

文章编号: 1000-5862(2012)03-0276-04

改进型 BP 神经网络自适应均衡器设计

闵秋应, 况庆强

(江西师范大学物理与通信电子学院, 江西 南昌 330022)

摘要: 针对传统的基于 BP 神经网络的自适应均衡器存在的缺陷, 提出了一种改进的 BP 神经网络自适应均衡器的 FPGA 设计方法. 从互连结构、训练法则及收敛算法等几个方面对 BP 网络进行改进, 通过加入协调器提高 BP 网络收敛速度和容错性能, 并根据 BP 网络的并行性能得到分块结构模型, 同时采用流水线技术使计算速度明显提高. 通过软件仿真, 验证了这种结构简单、收敛速度较快的 BP 网络的有效性和可行性, 并在 ModelSim 工具上完成了自适应均衡器的仿真验证.

关键词: BP 网络; 流水线结构; 自适应均衡

中图分类号: TN 914

文献标志码: A

0 引言

在数字无线通信系统中, 由于多径效应和信道带宽限制导致传输过程中不可避免地产生码间干扰, 使传输质量明显下降. 目前最常用的方法是采用自适应均衡技术降低码间干扰和噪声的影响来减少误码, 并跟踪时变信道, 解决时变波形的严重失真问题. 在时域中, 由于传统实现方法存在收敛速度慢且收敛性能取决于输入信号等问题^[1], 人们提出采用变换域的自适应均衡算法, 即通过域的变换(如离散余弦变换、离散傅立叶变换、离散正交小波变换)^[2-3]来改变输入信号的自相关矩阵的最大特征值与最小特征值之比, 用于提高收敛速度. 为了减少计算的复杂度, 基于小波域的滤波器又引入部分系数迭代更新的算法^[4-5], 相应的多小波及小波神经网络的方法也先后出现. 引入变换域后虽然收敛速度提高了, 但在这些算法中计算量过于庞大造成实时性能明显下降.

本文针对传统设计方法的不足, 在 BP 网络设计方法的基础上, 提出了一种改进的 BP 神经网络设计均衡器的硬件实现方法. 通过借助于神经网络的训练序列就能自适应调整均衡器的抽头系数, 达到自适应均衡的目的. 从优化硬件结构的角度考虑,

对神经网络的硬件实现方式进行了研究, 利用流水线技术, 找出合理的设计方案. 最后采用自顶向下的设计方法, 通过硬件 FPGA 优化, 利用 Altera 公司低成本的 Cyclone II 系列 EP2C35 芯片实现自适应均衡器技术. 这种均衡器性能良好, 特别适用于实时性能要求严格的信号处理系统, 在未来的无线通信系统中有着广阔的应用前景.

1 BP 网络改进算法

传统的反向传播算法(BP 算法)利用网络误差的平方和, 对网络误差函数的导数进行权值和阈值调整, 从而降低误差平方和. 从网络误差函数推导出输出层的误差导数或 δ 矢量, 隐层的 δ 矢量可由下一层的 δ 矢量导出. 这种 δ 矢量的反向传播正是 BP 的由来^[6-9], 即 $e(t) = H_d(w_t) - H(w_t)$, 其中 $H_d(w_t)$ 为系统期望, $H(w_t)$ 为系统的实际输出, 其计算性能指标为 $J = \frac{1}{2} \sum_{t=0}^{M-1} e^2(t)$.

根据梯度下降法得到神经网络权值的修正, 即 $w(k+1) = w(k) + 2\mu e(n)X(n)$, 其中 μ 为步长因子, 且 $0 < \mu < 1$.

但是这种 BP 算法的步长因子难以确定, 导致收敛的稳定性下降, 收敛速度变慢. 针对这一问题,

收稿日期: 2011-09-21

基金项目: 国家自然科学基金(60967002)资助项目.

作者简介: 闵秋应(1968-), 男, 江西南昌人, 高级工程师, 主要从事电子通信方面的研究.

本文提出一种改进的 BP 网络设计自适应均衡器, 有效地消除码间干扰, 具体的 7 个步骤如下:

(1) 对输入向量采用归一化预处理.

(2) 确定神经网络结构, 本文选择隐层有 10 个神经元, 输出层有单个神经元.

(3) 对网络输入到最末隐层之间的所有权值进行初始化: 首先在样本集中任意取 2 个不同的样本点 x_{h1} 、 x_{h2} , 计算网络隐层节点的梯度 $\nabla_w = w/\|w\| = (x_{h2} - x_{h1})/\|x_{h2} - x_{h1}\|$, 然后计算网络隐层节点敏感区的宽度 $D_a = \ln(\beta/(1-\beta))/\|w\| = \zeta_h \|x_{h2} - x_{h1}\|$, 其中 β 一般取 0.9 左右, ζ_h 为 [1, 3] 之间的随机数. 根据 ∇_w 和 D_a 可以计算出第 i 隐层节点的权值 $w_i = \|w_i\| \nabla_w = \frac{\ln(\beta/(1-\beta))}{D_a} \nabla_w$.

(4) 对网络输出层权值进行赋值. 由于网络输出层权值的变化对网络输出的影响最大, 可根据网络抗干扰性能对网络输出层权值进行设计. 对选择样本集中的特殊点求平均值 $x_a = \frac{1}{q} \sum_{i=1}^q x_i$, 其中 q 为特殊样点的个数. 计算各样点的差值, 用这些差值组成一个特殊样点集合, 即 $\Delta X = \{\Delta x_i, i=1, 2, \dots, q\}$.

将这些特殊样本集中加入到网络中得到输出 y^0 , 再根据特殊样本集找到相对应期望平均值的误差 e^0 , 即

$$e^0 = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1q} \\ e_{21} & e_{22} & \cdots & e_{2q} \\ \vdots & \vdots & & \vdots \\ e_{n1} & e_{n2} & \cdots & e_{nq} \end{bmatrix},$$

其中 p 、 n 分别为网络输出的个数和隐层节点的个数, 则网络输出层的权值为 $w^0 = 4e^0[y^0]^T[B]^{-1}$, 其中 $[B] = [y^0][y^0]^T$.

(5) 前向传播计算网络输出. 计算隐层节点的输出为

$$z_j = f\left(\sum_i w_{ji} x_i - \theta_j\right),$$

其中 z_j 为隐层节点的输出, w_{ji} 为隐层节点的权值, x_i 为输入向量, θ_j 为隐层节点阈值.

输出结点的输出为

$$y_l = f\left(\sum_j v_{li} z_j - \theta_l\right),$$

其中 y_l 为输出结点的输出, z_j 为隐层节点的输出, v_{li} 为输出结点的权值, θ_l 为输出结点的阈值.

(6) 进行网络训练更新权值. 计算网络误差为

$$E = \frac{1}{2} \sum_i (t_i - y_i)^2, \text{ 其中, } t_i \text{ 为目标矢量, } y_i \text{ 为输出层的输出矢量.}$$

根据自适应误差的方法对网络权值进行修正, 在误差反向传播路径上增加一个协调器, 并引入前次误差值来控制权值的修正, 可使得网络输出的误差自适应变化, 即

$$v_{ji}(k+1) = v_{\max}, w_{ij}(k+1) = w_{\max}, |\Delta e| > 30\%;$$

$$v_{ji}(k+1) = \gamma v_{ji}(k) + \eta \delta_j y_j + \lambda \frac{1}{k+1} \sum_{i=0}^k |e(k-i)|^2,$$

$$w_{ij}(k+1) = 0, 10\% < |\Delta e| < 30\%;$$

$$v_{ji}(k+1) = 0, w_{ij}(k+1) = w_{\min}, |\Delta e| < 10\%;$$

其中 $|\Delta e|$ 为误差变化率, γ 为泄漏因子, η 为记忆因子, λ 为协调器补偿因子以补偿权值增量.

当网络的误差变化率较大时, 网络输出层权值及隐层权值在协调器的控制下较大, 使网络迅速变化; 当误差在 [10%, 30%] 范围内时, 网络输出层权值在协调器的控制下减小, 同时停止隐层权值的修正, 以避免网络产生过冲反向误差; 当误差小于 10% 时, 隐层权值在协调器的控制下进行修正, 而停止网络输出层权值的修正, 使网络的综合误差达到允许值.

(7) 判断误差, 结束网络. 判断误差 ΔE 是否小于期望 E , 如满足要求则结束; 否则, 返回网络继续调整权值, 直到满足实际需求为止.

2 算法的仿真分析

通过 Matlab 对传统 BP 网络算法与本文中的改进算法进行仿真比较. 图 1 为 BP 网络的训练结果与理想输出的比较结果图, 其中 BP 网络的隐层神经元数目 $n=10$, 传统 BP 网络学习率为 $lr=0.05$, 最大迭代次数 $epochs=100$, 目标误差指标 $goal=1 \times 10^{-5}$. 图 1(a) 为传统 BP 网络算法的输入、输出关系图, 在相同的迭代条件下, 实际的输出与训练后的网络输出相比有明显的误差. 图 1(b) 为改进 BP 网络算法的输入、输出关系图, 从图可以看出, 在相同的条件下, 实际的输出与训练后的网络输出基本重合, 可以达到较好的效果. 显然, 本文改进算法的稳态误差是优

于传统 BP 神经网络的.

图 2 为 BP 网络误差与迭代次数关系曲线. 图 2(a)为传统 BP 网络算法的误差曲线, 在 100 次迭代后的误差仍大于 10^{-2} . 图 2(b)为本文提出的改进 BP 网络算法的误差曲线, 显然本文中提出的算法的收敛

速度优于传统的 BP 网络. 在训练初期, 网络的误差迅速减小; 训练至中后期, 添加的协调器使得网络能够进一步向期望输出靠近, 防止 BP 网络输出的过冲效应; 在最后阶段的误差曲线能够较稳定的靠近期望误差 E , 从而结束网络训练, 达到理想的输出效果.

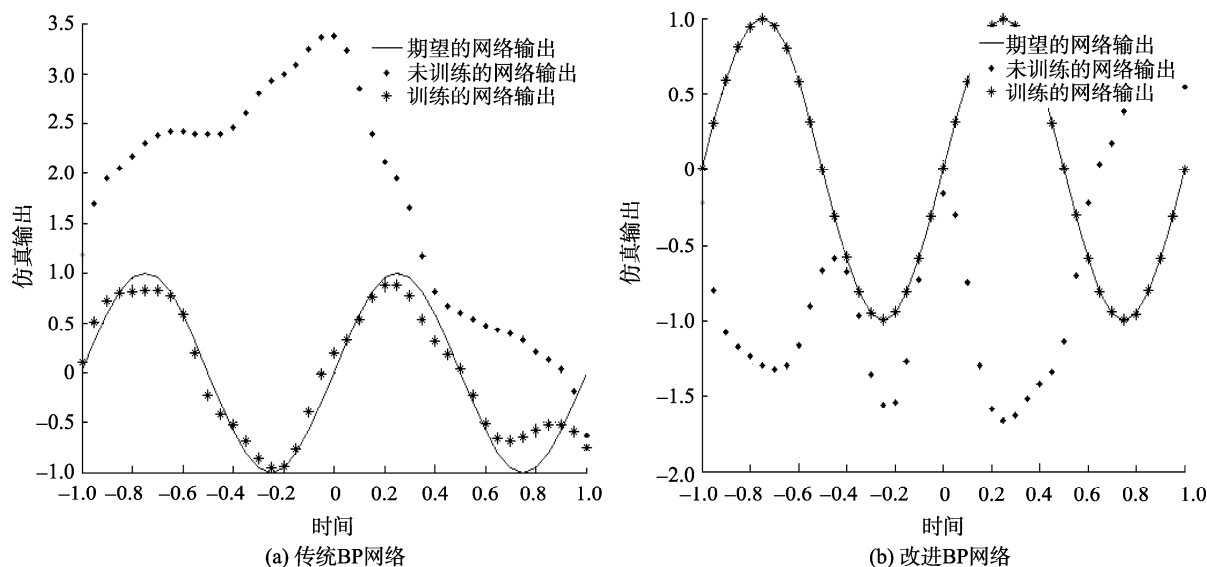


图 1 BP 网络的训练结果与理想输出的比较

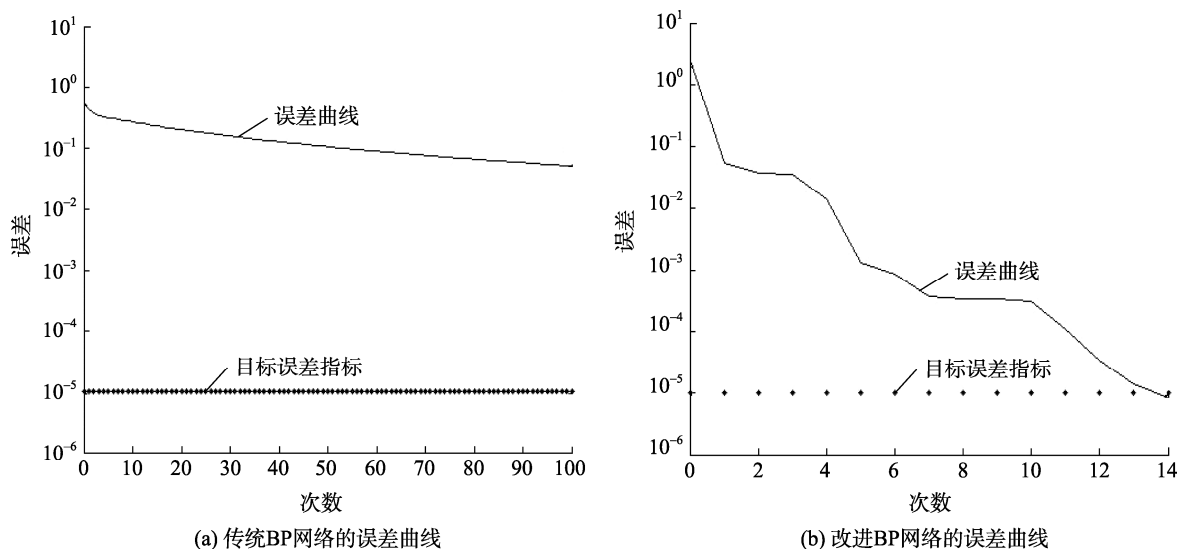


图 2 BP 网络的误差曲线

3 算法的 FPGA 实现

在设计过程中采用自顶向下的设计方法, 综合考虑每个模块的功能、结构、时序等问题^[10-11], 运

用 ModelSim 工具对神经网络各模块进行验证及优化. 完成分块实现自适应均衡器后, 在关键路径中加入额外的流水线以提高整个系统的工作频率, 实现了基于 Cyclone II 的 EP2C35 芯片的 FPGA 设计.

自适应均衡器的硬件结构仿真如图 3 所示。

图 3 为自适应算法采用 ModelSim 工具的仿真图, 其中 x_{in} 为输入信号, d_{in} 为期望信号, y_{out} 为输出信号, e_{out} 为输出误差信号, emu 为补偿因子, d 、 x_0 、 x_1 、 f_0 、 f_1 为存放系数的寄存器, p_0 、 p_1 、 $xemu_0$ 、

$xemu_1$ 为存放乘积的寄存器, y 、 e 为存放输出和误差的寄存器, $sxty$ 、 sxt_d 为存放期望和输出的寄存器。由于自适应均衡器的更新路径较长, 因此采用流水线技术和分块实现的自适应均衡器, 充分利用触发器资源从而大幅度提高了系统的工作速度。

Signal	Value
/test_fir_lms/lms/clock	1
/test_fir_lms/lms/rst	1
/test_fir_lms/lms/x_in	22
/test_fir_lms/lms/d_in	240
/test_fir_lms/lms/e_out	65487
/test_fir_lms/lms/y_out	5154
/test_fir_lms/lms/f0_out	173
/test_fir_lms/lms/f1_out	159
/test_fir_lms/lms/x0	22
/test_fir_lms/lms/x1	21
/test_fir_lms/lms/f0	173
/test_fir_lms/lms/f1	159
/test_fir_lms/lms/d	240
/test_fir_lms/lms/emu	234
/test_fir_lms/lms/p0	3255
/test_fir_lms/lms/p1	2840
/test_fir_lms/lms/xemu0	4977
/test_fir_lms/lms/xemu1	4740
/test_fir_lms/lms/sxty	40
/test_fir_lms/lms/sxt_d	65520
/test_fir_lms/lms/y	5154
/test_fir_lms/lms/e	65487

图 3 自适应均衡器仿真图

4 结论

本文提出的改进 BP 网络自适应均衡器实时性高, 仅需要迭代 20 次左右就可完成收敛。该算法通过初始化预处理和增加协调器来自适应的调整权值, 使得 BP 网络从原来全局反向传播式网络变成动态地局部反向传播式网络, 加快了网络的收敛速度。最后根据 BP 网络的快速并行能力和强有力的学习能力, 采用分块实现和流水线技术完成 FPGA 设计, 进一步提高自适应均衡器的参数灵活性及适应性, 这在数字通信领域中具有广泛的应用前景。

5 参考文献

- [1] Simon Haykin. 自适应滤波器原理 [M]. 郑宝玉, 译. 北京: 电子工业出版社, 2006: 159-179.
- [2] Wu Bingyan, Chen Qifa, Cheng Shixin. Performance of wavelet-transform-domain adaptive equalizer [J]. Journal of Southeast

University: English Edition, 2002, 18(1): 13-18.

- [3] 余丹, 潘奕娆, 丁树良, 等. 计算机化自适应诊断测验新的选题策略 [J]. 江西师范大学学报: 自然科学版, 2011, 35(5): 548-550.
- [4] 冯文江, 任志勇. 误差反馈的正交小波均衡器结构与算法研究 [J]. 系统仿真学报, 2007, 19(20): 4762-4764.
- [5] 孙兰清, 葛临东. 一种新的变步长常模盲均衡算法 [J]. 计算机仿真, 2008, 25(3): 107-110.
- [6] 杨家兴, 陈文. 神经网络自适应均衡器研究 [J]. 信息工程学院学报, 1994, 13(2): 9-17.
- [7] 贾丽会, 张修如. BP 算法分析与改进 [J]. 计算机技术与发展, 2006, 16(10): 101-107.
- [8] 吴梅, 杨华东, 林涛. 非线性系统的神经网络自适应滤波器 [J]. 计算机仿真, 2003, 20(3): 63-64.
- [9] 李辉, 蔡敏, 李宇, 等. 基于自适应粒子群优化算法的神经网络的优化研究 [J]. 江西师范大学学报: 自然科学版, 2010, 34(6): 632-635.
- [10] 金健, 陈涛. 采用 FPGA 实现基于 LMS 算法的自适应均衡器的设计研究 [J]. 航空电子技术, 2007, 38(4): 12-15.
- [11] 郭业才, 束南山. 基于 FPGA 的自适应均衡器设计与实现 [J]. 安徽理工大学学报: 自然科学版, 2007, 27(4): 33-36.

(下转第 316 页)