言

随着计算机技术、数据库技术和网络技术的飞速发展,图像信息量迅猛增加,依靠人工对海量图像进行分类和标注的管理方式已经远无法满足应用的需求。因此,利用计算机技术自动进行图像理解与处理成为目前的一个研究热点,它主要是对图像的语义解释,研究图像中包含的目标、目标之间的相互关系以及图像场景分类等问题^[1]。图像分类是计算机视觉、模式识别与机器学习领域中的基本任务,已广泛应用于人脸识别、行人检测、智能视频分析、场景物体识别、车牌识别,以及基于内容的图像检索等领域。

图像分类往往先使用图像特征提取器对图像进行特征提取,再用分类器对提取的特征进行分类。而分类器对图像的分类能力,往往取决于其提取图像特征的能力。传统的特征

提取方法,如主成分分析法(PCA)^[2-3]、线性判别分析法(LDA)^[4-5]、支持向量机(SVM)^[6]、K近邻(KNN)^[7]、前向神经网络(FNN)^[8],需要将数据转换成向量的形式作为输入。然而,对图像分类而言,将二维的图像转换成一维的形式,会破坏图像的空间结构,将直接影响图像的分类效果。文献[9]在 2DPCA^[10]和 2DLDA^[11]的启发下,在文献[12]的基础上提出了一个 2D-BP 网络,该网络将以向量数据为输入的单隐层前向神经网络推广到了以矩阵数据为输入的单隐层神经网络。

2D-BP 网络使得网络的输入保留了图像的空间结构,从而提高了网络的分类能力。然而,2D-BP 网络作为 2 层的网络,且网络第二层的节点数量根据图片类别固定,只有第一层节点的数量能够调整,使得网络的泛化能力有限。

回顾深度学习中卷积神经网络的发展^[13-18],不难发现,

本文受国家自然科学基金(61672477)资助。

顾哲彬(1992—),男,硕士,主要研究方向为机器学习、图像处理,E-mail:gzb@foxmail.com;曹飞龙(1965—),男,博士,教授,博士生导师,主要研究方向为智能计算、图像处理等,E-mail:feilongcao@gmail.com(通信作者)。

合理地加深网络能够有效地提高网络的特征提取能力。于是,我们很自然地想到:是否能够通过加深 2D-BP 网络来提升网络的特征提取能力?

通过加入全连接层的方式,对 2D-BP 网络进行了推广,形成了 3~5 层网络用于图像分类,分类能力对于 2D-BP 网络有明显的提升。

本文第 2 节介绍 2D-BP 网络;第 3 节提出基于 2D-BP 网络的多层 2D-BP 网络算法以及三通道 2D-BP 网络算法;第 4 节给出实验细节,通过实验分析不同结构下的网络性能;最后总结全文。

2 2D-BP 网络

2D-BP 网络是文献[9]在文献[12]的基础上提出的单隐层前向神经网络,2D-BP 网络的模型如式(1)所示:

$$y_k^{(2)} = \sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)}k^{(1)}} \sigma(u_k^{(1)T} X v_{k^{(1)}} + b_k^{(1)}) + b_k^{(2)} \quad (1)$$

其中,输入 $X = (x_{rc})_{m \times n}$ 是一个 $m \times n$ 维矩阵; $u_k^{(1)} = (u_{k^{(1)}1}, \dots, u_{k^{(1)}r}, \dots, u_{k^{(1)}m})^T$; $v_{k^{(1)}} = (v_{k^{(1)}1}, \dots, v_{k^{(1)}c}, \dots, v_{k^{(1)}n})^T$; $b^{(i)} = (b_1^{(i)}, \dots, b_k^{(i)}, \dots, b_{L^{(i)}}^{(i)})^T$ 是网络第 i 层的偏置, $i=1, 2$; σ 为 sigmoid 激活函数; $\beta_{k^{(1)}} = (\beta_{k^{(1)}1}, \dots, \beta_{k^{(1)}k^{(1)}}, \dots, \beta_{k^{(1)}L^{(1)}})^T$; $y_k^{(2)}$ 是网络输出 $y = (y_1, \dots, y_k^{(2)}, \dots, y_{L^{(2)}})^T$ 的第 $k^{(2)}$ 个分量; $L^{(i)}$ 是网络第 i 层的结点数量, $i=1, 2$ 。

2D-BP 网络保留了输入图像的空间结构,然而 2D-BP 网络作为单隐层前向神经网络,其泛化能力有限。

3 本文方法

本节提出基于 2D-BP 网络的改进网络,包括对 2D-BP 网络进行加深的多层 2D-BP 网络(见 3.1 节—3.3 节)以及对 2D-BP 网络进行加宽的三通道 2D-BP 网络(见 3.4 节),并给出网络的训练方法。

为了提升 2D-BP 网络的泛化能力,通过加入全连接层的方式对 2D-BP 网络进行推广。

3.1 3 层网络

对于 2D-BP 网络,3 层网络模型相对于式(1)添加了 1 层全连接层,如式(2)所示:

$$y_k^{(3)} = \sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(2)}k^{(3)}} \sigma\left(\sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)}k^{(1)}} \sigma(u_k^{(1)T} X v_{k^{(1)}} + b_k^{(1)}) + b_k^{(2)}\right) + b_k^{(3)} \quad (2)$$

其中, $L^{(1)}, L^{(2)}, L^{(3)}$ 分别是网络中第 1, 2, 3 层的节点数量, $\omega_{k^{(2)}k^{(3)}}$ 是第二层的第 $k^{(2)}$ 个结点与第三层的第 $k^{(3)}$ 个结点之间的权重。

记:

$$g_k^{(1)} = \sigma(u_k^{(1)T} X v_{k^{(1)}} + b_k^{(1)}) = \sigma\left(\sum_{c=1}^m \sum_{r=1}^n u_{k^{(1)}r} x_{rc} v_{ck^{(1)}} + b_k^{(1)}\right)$$

$$g_k^{(2)} = \sigma\left(\sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)}k^{(1)}} g_k^{(1)} + b_k^{(2)}\right)$$

$$g_k^{(3)} = \sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(2)}k^{(3)}} g_k^{(2)} + b_k^{(3)}$$

为网络中各层节点的输出,特别地, $g_k^{(3)} = y_k^{(3)}$ 。网络的损失函数为:

$$E = \frac{1}{2} \sum_{k^{(3)}=1}^{L^{(3)}} (y_k^{(3)} - t_k^{(3)})^2 \quad (3)$$

其中, $t = (t_1, \dots, t_k^{(3)}, \dots, t_{L^{(3)}})^T$ 为样本 X 的标签, $y = (y_1, \dots, y_k^{(3)}, \dots, y_{L^{(3)}})^T$ 是网络的输出。网络使用动量梯度下降法更新 y 中的权重 $\Theta = (U, V, \beta, b^{(1)}, b^{(2)}, b^{(3)})$ 以最小化损失函数式(3),其中, $U = (u_1, \dots, u_{k^{(1)}}, \dots, u_{L^{(1)}}), V = (v_1, \dots, v_{k^{(1)}}, \dots, v_{L^{(1)}}), \beta = (\beta_1, \dots, \beta_{k^{(1)}}, \dots, \beta_{L^{(1)}})$, 更新公式为:

$$\Delta \Theta_i = \gamma \Delta \Theta_{i-1} - (1 - \gamma) \eta \nabla \Theta \quad (4)$$

其中, i 是权重的迭代次数, γ 为动量系数, η 为学习率, $\nabla \Theta$ 为损失函数关于权重的偏导。为了求 $\nabla \Theta$, 对 $y_k^{(3)}$ 中的每个变量求导:

$$\frac{\partial E}{\partial b_k^{(3)}} = (g_k^{(3)} - t_k^{(3)})$$

$$\frac{\partial E}{\partial \omega_{k^{(2)}k^{(3)}}} = (g_k^{(3)} - t_k^{(3)}) g_k^{(2)}$$

$$\frac{\partial E}{\partial b_k^{(2)}} = \sum_{k^{(3)}=1}^{L^{(3)}} (g_k^{(3)} - t_k^{(3)}) \omega_{k^{(2)}k^{(3)}} g_k^{(2)}$$

$$\frac{\partial E}{\partial \beta_{k^{(2)}k^{(1)}}} = \sum_{k^{(3)}=1}^{L^{(3)}} (g_k^{(3)} - t_k^{(3)}) \omega_{k^{(2)}k^{(3)}} g_k^{(2)} g_k^{(1)}$$

$$\frac{\partial E}{\partial b_k^{(1)}} = \sum_{k^{(3)}=1}^{L^{(3)}} (g_k^{(3)} - t_k^{(3)}) \sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(2)}k^{(3)}} g_k^{(2)} \beta_{k^{(2)}k^{(1)}} g_k^{(1)}$$

$$\frac{\partial E}{\partial u_{k^{(1)}r}} = \sum_{k^{(3)}=1}^{L^{(3)}} (g_k^{(3)} - t_k^{(3)}) \sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(2)}k^{(3)}} g_k^{(2)} \beta_{k^{(2)}k^{(1)}} g_k^{(1)}$$

$$\sum_{c=1}^m x_{rc} v_{ck^{(1)}}$$

$$\frac{\partial E}{\partial v_{ck^{(1)}}} = \sum_{k^{(3)}=1}^{L^{(3)}} (g_k^{(3)} - t_k^{(3)}) \sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(2)}k^{(3)}} g_k^{(2)} \beta_{k^{(2)}k^{(1)}} g_k^{(1)}$$

$$\sum_{r=1}^n u_{k^{(1)}r} x_{rc}$$

$\nabla \Theta$ 如下:

$$\nabla b^{(i)} = \delta_i, i=1, 2, 3 \quad (5)$$

$$\nabla \omega^{(1)} = \delta_3 G^{(2)T} \quad (6)$$

$$\nabla \beta = \delta_2 G^{(1)T} \quad (7)$$

$$\nabla U = S_1 \circ (V^T X^T) \quad (8)$$

$$\nabla V = S_2 \circ (UX) \quad (9)$$

其中,

$$\delta_3 = G^{(3)} - t$$

$$\delta_2 = G^{(2)} \circ (E^{(2)} - G^{(2)}) \circ (\omega^{(1)T} \delta_3)$$

$$\delta_1 = G^{(1)} \circ (E^{(1)} - G^{(1)}) \circ (\beta^T \delta_2)$$

$$G^{(i)} = [g_1^{(i)}, \dots, g_k^{(i)}, \dots, g_{L^{(i)}}^{(i)}]^T, i=1, 2, 3$$

$$S_1 = [\delta_1, \dots, \delta_1]_{L^{(1)} \times m}$$

$$S_2 = [\delta_1, \dots, \delta_1]_{L^{(1)} \times n}$$

符号 \circ 代表按元素相乘, $E^{(i)} = [1, 1, \dots, 1]^T$ ($i=1, 2$) 是长度为 $L^{(i)}$ 的全 1 向量。

综上, 3 层 2D-BP 网络的训练算法如算法 1 所示。

算法 1 3 层 2D-BP 网络的训练算法

输入: 训练集 (P 个矩阵输入), 动量系数 γ , 学习率 η , 隐层结点数 $L^{(1)}$

和 $L^{(2)}$, 迭代总次数 Q

输出: 训练好的 3 层 2D-BP 网络

Begin

1. 初始化 3 层 2D-BP 网络式(2)的权重

2. for $q \leftarrow 1$ to Q do

3. for $p \leftarrow 1$ to P do

4. 通过式(5)–式(9)计算式(4)

5. $\Theta = \Theta + \Delta\Theta_i$

6. end for

7. end for

8. return 3层2D-BP网络

End

3.2 4层网络

在3层网络上加1层全连接层,得到4层网络。前3层节点的输出为:

$$g_k^{(1)} = \sigma(u_k^{(1)} X v_k^{(1)} + b_k^{(1)})$$

$$g_k^{(2)} = \sigma\left(\sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)},k^{(1)}} g_k^{(1)} + b_k^{(2)}\right)$$

$$g_k^{(3)} = \sigma\left(\sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(3)},k^{(2)}} g_k^{(2)} + b_k^{(3)}\right)$$

输出层的输出为:

$$\begin{aligned} y_k^{(4)} &= g_k^{(4)} \\ &= \sum_{k^{(3)}=1}^{L^{(3)}} \omega_{k^{(3)},k^{(4)}} g_k^{(3)} + b_k^{(4)} \\ &= \sum_{k^{(3)}=1}^{L^{(3)}} \omega_{k^{(3)},k^{(4)}} \sigma\left(\sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(2)},k^{(3)}} g_k^{(2)} + b_k^{(3)}\right) + b_k^{(4)} \\ &= \sum_{k^{(3)}=1}^{L^{(3)}} \omega_{k^{(3)},k^{(4)}} \sigma\left(\sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(2)},k^{(3)}} \sigma\left(\sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)},k^{(1)}} g_k^{(1)} + b_k^{(2)}\right) + b_k^{(3)}\right) + b_k^{(4)} \\ &= \sum_{k^{(3)}=1}^{L^{(3)}} \omega_{k^{(3)},k^{(4)}} \sigma\left(\sum_{k^{(2)}=1}^{L^{(2)}} \omega_{k^{(2)},k^{(3)}} \sigma\left(\sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)},k^{(1)}} \sigma(u_k^{(1)} X v_k^{(1)} + b_k^{(1)}) + b_k^{(2)}\right) + b_k^{(3)}\right) + b_k^{(4)} \end{aligned} \quad (10)$$

网络的损失函数为:

$$E = \frac{1}{2} \sum_{k^{(4)}=1}^{L^{(4)}} (y_k^{(4)} - t_k^{(4)})^2 \quad (11)$$

其中, $t = (t_1, \dots, t_{k^{(4)}}, \dots, t_{L^{(4)}})^T$ 为样本 X 的标签。对于网络的训练,采用2.1节中的动量梯度下降法。 $\Theta = (U, V, \beta, \omega^{(1)}, \omega^{(2)}, b^{(1)}, \dots, b^{(4)})$, 于是有 $\nabla \Theta$ 如下:

$$\nabla b^{(i)} = \delta_i, i=1, \dots, 4 \quad (12)$$

$$\nabla \omega^{(i)} = \delta_{i+2} G^{(i+1)T}, i=1, 2 \quad (13)$$

$$\nabla \beta = \delta_2 G^{(1)T} \quad (14)$$

$$\nabla U = S_1 \circ (V^T X^T) \quad (15)$$

$$\nabla V = S_2 \circ (UX) \quad (16)$$

其中, $\delta_4 = G^{(4)} - t$, $\delta_i = G^{(i)} \circ (E^{(i)} - G^{(i)}) \circ (\omega^{(i-1)T} \delta_{i+1}) (i=2, 3)$, $\delta_1 = G^{(1)} \circ (E^{(1)} - G^{(1)}) \circ (\beta^T \delta_2)$, $G^{(i)} = [g_1^{(i)}, \dots, g_{k^{(i)}}^{(i)}, \dots, g_{L^{(i)}}^{(i)}]^T (i=1, \dots, 4)$

$$S_1 = [\delta_1, \dots, \delta_1]_{L^{(1)} \times m}$$

$$S_2 = [\delta_1, \dots, \delta_1]_{L^{(1)} \times n}$$

3.3 多层网络

类似地,对于 N 层网络 ($N > 4$):

$$g_k^{(1)} = \sigma(u_k^{(1)} X v_k^{(1)} + b_k^{(1)})$$

$$g_k^{(2)} = \sigma\left(\sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)},k^{(1)}} g_k^{(1)} + b_k^{(2)}\right)$$

...

$$g_k^{(i-1)} = \sigma\left(\sum_{k^{(i-2)}=1}^{L^{(i-2)}} \omega_{k^{(i-2)},k^{(i-1)}} g_k^{(i-2)} + b_k^{(i-1)}\right) (i=4, \dots, N)$$

...

$$y_k^{(N)} = g_k^{(N)} = \sum_{k^{(N-1)}=1}^{L^{(N-1)}} \omega_{k^{(N-1)},k^{(N)}} g_k^{(N-1)} + b_k^{(N)} \quad (17)$$

网络的损失函数为:

$$E = \frac{1}{2} \sum_{k^{(N)}=1}^{L^{(N)}} (y_k^{(N)} - t_k^{(N)})^2 \quad (18)$$

同样地,使用动量梯度下降法训练网络, $\Theta = (U, V, \beta, \omega^{(1)}, \dots, \omega^{(N-2)}, b^{(1)}, \dots, b^{(N)})$, $\nabla \Theta$ 为:

$$\nabla b^{(i)} = \delta_i, i=1, \dots, N \quad (19)$$

$$\nabla \omega^{(i)} = \delta_{i+2} G^{(i+1)T}, i=1, \dots, N-2 \quad (20)$$

$$\nabla \beta = \delta_2 G^{(1)T} \quad (21)$$

$$\nabla U = S_1 \circ (V^T X^T) \quad (22)$$

$$\nabla V = S_2 \circ (UX) \quad (23)$$

其中,

$$\delta_N = G^{(N)} - t$$

$$\delta_i = G^{(i)} \circ (E^{(i)} - G^{(i)}) \circ (\omega^{(i-1)T} \delta_{i+1}) (i=2, \dots, N-1)$$

$$\delta_1 = G^{(1)} \circ (E^{(1)} - G^{(1)}) \circ (\beta^T \delta_2)$$

$$G^{(i)} = [g_1^{(i)}, \dots, g_{k^{(i)}}^{(i)}, \dots, g_{L^{(i)}}^{(i)}]^T (i=1, \dots, 4)$$

$$S_1 = [\delta_1, \dots, \delta_1]_{L^{(1)} \times m}$$

$$S_2 = [\delta_1, \dots, \delta_1]_{L^{(1)} \times n}$$

N 层2D-BP网络的训练算法如算法2所示。

算法2 N 层2D-BP网络的训练算法 ($N > 4$)

输入:训练集 P 个矩阵输入, 动量系数 γ , 学习率 η , 隐层结点数 $L^{(1)}, \dots, L^{(N-1)}$, 迭代总次数 Q

输出:训练好的 N 层2D-BP网络

Begin

1. 初始化 N 层2D-BP网络式(17)的权重

2. for $q \leftarrow 1$ to Q do

3. for $p \leftarrow 1$ to P do

4. 通过式(19)~式(23)计算式(4)

5. $\Theta = \Theta + \Delta\Theta_i$

6. end for

7. end for

8. return N 层2D-BP网络

End

3.4 三通道2D-BP网络

对于彩色图片 $X = (X^{(1)}, X^{(2)}, X^{(3)})$, $X^{(1)}, X^{(2)}, X^{(3)}$ 分别代表彩色图片 X 的R,G,B三通道。如果直接把彩色图像转化成灰度图像来分类,会丢失图像的颜色特征。我们修改了2D-BP网络的输入层,使得网络能够处理彩色图片的RGB信息。

记 $g_k^{(1)} = \sigma(\sum_{i=1}^3 (u_k^{(i)} X^{(i)} v_k^{(i)} + b_k^{(1)}))$ 为网络隐层的输出, $u_k^{(i)} = (u_{k^{(i)}1}^{(i)}, \dots, u_{k^{(i)}r}^{(i)}, \dots, u_{k^{(i)}m}^{(i)})^T$, $v_k^{(i)} = (v_{1k^{(i)}}^{(i)}, \dots, v_{ck^{(i)}}^{(i)}, \dots, v_{nk^{(i)}}^{(i)})^T$, $i=1, 2, 3$, $b_k^{(1)}$ 为隐层第 $k^{(1)}$ 个结点的偏置。

$$\begin{aligned} y_k^{(2)} &= g_k^{(2)} \\ &= \sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)},k^{(1)}} g_k^{(1)} + b_k^{(2)} \\ &= \sum_{k^{(1)}=1}^{L^{(1)}} \beta_{k^{(2)},k^{(1)}} \sigma\left(\sum_{i=1}^3 (u_k^{(i)} X^{(i)} v_k^{(i)} + b_k^{(1)}) + b_k^{(1)}\right) + b_k^{(2)} \end{aligned} \quad (24)$$

为网络的输出。网络的损失函数为:

$$E = \frac{1}{2} \sum_{k^{(2)}=1}^{L^{(2)}} (y_k^{(2)} - t_{k^{(2)}})^2 \quad (25)$$

使用动量梯度下降法更新权重 $\Theta = (U^{(1)}, U^{(2)}, U^{(3)}, V^{(1)}, V^{(2)}, V^{(3)}, \beta, b^{(1)}, b^{(2)})$, 其中, $U^{(i)} = (u_1^{(i)}, \dots, u_{k^{(i)}}^{(i)}, \dots, u_{L^{(i)}}^{(i)})$, $V^{(i)} = (v_1^{(i)}, \dots, v_{k^{(i)}}^{(i)}, \dots, v_{L^{(i)}}^{(i)}) (i=1, 2, 3)$ 。

$$\frac{\partial E}{\partial b_{k^{(2)}}} = (g_{k^{(2)}}^{(2)} - t_{k^{(2)}})$$

$$\frac{\partial E}{\partial \beta_{k^{(2)}k^{(1)}}} = (g_{k^{(2)}}^{(2)} - t_{k^{(2)}})g_{k^{(1)}}^{(1)}$$

$$\frac{\partial E}{\partial b_{k^{(1)}}} = \sum_{k^{(2)=1}^{l_2^{(2)}} (g_{k^{(2)}}^{(2)} - t_{k^{(2)}}) \sum_{k^{(1)=1}^{l_1^{(1)}} \beta_{k^{(2)}k^{(1)}} g_{k^{(1)}}^{(1)}$$

$$\frac{\partial E}{\partial u_{k^{(1)}r}} = \sum_{k^{(2)=1}^{l_2^{(2)}} (g_{k^{(2)}}^{(2)} - t_{k^{(2)}}) \sum_{k^{(1)=1}^{l_1^{(1)}} \beta_{k^{(2)}k^{(1)}} g_{k^{(1)}}^{(1)}$$

$$\sum_{c=1}^n x_{rc}^{(i)} v_{ck^{(1)}}^{(i)} (i=1,2,3)$$

$$\frac{\partial E}{\partial v_{ck^{(1)}}^{(i)}} = \sum_{k^{(2)=1}^{l_2^{(2)}} (g_{k^{(2)}}^{(2)} - t_{k^{(2)}}) \sum_{k^{(1)=1}^{l_1^{(1)}} \beta_{k^{(2)}k^{(1)}} g_{k^{(1)}}^{(1)}$$

$$\sum_{r=1}^m u_{k^{(1)}r}^{(i)} x_{rc}^{(i)} (i=1,2,3)$$

有 $\nabla \Theta$ 如下:

$$\nabla b^{(i)} = \delta_i, i=1,2 \quad (26)$$

$$\nabla \beta = \delta_2 G^{(1)T} \quad (27)$$

$$\nabla U^{(i)} = S_1 \circ (V^{(i)T} X^{(i)T}), i=1,2,3 \quad (28)$$

$$\nabla V^{(i)} = S_2 \circ (U^{(i)} X^{(i)}), i=1,2,3 \quad (29)$$

其中,

$$\delta_2 = G^{(2)} - t$$

$$\delta_1 = G^{(1)} \circ (E^{(1)} - G^{(1)}) \circ (\beta^T \delta_2)$$

$$G^{(1)} = [g_1^{(1)}, \dots, g_{k^{(1)}}^{(1)}, \dots, g_L^{(1)}]^T$$

$$S_1 = [\delta_1, \dots, \delta_1]_{L^{(1)} \times m}$$

$$S_2 = [\delta_1, \dots, \delta_1]_{L^{(1)} \times n}$$

三通道 2D-BP 网络的训练算法如算法 3 所示。

算法 3 三通道 2D-BP

输入:训练集(P个三通道矩阵输入),动量系数 γ ,学习率 η ,隐层结

数 $L^{(1)}$,迭代总次数 Q

输出:训练好的三通道 2D-BP 网络

Begin

1. 初始化三通道 2D-BP 网络式(24)的权重

2. for $q \leftarrow 1$ to Q do

3. for $p \leftarrow 1$ to P do

4. 通过式(26)~式(29)计算式(4)

5. $\Theta = \Theta + \Delta \Theta_i$

6. end for

7. end for

8. return 三通道 2D-BP 网络

End

4 实验与分析

本节将所提的网络与 2D-BP 网络进行比较,用实验来比较不同层数不同节点的网络之间的优劣。为了避免随机因素,每个实验重复 5 次,然后取平均值。实验均在拥有 32G 内存 Windows10 系统的计算机上使用 Matlab2015b 软件完成。使用带动量的 BP 算法,其中动量系数 $\gamma=0.8$,学习率 $\eta=0.01$ 。为了实验的可比性,动量系数、学习率与文献[9]保持一致。

4.1 手写数字识别

用 USPS 手写数据集^[13](如图 1 所示,包括 8000 个训练样本与 3000 个测试样本,每个样本为 16×16 大小的灰度图像)来测试多层 2D-BP 网络的性能。

由于通过对隐层节点数量的调整能够改变网络在 USPS 数据集上的表现,为了确定网络的最优结构,对 2~4 层网络的结构进行了训练。

对于 2 层网络,由于第二层的节点数量与标签维度一致,因此只能调节第一层节点的数量。第一层节点的数量从 100 开始逐步增加,测试网络结构对数据集的性能。



图 1 USPS 手写数据集

如表 1 所列,表 1 中的每行代表相应节点的网络迭代不同次数在 USPS 数据集上的测试精度,每列表示不同节点的网络在相应迭代次数下在 USPS 数据集上的测试精度,最后一列表示不同节点的网络在训练过程中的最好测试结果。如第一层节点数为 600 的网络在迭代至第 659 次时,测试精度达到最高 0.9674,则记为 0.9674,659。

表 1 2 层网络的训练结果

迭代数 \ 结点数	迭代数					
	10	100	200	500	1000	最好表现
100	0.902067	0.941700	0.948667	0.954633	0.952367	0.954800,519
200	0.902800	0.947700	0.955900	0.962700	0.962500	0.964000,744
300	0.903400	0.950600	0.957467	0.964267	0.965133	0.965667,960
400	0.904800	0.954400	0.961133	0.965533	0.964067	0.966133,635
500	0.905867	0.954800	0.960467	0.965733	0.965200	0.965933,569
600	0.899200	0.955533	0.961400	0.966867	0.965933	0.967400,659
700	0.897400	0.958867	0.964267	0.966533	0.966333	0.967267,434
1000	0.900000	0.958778	0.962111	0.966444	0.965333	0.967111,553
1300	0.855533	0.960333	0.964800	0.967400	0.965400	0.968533,425
2100	0.915000	0.960222	0.962111	0.963444	0.960222	0.964444,389

对于 3 层网络,由于第三层的节点数量与标签维度一致,因此需要调节第一、二层节点来确定较好的网络结构。

如表 2 所列,xxx-xxx 分别表示第一层和第二层节点的数量。比如,100-200 代表第一层节点数为 100,第二层节点

数为200,第三层节点数为10的3层网络。其中,最后一列表示不同节点的网络在训练过程中的最好测试结果,比如第一层节点数为600的网络在迭代到第659次时,测试精度达到最高0.9674,则记为0.9674,659。通过比较,第一层节点与第二层节点数分别为300,1700的3层网络在USPS数据集上的训练效果较好,精度达到了0.97475。其中,有的网络训练过程中出现了震荡,比如700-900-10的3层网络(见图2),数据就没有写入表中。

对于4层网络,则需要调节前3层节点来确定较好的网络结构。

如表3所列,xxx-xxx-xxx分别表示第一、二、三层节点的数量。其中,有的网络训练过程中出现了震荡,数据就没有写入表中。第一、二、三层节点数分别为300,300,700的4层网络在USPS数据集上的训练效果较好,精度达到了0.970133。

4层网络的测试精度(0.970133)小于3层网络的测试精度(0.97475)。图3是2层网络与3层网络之间的对比。

表2 3层网络的训练结果

迭代数 结点数	10	100	200	500	1000	最好表现
100-100	0.918933	0.959000	0.962267	0.957467	0.949000	0.963200,246
100-200	0.922267	0.960667	0.965000	0.964667	0.960533	0.965867,363
100-300	0.923000	0.962467	0.965733	0.964333	0.960800	0.966333,279
100-400	0.920733	0.963467	0.965000	0.965867	0.962067	0.966333,428
100-500	0.918400	0.962400	0.966933	0.967667	0.964267	0.968200,414
100-600	0.920267	0.964733	0.969133	0.970067	0.965933	0.971067,304
100-700	0.924733	0.964267	0.968533	0.968400	0.965533	0.969200,262
100-900	0.925267	0.964733	0.968533	0.968333	0.966133	0.969333,234
100-1100	0.924667	0.964267	0.969800	0.970400	0.966600	0.971133,375
100-1300	0.924800	0.963667	0.969467	0.970933	0.968067	0.971400,576
300-100	0.917800	0.958800	0.960533	0.960000	0.956267	0.961533,181
300-500	0.905067	0.966333	0.968667	0.968667	0.963267	0.969000,465
300-900	0.898917	0.967667	0.969750	0.967667	0.965167	0.970250,165
300-1300	0.878000	0.969600	0.971467	0.970400	0.968200	0.972600,307
300-1700	0.868333	0.969917	0.974417	0.973417	0.970500	0.974750,236
300-2100	0.874800	0.967000	0.970600	0.969667	0.967533	0.970733,210
300-2500	0.833000	0.967917	0.970750	0.969083	0.969000	0.971917,255
300-3300	0.734000	0.945800	0.957867	0.962933	0.959267	0.963800,427
500-500	0.893417	0.958500	0.960667	0.957167	0.951917	0.961583,309
500-900	0.850500	0.931583	0.958583	0.956500	0.938500	0.959500,213
500-1300						
700-500	0.913917	0.956417	0.956750	0.950333	0.949000	0.958000,142
700-900						
700-1700						
900-900	0.864889	0.883333	0.879333	0.669222	0.923556	0.934333,29

表3 4层网络的训练结果

迭代数 结点数	10	100	200	500	1000	最好表现
100-100-100	0.919200	0.960267	0.963200	0.961933	0.959867	0.963533,155
100-200-200	0.921133	0.961600	0.963067	0.959600	0.958467	0.964600,162
100-300-300	0.931917	0.962083	0.963167	0.960167	0.958333	0.964333,141
100-500-500	0.911000	0.962400	0.962800	0.960333	0.956267	0.963867,162
100-900-900						
200-900-900						
300-100-100	0.923467	0.962733	0.964933	0.965267	0.965067	0.965600,429
300-300-300	0.927250	0.966000	0.967917	0.965750	0.963250	0.968083,168
300-300-400	0.918222	0.965667	0.966333	0.964333	0.962889	0.967000,158
300-300-500	0.908600	0.965733	0.967867	0.964867	0.962600	0.968133,144
300-300-700	0.905400	0.968600	0.969733	0.965933	0.963667	0.970133,161
300-300-900	0.908600	0.966267	0.969133	0.968067	0.967000	0.969333,212
300-300-1100	0.911000	0.966667	0.968933	0.968067	0.965733	0.969067,205
300-300-1500	0.899067	0.964933	0.967867	0.965800	0.964800	0.968133,177
300-300-1900	0.887933	0.963667	0.964200	0.962400	0.961667	0.964733,148
300-400-300	0.920417	0.964167	0.965000	0.964583	0.963500	0.966167,147
300-400-400	0.914083	0.963500	0.967333	0.966500	0.964750	0.967833,217
300-400-500	0.920667	0.965250	0.967333	0.966333	0.965083	0.968250,300
300-500-300	0.923500	0.961583	0.960667	0.959583	0.959417	0.962750,123
300-500-500	0.912333	0.960267	0.963467	0.963933	0.962467	0.964667,285
300-900-900						
300-1300-1300						
300-1700-1700						

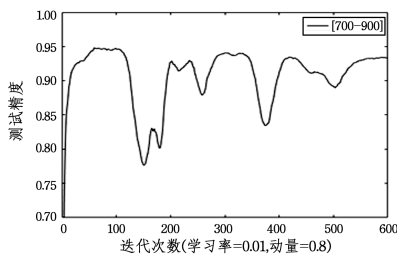


图 2 网络出现震荡

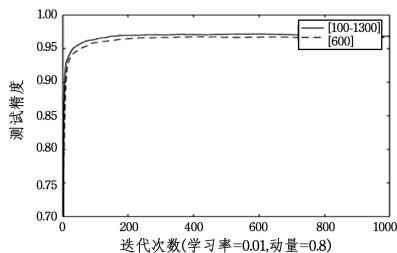


图 3 2层网络与3层网络之间的对比

通过实验发现,相对于2层的2D-BP网络,添加全连接层能够提高网络的性能,但是网络过深反而会降低网络的分类能力。3层的2D-BP网络在USPS数据集上获得了最好的效果。

4.2 CIFAR-10 数据集

我们使用 CIFAR-10 数据集^[4]来测试三通道 2D-BP 网络的性能。CIFAR-10 数据集包括 60000 张 32×32 像素的彩色图像,分成 10 类,每类 6000 张图片。有 50000 个训练图像和 10000 个测试图像。

由于 2D-BP 网络中不能直接输入彩色图片,我们先将 CIFAR-10 数据集中的图片转化为灰度图片,再用于 2D-BP 网络的训练,与三通道 2D-BP 网络进行比较。如图 4 所示,同为 256 个结点的三通道 2D-BP 网络与 2D-BP 网络,相对于 2D-BP 网络的测试精度 0.4392,三通道 2D-BP 网络的测试精度 0.5210 有明显的提升。

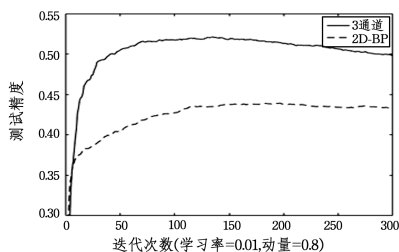


图 4 3 通道网络与 2D-BP 网络的对比

结束语 本文提出基于多层 2D-BP 网络的图像分类算法。相对于传统的神经网络,2D-BP 网络能够对图像给出一个行列之内有结构信息的先验。本文提出的算法对 2D-BP 网络进行了结构的调整,能够提升 2D-BP 网络对特征的提取能力,从而得到泛化能力更强的特征提取器。另外,对彩色图片进行分类时,本文提出的三通道 2D-BP 网络图像分类算法能够更好地对图片进行特征提取,达到更好的实验效果。多层 2D-BP 算法能够进一步使用正则化手段来提升性能。

参考文献

[1] 高隽,谢昭. 图像理解理论与方法 [M]. 北京: 科学出版社, 2009.

- [2] RICHARDS L E. Principal Component Analysis [J]. Journal of Marketing Research, 1988, 38(22): 41-64.
- [3] STONE J. Principal Component Analysis and Factor Analysis [M]. MIT Press, 2004: 129-135.
- [4] LU J, PLATANIOTIS K N, VENETSANOPOULOS A N. Face recognition using LDA-based algorithms [J]. IEEE Transactions on Neural Networks, 2003, 14(1): 195-200.
- [5] ZHENG W S, LAI J H, YUEN P C. GA-fisher: A new LDA-based face recognition algorithm with selection of principal components [J]. IEEE Transactions on Systems Man & Cybernetics Part B, 2005, 35(5): 1065-1078.
- [6] VAPNIK V N. The nature of statistical learning theory [M]. New York: Springer, 2000.
- [7] SHAKHAROVICH G, DARRELL T, INDYK P. Nearest-neighbor methods in learning and vision [J]. Pattern Analysis and Applications, 2008, 11(2): 221-222.
- [8] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer feed-forward networks are universal approximators [J]. Neural Networks, 1989, 2(5): 359-366.
- [9] DAI K, ZHAO J, CAO F. A novel algorithm of extended neural networks for image recognition [J]. Engineering Applications of Artificial Intelligence, 2015, 42(1): 57-66.
- [10] YAN X, YAN X, ZHANG L, et al. Feature extraction based on fuzzy 2DLDA [J]. Neurocomputing, 2010, 73 (10-12): 1556-1561.
- [11] SANGUANSAT P, ASDORNWISED W, JITAPUNKUL S, et al. Two-Dimensional Linear Discriminant Analysis of Principle Component Vectors for Face Recognition [J]. IEICE-Transactions on Information and Systems, 2006, E89-D(7): 2164-2170.
- [12] LU J, ZHAO J, CAO F. Extended feed forward neural networks with random weights for face recognition [J]. Neurocomputing, 2014, 136(1): 96-102.
- [13] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
- [14] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [C] // International Conference on Neural Information Processing Systems. Curran Associates Inc. 2012: 1097-1105.
- [15] ZEILER M D, FERGUS R. Visualizing and Understanding Convolutional Networks [C] // European Conference on Computer Vision. 2013: 818-833.
- [16] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [J/OL]. [2015-04-10]. <https://arxiv.org/pdf/1409.1556v6.pdf>.
- [17] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions [C] // IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2015: 1-9.
- [18] HE K, ZHANG X, REN S, et al. Deep Residual Learning for Image Recognition [C] // Computer Vision and Pattern Recognition. IEEE, 2016: 770-778.
- [19] HULL J J. A database for handwritten text recognition research [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(5): 550-554.
- [20] KRIZHEVSKY A, ALEX. Learning Multiple Layers of Features from Tiny Images [M]. Toronto: University of Toronto, 2009.