第 38 卷 第 11 期
Vol.38 No.11

计 算 机 工 程
Computer Engineering

2012 年 6 月
June 2012

·人工智能及识别技术· 文章编号：1000—3428(2012)11—0163—04 文献标识码：A 中图分类号：TP18

计算机博弈中估值算法与博弈训练的研究

吕艳辉，宫瑞敏

(沈阳理工大学信息科学与工程学院，沈阳 110159)

摘 要：计算机博弈程序难以处理局面估值问题。为此，结合时间差分算法和反向传播神经网络，设计一种局面估值算法 BP-TD(λ)，实现估值函数参数的自动调整。为提高博弈训练的性能，针对开局和中局，提出分阶段设置参数值的策略。以五子棋为应用背景，实现博弈系统 RenjuTD。实验结果表明，该算法可使程序的博弈水平得到较大提高。

关键词：计算机博弈；差分学习；反向传播神经网络；估值算法；增强学习；博弈训练

Study on Valuation Algorithm and Game Training in Computer Game

LV Yan-hui, GONG Rui-min

(College of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, China)

【Abstract】Situation valuation is the most difficult issue in all kinds of computer game programs. A valuation method named BP-TD(λ) is presented combining temporal difference algorithm and back propagation neural network, which can solve the problem of adjusting the parameter values of valuation function. On this basis, in order to enhance the performance of game training, the strategy of setting different parameter values is proposed for opening and middle game phases. The game system RenjuTD is implemented using Renju as application background. Experimental results show the game level of program is significantly improved.

【Key words】computer game; difference learning; back propagation neural network; valuation algorithm; reinforcement learning; game training

DOI: 10.3969/j.issn.1000-3428.2012.11.050

1 概述

计算机博弈作为人工智能研究领域的重要分支，包括棋盘表示、走法生成、搜索算法、估值函数等关键技术，其中，局面估值是一个难点。局面估值是一个很难量化研究的问题，通常需要领域专家人工调参[1-4]，存在静态估值函数依赖人类棋类知识和评估不够准确的问题。因此，一些自动调参方法被引入到计算机博弈中，然而，这方面的研究成果还很少。

本文以五子棋为背景，对计算机博弈中的一些关键技术进行研究。首先，通过使用反向传播(Back Propagation, BP)神经网络[5]作为局面估值函数的主体并引入时间差分(Temporal Difference, TD)算法[6-8]，实现了对估值函数参数的自动调整；其次，通过对开局和中局提出分阶段设置参数值的策略，以提高博弈训练的性能。

2 估值算法

2.1 BP-TD(λ)学习算法

BP-TD(λ)学习算法利用 BP 神经网络作为 TD(λ)(其中，λ为衰减因子)算法中的估值函数，通过 TD(λ)算法的自学习自动调整 BP 神经网络的权值。下面以五子棋为例，介绍该算法的实现过程。

设 $S_1, S_2, \cdots, S_t, \cdots, S_m$，$Z$ 代表五子棋中对弈的落子序列，在每个时刻 t 得到的局面观测状态为 S_t，落子序列的最终结果为 Z，分别为胜、平、负 3 种情况。对每一个状态 S_t，Agent 都能够产生与之相对应的预测序列 $P_1, P_2, \cdots, P_t, \cdots, P_m$，其中，$P_t$ 是基于状态 S_t 对 Z 的估计。这样，使用 BP 神经网络作为预测器，每一个 P_t 值都可以表示为关于 S_t 的函数，即 $P_t=P_t(\boldsymbol{w}, S_t)$，其中，$\boldsymbol{w}$ 是权值向量。

BP-TD(λ)算法的学习过程就是利用梯度法则修正 $\boldsymbol{w}$ 值以实现最终的正确预测，即 TD(λ)算法中的预测结果值能够通过修改 BP 神经网络中的 $\boldsymbol{w}$ 值得到，这种不断修正 $\boldsymbol{w}$ 值的过程即为学习的过程。在每个 t 时刻，权值 $\boldsymbol{w}$ 都会有一个变化量 Δw_t，Δw_t 只与前后 2 个相邻的预测值和过去的$\nabla_w P_t$值(即 P_t 对 $\boldsymbol{w}$ 的每个分量的偏导数)的和有关，这样 BP 神经网络中权值 $\boldsymbol{w}$ 的变化量 Δw_t 可以被不断地求出来。此外，考虑到距离 t 时间最近的时刻的预测值会有较高的可信度，引入衰减因子 λ，得到式(1)：

$$\Delta w_t = \alpha(P_{t+1} - P_t)\sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_t \tag{1}$$

式(1)即为 TD 预测值对当前局面的 BP 神经网络评估值进行反向传播调参的权值调整公式，其中，α 是 BP 神经网络的学习速率。

2.2 BP 神经网络的设计

下面以五子棋为应用模型，给出五子棋中 BP 神经网络的设计过程。

2.2.1 五子棋局面特征的提取

通过对五子棋的棋型特征进行提取，可以将五子棋棋盘上的所有直线按 0°、45°、90°、135° 方向分为 4 种情况，如图 1 所示。直线上交叉点的个数大于等于 5 的直线能够成为

基金项目：国家自然科学基金资助项目(60873010)；新世纪优秀人才支持计划基金资助项目(NCET-05-0288)

作者简介：吕艳辉(1971－)，女，副教授、博士，主研方向：人工智能，知识工程；宫瑞敏，硕士研究生

收稿日期：2011-08-01 **E-mail：**yanhuilv@126.com

有效直线向量；若交叉点的个数小于 5，则该向量为无效直线向量，黑白双方在这些无效直线向量上，无论怎么走棋都不会赢棋。

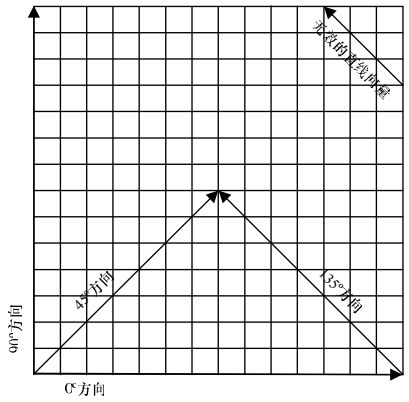

图 1 五子棋棋盘中直线向量的划分

假设五子棋使用大小为 15×15 的棋盘，整个棋盘上的有效直线向量共有 72 个。当五子棋对弈时，在每个有效直线向量上，根据五子棋特征恰好都能分割成 2 个不同的有效棋型。

2.2.2 BP 神经网络的设计过程

(1)网络结构整体设计

本文使用三层的 BP 神经网络。在使用 BP 神经网络作为五子棋的评估函数时，如何选择相邻 2 层之间的传输函数非常重要，因为它对网络的拟合性影响很大。由于五子棋的局面比较错综复杂，因此本文选择 2 层的 Sigmoid 函数作为 BP 神经网络相邻层间的转换函数。Sigmoid 函数能够任意地逼近到任何连续函数，同时，Sigmoid 函数在计算反向传播误差时更容易。

(2)输入设计

BP 神经网络的输入要用提取的时刻 t 的局面特征表示，所提取的局面特征应该具有对输出影响大、特征之间相关性很小或者互不相关的特点。

下面以死二、活二、死三和死四等非威胁棋型为例，说明对棋型进行特征编码的方法。例如，用 4 个输入单元就能够描述黑方的死三棋型的个数。当黑方死三棋型的个数为 0 时，用向量(1, 0, 0, 0)表示 4 个输入单元的取值；当黑方死三棋型的个数为 1 时，用向量(0, 1, 0, 0)表示；当黑方死三棋型的个数为 2 时，用向量(0, 0, 1, 0)表示；当黑方死三棋型的个数为 3 或 3 以上时，用向量(0, 0, 0, 1)表示。所以，一共需要使用 16 个输入单元来表示黑方的非威胁模型。同理，需要使用 16 个输入单元来表示白方的非威胁模型。此外，对威胁棋型个数的表示共需要 8 个输入单元，按进攻方分类的表示需要 4 个输入单元。这样，确定 BP 神经网络的输入层共有 44 个单元。

(3)隐藏层节点数的选取

通过使用一个隐藏层来增加神经数目的方法可以提高 BP 神经网络的训练精度，这种实现方式比增加大量的隐藏层简单得多。隐藏层的节点数如果选取太少，学习容量就会受到限制，以至于不能够存储训练样本包蕴含的所有规律；相反，若隐藏层节点数过多，会导致网络训练时间增加，同时样本中会存储一些非规律性的内容，网络泛化能力反而会被降低。因此，本文使用一个隐藏层并选取 14 个单元。

(4)输出设计

输出层用于表示对当前局面的估值，本文采用 1 个输出单元。输出值越接近于 1，表示当前局势越有利于黑方；反之，输出值越接近于 0，表示当前局势越有利于白方。走法生成器对着法进行排序的实质，就是利用所有输出值的大小，并据此形成最佳着法。

3 增强学习过程

用向量 $\boldsymbol{A}=(a_1, a_2,\cdots, a_n)$来表示棋盘状态，其中，$a_1$~$a_{22}$ 表示计算机方局面特征的数目；a_{23}~a_{44} 表示对手方局面特征的数目。设输出为 c，c=1 和 c=0 分别表示计算机和对手赢的输出状态，0<c<1 表示其余的棋局输出状态。每下一步棋都会改变棋盘状态，形成一个新的棋盘状态向量 $\boldsymbol{A}_t$，一盘棋如果下了 n 步，就有 n 个代表不同棋盘状态的向量 $\boldsymbol{A}_1, \boldsymbol{A}_2,\cdots, \boldsymbol{A}_n$。由于不能给所有的棋盘状态分别赋相应的一个评估值，因此只能给出最终状态 $\boldsymbol{A}_n$ 的最后评估值(即可以判断出赢输的状态)。若是由计算机取得胜利，则最后估值 R 取 1，若是对方赢得胜利，则 R 取 0，平局时 R 取 0.5。所以，对最终状态网络输出值，计算其偏差为：$\delta_n=R-P(\boldsymbol{A}_n)$。根据误差逆传播原理，用 δ_n 来修正 BP 网络的各连接权值，使 $P(\boldsymbol{A}_n)$接近于 R。修正后，网络对应 $\boldsymbol{A}_n$ 的输出是 $P'(\boldsymbol{A}_n)$。然后根据 TD 的预测原理，2 个相邻状态的评估值比较接近，因此，可得 $\boldsymbol{A}_{n-1}$ 状态的网络输出偏差为：$\delta_{n-1}=P'(\boldsymbol{A}_n)-P(\boldsymbol{A}_{n-1})$。同理，用 δ_{n-1} 修正网络权值，使 $P(\boldsymbol{A}_{n-1})$接近其后一状态输出值 $P(\boldsymbol{A}_n)$。这样，通过取棋局结束时的最后 2 个状态计算出误差，再根据式(1)修改权值，按上述方式进行训练，直到棋局记录全部训练完毕。

4 博弈训练的优化

在博弈训练中，衰减因子 λ 和学习速率 α 的取值有很大影响，鉴于此，本文提出分阶段设置不同参数值。在对弈过程中，设黑方为先落子方，其落子个数为 n，设一定值 $step$，当 n<$step$ 时，设 α 的值为 α_1，此时 λ 的值设为 λ_1；当 $n\geqslant step$ 时，设置 α 的值为 α_2，此时 λ 的值设为 λ_2；同时，设置参数步长$\Delta\alpha$ 和$\Delta\lambda$，使得 $\alpha_2=\Delta\alpha+\alpha_1$，$\lambda_2=\Delta\lambda+\lambda_1$。

由五子棋的特点，通常设定 step 的值为 8。在游戏开局阶段，学习速率设为 α_1，衰减因子设为 λ_1，采用随机的着法选择策略。设状态 s_t 为五子棋在时刻 t 的局面状态，集合 $\boldsymbol{A}$ 为处于状态 s_t 的全部合法的走法集合，$a_t\in\boldsymbol{A}$ 是集合 $\boldsymbol{A}$ 中的一个着法，该着法的后继状态记为 s_{t+1}，即 $s_{t+1}=\delta_t(s_t, a_t)$。若用 r_t 表示 BP 神经网络对 s_t 的后一个状态 s_{t+1} 的估计值，即 $r_t=\gamma(s_t, a_t)$，那么集合 $\boldsymbol{A}$ 中每个着法 a_t 都会有相应不同的评估值，把估值最高(也就是最优)的前 m 个着法保存起来作为后备着法，m 即为后备着法的窗口大小，其对应权值 $h(s_t, a_i, m)$按照式(2)进行定义：

$$h(s_t,a_i,m)=\frac{\gamma(s_t,a_i)}{\sum_{j=1}^{m}\gamma(s_t,a_j)} \tag{2}$$

若是随机着法，在选择其后继着法的过程中，要根据后备着法的权值依据概率大小随机进行。首先随机选择 BP 神经网络对每个不同局面前 m 个估计值最高的着法，当选择后一个着法时，要根据这 m 个后备着法，随机地选择下一个走法。但是，这种方式会产生一定的误差，在搜索过程中，博弈树会不断加深，这样，后期的着法可能有更高的评估值，使得所选择的着法不一定是最好的着法，因此，本文采取 m 取值逐渐减小的策略。

中局阶段设置学习速率 α 的参数值为 α_2，衰减因子 λ 的值为 λ_2，并根据极大极小原理选择最佳着法。因篇幅所限，

这里不再赘述。

5　基于增强学习的五子棋系统

5.1　系统实现

5.1.1　系统总体功能的实现

基于上面研究的计算机博弈策略，本文设计并实现了五子棋博弈系统 RenjuTD。该系统基于 C++语言，操作系统为 Windows XP Professional，硬件环境采用惠普双核 PC、2.80 GHz 的 Intel 处理器和 1 GB 内存。

RenjuTD 主要包括 5 个部分，即棋盘数据表示、搜索引擎、走法生成器、估值引擎和界面。与传统的 Renju 相比，RenjuTD 采用威胁空间搜索(Threat Space Search, TSS)算法，并且设计了一个 TD 学习器。下面给出 RenjuTD 系统的工作流程：首先初始化程序，然后开始进行自对弈训练。先随机选择一个开局状态，接着调用 TSS 算法，若搜索成功，则将该着法序列生成为制胜着法序列，并保存，利用这种着法落子直到结束；如果搜索不成功，生成后继着法及后继状态进行新的着法选择，根据 BP-TD(λ)强化学习算法调整整个 BP 神经网络的权值。程序重复此流程直至结束，则完成一次 TD 自学习过程。

5.1.2　自我学习训练

在系统的自我学习训练中，为在博弈过程中实现自对弈 TD 学习，要将黑白双方程序按相应的结构进行连接。在对弈过程中，使对弈双方进行相应控制，包括开始训练、取消训练等，并将博弈双方信息彼此进行传送。自我学习训练设置的界面如图 2 所示。

图 2　训练设置的界面

5.2　实验结果分析

5.2.1　BP 神经网络输入输出分析

下面使用目标函数 $f(x)=x$ 来检验 BP 神经网络是否能够正确地学习到网络的估值。使用数组 InPut[6]作为输入，数组 OutPut[6]作为输出，以测试在 6 个输出单元中能否重现 6 个输入单元。图 3 是样例 InPut[1]={100 000}在不同学习训练次数下输出值与输入值比较后的结果。可以看出，随着训练次数的增加，网络的输出越来越逼近输入值。

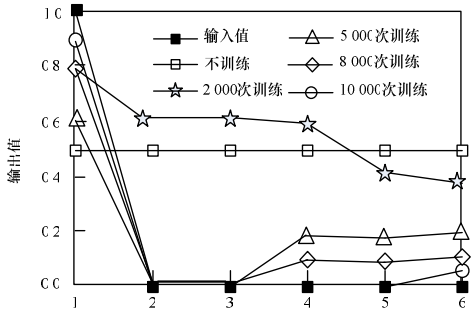

图 3　样例{100 000}在不同训练次数下的输出比较

5.2.2　性能分析

在五子棋开局阶段和中局阶段，需分别设置参数 α 和 λ 值。首先设置 *step* 的值为 8，在开局时，设置 α 的值为 α_1，λ 的值为 λ_1，并在中局阶段，设置 α 的值为 α_2，λ 的值为 λ_2。

令开局和中局的 α、λ 值整体表示为$(\alpha, \lambda)=((\alpha_1, \lambda_1),(\alpha_2, \lambda_2))$。选取 α 和 λ 的步长为 0.2 时，令$(\alpha, \lambda)_1=((0.2,0.3), (0.4, 0.5))$，$(\alpha, \lambda)_2=((0.2,0.4),(0.4,0.6))$；选取 α 和 λ 的步长为 0.1 时，令$(\alpha, \lambda)_3=((0.2,0.4),(0.3,0.5))$,$(\alpha, \lambda)_4=((0.4,0.4),(0.5,0.5))$；选取 α 和 λ 的步长为−0.2 时，令$(\alpha, \lambda)_5=((0.2,0.4),(0.1,0.3))$,$(\alpha, \lambda)_6=((0.4,0.4),(0.2,0.2))$。以上述$(\alpha, \lambda)$值，程序自学习训练 10 000 次后的结果如图 4~图 9 所示。

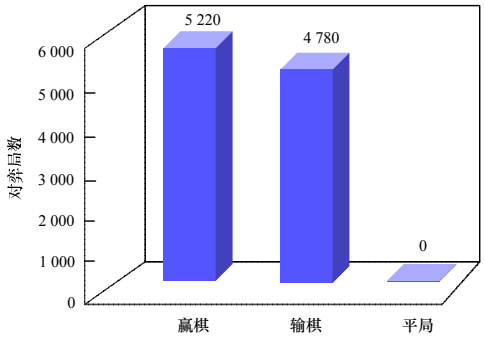

图 4　取值$(\alpha, \lambda)_1$自对弈结果

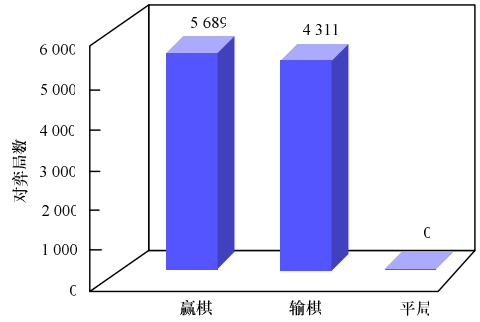

图 5　取值$(\alpha, \lambda)_2$自对弈结果

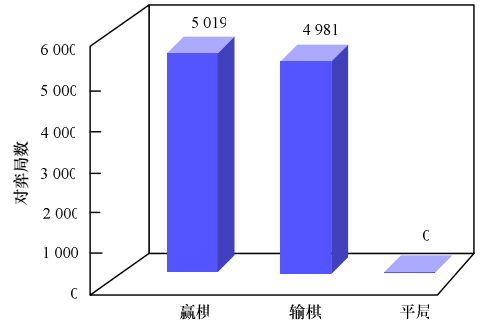

图 6　取值$(\alpha, \lambda)_3$自对弈结果

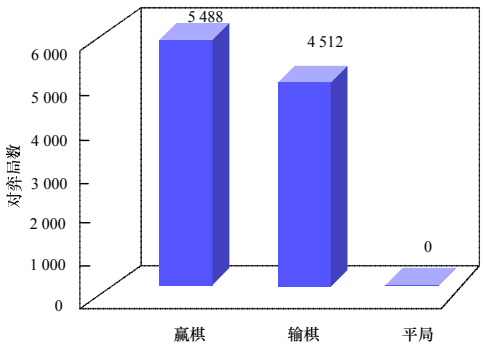

图 7　取值$(\alpha, \lambda)_4$自对弈结果

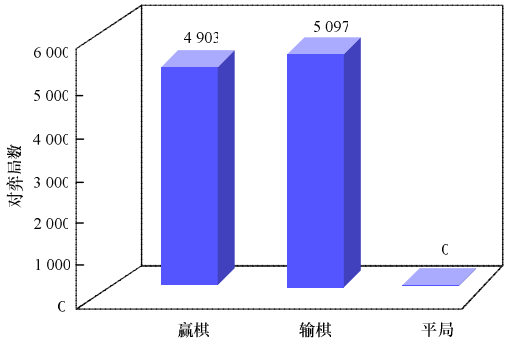

图 8 取值$(\alpha, \lambda)_5$自对弈结果

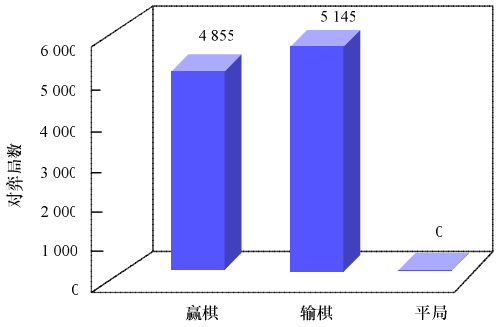

图 9 取值$(\alpha, \lambda)_6$自对弈结果

在自对弈 10 000 次后，RenjuTD 已经具备了一定的棋力，此后，α 的取值将逐渐递减，通过减缓学习速率，取得更好的训练效果。

从上面的自我学习训练结果可以看出，对于 α 和 λ 的值，若在中局阶段比开局阶段的值大，程序的棋力相对要高；反之，若中局阶段设置的值较小，程序棋力没有任何提升。此外，α 取值若变大将会使神经网络的收敛速度加快，当 α 一定时，λ 的值越小，相邻 2 个局面在时间上的相隔越远，彼此影响就越小，即对利用历史经验越少，TD 学习效果就越不理想。需要指出的是，α 取值不能过大，否则将会引起网络发生振荡，失去学习功能。随着学习任务的增加，α 的取值将会变小，从而充分利用历史经验训练网络。

5.2.3 系统性能比较

RenjuTD 使用$(\alpha, \lambda)_2$=((0.2,0.4),(0.4,0.6))自对弈 10 000 次后，参数改变为((0.1,0.1),(0.2,0.2))自对弈 5 000 次，此时，程序已经具备了一定的棋力。为了测试 RenjuTD 的性能，选择传统版本快乐五子棋衡量棋力的高低。两者对弈 3 000 次，其结果如图 10 所示。从图中可以看出，RenjuTD 与快乐五子棋对弈 2 000 次，取得的胜率为 70.5%，具有良好的学习效果。

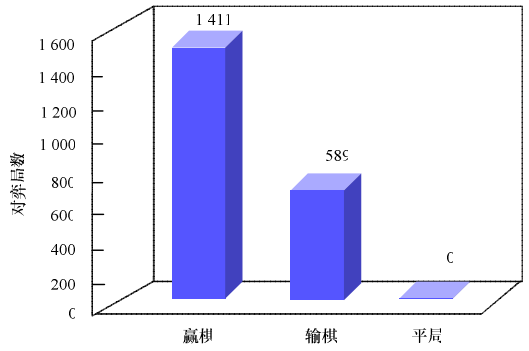

图 10 RenjuTD 与快乐五子棋对弈结果

6 结束语

局面估值是棋类博弈中决定棋力大小的主要因素之一。本文将 TD(λ)算法与 BP 神经网络相结合，应用到五子棋的局面估值中。同时，为了使 TD 学习更有效和更快速，针对五子棋的特点，提出一种分阶段设置参数值的策略，提高了估值网络的学习效率和准确性。实验结果表明，经过 10 000 盘的学习训练，程序的博弈水平明显得到提高。

参考文献

[1] 徐心和，邓志立，王 骄，等. 机器博弈研究面临的各种挑战[J]. 智能系统学报, 2008, 3(4): 289-293.

[2] Peter D. The Convergence of TD(lambda) for General Lambda[J]. Machine Learning, 1992, 8(34): 341-362.

[3] Xu Changming, Ma Zhongmin, Xu Xinhe. A Method to Construct Knowledge Table-base in k-in-a-row Games[C]//Proc. of ACM Symposium on Applied Computing. [S. l.]: ACM Press, 2009: 929-933.

[4] Wu Chen. A New Family of k-in-a-row Games[C]//Proc. of the 11th Advances in Computer Games Conference. Taipei, China: [s. n.], 2005: 88-100.

[5] Rumelhart D E, Hinton G E, Williams R J. Learning Representations by Back-propagating Errors[J]. Nature, 1986, 323(6088): 533-536.

[6] 王 骄，王 涛，罗艳红，等. 中国象棋计算机博弈系统评估函数的自适应遗传算法实现[J]. 东北大学学报：自然科学版, 2005, 26(10): 949-952.

[7] 徐长明，马宗民，徐心和，等. 面向机器博弈的即时差分学习研究[J]. 计算机科学, 2010, 37(8): 219-223.

[8] 宫瑞敏. 基于增强学习的计算机博弈策略的研究与实现[D]. 沈阳：沈阳理工大学, 2011.

编辑 张正兴

(上接第 162 页)

[6] Chen B, Zhu Qifeng, Morgan N. Learning Long-term Temporal Features in LVCSR Using Neural Networks[C]//Proc. of Conference on Spoken Language Processing. Jeju, Korea: [s. n.], 2005: 1233-1236.

[7] 李晨冲，董 滨，潘复平，等. 汉语普通话易混淆音素的识别[J]. 计算机工程, 2009, 35(23): 201-203.

[8] Ganapathy S, Thomas S, Hermansky H. Comparison of Modulation Features for Phoneme Recognition[C]//Proc. of IEEE International Conference on Acoustics Speech and Signal Processing. Dallas, USA: IEEE Press, 2010: 5038-5041.

[9] Ketabdar H, Bourlard H. Hierarchical Integration of Phonetic and Lexical Knowledge in Phone Posterior Estimation[C]//Proc. of IEEE International Conference on Acoustics Speech and Signal Processing. Las Vegas, USA: IEEE Press, 2008: 4065-4068.

[10] Le V B, Lamel L, Gauvain J L. Multi-style MLP Features for BN Transcription[C]//Proc. of IEEE International Conference on Acoustics Speech and Signal Processing. Dallas, USA: IEEE Press, 2010: 4866-4869.

编辑 索书志