基于 BP 神经网络的仿真线设计及其 FPGA 实现

张海燕　　李 欣　　田书峰
(中国海洋大学电子工程系　青岛　266100)

摘　要：该文提出了一种采用 BP 神经网络实现仿真线的方法。首先采用遗传算法优化神经网络结构，用离线训练后的 BP 神经网络逼近传输线的传递函数，然后用 STAM 算法以较少的存储空间实现 BP 神经网络的激励函数近似，进而用 FPGA 和 D/A 转换器进行硬件实现。文中基于 FPGA 对长度为 10000m，特性阻抗为 55Ω 的同轴电缆进行了仿真线的硬件实现，实验结果验证了该方法的有效性。该方法可以推广到传递函数未知的传输网络的仿真应用中。

关键词：仿真线；BP 神经网络；FPGA；STAM 算法
中图分类号：TN812　　**文献标识码**：A　　**文章编号**：1009-5896(2007)05-1267-04

Simulation Line Design and Its FPGA Realization Based on BP Neural Network

Zhang Hai-yan　　Li Xin　　Tian Shu-feng
(*Department of Electronic Engineering*, *Ocean University of China*, *Qingdao* 266100, *China*)

Abstract: A new method for simulation line realization based on Back Propagation Neural Network (BP NN) is presented in the paper. Applying Genetic Algorithm (GA) to optimize the neural network's structure, BP NN is trained to correspond the transfer function of simulation line. Activation function of NN is approximated with STAM (Symmetric Table and Addition Method) algorithms. A coaxial-cable which is 10000m long and 55Ω line characteristic impedance is simulated and realized by using FPGA and D/A converter. Experimental results show that the proposed approach can greatly reduce the memory of hardware realization. This method can be generalized to simulate the transmission network with unknown transfer function.

Key words: Simulation line; BP neural network; FPGA; STAM algorithms

1　引言

仿真线是仿效某种长度线路的衰耗频率特性，供模拟调测用的网络[1]。由于在长线传输的研究中不可能总是在实际传输线路上进行试验，所以常常使用仿真线来模拟长线的传输特性。仿真线实际上是一个低通滤波器，然而却很难通过反演的方法得到它的传递函数表达式，因此使用通常的模拟或数字滤波器的设计方法来实现仿真线比较困难。

本文基于 BP 神经网络用 FPGA 实现了仿真线设计，并详细讨论了以下关键技术：(1)采用 BP 神经网络逼近传输线的传递函数，其中利用遗传算法优化 BP 网络结构；(2)用 FPGA 实现 STAM 算法。

2　基于 BP 网络的仿真线设计

2.1　基于 BP 网络的仿真线设计原理

人工神经网络是模拟生物神经网络的人工智能系统，具有分布并行处理、非线性映射、自适应学习和容错等功能。目前在神经网络的实际应用中，绝大部分的神经网络模型使用的是BP神经网络或其变化形式。基于误差反传训练算法的多层前向神经网络即BP神经网络被证明具有较强的非线性映射和模式识别能力。Robert Hecht- Nielsen的研究结果表明，在一定的条件下，具有输入、输出和一个非线性隐层的三层BP神经网络，能够精确逼近在闭区间内的一个连续的非线性函数[2]。因此可以通过设计一个三层BP神经网络来逼近传输线的传递函数。不同的传输线具有不同的传递函数，首先根据所传输的具体信号通过实测的方法获得样本集，然后针对其特点构造BP神经网络，并训练它逼近该传输线的传递函数。仿真线设计的原理框图如图 1 所示。

本文针对科考船上的一根长度为 10000m，特性阻抗为55Ω的铠装同轴电缆，通过实测方法得到信源和信宿的信号波形数据，形成训练样本集和检验样本集，离线训练一个三层 BP 神经网络逼近该电缆的传递函数，然后使用 Altera 公司的 FPGA Stratix-II EP2S30 和 TI 公司的 10bits 高速 D/A

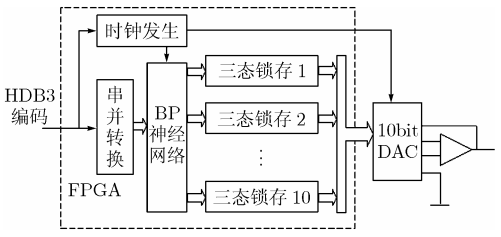

图 1　仿真线设计原理框图

2005-11-15 收到，2006-10-26 改回
国家 863 计划(2001AA612030)资助课题

转换器 THS5651A，实现上述同轴电缆的仿真线。

2.2 用遗传算法优化 BP 神经网络结构

实验中信源是 HDB3 编码的基带信号，该信号通过长线传输到达信宿时，由于传输线的衰耗，信号波形发生畸变，某一码元畸变后的输出波形不仅取决于该码元本身，而且也与该码元前、后数个码元有关，因此 BP 网络的输入层设计为一码元序列，设序列长度为 l。采用 l 个输入节点表示该码元序列，每个节点用 2bit 二进制数表示(00 表示 HDB3 编码中的 0，10 表示+1，01 表示−1)。BP 网络的输出层用来表示当前输入序列中的第 $\lceil (l+1)/2 \rceil$ 个码元对应的输出波形数据，由于该波形已经发生畸变，需要用一个模拟量来表示，所以使用一个高速 D/A 转换器对 BP 网络的输出层节点进行数模转换。综合考虑 FPGA 的资源和 D/A 转换器的速度及精度，确定 BP 网络的输出层节点数为 10 个，每个节点用 10bit 二进制数来表示，通过轮流选通图 1 中的 10 个三态锁存器的输出允许端来实现输出码元波形的 D/A 转换。

如何针对特定问题确定合适的网络结构一直是目前神经网络应用中的一个难点。目前常用试凑法或经验公式法对 BP神经网络结构进行寻优，其结果的偶然性比较大[3−5]。本文将遗传算法与BP神经网络相结合，利用遗传算法较强的全局寻优能力，优化神经网络的输入层和隐层结构。考虑到神经网络的硬件实现，其网络结构不宜过于复杂，故采用 8bit 的二进制编码描述一个网络结构的可行解，其中前 4bit表示输入层的节点数，解码取值范围为 3~10，后 4bit表示隐层的节点数，解码取值范围为 3~15。设置种群的个体数为 50，采用轮盘赌注选择，双点杂交以及 0，1 变异作为遗传算子，交叉概率取值为 0.7，变异概率取值为 0.01，在整个网络结构解空间中进行搜索、优化。

首先对遗传算法种群中的个体进行解码，获得相应的网络可行结构。基于 BP 算法使用训练样本对网络进行学习，学习误差达到 1E-3 时结束。然后用权重已确定的网络模型对检验样本进行测试，将检验样本的期望输出值与实际输出值的均方差作为网络的误差评价，其倒数作为适应度函数，具体表达式如下：

$$\text{Fitness} = \frac{1}{\sum_{x_i \in S} \left[D(x_i) - O(x_i) \right]^2} \tag{1}$$

其中 S 为检验样本集；当输入样本为 x_i 时，经过网络前向传播过程计算的实际输出为 $O(x_i)$，而相应的期望输出为 $D(x_i)$。

求解种群中个体的适应度，通过遗传算子的操作进行进化，产生新的子代。经过若干代遗传进化后，获得最适宜个体，解码后即对应网络的优化结构，在学习过程中同时也确定了该网络的连接权重。

由于铠装同轴电缆的高频衰耗非常严重(见图 3)，因此所传输的 HDB3 编码基带信号的码流选为 1Mbps。该信号在信宿端的波形变化剧烈，所以 BP 网络的激励函数选用了在振幅上比 Sigmoid 函数增长剧烈、其导数的取值范围也较大的双曲正切型函数 tanh，其表达式为

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2}$$

实验中选用了 1000 个训练样本，200 个测试样本，经遗传进化获得最优染色体为 01111001，解码得到输入层、隐层节点数分别为 7，9，因此本文采用 BP 神经网络的优化结构为 7-9-10，BP 学习算法结束时也获得了该网络的权重参数。

3 BP 网络的 FPGA 实现

由于 BP 网络采用离线训练的方式，所以在 FPGA 中只需要考虑前向传播过程的实现。对于 BP 网络的每一个隐层节点和输出节点，有

$$O_j = f\left(\sum W_{ij} \bullet x_i - \theta_j \right) \tag{3}$$

式(3)中，O_j 是节点的输出值，W_{ij} 是节点之间的连接权值，x_i 是节点的输入值，θ_j 是神经元阈值，$f(\bullet)$ 是 tanh 函数。

对于训练完毕的 BP 网络，式(3)中的 W_{ij} 和 θ_j 都是确定值，可以存放在 FPGA 的 RAM 区中。在式(3)的所有运算中，乘、加、减等运算的硬件实现较为简单，而 tanh 函数在 FPGA 上的实现则是问题的关键。

非线性函数的硬件实现方法主要有两大类：基于查找表的方法和基于多项式近似的方法。在基于多项式近似的方法中由于存在大量的乘法运算，硬件实时实现的复杂程度较高；对于基于查找表的方法，存在着表示精度与存储空间之间的矛盾，所以必须在满足待求解问题的表示精度的前提下，使用尽量少的硬件资源和尽量简单的运算。

本文采用一种被称之为“对称表求和”的算法[6,7](Symmetric Table Addition Method，STAM)。STAM 算法具有闭合解，适用于可微函数的近似。STAM算法采用多个并行查找表和随后的 1 个多操作数加法器实现函数逼近，所占用的存储空间比直接查表法小很多，而且查表之后的加法运算比较简单，运算速度也较高，很适合于硬件的高速实现。

3.1 STAM 算法

STAM 算法的基本思想是将原直接查表法中的一个查找表，按照预定的规则分成多个查找表，再将多个查找表的输出相加求和，得到最终结果。在 STAM 算法中，通过增加查找表的数目，可以有效节省总的查找表存储空间，而所增加的加法运算在实现的复杂度和速度两方面的开销也并不大。利用查找表数据的对称性和冗余特点，还可以进一步减小查找表所占用的存储空间。

为了实现对一个函数 $f(x)$ 的逼近，首先将具有 n 位输入的 x 分成 $m+1$ 个部分：$x_0, x_1, \cdots, x_m$，$x = x_0 + x_1 + \cdots + x_m$，其长度分别为：$n_0, n_1, \cdots, n_m$，$n = n_0 + n_1 + \cdots + n_m$。使用标识符 $x_{0:m}$ 表示 x_i 的连加关系，即

$$x_{0:m}=\sum_{i=0}^{m}x_i,\quad n_{0:m}=\sum_{i=0}^{m}n_i$$

由 x_0 分别和 $x_0,x_1,\cdots,x_m$ 构成 m 个并行的查找表，其中第 i 个查找表的输入值为 x_0 与 x_i 的位串连接，输出为 $a_{i-1}(x_0,x_i)$，m 个查找表的输出之和作为函数 $f(x)$ 的逼近值。因此函数 $f(x)$ 的逼近形式为

$$\tilde{f}(x)=\sum_{i=1}^{m}a_{i-1}(x_0,x_i) \tag{4}$$

从式(4)可知，STAM 算法的硬件实现需要有 m 个并行的查找表，和 1 个有 m 个输入端的多操作数加法器，其实现框图如图 2 所示。

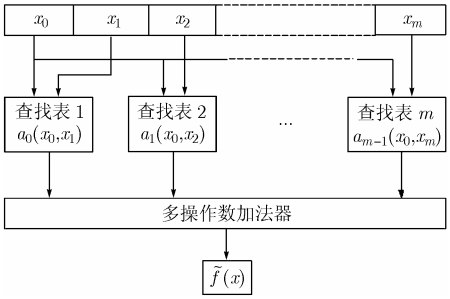

图 2　STAM 算法的实现框图

3.2 STAM 算法的查找表构造

基于 STAM 算法对函数进行逼近，需要使用函数 $f(x)$ 的 1 阶泰勒展开来构造查找表 $a_{i-1}(x_0,x_i),1\le i\le m$。

函数 $f(x)$ 在 $x=x_{0:1}+\delta_{2:m}$ 处的 1 阶泰勒展开为

$$\begin{aligned}f(x)&\approx f(x_{0:1}+\delta_{2:m})+f'(x_{0:1}+\delta_{2:m})(x_{2:m}-\delta_{2:m})\\&=f(x_{0:1}+\delta_{2:m})+f'(x_{0:1}+\delta_{2:m})\bullet\sum_{i=2}^{m}(x_i-\delta_i)\end{aligned} \tag{5}$$

式中 δ_i 被定义为 x_i 的取值区间的中间点，即

$$\delta_i=2^{-n_{0:i-1}-1}-2^{-n_{0:i}-1},\quad 1\le i\le m \tag{6}$$

为了简化查找表的构造，用常数 δ_1 代替 x_1，这会引入一个绝对误差，但是可以将其控制在精度容许的范围之内[6]。x_1 被 δ_1 替代后，式(5)改变为

$$f(x)\approx f(x_{0:1}+\delta_{2:m})+f'(x_0+\delta_{1:m})\cdot\sum_{i=2}^{m}(x_i-\delta_i) \tag{7}$$

由式(7)中的第 1 项构造查找表 1：$a_0(x_0,x_1)$，其余 $m-1$ 项分别构造第 2~m 个查找表 $a_{i-1}(x_0,x_i),\quad 2\le i\le m$：

$$a_0(x_0,x_1)=f(x_{0:1}+\delta_{2:m}) \tag{8}$$

$$a_{i-1}(x_0,x_i)=f'(x_0+\delta_{1:m})(x_i-\delta_i),\quad 2\le i\le m \tag{9}$$

式(8)构造的查找表 $a_0(x_0,x_1)$，其输入值的位数为 n_0+n_1。式(9)所构造的其余 $m-1$ 个查找表 $a_{i-1}(x_0,x_i)$，由于 δ_i 被定义为 x_i 的取值区间的中间点，故查找表中的系数值具有对称性，即 $a_{i-1}(x_0,x_i)$ 与 $a_{i-1}(x_0,2\delta_i-x_i)$ 互为补码，其输入值的位数可以减为 n_0+n_i-1，从而使这 $m-1$ 个查找表的存储空间节省了一半。

另外，对于 $m-1$ 个查找表 $a_{i-1}(x_0,x_i)$，其系数值要远小于第 1 个查找表 $a_0(x_0,x_1)$，这会产生多个前导 0($a_{i-1}(x_0,x_i)<0$ 时为前导 1)，其个数可根据下式进行推算：

$$\left|a_{i-1}(x_0,x_i)\right|<\left|f'(x_0+\delta_{1:m})\right|\cdot 2^{-n_{0:i}-1}$$

这些前导 0(或 1)没有必要存储在查找表中，而是可以在执行加法操作之前，根据 x_i 的最高位进行扩展来获得，由此又可以节省一部分存储空间。

根据 STAM 算法构造查找表来逼近函数 $f(x)$，存在一定的误差，但是能够对其进行控制，使误差小于查找表输出的最小表示精度 1 ulp(unit in the last place)，具体分析请参考文献[6]。

3.3 tanh 函数的逼近及其表示精度

使用上述 STAM 算法对 BP 神经网络的激励函数 tanh 函数进行逼近。在此过程中需要利用 tanh 函数 1 阶泰勒展开，设展开点为 $x^*=x_{0:1}+\delta_{2:m}$：

$$\begin{aligned}\tanh(x)&\approx\tanh(x^*)+\tanh'(x^*)(x-x^*)\\&=\frac{e^{x^*}-e^{-x^*}}{e^{x^*}+e^{-x^*}}+\frac{4(x-x^*)}{(e^{x^*}+e^{-x^*})^2}\end{aligned} \tag{10}$$

以下详细分析 STAM 算法查找表 $a_{i-1}(x_0,x_i)$ 的输入范围和输入、输出的表示精度等问题。由于 tanh 函数关于原点对称，而其导数 $\tanh'(x)$ 关于纵轴对称，所以只需要考虑 $x\ge 0$ 的情况。当 $x\to\infty$ 时，$\tanh(x)\to 1$，$\tanh'(x)\to 0$，因此查找表可以在某一个区间内逼近 $\tanh(x)$ 和 $\tanh'(x)$。既要保证对于任意的输入 x，查找表输出值的绝对误差均小于最小表示精度 1 ulp，又要尽量减少查找表的长度以节省存储空间，所以必须选择一个合适的输入值上限。设对于函数 $\tanh(x)$ 和 $\tanh'(x)$，分别用 x_a 和 x_b 表示。

假设查找表的输出值为 p 位小数，则最小表示精度为 2^{-p}，即 $1\text{ ulp}=2^{-p}$。设当 $x\ge x_a$ 时，$\tanh(x)$ 可以被近似为 $1-2^{-p}$，所以 x_a 应当满足

$$1-\frac{e^{x_a}-e^{-x_a}}{e^{x_a}+e^{-x_a}}<2^{-p} \tag{11}$$

为了便于硬件实现，x_a 选取满足式(11)的最小整数，

$$x_a=\left\lceil\frac{1}{2}\ln(2^{p+1}-1)\right\rceil \tag{12}$$

同理，设当 $x\ge x_b$ 时，$\tanh'(x)$ 可以被近似为 0，

$$x_b=\left\lceil\cosh^{-1}(2^{p/2})\right\rceil \tag{13}$$

这样就可以分别在区间 $[0,x_a]$ 和 $[0,x_b]$ 上逼近 $\tanh(x)$ 和 $\tanh'(x)$，同时保证了对于任意的输入 x，查找表输出值的绝对误差值小于 1 ulp。

考虑到构造查找表和加法器输出时产生的舍入误差，查找表中系数值的表示精度需要在最终输出结果的精度基础上增加 $2+\log_2(m-1)$ 位[8]。

在本文的实例中，选择查找表输入值和输出值的小数位 $p=10$，计算可得 $x_a=4$ 和 $x_b=4$，因此查找表输入 x 的整数位 $k=\log_2(x_a)=2$，其位长 $n=k+p=2+10=12$。STAM 算法中参数选择为 $m=3,\ n_0=4,\ n_1=2,\ n_2=n_3=3$，$2+\log_2(m-1)=3$，所构造的 3 个查找表 $a_0(x_0,x_1)$，$a_1(x_0,x_2)$ 和 $a_2(x_0,x_3)$，其输入 x 的位数均为 6 位，存放在查

找表中的系数值精度为 13 位小数。

对于上述 3 个查找表，其最低有效位均为 1，不必存储；由于有 $0 \le \tanh(x) \le 1$ 和 $0 \le \tanh'(x) \le 1$ 成立，整数位不必存储；对于查找表 $a_1(x_0, x_2)$ 和 $a_2(x_0, x_3)$，分别有 4 个和 7 个前导 0，也不必存储。因此 3 个查找表的位长分别为 12，8 和 5 位，所占用的存储空间为：$2^6 \times (12+8+5)=1600$ 位。如果使用直接查表法在 $[0,4]$ 区间实现 10 位小数的精度，所需要的存储空间则为 $2^{12} \times 10=40960$ 位。可见表示精度为 10 位小数时，STAM算法的存储空间大约为直接查表法的 1/25。

4 实验结果

4.1 STAM 算法的 Modelsim 仿真

用Modelsim对上述3个查找表以及加法器的VHDL描述进行仿真，结果显示，$\tanh(x)$ 函数的仿真输出值与理论值之间的绝对误差均小于 1 ulp=2^{-10}(≈0.0009765625)。表 1 列举了其中 5 组数据，分别包含：输入值 x，$\tanh(x)$ 函数的仿真输出值，理论值和绝对误差。

表 1 $\tanh(x)$函数的部分仿真结果

输入值 x	仿真输出值	理论值	绝对误差
0.1269531250	0.1261475975	0.1262754549	0.0001278574
0.5244140625	0.4815655179	0.4810996071	0.0004659109
1.0654296875	0.7884550059	0.7877331282	0.0007218778
1.5205078125	0.9090798315	0.9087861134	0.0002935698
3.7148437500	0.9988123281	0.9988139579	0.0000016298

4.2 仿真线的实验结果

为了利用 FPGA 中的乘法器快速实现神经元中的乘法操作，选用内部具有 128 个 9 位乘法器的 Stratix-II EP2S30F484，与 10bit 高速 D/A 转换器 THS5651A 配合，对图 1 的设计进行了实现。EP2S30 内部 9 位乘法器的利用率为 94%，自适应查找表 ALUTs 的利用率为 26%，说明采用 STAM 算法逼近激励函数比较节省硬件资源。

实验结果表明，该仿真线与实际同轴电缆的传输特性非常接近。图 3 中的实线和虚线分别是 10000m 同轴电缆和仿真线的幅频特性曲线。图 4(a)是 HDB3 编码为“1000110111 01000100011001100011”的一段信源信号，图 4(b)是与之对应的信宿信号波形，仿真线的输出信号波形为图 4 (c)，图 4 (d)是仿真线输出信号与信宿信号的误差，其信号幅度绝对误差小于 0.1V，相对误差小于 5%。

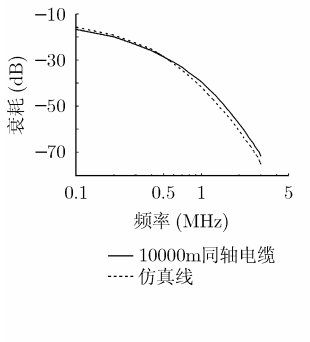

图 3 10000m 同轴电缆及其仿真线的幅频特性

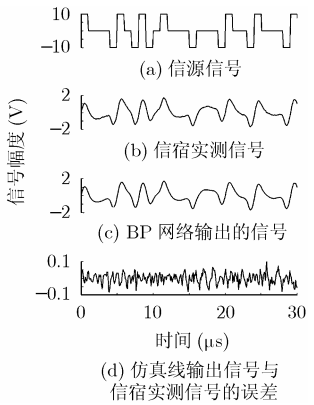

图 4 仿真实验结果的部分波形

5 结束语

本文采用训练 BP 神经网络逼近传输线传递函数的方法实现仿真线。在 FPGA 实现中，采用 STAM 算法，有效地节省存储空间，并对长度为 10000m，特性阻抗为 55 Ω 的同轴电缆进行了硬件实现。使用该方法实现的仿真线，已经成功地应用于国家 863 项目“6000 米海底有缆观测与采样系统——电视抓斗的研制”课题的数字图像传输研究中。

对于不同的传输线，传递函数不同，因此所构造的 BP 网络也有所不同，采用同样的原理，可以实现针对特定电缆的仿真线。对于高速传输的信号，只要能够在信宿端接收到信号，并且 FPGA 和 D/A 转换器的工作频率可以满足高速信号的要求，仍然可以在 FPGA 中实现仿真线。本文所提出的方法，可以推广到传递函数未知的传输网络的仿真应用中。

参 考 文 献

[1] 中华人民共和国国家标准：常用电信设备名词术语 GB1417-1978，北京: 技术标准出版社，1979.

[2] Nielsen R H. Theory of the backpropagation neural network. Proceedings of the International Joint Conference on Neural Networks. Washington, USA. 1989, 1: 593–605.

[3] 李倩，王永县，朱友琴. 人工神经网络混合剪枝算法. 清华大学学报, 2005, 45(6): 831–834.

[4] 杨建国，翁善勇，赵虹等. 采用遗传算法优化的煤粉着火特性 BP 神经网络预测模型. 动力工程，2006, 26(1): 81–83.

[5] 陈作炳，艾春庭，夏雪峰. BP 神经网络仿真软件. 计算机仿真, 2001, 18(4): 23–24.

[6] Stine J E and Schulte M J. The symmetric table addition method for accurate function approximation. *Journal of VLSI Signal Processing*, 1999, 21(2): 167–177.

[7] Schulte M J and Stine J E. Accurate function approximations by symmetric table lookup and addition. Proceedings of the 11th International Conference on Application-Specific Systems, Architectures and Processors. Zurich, Switzerland. 1999: 144–153.

[8] Sarma D D and Matula D W. Measuring the accuracy of ROM reciprocal tables. Proceedings of the 11th Symposiumon Computer Arithmetic. Windsor, Ontario, Canada. 1993: 95–102.

张海燕： 女，1968 年生，副教授，中国电子学会高级会员，研究领域为计算智能、信号与信息处理等.

李 欣： 男，1959 年生，教授，中国电子学会高级会员，研究领域为数字通信、嵌入式系统等.

田书峰： 男，1978 年生，硕士生，研究领域为嵌入式系统.