

# 基于 PSO 的 BP 神经网络学习算法

王爱平, 江 丽

(安徽大学计算机科学与技术学院, 合肥 230601)

**摘 要:** 针对标准反向传播(BP)算法收敛速度慢和易陷入局部极值等缺陷, 提出一种基于粒子群优化的 BP 神经网络学习算法。采用标准 BP 梯度下降法调整权值, 利用粒子群优化算法进行网络权值及阈值的修正。将该算法与标准 BP 算法及传统基于粒子群优化 BP 网络算法进行仿真比较。实验结果表明, 该算法能够克服标准 BP 算法的缺点, 性能优于其他 2 个 BP 网络优化模型。

**关键词:** 神经网络; 反向传播算法; 粒子群优化; 梯度下降法; 函数拟合

## BP Neural Network Learning Algorithm Based on Particle Swarm Optimization

WANG Ai-ping, JIANG Li

(School of Computer Science and Technology, Anhui University, Hefei 230601, China)

**【Abstract】** For the standard Back Propagation(BP) algorithm usually has the limitations of slow convergence and local extreme values, a new BP neural network learning algorithm based on Particle Swarm Optimization(PSO) is proposed. The main idea of the model is to modify weight and threshold using PSO based on the weight adjustments of gradient descent method in BP algorithm. It evaluates the model by using simulation test of five typical complex functions and compares it with other two models including standard BP network and traditional PSO based BP network. Experimental results show that it can overcome the limitations of slow convergence and local extreme values for BP network and perform better than other two kinds of optimized BP network models.

**【Key words】** neural network; Back Propagation(BP) algorithm; Particle Swarm Optimization(PSO); gradient descent method; function fitting

DOI: 10.3969/j.issn.1000-3428.2012.21.052

### 1 概述

反向传播(Back Propagation, BP)网络是一种按误差逆向传播算法训练的多层前馈网络。由于它具有大规模并行处理、分布式信息存储、良好的自组织自学习能力及原理简单、容易实现等特点, 因此在众多领域得到广泛应用。误差逆向传播学习算法(也称 BP 算法)是神经网络中最常用、最有效的一种方法<sup>[1]</sup>。

这种算法通过误差反向传播不断调整网络的权值和阈值, 使网络的误差平方和最小。尽管 BP 神经网络具有很多显著的特点, 但是也存在固有的缺陷: 容易陷入局部极小, 收敛速度较慢, 网络泛化能力较弱。因此, 许多学者提出了改进的学习算法<sup>[2-6]</sup>。

粒子群优化(Particle Swarm Optimization, PSO)算法是一种群体自适应搜索优化算法<sup>[6-9]</sup>, 其主要思想是通过种群中个体之间的协作和信息共享寻找最优解。系统初始化为一组随机解, 通过迭代搜寻最优值。该算法实现简单,

需要调整的参数少, 收敛速度快。目前, PSO 算法已被广泛应用于神经网络训练、函数优化、故障诊断、模糊系统控制及数据挖掘等领域。

本文同时考虑 PSO 算法与 BP 网络算法, 提出一种基于 PSO 算法的 BP 网络学习新算法。该算法在标准 BP 的梯度下降法调整权值及阈值的基础上, 再引入粒子群中粒子位置更新方法进行权值及阈值的修正。由于 PSO 算法是一种群体寻优算法, 它的多点搜索方式使其具有全局搜索能力, 因此这种权值及阈值修正可以弥补标准 BP 算法收敛速度慢及容易陷入局部极小值等缺陷。对 5 个典型的复杂函数应用新算法进行了函数拟合仿真实验, 并与标准的 BP 算法和传统的基于 PSO 的 BP 算法进行了比较。

### 2 基于 PSO 算法的 BP 网络新模型

#### 2.1 PSO 算法简介

在 PSO 算法中, 每个“粒子”代表优化问题的一个潜在解, 首先初始化一群随机粒子, 然后每个粒子通过跟

**基金项目:** 国家自然科学基金资助项目“熵理论在非高斯随机系统的故障诊断和容错控制中的应用”(61074071)、“非高斯随机分布控制系统的集成故障诊断与容错控制研究”(61104022)

**作者简介:** 王爱平(1956—), 女, 教授, 主研方向: 人工智能, 数据库技术, 容错控制; 江 丽, 硕士研究生

**收稿日期:** 2012-01-04 **修回日期:** 2012-03-11 **E-mail:** jiangli\_anda@163.com

踪个体极值(粒子本身所找到的最优解)和全局极值(整个种群当前找到的最优解)更新自己直到找到最优解<sup>[6-7,9]</sup>。

假设一个群体由  $m$  个粒子组成, 每个粒子的信息用  $d$  维向量表示, 则第  $i$  个粒子的位置向量和速度向量分别为  $X_i=(x_{i1}, x_{i2}, \dots, x_{id})$  和  $V_i=(v_{i1}, v_{i2}, \dots, v_{id})$ , 第  $i$  个粒子所经过的历史最佳位置向量记为  $P_i=(p_{i1}, p_{i2}, \dots, p_{id})$ , 群体所经历过的历史最佳的位置向量记为  $P_g=(p_{g1}, p_{g2}, \dots, p_{gd})$ 。在每一次迭代过程中, 粒子通过跟踪个体极值  $P_i$  和全局极值  $P_g$  更新自己的速度和位置。具体的更新公式为:

$$v_{im}(t+1) = \eta(t) \cdot v_{im}(t) + c_1 r_1 (p_{im} - x_{im}(t)) + c_2 r_2 (p_{gm} - x_{im}(t)) \quad (1)$$

$$x_{im}(t+1) = x_{im}(t) + v_{im}(t+1) \quad (2)$$

其中,  $m=1, 2, \dots, d$ ;  $\eta(t)$  为惯性权值, 一般采用线性惯性权;  $t$  为迭代次数;  $c_1$ 、 $c_2$  为学习因子;  $r_1$ 、 $r_2$  为 2 个随机数, 一般在 0~1 之间取值。

## 2.2 BP 网络权值优化方法

BP 神经网络模型拓扑结构包括输入层(input layer)、隐层(hidden layer)、输出层(output layer)。因此, 网络参数由隐结点与输出结点的权值( $w_{kj}(t)$ )、输入结点与隐结点的权值( $u_{jh}(t)$ )及输出结点与隐结点的阈值( $\theta_k, \hat{\theta}_j$ )组成。它们的更新公式如下所示<sup>[1,6]</sup>:

$$w_{kj}(t+1) = w_{kj}(t) + \alpha \delta_k H_j \quad (3)$$

$$u_{jh}(t+1) = u_{jh}(t) + \alpha \sigma_j I_h \quad (4)$$

$$\theta_k(t+1) = \theta_k(t) + \beta \delta_k \quad (5)$$

$$\hat{\theta}_j(t+1) = \hat{\theta}_j(t) + \beta \sigma_j \quad (6)$$

其中,  $H_j$  为隐结点  $j$  的输出;  $I_h$  为从输入结点  $h$  输入的信号;  $\delta_k$  和  $\sigma_j$  为输出层结点  $k$  和隐结点  $j$  的误差;  $\alpha$ 、 $\beta$  为学习参数, 通常在 0.1~0.9 内取值。

上述标准的 BP 网络按照梯度下降法进行寻优, 这种方法常常收敛速度较慢且容易陷入局部最优。针对该问题, 各种不同的改进方法相继被提出, 将 PSO 优化方法与 BP 网络算法相结合是其中较为典型的改进之一<sup>[4-6,8-9]</sup>。这种做法通常是将 BP 神经网络的连接权值和阈值看作粒子群中粒子的位置向量  $X$  的元素; 然后利用粒子群优化方法代替 BP 网络的梯度下降法实现网络连接权值和阈值的优化。这种算法进行网络权值调整大多直接采用 PSO 迭代算法进行优化, 没有利用 BP 算法的梯度下降法。

本文将 PSO 方法与 BP 算法相结合, 提出一种基于 PSO 的 BP 网络优化新模型。不同于传统的方法, 本文方法在网络本身梯度下降法调整权值的基础上进一步采用 PSO 算法进行权值修正。

## 2.3 基于 PSO 的 BP 神经网络算法

### 2.3.1 基于 PSO 算法的 BP 网络权值更新

将 BP 网络的权值及阈值视为 PSO 算法中各个粒子的位置分量, 则在网络训练过程中, 相继 2 次权值的改变可视为粒子位置的改变。设 BP 网络的输入层、隐层及输出层的结点数量分别为  $n_i$ 、 $n_h$ 、 $n_o$ , 则第  $i$  个粒子的位置所确定的网络表示为:

$$X_i = (x_{i1}, x_{i2}, \dots, x_{id}) = (w_{11}, w_{12}, \dots, w_{n_i n_h}, u_{11}, u_{12}, \dots, u_{n_h n_i},$$

$$\theta_1, \theta_2, \dots, \theta_{n_o}, \hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_{n_h}) \quad (7)$$

其中,  $w_{kj}(k=1, 2, \dots, n_o; j=1, 2, \dots, n_h)$  为隐结点  $j$  与输出层结点  $k$  的连接权值;  $u_{jh}(j=1, 2, \dots, n_h, h=1, 2, \dots, n_i)$  为输入结点  $h$  与隐结点  $j$  的连接权值;  $d=n_o n_h + n_h n_i + n_o + n_h$ 。本文将粒子位置更新与 BP 网络梯度下降方法相结合, 从而实现一种新的 BP 网络权值更新方法。

记  $l_1=n_o n_h$ ,  $l_2=l_1+n_h n_i$ ,  $l_3=l_2+n_o$ ,  $l_4=l_3+n_h$ , 则第  $i$  个粒子的位置更新如下:

$$v_{im}(t+1) = \eta \cdot v_{im}(t) + c_1 r_1 (p_{im} - x_{im}(t)) + c_2 r_2 (p_{gm} - x_{im}(t)) \quad (8)$$

$$x_{im}(t+1) = \begin{cases} x_{im}(t) + v_{im}(t+1) + \alpha \delta_k H_{j_1} & \text{if } m \in (0, l_1] \\ x_{im}(t) + v_{im}(t+1) + \alpha \sigma_{j_2} I_h & \text{if } m \in (l_1, l_2] \\ x_{im}(t) + v_{im}(t+1) + \beta \delta_k & \text{if } m \in (l_2, l_3] \\ x_{im}(t) + v_{im}(t+1) + \beta \sigma_{j_3} & \text{if } m \in (l_3, l_4] \end{cases} \quad (9)$$

其中,  $m=1, 2, \dots, d$ ;  $k=\lceil m/n_h \rceil$ ;  $j_1=m-(k-1) \cdot n_h$ ;  $j_2=\lceil (m-l_1)/n_i \rceil$ ;  $h=m-l_1-(j_2-1) \cdot n_i$ ;  $k_1=m-l_2$ ;  $j_3=m-l_3$ ;  $H_{j_1}$  为隐结点  $j_1$  的输出;  $I_h$  从输入结点  $h$  输入的信号;  $\delta_k$  和  $\sigma_{j_2}$  分别为输出层结点  $k$  和隐结点  $j_2$  的误差信号;  $\alpha$ 、 $\beta$  为学习率;  $r_1$ 、 $r_2$  为  $[0, 1]$  内的随机数;  $c_1$ 、 $c_2$  为常数;  $t$  为当前迭代次数;  $\lceil \cdot \rceil$  为取上整函数。

上述 BP 网络权值的更新过程中既考虑了 BP 本身的梯度下降更新又利用了粒子群自身的更新过程, 是一种严格意义上的 PSO 与 BP 算法相结合的网络训练算法。

### 2.3.2 算法流程

本文 PSO-BP 算法的具体实现如下:

#### 步骤 1 初始化

(1) 初始化 BP 网络结构, 包括设定网络的输入层、隐含层、输出层的神经元个数和学习率  $\alpha$ 、 $\beta$  及训练样本的输入和输出。

(2) 初始化粒子群, 包括粒子的规模  $N$  及每个粒子的位置向量及速度向量、每个粒子的个体极值和全局最优值、迭代误差精度  $\varepsilon$ 、常系数  $c_1$  和  $c_2$ 、最大惯性权值  $\eta_{\max}$ 、最小惯性权值  $\eta_{\min}$ 、最大速度  $v_{\max}$  及最大迭代次数等。

#### 步骤 2 迭代更新

(1) 利用式(8)更新每个粒子的速度, 并判断更新后的速度是否大于最大速度  $v_{\max}$ , 若大于最大速度  $v_{\max}$ , 则更新后的速度就取值为最大速度  $v_{\max}$ , 否则, 保持不变。

(2) 利用式(9)更新每个粒子的位置。

(3) 利用式(10)计算各个粒子的适应度值:

$$f_i = \frac{1}{n_i} \sum_{q=1}^{n_i} (O_{iq} - T_{iq})^2 \quad (10)$$

其中,  $n_i$  为训练样本的个数;  $O_{iq}$ 、 $T_{iq}$  分别为训练样本  $q$  在第  $i$  粒子的位置所确定的网络权值及阈值下的网络实际输出和期望输出。

(4) 计算粒子群的全局最小适应值  $f_g = \min\{f_1, f_2, \dots, f_N\}$ ; 若当前迭代次数达到最大迭代次数或  $f_g < \varepsilon$  (网络的训练误差达到精度要求), 则迭代停止, 转到步骤(3); 否则, 计

算各个粒子的个体极值  $P_i$  和全局极值  $P_g$  位置, 转到迭代更新的步骤(1)继续更新粒子的速度和位置。

**步骤 3** 输出全局极值  $P_g$  的位置所确定的网络权值和阈值, 算法结束。

3 仿真实验及分析

3.1 测试函数

实验选用以下 5 个典型的复杂函数:

- (1)  $f_1 = 100 \times (x_1^2 - x_2)^2 + (1 - x_1)^2, x_i \in [-10, 10], i = 1, 2$
- (2)  $f_2 = x_1^2 + x_2^3 - x_1 x_2 x_3 + x_3 - \sin x_2^2 - \cos(x_1 x_3^2), x_i \in [-2\pi, 2\pi], i = 1, 2, 3$
- (3)  $f_3 = \frac{1}{4\,000}(x_1^2 + x_2^2) - \cos x_1 \cos(\frac{x_2}{\sqrt{2}}) + 1, x_i \in [-10, 10], i = 1, 2$
- (4)  $f_4 = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} + 0.5, x_i \in [-5, 5], i = 1, 2$
- (5)  $f_5 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), x_i \in [-5, 5], i = 1, 2$

3.2 函数模拟及误差分析

利用本文的 BP 网络的训练方法, 实现对上述 5 个复

杂函数的模拟。

BP 网络的结构设定如下: 输入层神经元个数就是函数的自变量个数, 输出层的神经元个数是函数因变量个数, 隐含层神经元个数采用经验试凑法。PSO 算法的初始化种群个体数目设为  $N=40$ ; 最大迭代次数设为 1 000; 最小训练停止误差  $\varepsilon=10^{-6}$ ; PSO 算法的学习因子  $c_1=c_2=1.5$ ; 最大惯性权值  $w_{\max}=0.9$ ; 最小惯性权值  $w_{\min}=0.3$ ; 最大速度  $v_{\max}=1$ ; 初始样本生成: 对于每个待拟合的函数, 在自变量取值范围随机生成  $n_t=300$  个训练样本,  $n_s=100$  个测试样本。表 1 显示了不同方法的拟合误差结果。其中,  $E_t$ 、 $E_s$  分别为训练样本和测试样本的均方误差, 即:

$$E_t = \frac{1}{n_t} \sum_{q_1=1}^{n_t} (O_{q_1} - T_{q_1})^2$$
$$E_s = \frac{1}{n_s} \sum_{q_2=1}^{n_s} (O_{q_2} - T_{q_2})^2$$

其中,  $O_{q_1}$  和  $T_{q_1}$  分别为训练样本  $q_1$  的网络实际输出和期望输出;  $O_{q_2}$  和  $T_{q_2}$  分别为检验样本  $q_2$  的网络实际输出和期望输出。

表 1 3 种优化算法用于 5 个函数的误差比较

算法	误差	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$
BP	$E_t$	0.028 80	0.022 30	0.272 90	0.489 10	0.143 30
	$E_s$	0.311 80	0.035 50	0.212 50	0.485 00	0.155 70
TPSO-BP	$E_t$	0.015 00	0.020 80	0.269 40	0.466 80	0.096 50
	$E_s$	0.025 16	0.032 70	0.214 60	0.475 70	0.116 70
PSO-BP	$E_t$	1.82e-005	3.71e-005	0.146 80	0.325 70	0.055 70
	$E_s$	9.88e-004	5.86e-005	0.167 50	0.338 90	0.086 80

图 1~图 5 显示了 3 种算法在各个函数收敛结果。

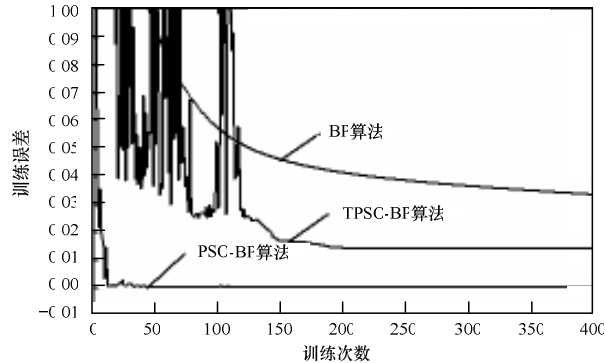


图 1 3 种算法在函数  $f_1$  上的收敛结果

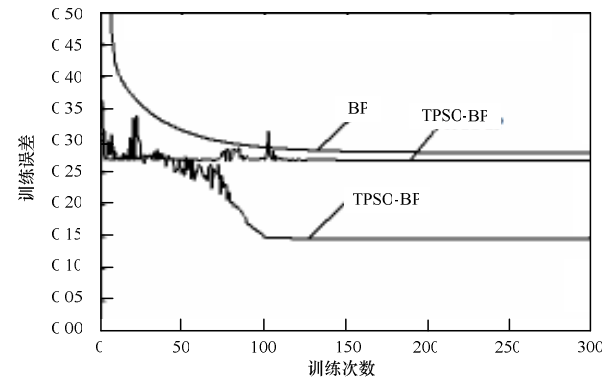


图 3 3 种算法在函数  $f_3$  上的收敛结果

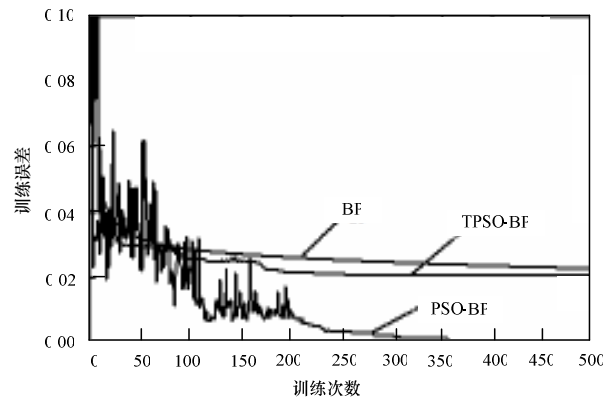


图 2 3 种算法在函数  $f_2$  上的收敛结果

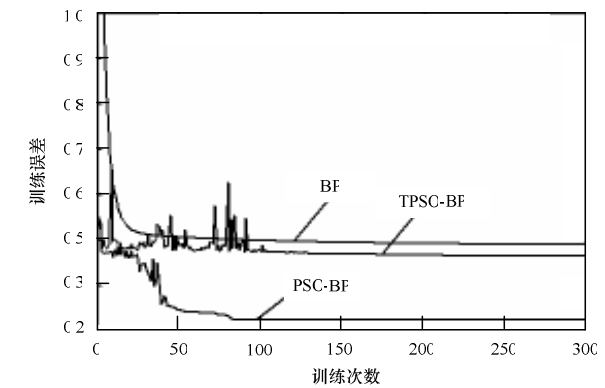


图 4 3 种算法在函数  $f_4$  上的收敛结果

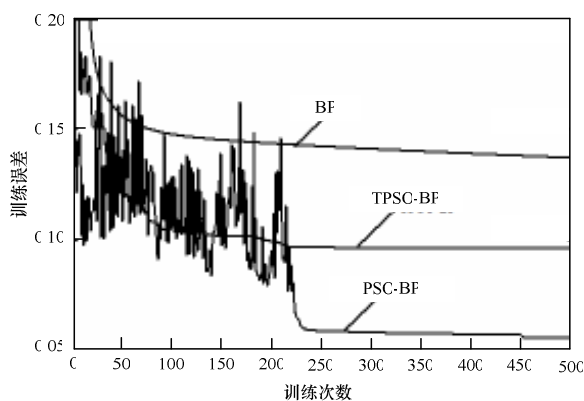


图5 3种算法在函数 $f_5$ 上的收敛结果

在实验中,将本文算法(PSO-BP)与传统的基于粒子群优化的BP算法(TPSO-BP)<sup>[4]</sup>及标准的BP做了比较。从实验结果可以看出:(1)相比于其他2种算法,本文算法能获得较好的函数逼近效果,表明其能够在一定程度上克服传统BP网络易限于局部极值的缺陷。(2)无论是训练误差还是检验误差,本文算法能够获得较小的误差值,尤其是函数 $f_1$ 和 $f_2$ 。因此,本文算法具有较强的函数拟合能力,能够较大地提高TPSO-BP及标准BP网络的收敛精度,克服传统方法收敛精度低的缺陷。(3)从图1可以看出,对于复杂的函数,本文算法的迭代次数均少于其他2种BP网络优化算法的迭代次数,尤其远少于标准BP的迭代次数,表明本文算法收敛速度高于标准BP算法及传统TPSO-BP算法的,进一步增强了网络的泛化能力。

#### 4 结束语

本文针对传统BP算法收敛速度慢、容易陷入局部极小值以及泛化能力弱等缺点,提出了一种基于粒子群优化

算法优化BP网络的新算法。该算法在标准BP的梯度下降法的基础上,进一步引入PSO算法进行权值修正。相比于传统的方法,本文算法能够较好地克服易陷入局部极值、收敛速度慢等缺点,提高BP算法的学习能力和泛化能力。

#### 参考文献

- [1] 陈果. 神经网络模型的预测精度影响因素分析及其优化[J]. 模式识别与人工智能, 2005, 18(5): 528-534.
- [2] Li Zuoyong, Yi Yongzhi. Quantitative Relation Between Learning Ability and Generalization Ability of BP Neural Network[J]. Acta Electronica Sinica, 2003, 31(9): 1341-1344.
- [3] 高雪鹏, 丛爽. BP网络改进算法的性能对比研究[J]. 控制与决策, 2001, 16(2): 167-171.
- [4] 宋乃华, 邢清华. 一种新的基于粒群优化的BP网络学习算法[J]. 计算机工程, 2006, 32(14): 181-183.
- [5] 高海兵, 高亮, 周驰, 等. 基于粒子群优化的神经网络训练算法研究[J]. 电子学报, 2004, 32(9): 1572-1574.
- [6] 李祚泳, 汪嘉杨, 郭淳. PSO算法优化BP网络的新方法及仿真实验[J]. 电子学报, 2008, 36(18): 2224-2228.
- [7] Kennedy J, Eberhart R. Particle Swarm Optimization[C]//Proc. of IEEE Int'l Conf. on Neural Networks. Piscataway, USA: IEEE Press, 1995: 1942-1948.
- [8] Che Z H. PSO-based Back-propagation Artificial Neural Network for Product and Mold Cost Estimation of Plastic Injection Molding[J]. Computers & Industrial Engineering, 2010, 58(4): 625-637.
- [9] 潘昊, 侯清兰. 基于粒子群算法的BP网络学习研究[J]. 计算机工程与应用, 2006, 42(16): 41-43.

编辑 张正兴

(上接第192页)

#### 参考文献

- [1] Yilmaz A, Javed O, Shah M. Object Tracking: A Survey[J]. ACM Computing Surveys, 2006, 38(4): 1-7.
- [2] 卞志国, 金立左, 费树岷. 特征融合与视觉目标跟踪[J]. 计算机应用研究, 2010, 27(4): 1537-1539.
- [3] 张辉, 赵保军, 唐林波, 等. 基于自适应多特征整合的红外目标跟踪[J]. 光学学报, 2010, 30(5): 1291-1296.
- [4] 陈善静, 杨华, 曾凯, 等. 基于多特征融合的粒子滤波算法研究[J]. 计算机工程, 2011, 37(7): 178-180.
- [5] 管志强, 陈钱, 钱惟贤, 等. 一种基于算法融合的红外目标跟踪方法[J]. 光学学报, 2008, 28(5): 860-865.
- [6] Collins R, Liu Yanxi, Leordeanu M. On-line Selection of Discriminative Tracking Features[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2005, 27(10): 1631-1643.
- [7] Yin Zhaozheng, Porikli F, Collins R. Likelihood Map Fusion for Visual Object Tracking[C]//Proc. of the IEEE Workshop on Applications of Computer Vision. [S. l.]: IEEE Press, 2008.
- [8] Avidan S. Ensemble Tracking[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2007, 29(2): 261-262.
- [9] 程建, 周越, 蔡念, 等. 基于粒子滤波的红外目标跟踪[J]. 红外与毫米波学报, 2006, 25(2): 113-117.
- [10] Arulampalam M S, Maskell S, Gordon N, et al. A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking[J]. IEEE Trans. on Signal Processing, 2002, 50(2): 174-188.
- [11] 冯祖仁, 吕娜, 李良福. 基于最大后验概率的图像匹配相似性指标研究[J]. 自动化学报, 2007, 33(1): 1-8.
- [12] Ojala T, Pietikainen M, Maenpaa T. Multiresolution Gray Scale and Rotation Invariant Texture Analysis with Local Binary Patterns[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, 24(7): 971-987.
- [13] Tan Xiaoyang, Triggs B. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions[C]//Proc. of AMFG'07. Rio de Janeiro, Brazil: [s. n.], 2007.

编辑 顾姣健

