

一个基于 BP 网络的字符识别器^①

杨 戈^{1,2}, 谢思路¹

¹(北京师范大学珠海分校 信息技术学院, 珠海 519087)

²(北京大学深圳研究生院 深圳物联网智能感知技术工程实验室, 深圳 518055)

摘 要: 本文简单介绍了人工神经网络的背景知识, 提出了一种利用传统 BP(Back-Propagation 误差逆传播)网络识别印刷字符的方法, 用 C 语言对其进行了实现. 在进行了大量实验之后, 实验结果表明该字符识别器具有较好的有效性和正确性, 能够在合理的误差范围内以较好的效率成功识别字符.

关键词: 人工神经网络; BP 网; 图像识别; C 语言

Character Recognizer Based on BP Network

YANG Ge^{1,2}, XIE Si-Lu¹

¹(Beijing Normal University (Zhuhai), College of Information Technology, Zhuhai 519087, China)

²(Engineering Lab on Intelligent Perception for Internet of Things (ELIP), Shenzhen Graduate School, Peking University, Shenzhen 518055, China)

Abstract: This paper introduced the background knowledge of Artificial Neural Network and a way to use BP network to recognize printed characters was proposed, which was implemented with C language. After a lot of experiments, the experimental results show that the character recognizer has good validity and correctness. Besides, the printed characters can be recognized successful within the reasonable range of error.

Key words: artificial neural networks; back-propagation network; image recognition; C language

人工神经网络(ANN Artificial Neural Networks)理论、实现方式及算法是人工智能(AI Artificial Intelligence)研究的重要课题之一, 也是智能控制的重要分支领域. 自从人们认识到人脑计算与传统的数字计算机相比是完全不同的方式开始, 关于人工元神经网络(一般称之为“神经网络”(neural network))的研究工作就开始了^[1]. 人脑是一个高度复杂的、非线性的和并行的计算机, 能够组织它的组成成分, 即神经元. 人工神经网络的研究在一定程度上受到了生物学的启发, 因为生物的学习系统是由相互连接的神经元组成的异常复杂的网络, 而人工神经网络与此大体相似^[2]. 神经网络是由简单处理单元构成的大规模并行分布式处理器, 天然地具有存储经验知识和使之可用的特性, 以人工的方法模拟人脑的功能, 有助于加深对思维及智能的认识. 自 40 年代以来,

人们在研究人脑机理的基础上, 广泛开展模仿脑模型的人工神经网络理论的研究, 由于人工神经网络具有与人脑相似的高度并行性、良好的容错性和联想记忆功能、自适应和自学习能力等特点, 尤其是以 BP 网络(Back-Propagation 误差逆传播)为代表的神经网络具有良好的自学习能力、强大的分类能力、容错能力, 可以实现输入到输出的非线性映射.

由于人工神经网络所具有的以上特点, 所以它在以下的一些主要领域中得到了应用:

① 模式识别与图像处理印刷体和手写体字符识别、语音识别、指纹、人脸识别、RNA 与 DNA 序列分析、癌细胞识别、目标检测与识别、心电图、脑电图分类、油气藏检测、加速器故障检测、电机故障检测、图像压缩复原. 其中人脸检测作为智能人机交互

① 基金项目:国家自然科学基金(60875050,61272364);国家高技术研究发展计划(863 计划)(2006AA04Z247);广东省自然科学基金(9151806001000025);

深圳市科技计划项目及基础研究计划项目(JC201005270275A);深圳市战略性新兴产业发展专项资金项目(JCYJ20120614144655154);

北京师范大学珠海分校科研创新团队(多媒体传输与计算机视觉研究团队,201251006);北京师范大学珠海分校教改项目(201329)

收稿时间:2013-04-03;收到修改稿时间:2013-07-02

中的一个重要环节, 受到了研究领域的普遍重视, 并取得了很大的进展^[3].

② 控制及优化化工过程控制、机械手运动控制、运载体轨迹控制以及电弧炉控制.

③ 金融预测与管理股票市场预测、有价证券管理、借贷风险分析以及信用卡欺骗检测.

④ 通信自适应均衡、回声抵消、路由选择、导航以及多媒体处理系统.

⑤ 其他如知识发现和数据挖掘、气象与地球科学等.

本论文主要介绍了 BP 网络, 并设计和实现了基于传统 BP 网络和基于改进 BP 网络的字符识别器, 用 C 语言对其进行了实现并验证.

1 基于传统BP网络的字符识别器的设计

本文中设计的字符识别器是可以实现对字符进行识别的方法. 例如现在要利用一 BP 网络来识别三种图形, 如图 1 的 (a), (b), (c) 所示. 识别过程为系统经过基于 BP 算法的反复学习之后正确识别出图形. 将图形以数组的方式输入, 通过隐藏层节点映射到一组输出模式, 使输出模式等同于输入模式.

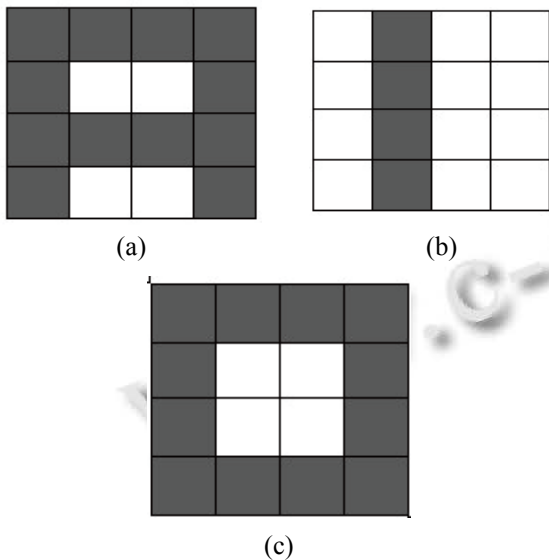


图 1 待识别图形

BP 网络(Back Propagation NN)即进行误差校正的多层前馈网络, BP 网络是利用实际输出与期望输出之差对网络的各层连接权由后向前逐层进行校正的一种计算方法, 它适用于任意多层的网络. 图 2 所示是一

个三层 BP 网络.

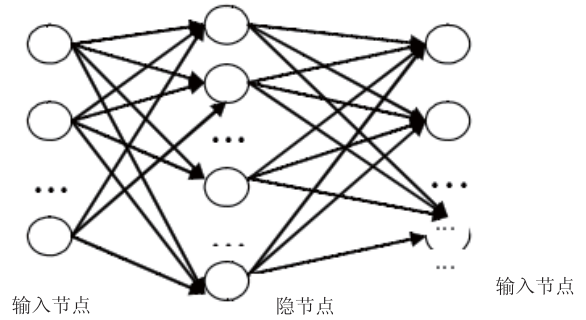


图 2 三层 BP 网络

BP 网可以看作是一个从输入到输出的高度非线性映射, 具有的主要特点体现在信息处理的并行性, 分布式的信息存储, 自组织和自适应性, 具有很强的学习和联想功能以及容错性并且由此而来的泛化能力. BP 神经网络已经成为神经网络的重要模型之一, 在很多领域都得到了应用.

BP 网络的学习过程主要由四部分组成:

- ① 输入模式顺传播, 即输入模式由输入层经中间层向输出层传播计算;
- ② 输出误差逆传播, 即输出的误差由输出层经中间层传向输入层;
- ③ 循环记忆训练, 即模式顺传播与误差逆传播的计算过程反复交替循环进行;
- ④ 学习结果判别, 即判定全局误差是否趋向极小值.

设图 2 中的网络的输入层有 n 个神经元, 输出层有 q 个神经元, 中间层有 p 个神经元.

通常人工神经网络的学习方式是“对权值的学习”^[4]. BP 网络的学习过程是有教师学习方式的权值学习, 训练集包含 M 个样本, 取第 k 个训练样本, 对于输入层和中间层而言, 设输入层的输入为 $a_{k,i}$ 则:

$$net_{k,j} = \sum_{i=1}^n W_{i,j} a_{k,i} - \theta_j \tag{1}$$

在(1)式中, $i=1,2,\dots,n$, $j=1,2,\dots,q$ (下同). $net_{k,j}$ 表示中间层的神经元 j 接收到的总输入, θ_j 表示神经元 j 的阈值, $W_{i,j}$ 表示输入层 i 与中间层 j 之间的连接权值. BP 算法中大多选用 S 型函数作为输出函数, S 型函数的优点在于它是可微分的^[5]. 其中 S 型函数如公式(2)所示:

$$f(x) = \frac{1}{1 + \exp(-x)} \tag{2}$$

则神经元 j 的实际输出 $b_{k,j}$ 如公式(3)所示:

$$b_{k,j} = f(\text{net}_{k,j}) = \frac{1}{1 + \exp(-\text{net}_{k,j})} \quad (3)$$

定义网络误差函数如公式(4)所示:

$$E = \sum_p E_p \quad (4)$$

其中

$$E_p = \frac{1}{2} \sum (d_{k,j} - b_{k,j})^2 \quad (5)$$

在公式(5)中 $d_{k,j}$ 是对单个训练样本, 单元 j 的期望输出^[6]. 训练网络的目的是找到一组权重, 使误差函数极小化. 由于最小误差要求权变换是负梯度方向,

所以应使 $W_{i,j}$ 的调整量 $\Delta W_{i,j}$ 与 $\frac{\partial E_p}{\partial W_{i,j}}$ 的负值成正比

例变化, 故两层之间的调整量应为:

$$\Delta W_{i,j} = -\alpha \frac{\partial E_p}{\partial W_{i,j}} \quad (6)$$

令 $-\frac{\partial E_p}{\partial W_{i,j}} = C_{p,j}$, 则

$$\frac{\partial E_p}{\partial W_{i,j}} = \frac{\partial E_p}{\partial \text{net}_{p,j}} * \frac{\partial \text{net}_{p,j}}{\partial W_{i,j}} = \frac{\partial E_p}{\partial W_{i,j}} * a_{k,i} = -C_{p,j} * a_{k,i}$$

得到:

$$\Delta W_{i,j} = -\alpha C_{p,j} * a_{k,i} \quad (7)$$

在公式(7)中, $a_{k,i}$ 即输入层的输入, $\alpha > 0$ 表示学习系数.

而中间层与输出层之间的计算过程也类似于输入层与中间层之间的计算过程, 故这里不再赘述.

BP 网络的优点如下:

- ① 对问题的先验知识要求少;
- ② 可实现对特征空间较为复杂的划分;
- ③ 适合用高速并行处理系统来实现.

2 基于传统BP网络的字符识别器的实现

该字符识别器的 BP 网络结构的设定如下:

① 神经元个数的选取

对于 BP 神经网络, 其结构的确定包括了输入层、输出层节点个数的确定和隐藏层层数、隐藏节点个数的确定. 其中输入层和输出层节点个数是根据实际问题确定的. 在本例中, 根据图 12 的待识别图形可以看出输入层的节点个数应设为 16 个(因为 16 个节点可以表示出一个字母), 而输出层节点个数设为 3 个, 表示

期望的三个不同的输出. 关于隐藏层数和隐藏节点个数的确定一直是人工神经网络设计的难点, 这是因为隐藏层神经元数与所需解决的问题的特定要求、输入层和输出层神经元的数量以及训练样本的数量等因素有直接关系, 故隐藏层神经网络的数量问题没有精确解只有次优解. 隐藏层神经元的数量不能太少, 太少会使得网络解决问题的精度不够. 而最关键的是隐藏层的神经元数量不能过多, 在过多的情况下对于样本学习的精度很容易达到, 但是网络很可能出现过拟合, 造成泛化能力下降, 容错性差. 通常是通过经验来确定隐藏层的神经元个数, 在本例中将其设定为 8 个.

② 初始权值的选取

由于系统是非线性的, 初始权值对于学习是否达到局部最小、是否能够收敛以及训练时间的长短关系很大. 如果初始权值太大, 使得加权后的输入落在激活函数的饱和区, 从而使得调节过程几乎停顿下来. 所以, 一般总是希望经过初始加权后的每个神经元的输出值都接近于零, 这样可以保证每个神经元的权值都能够在她们的激活函数变化最大之处进行调节^[7]. 所以, 一般取初始权值在 $(-1, 1)$ 之间的随机数.

③ 学习系数的选取

学习系数决定每一次循环训练中所产生的权值变化量. 太高的学习系数可能导致系统的不稳定, 但太小的学习系数将导致训练时间较长, 收敛速度很慢, 不过能保证网络的误差值不超出误差表面的低谷而最终趋于最小误差值^[8]. 所以在一般情况下, 倾向于选取较小的学习系数以保证系统的稳定性. 在本例中将学习系数选取为 0.01.

④ 允许最大学习次数的选取

允许最大学习次数选取的关键是要保证在其范围内能够完成正确识别的功能, 即在允许的学习次数内, 误差值能够达到期望的精度, 小于允许最大误差值. 当在允许最大学习次数的范围内, 不能够达到期望精度, 则学习失败. 在本例中, 先将其值设定为 3000.

本例中的具体设定如下:

首先将这三个图形用三个输入模式向量 A_1 , A_2 , A_3 来表示, 其中有颜色的部分为 1, 没有颜色的部分为 0. 即:

$$X_1 = [1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1]^T$$

$$X_2 = [0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0]^T$$

$$X3 = [1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1]^T$$

这三个输入模式所对应的希望输出为:

$$D1 = [1, 0, 0]^T$$

$$D2 = [0, 1, 0]^T$$

$$D3 = [0, 0, 1]^T$$

该例对应的 BP 网络的输入层有 16 个神经元, 输出层有 3 个神经元, 隐含层有 8 个神经元. 其网络结构如图 4 所示.

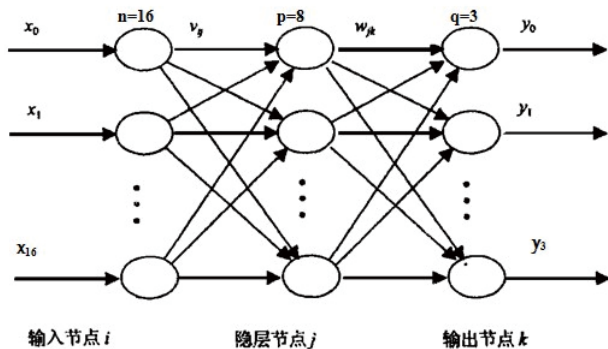


图 3 三层 BP 网络

这里设定最大允许误差为:

$$E_{max} = 0.01$$

误差的计算方法按照平方和误差计算:

$$E = \frac{1}{2} \sum (d - y)^2$$

学习系数取 $\alpha = \beta = 0.1$.

综上所述, 实验参数的设定如表 1 所示.

表 1 识别器的实现参数

输入层神经元个数	16
隐藏层神经元个数	8
输出层神经元个数	3
训练样本个数	3
允许最大学习次数	3000
学习系数	$\alpha = \beta = 0.1$
允许最大误差值	0.01

本节根据第 1 节中的设计进行了识别器的实现. 字符识别器的算法伪代码如下:

```

头文件;
main(){
声明变量;
初始化输入样本 x1, x2, x3;
初始化期望输出 y1, y2, y3;
初始化输入层和中间层的连接权 w;
    
```

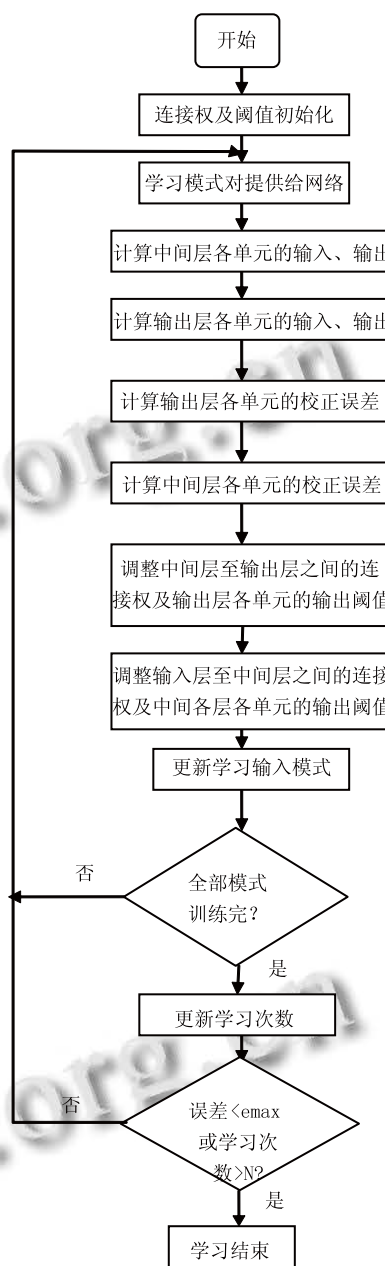


图 4 BP 算法流程图

```

初始化中间层和输出层的连接权 v;
初始化中间层和输出层的阈值 t1,t2;
do{//一次循环表示一次学习过程
while 样本数 <= 3 { //每一次学习都要训练三个样本
选定要学习的样本, 将样本赋值给 x0;
用 S 型激励函数计算中间层的实际输出 bj;
用 S 型激励函数计算输出层的实际输出 ct;
计算误差值 er;
    
```

计算输出层各单元的校正误差 dt ;

计算中间层各单元的校正误差 ej ;

计算下一次学习时中间层和输出层之间的新连接权和新阈值;

计算下一次学习时输入层和中间层之间的新连接权和新阈值;

样本号 $cp++$;

}

学习次数 $cnt++$;

}while 误差值 $er <$ 允许最大误差值 E_{max} 或者学习次数 < 3000 ;

if $er < e_{max}$ {

学习成功;

输出输入层和中间层之间的连接权;

输出中间层和输出层之间的连接权;

输出完成学习后中间层的新阈值;

完成学习后输出层的新阈值;

}

else{

学习失败;

}

3 实验结果分析

3.1 实验环境

实验运行的具体的实现环境见表 2.

表 2 实验环境

操作系统	Windows 7 旗舰版
内存大小	2.0 GB
硬盘大小	320GB
软件版本	C-Free 5

3.2 实验结果分析

为了验证识别器识别字符的成功率, 本文进行了 4000 次的大量实验, 但由于篇幅所限, 在此只列出部分的实验结果(见表 3).

由表 3 所记录的实验结果可以看出, 该算法的运行结果较为稳定, 基本上在允许的范围内都能够实现识别字符的功能, 成功完成学习的次数以及误差均存

在于合理的范围之内. 其中在 3.2 节中设定误差值应小于 0.01, 允许最大学习次数为 3000 次.

表 3 4000 次实验结果记录(部分记录)

实验序号	完成学习的次数	误差值	结果分析
1	609	0.00999881	成功识别
2	664	0.00998786	成功识别
3	856	0.00998983	成功识别
4	699	0.00999482	成功识别
5	842	0.00999512	成功识别
6	889	0.00999316	成功识别
7	691	0.00998495	成功识别
8	662	0.00999111	成功识别
9	694	0.00999079	成功识别
10	724	0.00998688	成功识别
50	676	0.00998808	成功识别
100	712	0.00999642	成功识别
500	569	0.00998729	成功识别
1000	548	0.00997927	成功识别
2000	622	0.00999418	成功识别
3000	685	0.00999457	成功识别
4000	852	0.00999161	成功识别

基于传统 BP 网络的识别器的运行结果如图 5 所示.

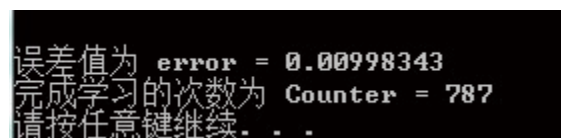


图 5 程序运行结果

从图 5 可以看出, 此次实验中学习次数为 787 次, 在规定的学习次数内(3000)完成了学习. 并且误差值小于允许最大误差值(0.01). 所以此次实验完成了识别字符的任务, 实际输出在可接受的误差内等于了实际输出.

3.3 改变学习次数实验结果分析

在允许最大学习次数为 3000 次的情况下, 让程序自动进行 1000 次反复试验, 试验结果如图 6 所示. 此时学习成功率为 100%.

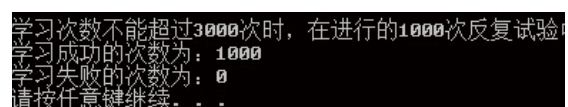


图 6 1000 次实验(允许最大学习次数为 3000)

当将允许最大学习次数缩小为 1000 次时, 让程序自动进行 1000 次反复试验, 试验结果如图 7 所示. 此时学习成功率为 91.8%.

```
学习次数不能超过1000次时, 在进行的1000次反复试验
学习成功的次数为: 918
学习失败的次数为: 82
请按任意键继续. . .
```

图 7 1000 次实验(允许最大学习次数为 1000)

总结两次实验, 可以看出, 当学习次数可以多达 3000 次时基本能够保证正确识别字符; 当将允许最大学习次数缩小为 1000 次时, 并不能保证完成识别字符的功能, 但是能够缩短程序的运行时间, 当程序需要识别的输入模式对的数量庞大时, 这种时间的节省有着十分重要的意义.

4 结语

本文用 BP 算法设计与实现了一个字符识别器, 但这种传统的 BP 算法具有一定的缺陷, 因此本文还提出了几种改进方法, 并实现了其中一种改进方法, 即累积误差校正算法, 它能够有效的缩短学习所需时间.

本文识别器完成了识别 3 个字符的功能, 同理可以扩展识别 26 个英文字母、0-9 数字或其他字符, 如果附加其他功能模块, 如翻译功能, 即可通过计算机

翻译印刷书本中英文单词、短语等, 对于盲人等特定人群有着特别意义^[9].

参考文献

- 1 Simon Haykin. 神经网络与机器学习. 北京: 机械工业出版社, 2011: 2-10.
- 2 Tom M. Mitchell. 机器学习. 北京: 机械工业出版社, 2003: 46-89.
- 3 叶航军, 白雪生, 徐光祐. 基于支持向量机的人脸姿态判定. 清华大学学报(自然科学版), 2003, 43(1): 67-70.
- 4 Stuart Russell, Peter Norvig. 人工智能——一种现代方法(第二版). 北京: 人民邮电出版社, 2004: 23-40.
- 5 危辉. 基于结构学习和迭代自映射的自联想记忆模型. 软件学报, 2002, 13(3): 438-440.
- 6 张泉灵, 王树青. 基于神经网络的非线性预测函数控制. 浙江大学学报(工学版), 2001, 35(5): 497-501.
- 7 王旭, 王宏, 王文辉. 人工神经网络原理与应用. 沈阳: 东北大学出版社, 2000: 19-38.
- 8 张宇, 潘国腾, 谢伦国. 一种基于人工神经网络的条件分支预测算法. 华中科技大学学报(自然科学版), 2005, 33(增刊): 101-103.
- 9 陆琼瑜, 童学峰. BP 算法改进的研究. 计算机工程与设计, 2007, 28(3): 648-650.

(上接第 118 页)

参考文献

- 1 张伟, 梅宏. 一种面向特征的领域模型及其建模过程. 软件学报, 2003, 14(8): 1345-1356.
- 2 Moon M, Yeom K. An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line. IEEE Trans. on software Engineering, 2005, 31(7): 551-569.
- 3 Jones L, Northrop L. Clearing the way for software product line success. Software, IEEE, 2010, 27(3): 22-28.
- 4 王孟伟, 管群. 产品线可变性的需求方法研究. 计算机工程与设计, 2011, 32(11): 3725-3728.
- 5 Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. Feature-oriented domain analysis(FODA) feasibility study. CMU/SEI-90-TR-21, Pittsburgh: Software Engineering Institute, Carnegie Mellon University, 1990: 1-10.
- 6 Kang KC, Kim S, Lee J, Kim K, Shin E, Huh M. FORM: A feature-oriented reuse method with domain-specific reference architectures. Annals of Software Engineering, 1998, 5: 143-168.
- 7 Griss ML, Favaro J, D' Alessandro M. Integrating feature modeling with the RSEB. Proc. of the Fifth International Conference on Software Reuse. Victoria, IEEE Computer Society, 1998: 76-85.
- 8 Pohl K, Bockle G, van der Linden F. Software Product Line Engineering. Berlin, Springer-Verlag, 2005: 72-86.
- 9 Buhne S, Lauenroth K, Pohl Klaus. Why is it not Sufficient to Model Requirements Variability with Feature Models. Proc. of Workshop: Automotive Requirements Engineering, 2004, 9: 5-12.
- 10 赵海燕, 张伟, 麻志毅. 面向复用的需求建模. 北京: 清华大学出版社, 2008: 127-145.