

自顶向下优化神经网络的方法

杨钟瑾

(广东商学院 信息学院, 广东 广州 510320)

摘要: 介绍了一种自动优化神经网络的新启迪方法. 这种启迪方法综合采用了快速自顶向下优化神经网络结构算法、动态优化学习参数算法和快速交叉校验算法. 首先, 快速自顶向下优化神经网络结构算法自动地优化神经网络结构; 其次, 动态优化学习参数算法动态地调整学习参数和选取优化的学习参数; 随后, 快速交叉校验算法的引入能有效地防止过度适应问题. 实验结果显示, 这种启迪方法与其它算法相比, 具有更强的归纳性能、优化的网络结构和更快的学习速度.

关键词: 神经网络; 神经中枢手术优化; 自顶向下; 学习参数优化; 交叉校验; 归纳

中图分类号: TP183

文献标识码: A

文章编号: 1007-7162(2006)03-0095-07

引言

神经网络属于智能机器, 通过学习可以获取知识和存储知识, 并且能有效地解决问题, 从而吸引众多的研究者与应用者. 前馈神经网络是解决非线性问题的极好模型^[1], 他们的训练通过梯度下降法优化路径来实现^[2]. 在历史上, 人们经常使用反向传播算法 (BP 算法) 来训练前馈神经网络^[3-4]. BP 算法是被 Rumelhart, Hinton 和 Williams 发现和提出^[5]的. BP 算法有致命的缺点: 容易陷入局部最小值, 对初始条件过于敏感和收敛速度缓慢. 可是, 如果 BP 算法具备适合的初始条件、适当的网络结构和优化的学习参数, 那么它就能充分发挥优势而出色地完成各项事务. 随便给出初始条件、学习参数和随意猜测网络结构的情形, 都会导致迟缓的、繁杂的学习过程. 因此, 一种高性能的网络优化方法值得探索与研究.

理论研究^[6]和仿真试验^[7]都充分证明了以下两种事实: 太大的网络结构容易造成归纳性差; 太小的网络结构则造成学习能力低. 前馈神经网络通过使连接权数量最小化来实现结构优化. 众所周知, 单一的神经网络结构不可能适合所有问题^[8]. 近年来, 相继有一些结构优化方法被提出: (1) 规则化, 倍受推崇的方法有交叉校验^[9]; (2) 自底向上方法, 有著名的瀑流关联算法^[10]; (3) 自顶向下方法, 神经中枢手术优化算法^[11]就是其中一种有代表性的方法; (4) 遗传算法^[12]. 其中, 自顶向下方法具有快速收敛和低敏感于初始条件的长处. 优化的学习参数对神经网络的性能具有深远的影响. 学习参数的优化包括: (1) 学习率的优化; (2) 动量因子优化. 在神经网络学习过程中, 学习率是控制连接权改变幅度的重要参数. 在选择学习率时应谨而慎之, 太小的学习率使网络收敛迟缓, 太大的学习率令网络不稳定而导致不收敛^[13]. 动量因子的采用有助于减少“锯齿”现象, 错误选择动量因子将导致网络收敛缓慢. 苛求可以解决所有问题的单一学习参数也是不现实的^[1]. 优化学习参数算法可以动态调整和自动获取优化的学习参数. 在

收稿日期: 2006-04-10

作者简介: 杨钟瑾 (1966-), 男, 讲师, 博士, 主要研究方向为信息分析、预测控制与人工智能等.

通常 BP 算法学习进程中, 随着网络学习的继续和深入, 神经网络所拥有的输入 - 输出映射性能也由简单变为复杂; 但是, 网络学习若不能在某时刻正确停止, 则过多的学习就会引发过度适应问题. 交叉校验方法是预防过度适应问题出现的良方.

由于常规 BP 算法的固有缺陷和影响神经网络性能的诸多因素, 若仅仅采取单一、片面的算法优化神经网络, 虽然能加快网络的收敛速度, 但是陷入局部极小点的现象时有发生, 因而也极大地影响神经网络的归纳性能. 为了改进网络的全局搜索性能、加快学习速度和增强归纳性能, 本文提出了一种综合采用相关有效算法的新启迪方法——自顶向下优化神经网络方法, 并从理论基础、方法研究、实验仿真与结果讨论等方面进行论述.

1 理论基础

反向传播算法运用迭代梯度下降法来搜索误差曲线上的最小值点, 由此来最小化期望输出与实际输出的均方误差. BP 算法由通过不同网络层的两个过程组成: 前向传播过程和反向传播过程. 假设拥有 L 层的前馈神经网络有 m_1 个输入神经元和 m_L 个输出神经元, $e_k(n)$ 代表第 k 个输出神经元在第 n 个样本训练时的输出误差, 定义第 n 个样本训练时的均方误差为

$$E(n) = \frac{1}{2} \sum_{k=1}^{m_L} e_k^2(n). \quad (1)$$

在 BP 算法的单序列模式学习过程中, 网络中的各连接权因输入样本的变化而调整, 连接权改变值与 $E(n)$ 的偏导数成正比关系. 按梯度下降原则, 第 k 个输出神经元的局部梯度 $\delta_k(n)$ 为

$$\delta_k(n) = e_k(n) f'(v_k(n)), \quad (2)$$

式 (2) 中, $v_k(n)$ 代表第 k 个输出神经元在第 n 个样本训练时的局部域值, $f(\cdot)$ 代表 S 形激活函数. 按梯度下降原则, 最后一隐含层第 j 个输出神经元的局部梯度 $\delta_j(n)$ 为

$$\delta_j(n) = f'(v_j(n)) \sum_{k=1}^{m_L} \delta_k(n) w_{kj}(n). \quad (3)$$

依据增量规则, 连接权值调整为

$$w_{ji}(n+1) = w_{ji}(n) + \alpha w_{ji}(n-1) + \eta \delta_j(n) y_i(n). \quad (4)$$

相关偏置值调整为

$$b_j(n+1) = b_j(n) + \eta \delta_j(n). \quad (5)$$

式 (4)、式 (5) 中, $w_{kj}(n)$ 代表在第 n 个样本训练时, 连接第 k 个神经元与第 j 个神经元的权值; $w_{ji}(n)$ 代表在第 n 个样本训练时, 连接第 j 个神经元与第 i 个神经元的权值; $y_i(n)$ 代表第 i 个神经元在第 n 个样本训练时的输出值; $b_j(n)$ 代表第 j 个神经元在第 n 个样本训练时的偏置值; α 代表动量因子; η 代表学习率.

在 BP 算法的批处理模式学习过程中, 网络中的各连接权仅在所有输入样本完成一回合的学习后才进行调整. 在这里, N 代表训练样本的数量, 定义经过一回合学习后的均方误差为

$$E_{av}(T) = \frac{1}{N} \sum_{n=1}^N E(n) = \frac{1}{2N} \sum_{n=1}^N \sum_{k=1}^{m_L} e_k^2(n) \quad (6)$$

依据增量规则, 连接权值调整为

$$w_{ji}(T+1) = w_{ji}(T) + \alpha w_{ji}(T-1) + \frac{\eta}{N} \sum_{n=1}^N \delta_j(n) y_i(n) \quad (7)$$

相关偏置值调整为

$$b_j(T+1) = b_j(T) + \frac{\eta}{N} \sum_{n=1}^N \delta_j(n). \quad (8)$$

式(7)、式(8)中 $w_{ji}(T)$ 代表在第 T 回合训练时, 连接第 j 个神经元与第 i 个神经元的权值; $b_j(T)$ 代表第 j 个神经元在第 T 回合训练时的偏置值. 当 $T=1$ 时, $w_{ji}(0)$ 与 $w_{ji}(1)$ 分别为小的随机值.

遵循神经中枢手术优化(OBS)算法的思想, 快速自顶向下优化神经网络结构(FOBS)算法^[14]根据误差函数的二阶导数结果, 从“平衡网络复杂性与训练误差之间的性能关系”的基本思想出发, 然后对网络结构进行优化. 在 OBS 算法中, L_i 是“ w_i 为特殊权”下的极值,

$$L_i = \frac{w_i^2}{2[\mathbf{H}^{-1}]_{ii}}. \quad (9)$$

事实上, L_i 极值表示除去 w_i 特殊权后所引起的均方误差增量; $[\mathbf{H}^{-1}]_{ii}$ 表示第 i 个元素的赫森矩阵的逆矩阵. 由于极值 L_i 与 w_i^2 成正比关系, 因此, 越小的权值对均方误差的影响就越小. 这样与最小极值 L_i 相关的特殊权 w_i 就是将除去的连接权.

2 自顶向下优化启迪方法

自顶向下优化启迪方法首先通过快速自顶向下优化神经网络结构算法自动获得优化结构的神经网络, 然后经由动态优化学习参数算法来动态优化学习参数, 接着采用快速交叉校验(FCV)算法^[15]有效防止过度适应问题.

2.1 快速自顶向下优化神经网络结构算法

在众多的自顶向下方方法中, 神经中枢手术优化算法出类拔萃. 在 OBS 算法中, 计算赫森矩阵的逆矩阵 \mathbf{H}^{-1} 是个关键问题. 假设网络已经训练并趋于局部极小点, 运用求逆矩阵方法, 对于只有单个输出神经元的情况, 得出求赫森逆矩阵的迭代公式

$$\mathbf{H}^{-1}(n) = \mathbf{H}^{-1}(n-1) - \frac{\mathbf{H}^{-1}(n-1)\mathbf{f}(n)\mathbf{f}^T(n)\mathbf{H}^{-1}(n-1)}{1 + \mathbf{f}^T(n)\mathbf{H}^{-1}(n-1)\mathbf{f}(n)}, \quad (10)$$

$$\mathbf{f}(n) = \frac{1}{\sqrt{N}} \frac{\partial F(\mathbf{W}, \mathbf{X}(n))}{\partial \mathbf{W}}. \quad (11)$$

同理, 对于多个输出神经元的情况, 有

$$\begin{aligned} \mathbf{H}_{k+1}^{-1}(n) &= \mathbf{H}_k^{-1}(n) - \frac{\mathbf{H}_k^{-1}(n)\mathbf{f}_{k+1}(n)\mathbf{f}_{k+1}^T(n)\mathbf{H}_k^{-1}(n)}{1 + \mathbf{f}_{k+1}^T(n)\mathbf{H}_k^{-1}(n)\mathbf{f}_{k+1}(n)}, \\ \mathbf{H}_k^{-1}(n) &= \mathbf{H}_{m_L}^{-1}(n-1) - \frac{\mathbf{H}_{m_L}^{-1}(n-1)\mathbf{f}_k(n)\mathbf{f}_k^T(n)\mathbf{H}_{m_L}^{-1}(n-1)}{1 + \mathbf{f}_k^T(n)\mathbf{H}_{m_L}^{-1}(n-1)\mathbf{f}_k(n)}, \end{aligned} \quad (12)$$

$$\mathbf{f}_{k+1}(n) = \frac{1}{\sqrt{N}} \frac{\partial F_{k+1}(\mathbf{W}, \mathbf{X}(n))}{\partial \mathbf{W}}, \quad (13)$$

$$\mathbf{f}_k(n) = \frac{1}{\sqrt{N}} \frac{\partial F_k(\mathbf{W}, \mathbf{X}(n))}{\partial \mathbf{W}}, \quad (14)$$

其中, \mathbf{W} 为权向量; $\mathbf{X}(n)$ 为第 n 个样本训练时的输入向量; F 为输入-输出映射函数. 后面将用到的 \mathbf{I}_i 表示除了第 i 个元素为 1 而其它元素全为零的特殊单位向量矩阵.

在 OBS 算法中, 正确选择初始网络与优化学习, 改进归纳性息息相关. 对于初始网络, 可以

有无数选择,但其中仅有少许适合实际问题.与所给问题的复杂性相比较而言,若所选择的初始网络过小,则所得网络的归纳性将很差;然而,如果所选的初始网络过大,又会浪费资源和时间.一旦能选取适当的初始网络,那么将促使所得网络更简单、学习速度更快和归纳性更好.虽然具有 S 形函数的非线性三层神经网络结构可以逼近任何的非线性函数;但是,在这种只有单层隐含层的神经网络里,各神经元相互影响.面对复杂情况,这种相互影响成为改进函数逼近性的障碍;可是,拥有两层隐含层的神经网络更易于改进函数逼近性. Blum 和 Li^[16] 实验证明:拥有两层隐含层的神经网络,具有全局函数逼近性.由此,在实验仿真过程中,只针对四层神经网络结构进行研究,并特定所有的隐含神经元使用 S 形函数.快速 OBS 算法起始于适当复杂的初始网络,该初始网络有如下结构:拥有 m_1 个输入神经元的输入层,其中 m_1 视实际情况而定;拥有 $H1N$ 个隐含神经元的第一隐含层,其中 $H1N = 2^{\lceil \log_2(m_1 + 1) \rceil} + (2^{\lceil \log_2(m_1 + 1) \rceil} - 2)D_1 + 1$, D_1 代表输入变量的维数;拥有 $H2N$ 个隐含神经元的第二隐含层,其中 $H2N = 2^{\lceil \log_2(m_1 + 1) \rceil} m_1 + 1$;输出层有 m_4 个输出神经元,其中 m_4 由实际情况定值.

快速 OBS 算法的概要为:

步骤 1 训练适当复杂的初始神经网络,使均方误差趋于局部极小值.

步骤 2 计算 H^{-1} .

步骤 3 计算查找与最小极值 L_i 相关的特殊权 w_i .若极值 L_i 远小于均方误差 E_{avg} ,则除去特殊权 w_i ,然后执行步骤 4;否则执行步骤 5.

步骤 4 根据公式: $\Delta W = \frac{w_i}{[H^{-1}]_{ii}} H^{-1} \cdot l_i^T$, 对网络中所有权值进行调整,然后执行步骤

2

步骤 5 当网络中没有可除去的特殊权时,终止该过程.

2.2 动态优化学习参数算法

BP 算法运用梯度下降法来搜索误差曲面上的最小值点,同时根据搜索结果不断调整相关权值并记忆.然而,常规 BP 算法的收敛速度极其缓慢,其中缘由是神经元所采用的激活函数具有饱和属性.一旦输出神经元处于饱和区域,即使输出误差再大,但其相关递减梯度的取值还是非常小.在这种情况下,如果仍然采用一成不变的学习参数,那么相关权值的调整尺度会很小.在误差曲面上存在许多局部极小点,若不能采取有效方法,则容易陷入局部极小点而错失全局最小点,从而使神经网络不能全局收敛而减弱归纳性能.

学习参数的取值因时而异.当迭代点处于陡峭的碗状误差曲面上时,为了防止权向量的过度调整而偏离准性,这时应保持小的学习率和相配的动量因子;但学习进程在平缓的误差曲面区域时,为了加快学习速度而增大梯度,因此采用较大的学习率和相一致的动量因子.

优化学习参数算法的概要为

步骤 1 在批处理模式学习下,运用 BP 算法计算经过 P 回合学习后的平均均方误差 E_{avg}

$$E_{avg} = \frac{1}{P} \sum_{t=1}^P E_{avg}(T), \quad (15)$$

步骤 2 在单序列模式学习下,基于 BP 算法的前向传播过程,利用公式 (1) 计算第 n 个样本学习后的均方误差 $E(n)$.

步骤 3 计算学习率的估计敏感度 $S(n)$,

$$S(n) = |E(n) - E_{avg}|, \quad (16)$$

步骤 4 计算第 n 个样本学习时的优化学习率 $\eta(n)$,

$$\eta(n) = \eta(0) / [1 + b e^{-s(n)}]. \quad (17)$$

步骤 5 计算第 n 个样本学习时的优化动量因子 $\alpha(n)$,

$$\alpha(n) = \alpha(0) + c[\eta(0) - \eta(n)]. \quad (18)$$

在以上的算法中, b, c 分别代表协调参数, 取值由实际研究对象而定.

2.3 快速交叉校验算法

交叉校验 (CV) 方法能有效防止过多学习所引起的过度适应问题. 交叉校验算法在学习过程中, 由比较与评估候选模型来选择最优模型.

首先用 BP 算法对候选模型进行训练评估, 然后将已训练的神经网络进行校验, 目标是选取具有最小归纳性误差的网络优化模型. 为了实现该目标, 训练样本数据集被随机地分割为两个相互独立子集: 评估子集和校验子集. 在此, 用 D 代表整个训练样本数据集, T^m 代表第 m 轮训练时的评估子集, V^m 代表第 m 轮训练时的校验子集. 假设训练样本数据集包含 N 种样本, r 是决定训练样本数据集 D 分割的参数 ($0 < r < 1$). 这样, $|T^m| = [(1-r)N]$, $|V^m| = [rN]$. 它们满足

$$(\forall m) \quad (1 \leq m \leq t) \quad T^m \subset D \wedge V^m \subset D, \quad (19)$$

$$D = T^m \cup V^m, \quad (20)$$

$$T^m \cap V^m = \phi. \quad (21)$$

在这里, t 代表整个训练次数. 每一个评估子集 T^m 进一步分割成 5 个独立子集 T_i^m , 关系为

$$(\forall i) \quad (1 \leq i \leq 5) \quad T_i^m \subset T^m, \quad (22)$$

$$T^m = \bigcup_{i=1}^5 T_i^m, \quad (23)$$

$$(\forall i, j) \quad (1 \leq i, j \leq 5 \wedge i \neq j) \quad T_i^m \subset T^m \wedge T_j^m \subset T^m \wedge T_i^m \cap T_j^m = \phi. \quad (24)$$

同样, 每一个校验子集 V^m 进一步分割为 2 个独立子集 V_q^m , 关系为

$$(\forall q) \quad (1 \leq q \leq 2) \quad V_q^m \subset V^m, \quad (25)$$

$$V^m = \bigcup_{q=1}^2 V_q^m, \quad (26)$$

$$(\forall q, s) \quad (1 \leq q, s \leq 2 \wedge q \neq s) \quad V_q^m \subset V^m \wedge V_s^m \subset V^m \wedge V_q^m \cap V_s^m = \phi. \quad (27)$$

衡量 A_l 网络模型性能好坏的标准是依据其平均归纳性误差的大小, 平均归纳性误差 E_{av} 有以下等式:

$$E_{av}(A_l^m) = \frac{1}{10} \sum_{l=1}^2 \sum_{i=1}^5 E_{qi}^m \quad (28)$$

其中, E_{qi}^m 表示 A_l 网络模型在第 m 轮训练时, 用评估子集 T_i^m 训练后, 再经校验子集 V_q^m 进行校验所得的均方差.

常规 CV 算法的主要问题是候选模型范围过大. 由于资源和时间的因素以及梯度递减算法固有缺陷的限制, 不得不考虑候选模型的范围. 为了使 CV 算法更直接和更高效, 笔者提出了定义候选模型适当范围的方法. 快速 CV 算法的概要为

步骤 1 确定初始神经网络与明确候选模型的范围; 将初始神经网络作为当前候选模型.

步骤 2 用评估子集 T_i^m 训练当前候选模型, 然后经校验子集 V_q^m 进行校验并计算平均归纳性误差 E_{av} .

步骤 3 用评估子集 T_i^m 训练新候选模型, 然后经校验子集 V_q^m 进行校验并计算平均归纳性误差 E_{av} .

步骤 4 比较当前候选模型与新候选模型的平均归纳性误差 E_{avg} 的大小, 将 E_{avg} 为小的候选模型保存并作为当前候选模型.

步骤 5 在确定的范围内是否仍有新的候选模型? 若有, 则返回执行步骤 3; 否则终止, 所得的当前候选模型就是要选定的优化网络结构.

3 仿真实验

在实际问题的开始, 最优网络是个未知数. 笔者的目标是获得结构简单、学习快速和归纳性能良好的神经网络. 在此, 主要用归纳性误差作为衡量网络性能优劣的标准.

为了考察自顶向下优化神经网络启迪方法的性能, 选用在深圳证券市场交易的公司股票为研究对象. 数据来源于各大公司在深圳证券市场交易的原始数据, 数据记录时间为: 从 1991 年 1 月 2 日至 2004 年 8 月 20 日. 实验对象是股票涨或跌的问题, 目标是分清股票的走势. 在仿真实验中, 用“1”表示涨势, 用“0”表示跌势. 实验对象有八个真值输入神经元 (OPF 当日开盘价; HPT 当日最高价; LPT 当日最低价; CPT 当日收盘价; TMT 当日成交总金额; TVT 当日总成交量; CIT 当日综合指数收盘指数; CMT 当日深市成交总金额.) 和一个输出神经元 (SCT 明日走势).

神经网络在采用样本学习之前, 首先进行权值和偏置值的初始化, 其次经由 FOBS 算法优化网络结构, 由此得到优化网络结构 (输入神经元数目—第一隐含层神经元数目—第二隐含层神经元数目—输出神经元数目) 为: 8—17—12—1; 然后运用动态优化学习参数算法分别对网络的学习率和动量因子优化, 在仿真实验中, 协调参数 b 与 c 的取值为: $b = 0.16$ $c = 0.0003$. 随后在优化神经网络结构和优化学习参数的基础上, 选用从 2001 年 7 月至 2004 年 3 月的原始数据作为实验数据, 针对每个公司所用的样本集包含 600 种样本, r 的取值范围为: $0.15 \leq r \leq 0.3$. 在这里, 评估子集 T^m 包含 500 种样本, 校验子集 V^m 包含 100 种样本. 根据实际情况, 实验选定的候选模型范围: 学习全局均方误差 $E_{avg} < 0.126$ 的神经网络, 最后运用 FCV 算法选定优化神经网络. 在实验过程中, 每次仿真实验都进行 10 次.

通过仿真实验获得启迪方法的相关实验数据, 并将这些数据与常规 OBS 算法和 FOBS 算法进行比较. 实验所得结果见图 1 和表 1. 结果表明, 启迪方法有优化的网络结构、网络收敛速度更快、归纳性能更好的优良性能.

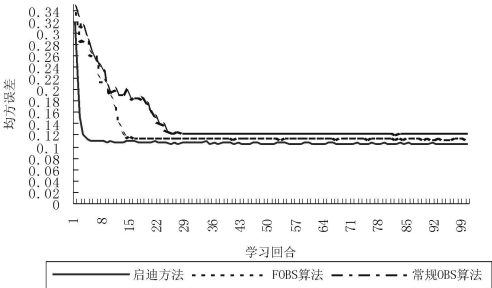


图 1 均方误差对照图

表 1 几种算法的性能参数对照表

问题	方法	优化网络结构	E_{avg}	T_{avg}
股票	OBS 算法	8—17—12—1	0.1282	30
涨或	FOBS 算法	8—17—12—1	0.1206	16
跌	启迪方法	8—17—12—1	0.1137	5

E_{avg} : 平均均方误差, T_{avg} : 平均学习的回合

4 结论

本文对神经网络优化算法进行研究, 提出和阐述综合采用相关优化算法的启迪方法. 实验结果显示, 相关优化算法的综合采用, 令自顶向下优化神经网络方法有效加快网络学习和提高

网络归纳性能. 在解决非线性、非确定的复杂问题时, 虽然神经网络的学习时间、收敛性和归纳性由许多因素决定, 但其主要因素为: (1) 输入的正常化与学习参数的初始化; (2) 神经网络结构的优化; (3) 学习参数的优化; (4) 尽早结束学习. 若所采用的方法缺乏系统性, 则难于极大地提高神经网络性能. 因此, 未来的研究重点是系统、全面地探究一系列相关优化算法, 从而提出极其有效的综合方法.

参考文献:

- [1] Haykin S. Neural Networks: A Comprehensive Foundation[M]. New York: Macmillan, 1994.
- [2] Fletcher R. Practical Methods of Optimization[M]. New York: Wiley, 1990.
- [3] 史忠科. 神经网络控制理论[M]. 西安: 西北工业大学出版社, 1997.
- [4] Hagan M T, Demuth H B, Beale M H. 神经网络设计[M]. 戴葵等译. 北京: 机械工业出版社, 2002.
- [5] Rumelhart D E, Hinton G E, Williams R J. Learning internal representations by error propagation[C] // Rumelhart D E, McClelland J L, Eds. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Cambridge, MA: MIT Press, 1986, 1: 318-362.
- [6] Kung S, Fallside F, Sorenson J A, et al. Neural networks for signal processing[J]. Proceeding of the 1992 IEEE Workshop, 1992, 2: 255-266.
- [7] Guyon I, Wang P S P. Special issue on neural networks and pattern recognition[J]. Pattern Recognition Artificial Intelligence, 1993, 7(4): 849-872.
- [8] Friedman J H. An Overview of Predictive Learning and Function Approximation[C] // Friedman J H, Wechsler H, Eds. From Statistics to Neural Networks: Theory and Pattern Recognition Applications. Proceeding of the ASI Conference, Subseries F. New York: Springer-Verlag, 1994.
- [9] Moody J. Prediction Risk and Architecture Selection for Neural Networks[C] // Cherkassky V, Friedman J, Wechsler H, Eds. From Statistics to Neural Networks: Theory and Pattern Recognition Applications. NATO ASI Series F. New York: Springer-Verlag, 1994, 136: 147-165.
- [10] Fahman S E, Lebiere C. The cascade-correlation learning architecture[C] // Touretzky D S, Ed. Advances in Neural Information Processing Systems. San Mateo, CA: Morgan Kaufmann, 1990, 2: 524-532.
- [11] Hassibi B, Stork D, Wolff G. Optimal brain surgeon and general network pruning[J]. IEEE International Conference on Neural Networks, 1993, 1: 293-299.
- [12] Holland J H. Adaptation in Natural and Artificial Systems[M]. Ann Arbor: MI University of Michigan Press, 1975.
- [13] Xiao Yu-Hu, Guo Chen-An, Xin Cheng-Shi. Dynamic learning rate optimization of the backpropagation algorithm[J]. IEEE Transaction on Neural Networks, 1995, 6(3): 669-677.
- [14] 杨钟瑾, 史忠科. 快速自顶向下优化神经网络结构的方法[J]. 系统仿真学报, 2005, 17(1): 162-165.
- [15] 杨钟瑾, 史忠科. 快速选定神经网络优化结构的方法[J]. 计算机科学, 2005, 32(4): 81-83.
- [16] Blum E K, Li L K. Approximation theory and feedforward networks[J]. Neural Net, 1991(4): 511-515.

(下转第 107 页)

- [4] 缪淮扣, 高晓雷, 李刚. 结构化方法、面向对象方法和形式方法的比较与结合 [J]. 计算机工程和科学, 1999, 21(4): 28-29.
- [5] 朱涛. 软件开发中两种常用方法的分析与研究 [J]. 固原师范学院学报 (自然科学版), 2005, 26(3): 39-41.
- [6] 张莉, 裴国永. 结构化方法与面向对象方法的比较分析 [J]. 陕西师范大学学报 (自然科学版), 2001, 29(2): 29-30.
- [7] 李红梅, 韩逢庆, 纪钢. 程序设计中结构化方法与 OO 方法的结合 [J]. 软件技术, 2002, 21(1): 31-32.
- [8] 贺启宝, 王军. 面向对象方法与结构化方法 [J]. 佛山科学技术学院学报 (自然科学版), 2004, 22(4): 45.
- [9] 严静, 周立新, 潘云鹤. 面向对象方法与结构化方法相结合的 MIS 系统开发技术 [J]. 计算机应用研究, 1999 (5): 25-26.
- [10] 杨文龙, 姚淑珍, 吴芸. 软件工程 [M]. 北京: 电子工业出版社, 1999.

The Combination of Structured Method and Object-oriented Method

CHENG Peng LI Dai-ping YANG Wen-wei

(Faculty of Computer Guangdong University of Technology, Guangzhou 510090, China)

Abstract Structured method and object-oriented method are the most popular software development methods today. This paper compares them with examples, discusses the characteristics and disadvantages which mainly exist in maintenance of structured method and cognition of object-oriented method, then analyzes the feasibility of integrating them into one to develop software, and gives the conclusion that integrated method can take greater effect than single method in certain situation.

Key words structured method, object-oriented method, combination

A Heuristic Approach for Improving Performance of Neural Network with the Optimal Brain Surgeon

YANG Zhong-jin

(School of Information Science and Technology, Guangdong University of Business Studies, Guangzhou 510320, China)

Abstract A new heuristic approach is proposed for the automated optimization of neural network. It combines fast optimal brain surgeon, dynamic optimization of learning parameters and fast cross-validation. The fast optimal brain surgeon optimizes architecture of neural network automatically. The dynamic optimization of learning parameters is able to dynamically vary its learning parameters and select optimal learning parameters. The fast cross-validation provides improvements of avoiding overfitting problem effectively. Experimental results demonstrate that with the heuristic approach, considerable performance gains are obtained compared to the other algorithms. This includes better generalization, optimal network architecture, and faster learning.

Key words neural network, optimal brain surgeon, top-down, learning parameters optimization, cross-validation, generalization