

有限精度的 BP 神经网络高速算法*

罗莉 胡守仁

(国防科技大学计算机学院 长沙 410073)

摘要 精度要求是神经网络硬件实现的一个关键问题。本文提出了一种有限精度的 BP 高速算法, 通过实例验证, 该算法适于低精度, 且收敛速度快。

关键词 BP 学习算法, 代价函数, 活跃函数, 有限精度计算, 神经网络硬件

分类号 TP18

A High Speed BP Algorithm for the Feed-forward Multilayer

Neural Network Using Limited Precision Calculation

Luo Li Hu Shouren

(Department of Computer Science, NUDT, Changsha, 410073)

Abstract The precision required for neural network algorithm is an important question for hardware architects. This paper gives an excellent high speed BP algorithm for feed-forward multilayer neural network using limited precision calculation. The experimental results show that the algorithm improves convergence speed with reduced precision bits.

Key words back-propagation, feed-forward multilayer neural network, cost function, activation function, limited precision calculation, neural network hardware

精度要求是神经网络硬件实现的一个关键问题。目前通用数字计算机上的神经网络仿真需 32 位浮点或 64 位双精度环境下进行, 而现有的 VLSI 模数或全数字的神经网络集成电路很难做到中等精度以上, 高精度的实现需更复杂电路和更大的芯片面积, 经济、工艺也不容许。通常网络模型学习算法的低精度实现已失去了原有的一些性能(如收敛性、准确性)。研究有限精度的神经网络计算意义十分重大, 因为设计高速神经网络硬件时可以设计更多的神经元和连接权值, 使网络的速度、容量与成本之比大大提高。

BP 网络是应用最广泛的一种人工神经网络, 在各门学科领域中都具有很重要的实用意义, 同时网络结构直观, 很适于用 VLSI 技术实现大规模神经网络专用芯片。本文提出了一种有限精度的高速 BP 算法, 在 XOR 问题试验, 迭代次数 45 次, 12 位定点计算误差为 $1.5E-3$ 。应用于故障诊断、数字识别问题, 学习算法鲁棒性也较强, 对运算精度要求较低, 是一种优质高效的学习算法, 算法的硬件实现具有较好性能价格比。

1 有限精度的 BP 算法

有限精度的神经网络计算, 数值的表示是 N 位定点形式, 1 位符号位, Q 位整数 ($Q \leq N-1$), S 位小数部分 ($S = N-1-Q$), 数值的范围是 $-2^Q \sim 2^Q - 2^{-S}$, 产生的计算误差有两种情况:

上溢: 大于 $2^Q - 2^{-S}$ 的值四舍五入为 $2^Q - 2^{-S}$, 小于 -2^Q 的值四舍五入为 -2^Q 。

下溢: 绝对值低于 2^{-S-1} 的值四舍五入为 0。

* 1997 年 3 月 28 日收稿

第一作者: 罗莉, 女, 1971 年生, 博士

在神经计算中, 上溢误差的忽略对结果影响并不大, 因为 sigmoid 函数的饱和特性; 但下溢误差的忽略产生影响较大。我们通过对传统方法的改进, 消去有限精度在算法中的不良影响, 同时加快算法的收敛速度, 采取了下述的五种措施。

(1) 在传统 BP 算法, 根据 Robert 的论证^[3], 一个 BP 三层网络可以实现任意精度, 近似任何连续函数, 取三层前馈网络, 即输出层、隐含层、和输入层, 设输出层第 k ($k=1, K$) 节点, 隐层第 j ($j=1, J$) 节点, 输入层第 i ($i=1, I$) 节点, 第 p ($p=1, M$) 组样本的输入、输出分别 $net_k^p, net_j^p, net_i^p = O_i$, 和 O_k^p, O_j^p, O_i^p 表示, 则:

$$\begin{cases} net_k^p = \sum_{j=1}^{J+1} w_{kj} \times o_j^p, & o_{j+1}^p = 1, \quad w_{k, J+1} = \theta_k \\ net_j^p = \sum_{i=1}^{I+1} w_{ji} \times o_i^p, & o_{i+1}^p = 1, \quad w_{j, I+1} = \theta_j \\ o_k^p = f(net_k^p), & o_j^p = f(net_j^p) \end{cases} \quad (1)$$

式中 w_{kj}, w_{ji} 为相邻两层神经元的连接权值, θ_k 代表第 k 节点阈值, θ_j 代表第 j 节点阈值, $f(\cdot)$ 为神经元的激活函数, $f(\cdot)$ 取 Sigmoid 函数。描述 BP 网络映射的逼近程度, 用代价函数 $E(\cdot)$ 表示, 常取误差平方和, 即

$$E = \frac{1}{2} \sum_{p=1}^M \sum_{k=1}^K (t_k^p - o_k^p)^2 \quad (2)$$

式中 t_k^p 为相应的期望输出值。BP 是有指导的学习, 学习的实质是不断修改权值, 使代价函数趋于零, 即 $t_k^p \approx o_k^p$ 取最陡下降法, 即

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}}, \Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \quad (3)$$

比例常数 η 称为步长或叫学习速率, 不难导出,

$$\begin{cases} -\frac{\partial E}{\partial w_{kj}} = \sum_{p=1}^M \delta_k^p o_j^p, & \delta_k^p = (t_k^p - o_k^p) \times o_k^p \times (1 - o_k^p) \\ -\frac{\partial E}{\partial w_{ji}} = \sum_{p=1}^M \delta_j^p o_i^p, & \delta_j^p = \left(\sum_{k=1}^K \delta_k^p \times w_{kj} \right) \times o_j^p \times (1 - o_j^p) \end{cases} \quad (4)$$

式中 δ_k^p, δ_j^p 称为等效误差分量。权值的修改过程是一个迭代过程, 即

$$\begin{cases} w_{kj}(n+1) = w_{kj}(n) + \eta \sum_{p=1}^M \delta_k^p \times o_j^p \\ w_{ji}(n+1) = w_{ji}(n) + \eta \sum_{p=1}^M \delta_j^p \times o_i^p \end{cases} \quad (5)$$

在公式 (4) 的计算中, 当 o_j^p 为 1 或 0, δ_j^p 的值为 0, 造成“假饱和”现象, 低精度度运算当下溢时, o_j^p 四舍五入为 0, 从而, 权值几乎不修改, 这导致了学习的不收敛性。

对 BP 算法进行修改, 改变代价函数的定义, 降低下溢误差的影响。因为是 BP 网络, 误差逆向传播, $\delta_j^p = \left(\sum_{k=1}^K \delta_k^p \times w_{kj} \right) \times o_j^p \times (1 - o_j^p)$, δ_k^p 与 δ_j^p 两者相比较, δ_k^p 的影响是更重要。基于上述两点的分析, 选取“信息熵”作为代价函数的定义, 用下述 (6) 代替 (2)。

$$E_H = - \sum_{p=1}^M \sum_{k=1}^K [t_k^p \times \log o_k^p + (1 - t_k^p) \times \log(1 - o_k^p)] \quad (6)$$

为什么作这样的选择? 因为 $\log 1 = 0$, (6) 式右边一、二两项分别对应于期望输出值 $t_k^p = 1$ 和 $t_k^p = 0$ 的实际输出值的贡献。若输出层的激活函数不变, 则 δ_k^p 大为简化了。

$$\delta_k^p = t_k^p - o_k^p \quad (7)$$

曾设想, 改变隐含层的激活函数, 但效果不佳。故所有神经元的激活函数仍取 Sigmoid 函数。因此 (2. 4) 式的第一式由 (7) 式代替, 而第二式保持不变。

(2) 为了抑制迭代过程的振荡, 根据 Rumelhart 的建议^[4], 在迭代公式 (5) 的右边增加一 momentum 项, 改善收敛, 即

$$a\Delta w_{kj}(n), \quad a\Delta v_j(n)$$

式中 a 为动量项常数, $0 \leq a \leq 1$ 。

(3) 考虑到 $\Delta E_n = E_{n+1} - E_n < 0$ 时, 为了加速收敛, 步长 η 大一点有利; 相反 $\Delta E_n > 0$ 时, η 适当减小一点。为此, 采取变学习速率法, 即

$$\begin{cases} \text{if } \Delta E_n < 0, & \text{then } \eta = \eta_1, \quad a = a_1 \\ & \text{else } \eta = \eta_2, \quad a = 0 \end{cases} \quad (8)$$

式中 η_1, η_2 和 a 是常数, 允许 η 可以大于 1。

(4) 过去 Caillon 等人强调限制连接权重的范围, 避免学习过程中的振荡^[8]。文献中一般指出, 权重初值在 $[-0.5, 0.5]$ 区间内随机选取, 我们的看法与此恰好相反, 权重初值随机设定范围太窄, 都在零值附近更易引起振荡或停止不前。我们根据对权值分布的了解, 采取在 $[-5, 5]$ 区间内随机地设定权的初值。试验证明, 这样做是有利加速收敛。

(5) 为防止或减少迭代初期的振荡, 对输入值 $o_i^l = \{0, 1\}$ 采取“两步到位”, 即初期让 $0 \rightarrow 0.2, 1 \rightarrow 0.8$, 迭代一定次数后, 代价函数降低到一定值, 再恢复 $0.2 \rightarrow 0, 0.8 \rightarrow 1$ 进行迭代, 至满足精度要求为止。

2 实验结果

XOR 问题是经典的前馈网络计算问题。建立三层前馈网络 2-2-1 结构, 2 个输入层神经元, 2 个隐含层神经元, 1 个输出层神经元, 误差以平方和定义。12 位定点计算, 迭代次数 45 次, 误差为 $1.5E-3$; 13 位定点计算, 迭代次数 45 次, 误差为 $4E-5$; 16 位定点计算, 迭代 48 次, 误差为 $3.1E-10$ 。浮点计算 (32 位), 迭代 40 次, 误差为 $6.19E-5$; 迭代 50 次, 误差为 $1.55E-38$ 。精度与迭代次数之间的关系如图 1, 精度与误差之间的关系如图 2。采用传统算法, 16 位定点计算精度很难收敛; 浮点计算, 迭代 2000 次, 误差为 0.001。

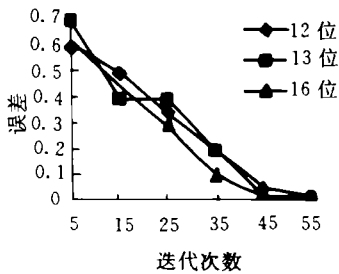


图 1 XOR 学习过程

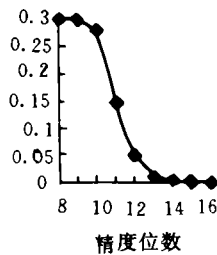


图 2 XOR 问题精度与误差之间的关系

所用定点数如附表定义:

附表

Q 位整数部分	S 位小数部分	N 位定点数
4	7	12
5	7	13
6	9	16

阿拉伯数字识别问题。建立三层前馈网络 88-16-10 结构。选择精度分别为 12 位、13 位、16 位。学习结果如图 3。16 位精度迭代 128 次, 13 位精度迭代 141 次, 12 位精度迭代 169 次, 达到误差为 0.1。采用传统算法, 16 位定点计算精度很难收敛; 浮点计算, 迭代 9000 次, 误差为 0.1。

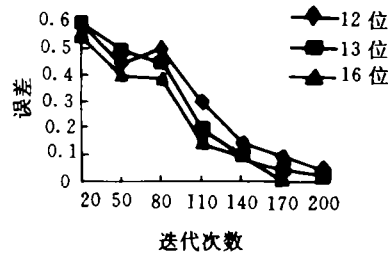


图3 数字识别学习过程

从图3上看到，算法收敛速度快，且对精度要求不高，12位精度就够了。

3 结论

提出了一种高速的BP有限精度算法，通过代价函数的重新定义减少了下溢误差的影响，加快了收敛速度，算法硬件实现更加有效：(1) 提高了模拟电路实现的灵活性，降低精度的收敛算法对模拟电路实现更加容易；(2) 减少了数字电路实现的芯片面积，数据宽度的降低，减少了权值RAM的容量要求，降低了通讯开销；(3) 增强了BP算法的适应性，BP算法可在较短的时间内学习新的样本。所用算法增强了硬件实现的灵活性，相比传统BP算法，较少的运算精度、较少的权值修改，较少的存储容量。

参考文献

- 1 郑君里，杨行。神经网络，北京：高等教育出版社，1992
- 2 张立明。神经网络的模型及其应用，上海：复旦大学出版社，1993
- 3 Robert Hecht-Nielsen. Theory of the Backpropagation Neural Network. IJCNN, 1989
- 4 Rumelhart D E. and Mclelland J L. Parallel Distributed Processing; Explorations in the Microstructure of Cognition. Bradford Books, MIT Press, 1986
- 5 Lehman A V. et al. Factors Influencing Learning in Backpropagation. Proceedings of IEEE Inter. conf. on Neural Networks, 1988
- 6 Le Cunn. Using Curvature Information to Improve Backpropagation. Neural Network Supplement, INNS Abstracts 1988
- 7 Stornetta W, Huberman B. An Improved Threelayer Backpropagation Algorithm. Proceedings of the IEEE First Inter. Conf. on Neural Networks, 1987
- 8 Cailon J. et. al. Constrained Backpropagation. Neural Network supplement; INNS Abstracts, 1988
- 9 Hush D, Salas J. Improving the Learning Rate of Back-propagation. Proceedings of the IEEE Inter. Conf. on Neural Networks, 1988
- 10 Sakauo, S Kohde, T. Reduction of Required Precision Bits for Back-Propagation Applied to Pattern Recognition. IEEE Trans. on Neural Networks, 1993, 42 (2)
- 11 Holt J and Huang Jeng-neng Finite Precision Error Analysis of Neural Network Hardware Implementations. IEEE Trans. on Computers, 1993, 42 (3)