# 基于 BP 神经网络的人体行为识别

**摘　要**：*针对人体行为识别问题，提出一种基于径向基函数（*BP*）神经网络的人体行为分类算法。首先，利用奇异值分解（*SVD*）算法提取视频每一帧的奇异值，将每一帧的奇异值按照行拼接起来即为一个视频的样本，样本按照行排成样本矩阵；然后，利用主成分分析（*PCA*）对得到的矩阵进行去相关并且降低维数，降低维数的矩阵再进行线性鉴别分析（*LDA*），使样本变得线性可分；最后，利用* BP *神经网络对样本进行分类；实验结果表明，与采用最近邻分类和 K 近邻分类（k*NN*）相比，所提算法具有更高的识别率。*

**关键词**：*人体行为识别*；SVD；PCA；LDA；BP *神经网络*

**中图分类号**：TP391.41

**文献标识码**：A

# Human action recognition based on back propagation neural network

**Abstract:** For human action recognition, an algorithm for classifying human action based on back propagation (BP) neural network was proposed. Firstly, singular value decomposition (SVD) was used to extract the singular value of each frame of the video. Then each row of a matrix was composed of the singular value of each video. Every single row of the matrix is a sample of human action. Secondly, the principle component analysis (PCA) algorithm was proposed to remove correlation and reduce dimension. Then the linear discriminant analysis (LDA) algorithm was applied to matrix processed by PCA to make samples linearly separable. Finally, the back propagation neural network was used as a classifier. The experimental results show that the proposed method, compared with nearest neighbor classifier and *K*-nearest neighbor (*k*NN) classifier, has a higher recognition rate.

**Key words:** human action recognition, SVD, PCA, LDA, back propagation neural network

## 1　引言

*近年来，计算机视觉的发展大大推动了人体行为分析的进步，取得了诸多成果。人体行为分析可分为人体动作识别和分析，其中人体动作识别是指对人体的运动模式进行分析和识别*[1]*。目前，针对人体行为识别的方法主要有模板匹配法、状态空间法以及基于模型的方法，其中，模板匹配法拥有快速的识别效率，但识别率较低且模板具有特定性；基于模型的方法很难寻找到好的模型。因此人体行为识别仍有许多问题亟需解决，研究可靠且稳定的识别方法具有较大意义。*Tu *等*[2] *利用结构模型，通过建立点头、摆头、举手、迈步等行为模型，准确地描述了上述几种行为，取得了较高的识别率；针对视频中人体行为尺度不同的问题，李妍婷等*[3]*提出了单目视频中多视角*

行为识别方法，获得了稳定的行为特征。

本文提出一种基于 BP 神经网络的人体行为识别方法，首先对视频进行对齐处理，从每个样本视频中提取每一帧的奇异值，然后将奇异值排成一行即为一个视频样本，样本矩阵由多个矩阵排成行构成。对得到的样本矩阵用 PCA 对其进行去除相关和降低维度，再由 LDA 使各个样本之间变得线性可分。对经过特征提取的样本，可以用来进行 BP 神经网络的训练，最后选择出部分样本进行测试。实验结果表明本文算法性能良好。

## 2　基于 BP 神经网络的人体行为识别算法

### 2.1　模式识别一般框架

一个经典的模式识别框架包含样本采集、预处理、特征选择与提取、特征变换和分类器，如图 1 所示。

本文采用简单的帧数对齐作为视频预处理算法，SVD 和 PCA 作为特征提取算法，LDA 进行特征变换，最后使用 BP 神经网络作为分类器。样本采用了 KTH activity 数据库[4]，该数据库包含 6 个动作，分别为走（walking）、打拳（boxing）、拍手（handclapping）、慢跑（jongging）、挥手（handwavin）、快跑（running），每个动作有 25 个样本。本文选取每个动作的 20 个样本作为训练集，另外 5 个样本作为测试集，图 2 为 walking 动作的关键帧。

### 2.2　视频预处理

视频样本的时间长度不一样会导致计算出来的矩阵每一样维度均不一样，有 2 种处理方法：1) 提取同样数量的视频的相关帧来对齐[5]；2)采用可以计算不同维度之间相关性的张量算法，如TCCA[6]。本文则是简单地计算了所有视频中最小的帧数，采用该帧数作为所有视频的帧数，这样就可以使得到的样本矩阵维度一致便于进行计算，经过计算得到最小的帧数 *minFrame*=232。

### 2.3　特征提取

特征提取是为了提取出感兴趣的信息，忽略噪声和不重要的信息。SVD 的奇异值可以反映图片像素矩阵行与列的相关性，这种相关性可以看作一种特征；同时 SVD 的特征可以用很小的维度表示一张图片。例如，一张 $M\times N$ 的图片的 SVD 特征有 $\min\{M,N\}$ 维。SVD 可以将一个矩阵分解为 3 个矩阵相乘，如(1)式所示。

$$\boldsymbol{X}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} \tag{1}$$

其中，$\Sigma=\mathrm{diag}\{\delta_1,\delta_2,\cdots,\delta_3\}$ 是每一帧视频的奇异值，$X$ 代表视频的每一帧，每一帧的分辨率为 160×6120。20 个训练视频组成的训练矩阵维度为(20×6)×6(232×120)=120×28 040，具体如(2)式所示。

$$\boldsymbol{Train}=\begin{matrix} a_{1,1} & \cdots & a_{1,28040} \\ \cdots & = & \cdots \\ a_{20,1} & \cdots & a_{20,28040} \end{matrix} \tag{2}$$

这样的高维度的矩阵直接进行分类，计算量显然是巨大的，而且提取出来的样本之间可能存在冗余，而 PCA 算法刚好可以同时解决这 2 个问题。PCA 的主要作用是去除矩阵行或列之间的相

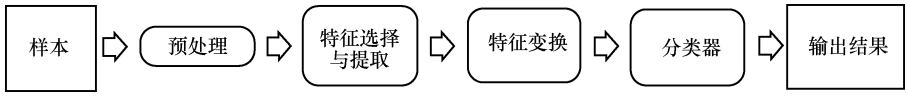


图 1　模式识别一般框架

图 2　walking 动作的关键帧

关性（本文去除的是维度之间的相关性），去除相关性后可以选择奇异值较大的维度作为新的特征，这样可以压缩维度，提高后续分类的运算速度。PCA 的目标就是为了找到一个投影矩阵 $\boldsymbol{P}$，使 $\boldsymbol{Train}$ 经过 $\boldsymbol{P}$ 投影后的 $\boldsymbol{Train\_prj}$ 矩阵各个样本之间相互正交，即 $\boldsymbol{Train\_prj}$ 的各行之间相互正交。那么投影后的协方差矩阵变为对角阵。推导过程如下。

$$\boldsymbol{Train\_prj} = \boldsymbol{P} \times \boldsymbol{Train} \tag{3}$$

$\boldsymbol{Train}$ 的协方差矩阵为 $\boldsymbol{C}_{\boldsymbol{Train}}$，$\boldsymbol{Train\_prj}$ 的协方差矩阵为 $\boldsymbol{C}_{\boldsymbol{Train\_prj}}$。

$$\boldsymbol{C}_{\boldsymbol{Train}} = \frac{1}{N}\boldsymbol{Train} \times \boldsymbol{Train}^{\mathrm{T}} \tag{4}$$

$$\begin{aligned}\boldsymbol{C}_{\boldsymbol{Train\_prj}} &= \frac{1}{N}\boldsymbol{Train\_prj} \times \boldsymbol{Train\_prj}^{\mathrm{T}} \\ &= \frac{1}{N}\boldsymbol{P} \times \boldsymbol{Train} \times \boldsymbol{Train}^{\mathrm{T}} \times \boldsymbol{P}^{\mathrm{T}} \\ &= \boldsymbol{P}\boldsymbol{C}_{\boldsymbol{Train}}\boldsymbol{P}^{\mathrm{T}}\end{aligned} \tag{5}$$

由于 $\boldsymbol{C}_{\boldsymbol{Train}}$ 是实对称阵，所以存在一个矩阵 $\boldsymbol{Q}$ 使

$$\boldsymbol{C}_{\boldsymbol{Train}} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}} \tag{6}$$

令 $\boldsymbol{P}=\boldsymbol{Q}^{\mathrm{T}}$，将式(6)代入式(5)中得 $\boldsymbol{C}_{\boldsymbol{Train\_prj}} = \boldsymbol{\Lambda}$。消除相关性后的训练矩阵 $\boldsymbol{Train\_prj}$，由于维度过于大，根据经验将维度压缩为 $nSamples-nClass$，就是样本数量减去类别数量。仍然用 $\boldsymbol{Train\_prj}$ 表示降维后的矩阵。

$$\boldsymbol{Train\_prj} = \boldsymbol{Train\_prj}(:,1:nSamples - nClass) \tag{7}$$

### 2.4 特征变换

特征变换的目的是使各个类别之间更容易区别，其一般的思想是从一个空间投影到另外一个空间，使原来不容易区分的类别变得容易区分。本文采用 LDA 作为特征变换算法，LDA 由 Fisher 提出，它的主要思想是将样本投影到一个低维的特征空间，使投影后的数据集的类内的样本与样本之间的距离最小（也就是说使样本越聚拢），使类与类之间的距离越分散。一般来说，经过 LDA 的特征变换后，样本都更加容易找到分界面。

一个数据集为 $X$，它的每个样本的维度为 $D$，类别为 $K$，对于多类问题需要假设 $D>K$。然后，引入 $D'$特征 $\boldsymbol{y}_k = \boldsymbol{w}_k^{\mathrm{T}}\boldsymbol{x}$,其中，$k=1,\cdots,D'$。写成矩阵形式（不包括偏置），于是有

$$\boldsymbol{y} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{x} \tag{8}$$

用类内散度矩阵来衡量类内方差。

$$\boldsymbol{S}_W = \sum_{k=1}^{K}\boldsymbol{S}_k \tag{9}$$

其中，

$$\boldsymbol{S} = \sum_{n\in c_k}(\boldsymbol{x}-\boldsymbol{m})(\boldsymbol{x}-\boldsymbol{m})^{\mathrm{T}} \tag{10}$$

$$\boldsymbol{m}_k = \frac{1}{N_k}\sum_{n\in c_k}\boldsymbol{x}_n \tag{11}$$

$N_k$ 是 $c_k$ 类的样本数。总体协方差矩阵 $\boldsymbol{S}_T$ 为

$$\boldsymbol{S}_T = \sum_{n=1}^{N}(\boldsymbol{x}_n-\boldsymbol{m})(\boldsymbol{x}_n-\boldsymbol{m})^{\mathrm{T}} \tag{12}$$

其中，

$$\boldsymbol{m} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n = \sum_{k=1}^{K}N_k\boldsymbol{m}_k \tag{13}$$

用类间散度矩阵 $\boldsymbol{S}_B$ 度量类与类之间的散布的情况，由于 $\boldsymbol{m}$ 由 $\boldsymbol{m}_k$ 线性表出，故 $\boldsymbol{S}_B$ 的秩至多为 $k-1$。因此可以在 $k-1$ 维子空间内进行分类。

$$\boldsymbol{S}_B = \sum_{k=1}^{K}N_k(\boldsymbol{m}_k-\boldsymbol{m})(\boldsymbol{m}_k-\boldsymbol{m})^{\mathrm{T}} \tag{14}$$

现在定义投影后的 $D'$维空间类似的矩阵。

$$\boldsymbol{s}_W = \sum_{k=1}^{K}\sum_{n\in c_k}(\boldsymbol{y}_n-\boldsymbol{u}_k)(\boldsymbol{y}_n-\boldsymbol{u}_k)^{\mathrm{T}} \tag{15}$$

$$\boldsymbol{s}_B = \sum_{k=1}^{K}N_k(\boldsymbol{u}_k-\boldsymbol{u})(\boldsymbol{u}_k-\boldsymbol{u})^{\mathrm{T}} \tag{16}$$

其中

$$\boldsymbol{u}_k = \frac{1}{N_k}\sum_{n\in c_k}\boldsymbol{y}_n, \boldsymbol{u} = \sum_{k=1}^{K}N_k\boldsymbol{u}_k \tag{17}$$

最终得到一个由 $\boldsymbol{s}_B$ 和 $\boldsymbol{s}_W$ 组成的标量，LDA 的目的是使 $\boldsymbol{s}_B$ 最小、$\boldsymbol{s}_W$ 最大。求得该标量最大值即可满足 LDA 的目标。令 $J(W)$为该标量，具体形式如式(18)所示。

$$\boldsymbol{J}(\boldsymbol{W}) = \boldsymbol{Tr}\left(\boldsymbol{s}_W^{-1}\boldsymbol{s}_B\right) \tag{18}$$

代入投影矩阵 $W$ 可得

$$\boldsymbol{J}(\boldsymbol{w}) = \boldsymbol{Tr}((\boldsymbol{W}\boldsymbol{S}_W\boldsymbol{W}^{\mathrm{T}})^{-1}(\boldsymbol{W}\boldsymbol{S}_B\boldsymbol{W}^{\mathrm{T}})) \tag{19}$$

对于式(19)，可以使用广义 Rayleigh 商来求解，对 $\boldsymbol{s}_W^{-1}\boldsymbol{s}_B$ 做广义特征值分解，取 $k-1$ 个最大特征值所对应的特征向量组成投影矩阵。

LDA 可以分为以下几个步骤。

1) 计算每个类的样本均值 $m_k$ 和总体均值 $m$。

2) 计算类内散度矩阵 $\boldsymbol{s}_W$ 和类间散度矩阵 $\boldsymbol{s}_B$。

3) 计算广义特征值分解，提取 $k-1$ 个最大特征值所对应的特征向量组成投影矩阵。

4) 计算投影后的样本点。

LDA 算法的伪代码如图 3 所示。

LDA 算法变换训练矩阵并且进行压缩，输入为训练矩阵为 train，测试矩阵 test，输出为特征向量组成的矩阵 train_lda,test_lda

1) 计算总体平均值 $m$

2) 初始化 $S_B$ 和 $S_W$

3) for $i$=1 to $nClass$ do

4) 计算每类均值 $m_k$

5) 计算 $\boldsymbol{s}_B = \sum_{k=1}^{K} \boldsymbol{N}_k (\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^{\mathrm{T}}$

6) end for

7) 计算 $\boldsymbol{s}_W^{-1}\boldsymbol{s}_B$ 的特征值和特征向量，并且按照特征值从大到小的顺序对特征向量进行排列

8) 投影压缩特征*$V$(:,1:$nDims$)

9) 投影压缩特征*$V$(:,1:$nDims$)

10) returntrain_lda test_lda

图 3　LDA 的伪代码

## 2.5　BP 神经网络分类器

### 2.5.1　BP 神经网络

BP 神经网络是一种全连接的前向网络，网络参数采用误差反向传播算法求解。典型的 BP 神经网络结构如图 4 所示，其中包括输入层、隐含层和输出层。输入层由输入数据构成，它是网络与外部世界连接的窗口；隐含层的每个节点是由输入层节点线性加权求和，再进行非线性变换得到的，其中权值可利用反向传播算法求得。隐层的数量可以很多，隐层数越多，网络相对越复杂，训练越耗时；最后一层是输出层，最终输出需要的结果。

输入层的输出等于整个网络的输入信号。

$$v_M^m(n) = x(n) \tag{20}$$

隐含层第 $i$ 个神经元输入等于 $v_M{}^m(n)=x(n)$的加权和。

$$u_I^i(n) = \sum_{m=1}^{M} w_{mi}(n) v_M^m(n) \tag{21}$$

假设 $f(\cdot)$为Sigmoid 函数，则隐含层第 $i$ 个神经元的输出为

$$v_I^i(n) = f(u_I^i(n)) \tag{22}$$

输出层第 $j$ 个神经元输入等于 $v_I^i(n)$的加权和。

$$u_J^j(n) = \sum_{i=1}^{I} w_{ij}(n) v_I^i(n) \tag{23}$$

输出层第 $j$ 个神经元的输出为

$$v_J^j(n) = g(u_J^j(n)) \tag{24}$$

为了使神经网络在误差大的时候收敛的速度更快，误差小的时候收敛的慢一些，采用了交叉熵误差函数（cross-entropy error function），式(25)是第 $j$ 个神经元的误差。

$$e_j(n) = -v_J^j(n) \ln d_j(n) - (1 - v_J^j(n)) \ln(1 - d_j(n)) \tag{25}$$

网络总误差为

$$e(n) = -\sum_{n=1}^{N} \{v_J^j(n) \ln d_j(n) + (1 - v_J^j(n)) \ln(1 - d_j(n))\} \tag{26}$$

### 2.5.2　基于 BP 神经网络的人体行为识别

将训练矩阵经过特征提取和特征变换后的矩阵用来训练 BP 神经网络调整权值。首先调整隐含层与输出层之间的权值 $w_{ij}$，根据最速下降法，应计算误差对 $w_{ij}$ 的梯度$\partial e(n)/\partial w_{ij}$，再沿着反方向进行调整。

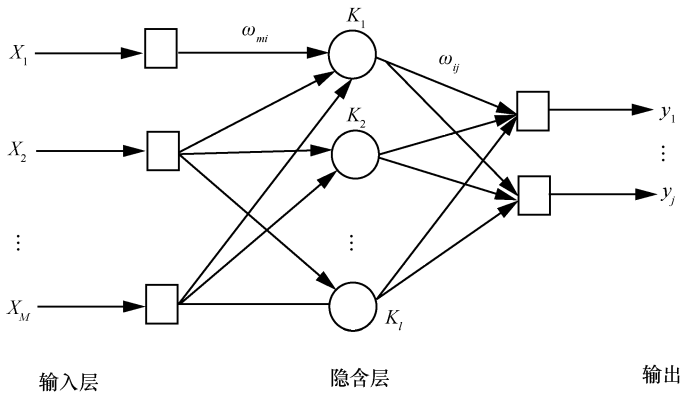


图 4　BP 神经网络拓扑结构

$$w_{ij}(n+1)=\triangle w_{ij}(n)+w_{ij}(n) \tag{27}$$

其中

$$\triangle w_{ij}(n)=-\eta\frac{\partial e(n)}{\partial w_{ij}} \tag{28}$$

根据微分的链式法则

$$\frac{\partial e(n)}{\partial w_{ij}}=\frac{\partial e(n)}{\partial e_j^J(n)}\frac{\partial e_j(n)}{\partial v_J^j(n)}\frac{\partial v^j(n)}{\partial u_J^j(n)}\frac{\partial u^j(n)}{\partial w_{ij}(n)} \tag{29}$$

得出梯度值为

$$\frac{\partial e(n)}{\partial w_{ij}}=-e_j(n)g'(u_J^j(n))v_I^i(n) \tag{30}$$

权值修正量为

$$\triangle w_{ij}(n)=\eta e_j(n)g'(u_J^j(n))v_I^i(n) \tag{31}$$

局部梯度定义为

$$\delta_J^j=-\frac{\partial e(n)}{\partial u_J^j(n)}=-\frac{\partial e(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial v_J^j(n)}\frac{\partial v_J^j(n)}{\partial u_J^j(n)} \tag{32}$$

权值修正量为

$$\triangle w_{ij}(n)=\eta\delta_J^j v^i(n) \tag{33}$$

在输出层，传递函数为线性函数，因此

$$g'(u_J^j(n))=1 \tag{34}$$

权值修正量为

$$\triangle w_{ij}(n)=\eta e_j(n)v_I^i(n) \tag{35}$$

同理得到

$$\triangle w_{mj}(n)=\eta\delta_I^i v^m(n) \tag{36}$$

至此，三层BP网络的一轮权值调整完成，训练好的神经网络可以用来对测试集进行分类。

## 3 实验结果分析

实验数据来自KTH数据库，BP神经网络的输入层为5个。对LDA来说，C类问题在C−1维空间内就能分类。隐含层20个节点，输出层6个节点对应6个不同类型的行为实验，比较不同的分类方法（最近邻分类法、K近邻分类法和BP神经网络分类器）的识别率。在Matlab 2015b上进行仿真，结果如表1所示。

K近邻分类的效果在样本之间有重叠的时候比最近邻分类器的效果好，经过实验发现，可以将准确率由63.33%提升至76.67%。而BP神经网络分类器的效果最好，它可以比较好地对各个类别之间的界限进行拟合，拟合的误差和隐含层的层数以及节点数有很大的关系。为了使训练时误差大的时候收敛快，类似于人脑的学习，将误差函数用方差误差函数调整为交叉熵误差函数，发现收敛速度明显加快。

**表1 不同分类方法的比较结果**

| 分类器 | 识别率 |
|---|---|
| 最近邻分类器 | 63.33% |
| K近邻分类器 | 76.67% |
| BP神经网络分类器 | 85.37% |

## 4 结束语

本文提出了一种基于BP神经网络的人体行为识别算法，采用SVD进行特征提取，提取出了视频中每一帧的奇异值信息作为特征。为了提高计算速度、减少数据量，采用了PCA去除冗余和压缩数据。采用LDA进行特征变换，将数据投影到低维空间，这样更容易找到类与类之间的界限。最后使用可以很好拟合分类面的BP神经网络作为分类器，在实际的测试中对KTH数据库中的6类行为的分类性能良好。由于视频是一个3阶张量，后面的工作为：1)对特征提取可以采用3D-Gabor滤波器[7]提取不同频率上尺度、不同方向的特征，一个二维的Gabor核函数由一个高斯函数和余弦函数相乘得到，可以使用核函数的相关参数调整需要的尺度和方向；2)采用STM分类器[8]或3D-CNN[9]对张量数据进行分类，这样可以避免将张量转换成向量而忽略数据本身之间的联系。这是由于传统图像、视频处理方法都是将图片向量化，这样将像素值割裂开的方法会丢失图像内容上像素值之间的关联性，张量的方法是将整张图片或视频看作整体进行处理，可以避免这种“割裂”操作。

## 参考文献：

[1] ADAM N R, ATLURI V, HUANG W K. Modeling and analysis of workflows using Petri nets[J]. Journal of Intelligent Information Systems, 1998, 10(2): 131-158.

[2] TU T Y, SHI Y X. Image capture and efficient storing in monitor system[J]. Application Research of Computers, 2005, 22(8): 241-242.

[3] *李妍婷, 罗予频, 唐光荣. 单目视频中的多视角行为识别方法*[J]. *计算机应用*, 2006, 26(7): 1592-1594.
LI Y T, LUO Y P, TANG G R. Activity recognition method of multiple view angles from monocular videos[J]. Journal of Computer Applications, 2006, 26(7): 1592-1594.

[4] SCHULDT C, LAPTEV I, CAPUTO B. Recognizing human actions: a local SVM approach[C]//The 17th International Conference on Pattern Recognition. 2004: 32-36.

[5] *许伟坚, 冯超, 李进锦, 等. 一种新的视频质量评价中帧对齐算法*[J]. *厦门大学学报*(*自然版*), 2012, 51(2):185-188.
XU W J, FENG C. A new frame alignment algorithm in video quality evaluation[J]. Journal of Xiamen University (Natural Science), 2012 , 51(2) :185-188.

[6] KIM T K, WONG S F, CIPOLLA R. Tensor canonical correlation analysis for action classification[C]//IEEE Conference on Computer Vision and Pattern Recognition. 2007: 1-8.

[7] WANG Y, CHUA C S. Face recognition from 2D and 3D images using 3D Gabor filters[J]. Image and Vision Computing, 2005, 23(11): 1018-1028.

[8] CAI D, HE X, WEN J R, et al. Support tensor machines for text categorization[J]. 2006.

[9] JI S, XU W, YANG M, et al. 3D convolutional neural networks for human action recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(1): 221-231.