

基于 $\Sigma - \Delta$ 调制的单比特非线性 BP 人工神经网络的硬件实现^①郭晓丹^② 孟 桥^③ 梁 勇

(东南大学射频与光电集成电路研究所)

(东南大学教育部射频与光电集成电路研究中心)

摘 要 提出了一种基于 $\Sigma - \Delta$ 调制(SDM)的单比特 BP 人工神经网络的硬件实现方法。设计了基于 $\Sigma - \Delta$ 单比特信号的非线性立方根运算单元,并以此为激活函数单元构建了 BP 人工神经网络,网络中各神经元的输入输出均为基于 $\Sigma - \Delta$ 调制的单比特信号。在此基础上实现 S 函数逼近和网络隐含层的非线性输出。同时采用低环路延时加法器、混合信号乘法器作为关键运算单元,减少了硬件消耗,提高了运算精度。最后在可编程门阵列(FPGA)上实现整个非线性 BP 人工神经网络,并通过函数逼近的实例验证了该网络的功能。

关键词 单比特,人工神经网络,非线性,BP 网络,Sigma-Delta,可编程门阵列(FPGA)

0 引 言

前馈型反向传播(back propagation, BP)人工神经网络能够实现非线性函数逼近,在模拟控制、智能预测等方面有着重要的应用。人工神经元和人工神经网络的电路实现方法可分为数字和模拟两类。模拟实现方式^[1,2]的优点是神经元电路简单,突触信息只要用一根数据线就可以传输,缺点是存储困难,对温度等外界环境的变化敏感,电路抗干扰性能差。与之相反,数字方式在权值存储和抗噪音干扰方面有着巨大的优势,大规模的计算电路都可以得以实现。

传统的 BP 人工神经网络的数字电路实现,大都采用多比特结构,即各神经元之间需要多位数据总线相连,同时神经元内部也需采用复杂的多位数字信号处理(DSP)运算单元。例如 Popescu 通过复杂可编程逻辑器件(CPLD)设计了 8 位的人工神经网络^[3]。Hariprasath 等也通过现场可编程门阵列(FPGA)实现了 2-2-1 多比特前馈型人工神经网络^[4]。但是这些多比特并行分布式结构实现的人工神经网络电路规模庞大,布线复杂,给多层网络的

实现带来很大的难度。单比特人工神经网络的提出在很大程度上缓解了这一矛盾,其无论是神经元之间还是神经元内部都采用一位二进制码来表示信息,因此网络的神经元之间也只需通过一根数据线相连。例如 Tomlinson 等人提出了一种基于统计脉冲串的人工神经网络结构^[5]。Maeda 等又研究了采用脉冲密度调制实现人工神经网络的硬件设计^[6]。但是这两种单比特计算方式运算精度有限,为此 Liang 等^[7-9]研究了运算精度较高的基于 $\Sigma - \Delta$ 调制(sigma-delta modulation, SDM)的单比特运算单元,并将其应用到了神经元中,以线性或硬极限函数作为激活函数实现了简单神经网络。同时文献[7-9]中的关键运算单元结构复杂,资源利用率低。本研究在此基础上提出了基于 $\Sigma - \Delta$ 调制的非线性运算单元,并以此为激活函数单元构建了 BP 人工神经网络。同时简化了基于 $\Sigma - \Delta$ 调制的关键运算单元结构,设计了包括低环路延时加法器、混合信号乘法器等模块。非线性运算单元利用可实现的立方根函数逼近了传统的 Log-Sigmoid 函数。整个 BP 神经网络在 FPGA 上得以实现。最后通过函数逼近验证了单比特神经网络的功能。

① 国家自然科学基金(61076118)资助项目。

② 男,1982 年生,博士生,研究方向:神经网络设计,混合信号电路设计,单比特信号处理等;E-mail:230099097@seu.edu.cn

③ 通讯作者 E-mail:mengqiao@seu.edu.cn

(收稿日期:2013-01-30)

1 基于 $\Sigma - \Delta$ 调制的单比特网络结构

目前能够实现将模拟输入或多比特输入转换为单比特码流的方式有多种,文献[10]指出了它们在神经网络应用中的优缺点。鉴于 $\Sigma - \Delta$ 调制技术的优势,本文则采用 $\Sigma - \Delta$ 调制器作为单比特信号的来源。 $\Sigma - \Delta$ 调制可以将双极性输入转换为二进制比特流,即利用逻辑‘1’表示一个正量,逻辑‘0’表示一个负量。 $\Sigma - \Delta$ 调制信号比特流输出如式

$$OUT = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} out(n) = In \quad (1)$$

所示,理论上可以无限近似输入的多比特信号或模拟信号^[11]。

可以证明,和普遍的随机逻辑调制相比, $\Sigma - \Delta$

调制比特流解调后模拟输出的方差是 $1/N^2$ 阶,而随机逻辑调制输出信号的方差为 $1/N$ 阶^[12]。在超采样率相同时, $\Sigma - \Delta$ 调制要比脉冲密度调制等有着更高的信噪比。由此可见在一定的精度要求下,整个神经网络所需的总字长也会更小,运算速度也会得到提高。

BP 神经网络是一种典型的前馈型网络,图 1 为本文提出的一个基于 $\Sigma - \Delta$ 调制比特流实现的 $1 - n - 1$ BP 人工神经网络系统框图。网络中的各神经元之间都是采用 $\Sigma - \Delta$ 调制比特流进行信息交流。而神经元本身则采用了 McCulloch 和 Pitt 提出的乘加模型。同时,根据 BP 网络的特点,隐含层中的神经元采用非线性激活函数单元。而输出层,则为线性函数。

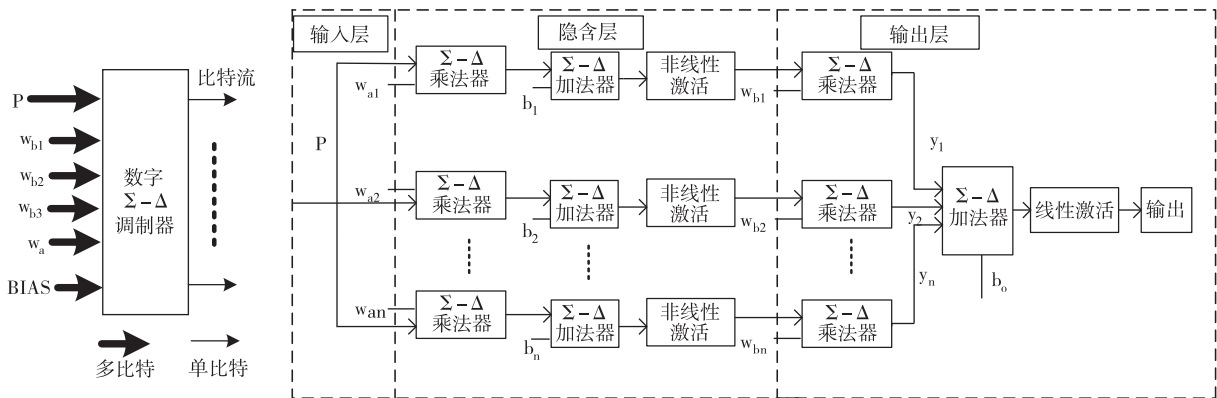


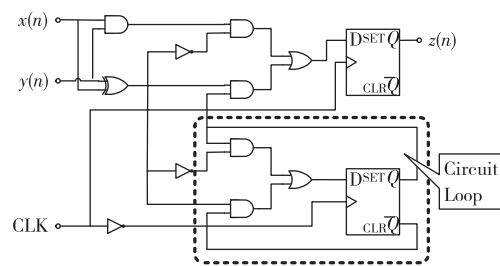
图 1 基于 $\Sigma - \Delta$ 调制的非线性 $1 - n - 1$ BP 人工神经网络结构框图

2 基于 $\Sigma - \Delta$ 调制的关键模块实现

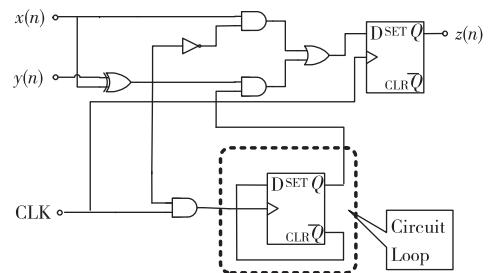
如何实现高精度的运算单元是系统实现的关键之一。Liang^[7,9]曾设计了加、乘运算单元,但它们结构复杂,运算速度不理想。本文将逐一简化结构,提高其运算速度和精度,并同时构建非线性激活单元。

2.1 低环路延时加法器

基于 $\Sigma - \Delta$ 比特流的加法运算单元有多种结构^[8,13-15]。在文献[7-9]中 Liang 采用的是文献[15]中的加法器结构,但此结构存在一个复杂的反馈环路,如图 2(a)所示。本文在不改变计算精度的情况下简化了环路结构,设计了一种低环路延时结构的比特流加法器,结构如图 2(b)所示,它能够很好地减少硬件消耗并提高运算速度。



(a) 复杂环路结构



(b) 低环路延时结构

图 2 回法器结构

从结构上分析,这两种加法器电路均能够实现如下的运算:

$$z(n) = \begin{cases} (x(n) + y(n))/2, & x(n) = y(n) \\ q(n), & x(n) \neq y(n) \end{cases}$$
$$q(n) = \begin{cases} q(n-1), & x(n) = y(n) \\ -q(n-1), & x(n) \neq y(n) \end{cases} \quad (2)$$

其中 $z(n)$ 是加法器的输出信号, $q(n)$ 代表了图中 D 触发器的输出。

为了验证模块的正确性,分别在两端输入 300kHz 和 500kHz 的正弦调制波。图 3 给出了图 2 加法器的输出频谱测试图,从中可以看出此加法器实现了比特流相加。

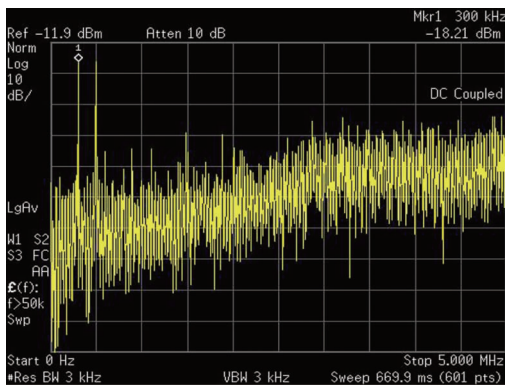


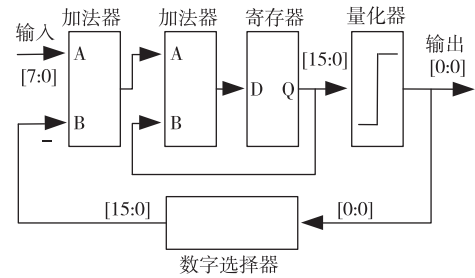
图 3 两种单比特加法器结构和输出频谱

2.2 数字 $\Sigma - \Delta$ 调制器

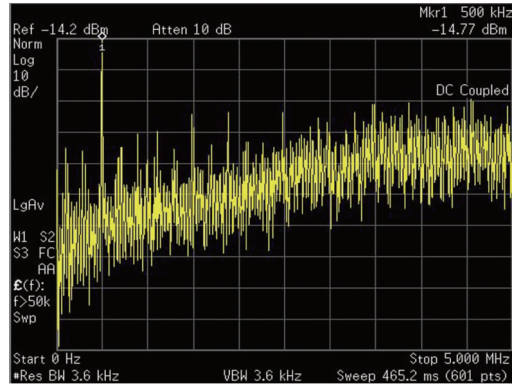
本文设计了一阶 $\Sigma - \Delta$ 调制器,其可以将 8 比特数据调制成单比特码流。由此输入和权值只需要 8 位寄存器就可以实现,而且只需要按照标准二进制方式修改权值。图 3(a) 给出了数字 $\Sigma - \Delta$ 调制器的框图。调制器输入幅度为 $[-127, 127]$, 输入的最高位为符号位。图 3(b) 给出了利用数字调制器输出 500kHz 正弦波的频谱测试波形。从频谱中可以明显看到 $\Sigma - \Delta$ 调制技术的特点,即量化噪声被从低频部分挤压到了高频部分。

2.3 混合信号乘法器

文献[15,16]提出一种 L 级结构乘法器(Liang 采用此结构),但其精度都不高。由于权值寄存器中为多比特信号,因此本文提出一种混合信号乘法器来实现权值相乘。其结构如图 4(a) 所示,其中乘法运算仅需采用一个选择开关实现,即利用 $X(n)$ 控制 $S_y(t)$ 的输入。其中数字 $\Sigma - \Delta$ 调制器(SDM)采用的是上文中的结构。

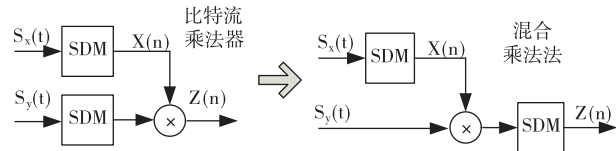


(a) 结构框图

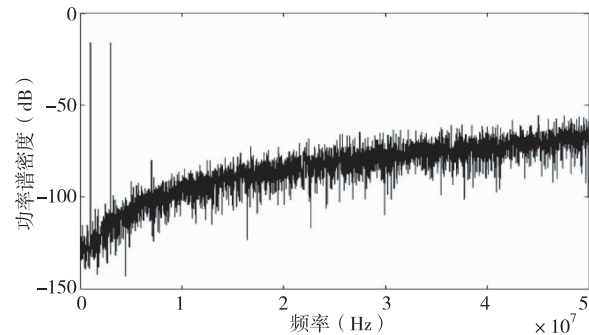


(b) 输出频谱

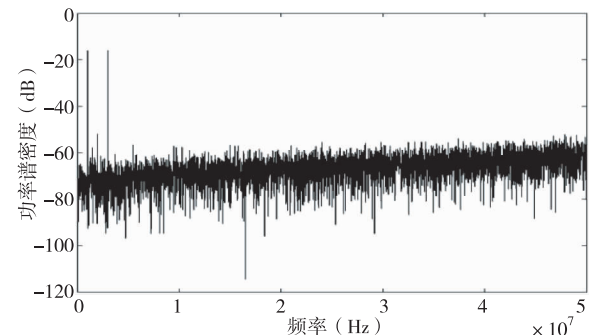
图 3 数字调制器结构框图和输出频谱



(a) 两种乘法器结构比较



(b) 混合乘法器的频谱



(c) L-level 乘法器的频谱^[15]

图 4 两种乘法器的比较

等式(3)和(4)分别给出了 L 级乘法器和本乘法器的输出功率:

$$\begin{aligned} Z_1(\omega) &= (S_x(\omega) + E_x(\omega)) * (S_y(\omega) + E_y(\omega)) \\ &= S_x(\omega) * S_y(\omega) + S_y(\omega) * E_x(\omega) \\ &\quad + S_x(\omega) * E_y(\omega) + E_x(\omega) * E_y(\omega) \end{aligned} \quad (3)$$

$$\begin{aligned} Z_2(\omega) &= (S_x(\omega) + E_x(\omega)) * S_y(\omega) + E_y(\omega) \\ &= S_x(\omega) * S_y(\omega) + E_x(\omega) * S_y(\omega) + E_y(\omega) \end{aligned} \quad (4)$$

其中 $S_x(w)$ 和 $S_y(w)$ 为信号功率,而 $E_x(w)$ 和 $E_y(w)$ 为由 $\Sigma\Delta$ 调制器带入的量化噪声。可以将量化噪声视为白噪声,它们的频域表达式为 $E(w) =$

$2e_{\text{rms}} \sqrt{\frac{2}{f_s}} \sin(\frac{w}{2f_s})$, $e_{\text{rms}} = \delta / \sqrt{12}$, 因此比较 $Z_1(w)$ 和 $Z_2(w)$ 可以看出,混合信号乘法器在运算过程中通过卷积引入的噪声项要小很多。

这一点也可以通过快速傅立叶变换(FFT)计算得出。现输入信号频率分别为 1MHz 与 2MHz,幅度均为 0.8,系统时钟频率为 1GHz。输出信号的频谱分别如图 4(b)和图 4(c)所示。经计算混合信号乘法器的输出信号 SNR 为 49.48dB,而 L 级结构乘法器只有 25.53dB。由此可见混合信号乘法器运算精度要高很多。

2.4 $\Sigma - \Delta$ 比特流非线性激活函数单元

文献[7-9]设计的神经网络采用线性和硬极限函数作为激活函数,而 BP 人工神经网络隐含层的激活函数应为非线性函数,所以本文利用单比特立方根函数设计了非线性运算单元,其结构如图 5(a)所示。图中的子乘法器采用了文献[15]的 L-level 结构。而子乘法器前增加的多个 D 触发器,对输入的三路相同的信号起一个错位作用。实验表明这样能够更好地优化输出信噪比。

通过整个比特流非线性单元建立的 z 域模型,可得

$$\begin{cases} (X(z) - T(z)^3) \cdot \frac{z^{-1}}{1 - z^{-1}} + E(z) = T(z) \\ Z(z) = (1 + T(z))/2 \end{cases} \quad (5)$$

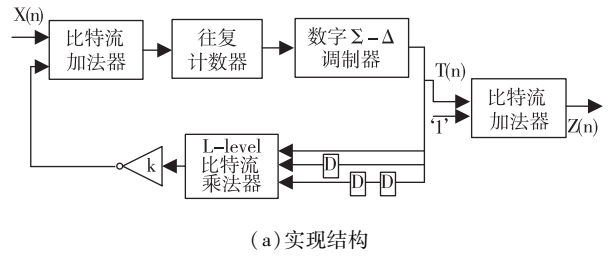
其中 $E(z)$ 为经过噪声整形的量化噪声。因此,滤波后该电路结构的时域输出满足下式:

$$\begin{cases} t(n+1) = t(n) + [x(n) - t(n)^3] \\ z(n) = (1 + t(n))/2 \end{cases} \quad (6)$$

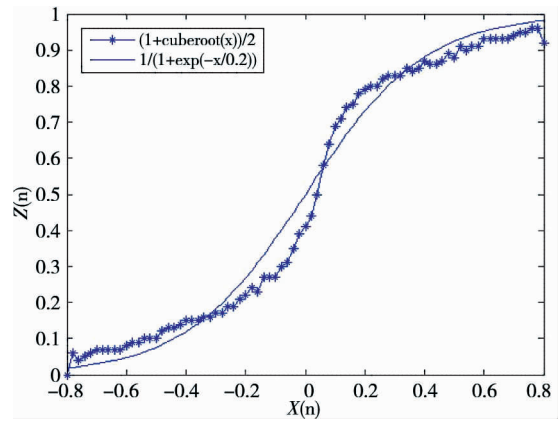
通过上式可以看出本电路结构中的 $T(n)$ 实现

了 $x(n)$ 的立方根函数,输出 $Z(n)$ 则将 $T(n)$ 映射到了 $[0, 1]$ 之间,并以此逼近 Log-Sigmoid 函数。

图 5(b)比较了传统的 Logsig 函数和所设计非线性激活函数单元的解调输出值。由模块的 I/O 输出曲线可见,虽然这里实现的 Sigmoid 函数不十分理想,但其在 $[-1, 1]$ 内同样具有连续递增性和有界性。因此以此函数作为激活函数的 BP 网络也是能够实现任意函数逼近的^[17]。同时它和 Log-Sigmoid 函数的误差也并不影响 BP 网络的实现,通过神经网络在训练过程中的自适应特征,依然可以使系统的输出达到设定目标。



(a) 实现结构



(b) I/O 曲线 vs Log-sigmoid 函数

图 5 非线性函数实现结构及仿真曲线

3 实例验证

函数逼近是非线性 BP 神经网络的重要应用。我们在 FPGA 上按照本文所设计的关键运算单元构建图 2 所示的 BP 网络系统,并在此基础上利用此系统完成对平方项函数和 sine 函数的逼近。本系统共应用 9 个神经元。在经过电路验证后,整个系统在 Xilinx Spartan-3 XC3S400 上的资源使用结果如表 1 所示。

表 1 比特流 BP 神经网络在 FPGA 上的实现

结 构	1 in 8 hidden 1 out
SDM 的信噪比	6.4bit
LUTs 数量	2211(31%)
触发器数量	1491(21%)
接口数量	4(1%)

3.1 训练 1:逼近平方项函数

这里首先验证网络可以逼近典型的平方项函数。选择目标函数如下:

$$Y(n) = (X(n) - 0.1)^2/2 + 1/8 \quad (7)$$

P 是在 $[-1,1]$ 之间平均选取的 19 个训练对象。 $P = [-0.9, -0.8, -0.7, -0.6, -0.5,$

$$-0.4, -0.3, -0.2, -0.1, 0, 0.1, 0.2, \\ 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]$$

先由原函数计算出参考标准值 (Teaching Signal), 并在权值寄存器中随机设定初始值后, 我们可以根据 BP 算法对网络进行训练。根据 BP 算法, 可以得出等式

$$S_2 = -2 \times err$$

$$S_1 = \frac{\partial F_{\text{sigmoid}}}{\partial x} \times W_b \times S_2$$

$$W_m(k+1) = W_m(k) - \alpha \cdot S \cdot F_{\text{out}}$$

$$\text{Bias} = \text{Bias} - \alpha \cdot S_2 \quad (8)$$

中的权值修改公式。其中 S_1, S_2 分别为第一层和第二层的误差反向传递系数, 学习参数 α 本文设定为 $1/4$ 。

图 6 给出了训练过程中的误差收敛曲线。由图可见网络训练过程良好, 误差收敛。

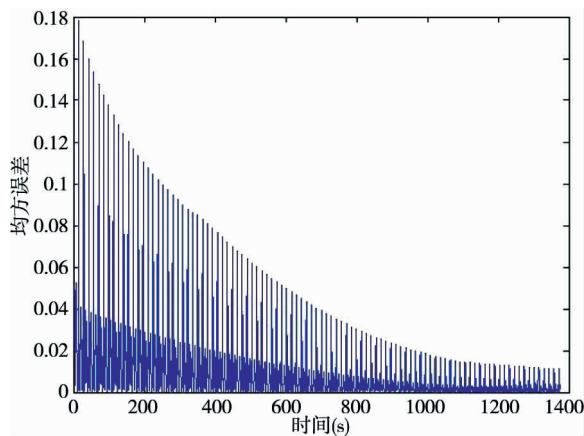


图 6 误差收敛曲线

图 7 是网络权值确定后, 输入线性波时的 FPGA 的仿真输出; 图 8 是同时段的测试波形。虽然测试中, 系统存在竞争冒险的现象, 但是从整体上, 图 8 保留了图 7 中输出比特流的变化趋势。

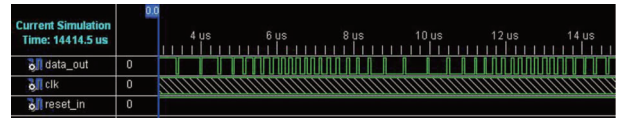


图 7 逼近平方函数的 FPGA 仿真输出

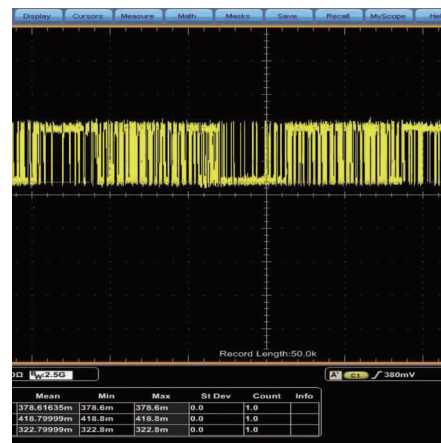


图 8 逼近平方函数 FPGA 输出波形

如将上文 19 个训练点逐一输入神经网络系统, 并解调输出码流, 可以得到图 9 中的 I/O 输出曲线。和原函数标准波形比较, 可以发现本系统实现了对平方项函数的逼近。通过计算此时全局均方误差约为 0.017。

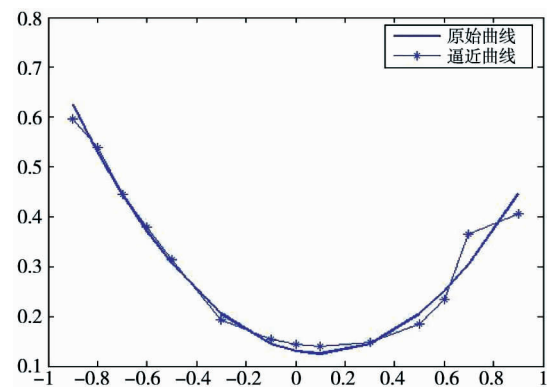


图 9 网络输出 vs 原平方函数

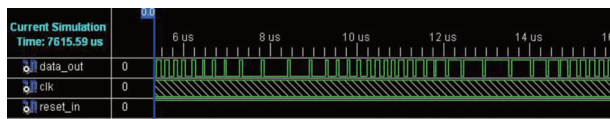
3.2 训练 2:逼近 sine 函数

正弦函数一直是验证函数逼近的重要函数之一。首先选取频率较低的正弦函数作为目标函数:

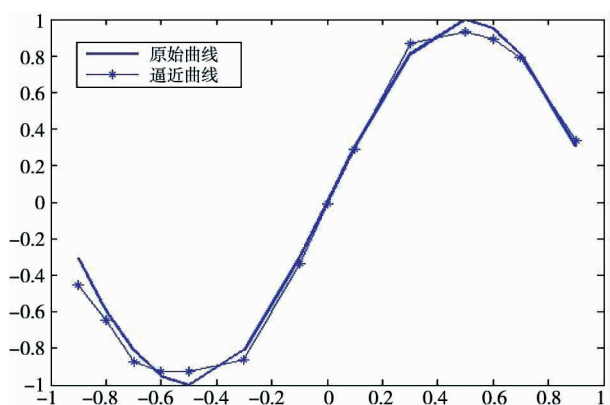
$$Y(n) = \sin(\pi \times X(n)) \quad (9)$$

输入范围为 $[-1, 1]$, 此时正弦函数输出有两个极值点和一个拐点。

图 10(a) 是 FPGA 的仿真输出, 而图 10(b) 是逐点输入后网络输出和 $\sin$ 函数的对比波形。从图中可以看出, 网络能够实现逼近等式 (9) 中的正弦函数。全局均方误差约为 0.028。



(a) FPGA 仿真输出



(b) 网络输出 vs 原 Sine 函数

图 10 神经网络逼近 Sine 函数

在传统 BP 人工神经网络理论中, 若目标函数的复杂度上升, 由于初始值的不确定性, 则容易导致整个网络系统在训练过程中落入局部最小值, 而非全局最小值。但这一局限性可以通过增加隐含层中神经元的数量加以克服。以下网络实验可以证明这一性质。

若另设立复杂目标函数如等式

$$Y(n) = \sin(\frac{3}{2}\pi \times X(n)) \quad (10)$$

则 1-8-1 网络在通过训练确定权值时, 容易落入局部最小值。若我们增加隐含层神经元数目, 采用 1-16-1 网络实现函数逼近, 则很容易找到更小误差的逼近状况。图 11 描述了 8 神经元和 16 神经元两种情况下的比较结果, 证明了上述分析的正确性。

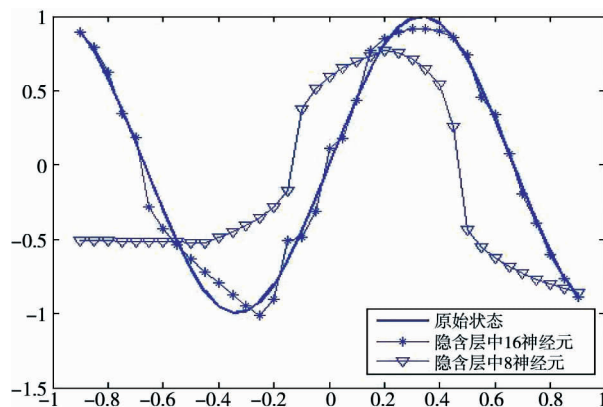


图 11 两种网络逼近 $\sin[(\frac{3}{2})\pi x]$

4 结 论

综上所述, 本文设计了基于 $\Sigma - \Delta$ 调制的并行分布式非线性 BP 人工神经网络。构建了低环路结构的 $\Sigma - \Delta$ 单比特加法器和混合信号乘法器, 不但减少了硬件消耗, 而且提高了运算速度和精度, 同时通过立方根运算实现了 Logsig 非线性函数激活单元。最后利用 FPGA 实现了整个网络系统, 并通过此网络分别逼近了平方项函数和 Sine 函数。

致 谢

感谢东南大学射频与光电集成电路研究所为本课题提供的仿真及测试环境, 并感谢研究所王志功教授和李伟老师为网络系统设计提供的指导帮助。

参考文献

- [1] Rafal D, Tomasz T, Witold P. Current-Mode Analog Adaptive Mechanism for Ultra-Low-Power Neural Networks. *IEEE Trans on Circuits and Systems II: Express Briefs*, 2011, 58(1): 31-35
- [2] Gatet L, Helene T B, Marc L. Analog neural network implementation for a real-Time surface classification application. *IEEE Sensors Journal*, 2008, 8(8): 1413-1421
- [3] Popescu S. Hardware implementation of fast neural networks using CPLD. In: *Proceedings of the 5th Seminar on Neural Network Applications in Electrical Engineering*, Belgrade, Serbia, 2000. 121-124
- [4] Hariprasath S, Prabakar T N. FPGA implementation of multilayer feed forward neural network architecture using VHDL. In: *International Conference on Computing, Communication and Applications (ICCCA)*, Dindigul, India, 2012. 1-6
- [5] Tomlinson M S J, Walker D J, Sivilotti M A. A digital neu-

- ral network architecture for VLSI. In: International Joint Conference on Neural Networks, San Diego, USA, 1990. 1-3
- [6] Maeda Y, Tada T. FPGA implementation of a pulse density neural network with learning ability using simultaneous perturbation. *IEEE Trans on Neural Networks*, 2003, 14 (3): 688-695
- [7] Liang Y, Meng Q, Guo X D, et al. Design of bit-stream neuron based on direct Sigma-Delta signal process. In: International Conference on Wireless Communications & Signal Processing, Nanjing, China, 2009. 1-3
- [8] Liang Y, Wang Z G, Meng Q, et al. Bit-stream linear artificial neural networks based on Sigma-delta modulation. *High Technology Letters*, 2012, 18(2): 120-123
- [9] Liang Y, Wang Z G, Meng Q, et al. FPGA implementation of bit-stream neuron and perceptron based on sigma delta modulation. *Journal of Southeast University (English Edition)*. 2012, 28(3): 282-286
- [10] Reyneri L M. A performance analysis of pulse stream neural and fuzzy computing systems. *IEEE Trans on Circuits and Systems II: Analog and Digital Signal Processing*, 1995, 42(10): 642-660
- [11] Schreier R, Temes G C. Delta-Sigma Data Converters. 北京: 科学出版社, 2007. 7-35
- [12] Cheung K F, Tang P Y H. Sigma-delta modulation neural networks. In: IEEE International Conference on Neural Networks, San Francisco, USA, 1993. 489-493
- [13] Katao T, Hayashi K, Fujisaka T, et al. Sorter-based sigma-delta domain arithmetic circuits. In: 18th European Conference on Circuit Theory and Design, Seville, Spain, 2007. 679-682
- [14] Fujisaka H, Kamio T, Ahn C J, et al. Sorter-based arithmetic circuits for sigma-delta domain signal processing-Part I: addition, approximate transcendental functions, and log-domain operations. *IEEE Trans on Circuits and Systems I: Regular Papers*, 2012, 59(9): 1952-1965
- [15] Fujisaka H, Kurata R, Sakamoto M, et al. Bit-stream signal processing and its application to communication systems. *IEE Proceedings-Circuits, Devices and Systems*, 2002, 149 (3): 159-166
- [16] Ng C W, Wong N, and Ng T S. Efficient FPGA implementation of bit-stream multipliers. *Electronics Letters*, 2007, 43(9): 496-497
- [17] Castro J L, Mantas C J, Benítez J M. Neural networks with a continuous squashing function in the output are universal approximators. *Neural Networks*, 2000, 13(6): 561-563

Hardware implementation of single-bit nonlinear BP neural network based on sigma-delta modulation

Guo Xiaodan, Meng Qiao, Liang Yong

(Institute of RF- & OE-ICs, Southeast University)

(Engineering Research Center of RF-ICs & RF-Systems, Ministry of Education, Southeast University)

Abstract

A new method for implementation of the hardware of single-bit feed-forward BP artificial neural networks by using sigma-delta modulation (SDM) was presented. The nonlinear cube-root calculation unit based on single-bit streams was designed, and by taking it as the activation function, the BP artificial neural network was constructed. The signals from the network's input and output of each neuron were represented by sigma-delta modulated single-bit streams. As the activation function in the hidden layer should be nonlinear, the log-Sigmoid squashing function approximation was obtained through the design of a cube-root computation module based on sigma-delta bit stream. A little-loop-delay adder and a hybrid-signal multiplier were also presented as the key function elements of the system to offer low hardware consumption and high precision. The neurons and the whole BP neural network were implemented and simulated on a field programmable gate array (FPGA). Two examples of function approximation successfully demonstrate that the sigma-delta bit stream technique is viable for the hardware implementation of BP neural networks.

Keywords: single-bit, artificial neural network, nonlinear, BP network, sigma-delta, field programmable gate array (FPGA)