

基于 NIOS 的 ANN 语音识别系统^①

ANN Speech Recognition System Implemented in NIOS

罗 希 刘锦高 (华东师范大学 信息科学与技术学院 上海 200241)

摘 要: 现在嵌入式系统对语音识别的需求越来越清晰, 比如移动通信设备和智能家居等。目前大多数实用语音识别系统都要求强大的 CPU 计算能力, 为了平衡嵌入式系统计算能力的不足, 采用 SOPC 思想, 在 Altera 公司的 NIOS 软核上用软件实现系统程序流程控制, 在 FPGA 上用硬件实现高强度的数学计算。这种语音识别系统的效率相比纯软件实现的系统有较大提升, 比较适合在嵌入式系统中推广。把 FFT 在 FPGA 上以硬件实现, 流程控制和模式匹配则采用软件完成。实验结果证明此软硬结合的思路有理想的效果。

关键词: NIOS SOPC FFT 硬件实现 ANN 语音识别

目前家用电脑上的大词汇量语音识别系统已经能够较理想的按人类正常语速完成语音录入的功能, 电影《谎言之躯》也很好的展示了这一点。在计算能力略显不足的嵌入式系统中, 特别是在识别词汇量需求不大的情况下, 可以考虑较简单有效的识别算法和转换计算任务到硬件的思路, 即所谓合适才是最好的。

1 引言

如何描述一段语音片段, 是语音识别的第一步。一般情况下通过对采样后的声音片段提取一些特征参数来实现描述。常见的声音特征包括: 短时能量(Short Time Energy), 过零率(Zero Cross Rates), 短时频谱包络(Short Time Spectral Envelope), 线性预测系数(Linear Predictive Coefficients, LPC), 梅尔倒谱频谱系数(Mel-Frequency Cepstral Coefficients, MFCC)等。提取到这些语音信号的特征参数后, 接下来要考虑如何来对这些特征参数和参考模板进行匹配, 比较成熟的方法是: 隐马尔科夫模型(HMM)和人工神经网络(ANN)。考虑本文系统词汇量要求不大, 且在嵌入式系统中实现, 所以选择了较为简单的反向传播人工神经网络(BP ANN)。

2 设计方案

整个系统的逻辑流程如图 1 所示:

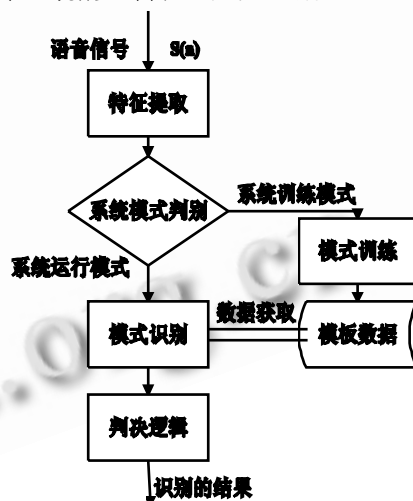


图 1 系统逻辑流程图

2.1 语音端点检测

系统采用 8KHz 采样, 端点检测用过零率和短时能量特征实现。

2.1.1 信号分帧和参数计算

对信号进行分帧之前, 为了消除低频干扰, 特别是工频干扰, 先让信号通过一个预加重滤波器 $1-0.937Z^{-1}$ 。

^① 基金项目: 上海市科委重点攻关项目(075115002)

收稿时间: 2009-03-18

为了后续处理 FFT 计算时的方便, 选用 256 个点为一帧, 帧移为 80。

为了简化系统, 选用一帧信号的绝对值总和为短时能量。

$$M = \sum_{n=0}^{256} |x(n)|$$

为了防止随机波动, 过零率的计算还要考虑最低门限。

$$zcr = \sum_{n=0}^{255} [x(n) * x(n+1) < 0] \& [(|x(n)| - |x(n+1)|) > Min]$$

2.1.2 端点检测状态机

端点检测程序类似施密特触发器的滞回曲线。为短时能量和过零率两个参数分别设定高低两个门限。端点检测状态机有三个状态: 静音段, 过渡段, 语音段。初始化为静音段, 在静音段中, 任何一个参数超过了低门限, 则标记起始点, 并进入过渡段。如果两个参数在过渡段中小于低门限, 判断为噪音, 回到静音段。如果有一个参数超过高门限则进入语音段。进入语音段后, 若两个参数均小于低门限则开始判断是否停顿。如果小于低门限的帧数在认可的停顿帧数范围内, 则判断为语音信号的自然停顿, 继续停留在语音段, 否则判断为语音结束, 标记结束点。若最后获取的语音帧数少于系统认可的最小值, 仍然认为是噪音。

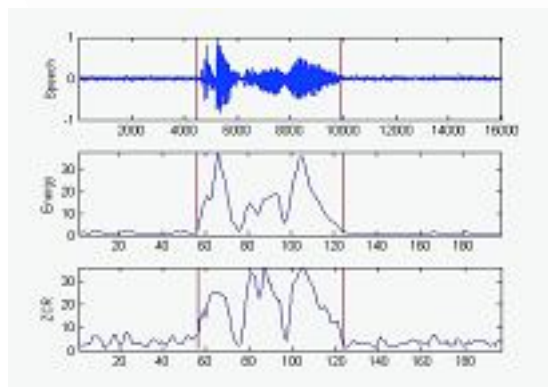


图 2 语音 Continue 的端点检测结果

2.2 MFCC 参数提取

语音识别系统从某种程度上来说是在模仿人的听觉系统, 而人的听觉系统又是一个特殊的非线性系统。研究表明, 人耳对声音的感知强度和声音频率的对数成正比。于是引出了 Mel 频率的概念, 在 Mel 频域中,

人耳对语音的感知强度与 Mel 频率成正比。

Mel 频率 f_{mel} 和线性频率的转换关系定义为^[1]:

$$f_{mel} = 2595 \log_{10} \left(1 + \frac{f}{700} \right)$$

MFCC(Mel-scaled Frequency Cepstrum Coefficient)就是在 Mel 频域上提出来的。

把原始信号通过 FFT 得到线性频谱后, 再通过一组带通滤波器转换成 Mel 频率下的功率谱。这组带通滤波器在线性频谱上是均匀分布的 Hamming 窗滤波器, 但是在 mel 频域上均匀分布。F(m)为滤波器组中第 m 个滤波器的中心频率, 在频率较低的时候 F(m)间隔比较小, 随着频率的增大, 间隔也增大, 这正好满足了 Mel 频率和线性频率的转化关系。

实现的流程为:

- ① 为了克服吉布斯现象, 数据先通过 Hamming 窗。
- ② 利用 FPGA 上的 FFT 硬件模块, 得到线性功率谱。
- ③ 通过 Mel 滤波器组, 转换成 Mel 频率下的功率谱。
- ④ 把每个滤波器的输出叠加, 24 个滤波器得到 24 个参数, 这就是每帧信号的初步语音特征参数。
- ⑤ 对这 24 个参数取对数, 然后进行离散余弦变换(DCT)转换到倒谱域, 得到 12 个参数。

$$c(k) = \sum_{m=0}^{24} s(m) \cos(\pi k (m + 1/2)/24), k \in [1, 12]$$

- ⑥ 最后把这 12 个参数通过倒谱提升窗口, 就得到了这帧信号的 MFCC 参数。

窗口函数采用:

$$W_m = 1 + \frac{k}{2} \sin\left(\frac{\pi m}{k}\right), k \in [1, 12]$$

最后借鉴线性归一化把特征参数归一化到统一的长度。获得的参数为 M 帧{R(1), R(2), ..., R(M)}, 归一化到 N 帧^[2]。

$$R(N) = \begin{cases} R(m), n=m=1 \\ \left(1 - \frac{nM}{N} + m\right)R(m) + \left(\frac{nM}{N} - m\right)R(m+1), m \leq \frac{nM}{N} \leq m+1, n=2, 3, \dots, N \end{cases}$$

2.3 神经网络

反向传播神经网络至 1986 年 Rumelhart 等提出以来影响相当广泛, 在识别模板库不是很大的情况下,

依然在很多领域受用。

BP 算法用一句话概括就是：从后向前(反向)逐层传播输出层的误差，以间接算出隐层误差。算法分为两个阶段：第一阶段(正向计算)输入数据从输入层经过隐层逐层计算各单元的输出值；第二阶段(误差反向传播)输出误差逐层向前算出隐层各单元的误差，并用此误差修正前层权重。

学习过程为^[3]：

- ① 初始化权重：随机的给全部权重赋以初始值，一般初始值范围在-1~1 之间；
- ② 给定输入 $X = \{x_i\}$ 和目标输出 $Y = \{y_j\}$ ；
- ③ 逐层计算实际输出 $O = \hat{Y} = \{\hat{y}_j\}$ ；
- ④ 修正权重：从输出层开始，将误差信号反方向传播，通过修正各权值使误差最小。

对输出层计算 δ_j , $\delta_j = (y_j - \hat{y}_j) \hat{y}_j (1 - \hat{y}_j)$,

从后向前计算各隐层 δ_j , $\delta_j = \hat{y}_j (1 - \hat{y}_j) \sum w_{jk} \delta_k$, 其中 w_{jk} 是从前一层 j 节点到后一层 k 节点的权重。计算并保存各权重的修正量, $\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) + \eta \delta_j y_i$,

其中 α 是惯性项系数，其值在很多应用中取 0.9~1 之间， $\alpha \geq 1$ 时不收敛； η 是步长，对收敛性影响很大，且对于不同问题其最佳值相差也很大，通常取 0.1~3 之间；最后，修正权重 $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$,

- ⑤ 当循环到误差达到精度要求后输出结果，否则回到②。

3 系统实现

片上系统由 NIOS 软核 CPU，FFT 模块，音频芯片 WM8731 的接口模块，通用的 KEY 和 LCD 接口，FLASH 模块接口等组成。

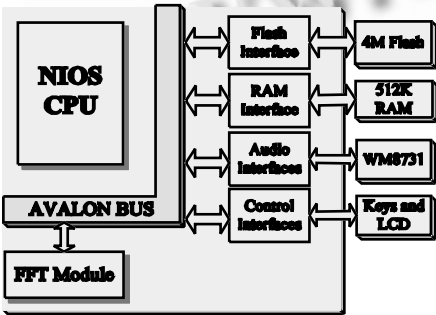


图 3 系统结构图

整个系统包括一块 FLASH 芯片，用于存储 FPGA 描述文件，NIOS 软核程序和系统初始化所需数据。

SDRAM 作为系统内存。音频芯片 WM8731，其采样速率从 8kHz-96kHz 可根据外部时钟和内部的分频系数寄存器来决定。CPU 通过 I2C 总线读取 FIFO 中的数据。

FFT 采用 ALTERA 提供的 IP 核实现，在 IP 核的外面做了一个接口，使其能挂在 AVALON 总线上，从而让 NIOS 能够方便访问其数据。

FFT 的输入时序：

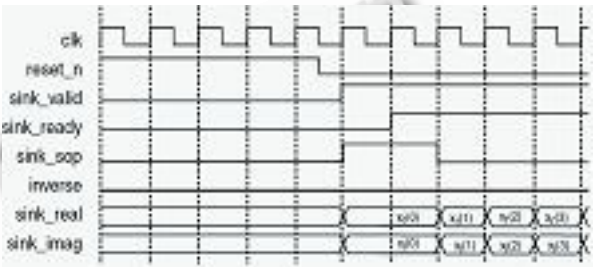


图 4 FFT 模块输入时序

其中只有 sink_ready 是输出,由 FFT 模块给出,用于响应输入的 sink_valid 信号。

sink_valid 用以表明外部有稳定的数据待输入，sink_sop 的有效表明这是输入数据帧的第一个数据，其时长保持和第一个数据的时长一样，以后依次输入其它数据。

FFT 的输出满足以下时序：

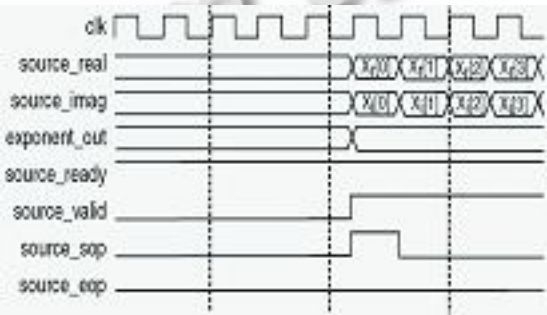


图 5 FFT 模块输出时序

其中只有 source_ready 是输入，用以响应 source_valid 信号，让 FFT 模块开始输出数据。

在硬件模块中判断 source_ready 是否有效，如果无效，就和软件做一个协议，把一个表明输出数据无效的信号放在总线上，这个信号不能出现在数据中。如果有效，则把数据的虚部放在总

(下转第 165 页)

线的高 16 位上, 实部放在总线的低 16 位上, 供 NIOS 读取。

4 总结

整个系统在 FPGA 上用 NIOS 实现, 其按需定制的特性使得硬件设计灵活可靠, 将来硬件部分如需改动也会相当方便。

本文作者创新点: 很好的结合了软件的控制优势和硬件的快速并行计算能力, 节约了单独的 CPU 成本。率先在 NIOS 上利用 ANN 实现完整实用的语音识别系统。

整个系统运行良好, 目前已在智能家居项目中承担了对用户语音命令输入进行识别的任务。

参考文献

- 1 何强, 何英. MATLAB 扩展编程. 北京: 清华大学出版社, 2002. 330 - 339.
- 2 Lee L, Rose RC. Speaker normalization using efficient frequency warping procedures. IEEE Int. Conf on Acoustics, Speech and Signal Processing. Atlanta. 1996. 353 - 356.
- 3 边肇祺, 张学工. 模式识别. 北京: 清华大学出版社, 2000. 250 - 271.
- 4 Rabiner L, Juang BH. Fundamentals of Speech Recognition. Prentice Hall PTR, 1993. 11 - 54.