

基于 FPGA 的卷积神经网络浮点激励函数实现

李施豪, 应三丛

(四川大学 计算机学院 视觉合成图形图像技术国家重点学科实验室, 四川 成都 610065)

摘 要: 卷积神经网络因深度学习概念的提出再一次被研究人员所重视, 激励函数是卷积神经网络的一个重要组成部分, 选取了 sigmoid 函数作为实验对象, 讨论了当前几种可行的逼近方法, 最终采用分段四阶多项式拟合 sigmoid 函数. 在 FPGA 上使用 Verilog 硬件描述语言设计了并行电路, 并采集了数据集进行 FPGA 与 CPU 版本 caffe 库进行运算效率对比. 实验结果表明, 此种方法误差小效率高, FPGA 在深度学习领域有着广阔的应用前景.

关键词: 卷积神经网络; 激励函数; FPGA; 多项式逼近

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2017)10-0105-05

The Implementation of Activation Function of CNN Based on FPGA Using Single Precision Floating-Point-Unit

LI Shi-hao, YING San-cong

(National Key Laboratory of Fundamental Science on Synthetic Vision,
School of Computer Science, Sichuan University, Chengdu 610000, China)

Abstract: Convolutional Neural Network (CNN) was gotten attention because of the presentation of Deep-Learning. It has large research value for Deep-Learning in using FPGAs. The activation function is the most necessary part of CNN. In this paper, Sigmoid function was chosen as the experiment object. The approximation ways of Sigmoid Function were listed and analysed, the piecewise forth-order approximation was the best way to fit Sigmoid Function. Parallel calculation circuit was designed by Verilog HDL on FPGAs, collecting dataset and inputing them to FPGA Platform and CPU Platform. The experiment result set forth that this solution had high efficiency and low error with rosy prospect about Deep-Learning and FPGA.

Key words: convolutional neural network; activation function; FPGA; polynomial approximation

1 引言

卷积神经网络是深度学习领域一种重要网络模型. 在集成电路性能发展迅猛的今天, 计算机处理速度不断提升, 神经网络的发展步入了一个新的阶段^[1]. 当前的网络模型主要依靠软件实现, 运行于操作系统之上. 其功耗、效率等方面的缺点日益凸显, 改良并完善网络模型的需求日益增长^[2].

当前的神经网络模型都是基于 C/C++、Python、Matlab 以及 Lua 等串行代码实现. 在数据量如此庞大的现在, 其性能瓶颈尤为明显^[2]. 而且, 嵌

入式平台对于深度学习的需求日益突出. 将深度学习移植到嵌入式平台是当前深度学习的一个重要研究方向^[2]. 本文尝试采用 FPGA 芯片将神经网络中一个激励函数模块进行硬件描述语言编码, 并将其运行在硬件电路.

用户可以通过硬件描述语言将设计的数字系统电路逻辑烧录到 FPGA 芯片中实现并行计算. 使用 FPGA 作为卷积神经网络的实现载体, 不仅能提高网络实时性, 还具有低功耗、可重构等优点, 有助于深度学习推广到嵌入式平台运行.

本文选取了卷积神经网络的激励函数部分, 在

FPGA 上设计了并行计算电路. 鉴于 CPU 的 caffe 库^[3]是单精度浮点实现,故本文在 FPGA 上依然采用相同的单精度浮点运算,并在其之后又对研究工作进行了展望.

2 卷积网络激励函数简介

神经元模型是卷积神经网络的最基本模型,如图 1 所示. 其中,激励函数是卷积网络神经元中最重要的组成部分. 函数的作用是将输入进行一种空间的转化,将网络中的输出值变换到一个有限的范围内.

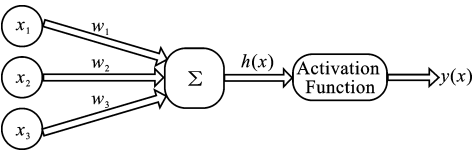


图 1 单个卷积神经网络神经元模型

计算公式如下:

$$W = [w_1, w_2, \cdots, w_n]^T \tag{1}$$

$$X = [x_1, x_2, \cdots, x_n]^T \tag{2}$$

$$h(x) = W^T X \tag{3}$$

$$y(x) = Act(h(x)) \tag{4}$$

卷积神经网络的激励函数具有非线性、可微性、单调性以及有限的输出范围. 神经网络提出至今,研究人员提出了多种可作为神经网络的激励函数,如线性函数 $y=x$ 、sigmoid 函数、双曲正切函数、反正切函数、ReLU 函数等等. 其中 sigmoid 函数是在业界得到广泛认可,可应用于实际工程模型的神经网络激励函数. 其函数图像如图 2 所示,函数值范围 $(0,1)$,输入值为 0 时函数值为 0.5. 函数的导数如图 3 所示,其值在自变量为 0 处最大,即上升速度最快. 函数两端无限趋近于 0 和 1,函数变化率无限趋近于 0.

3 FPGA 实现激励函数

FPGA 实现各种数学函数求值,通常的做法有泰勒级数展开、Look-Up-Table 查找表法、多项式插值逼近、CORDIC 算法等.

Taylor 公式展开法需要较高的阶数,需要大量级联的乘法器. 级联乘法器要实现复用促进并行化的工程实现难度和开销过大,如今大数据处理的时代此种方法不适用范围在 FPGA 上.^[4]

LUT 方法最为直接,也是最易实现的方法. 将激励函数根据自变量的不同取值,存储对应的函数

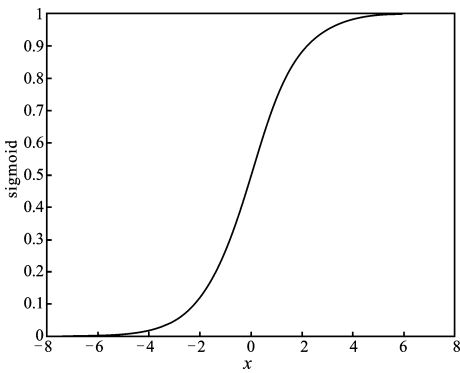


图 2 sigmoid 函数图像

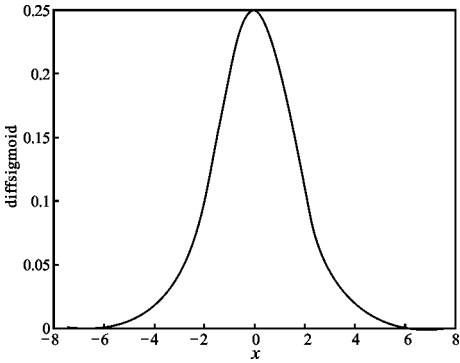


图 3 sigmoid 导数图像

值到 FPGA 的存储器中. 自变量取值越精细,表内的输出值越高. 可用于所有函数,实现方式简单易行,并且只需一个时钟周期便可得到结果. 但是若实际情况要求精度高,则表的尺寸会过大,甚至溢出存储器可容纳的范围,资源的开销也不容小觑.

单一多项式插值拟合法依然需要较高的阶数. 经过 MATLAB 以及之前的研究工作表明,至少需要六阶多项式才能得到较好的拟合函数. 大量乘法器以及多个加法器,使得单一多项式插值拟合法引起较大的输出延迟以及 FPGA 的片上资源的消耗.^[4]

CORDIC 算法在 FPGA 偏上资源开销较为合理,但是其精度较差,定点实现和浮点实现的结果精度差异过大. 在神经网络中收敛速度慢甚至不收敛的问题一直以来仍未得到解决.^[5]

本文所使用的方法是分段四阶多项式逼近的方法,并且采用浮点运算器实现卷积神经网络的 sigmoid 激励函数. 将 sigmoid 分为多段,每段采用四阶多项式逼近. 这样既解决了资源开销的问题,在实现方法上也简单易行. 浮点运算与理论值基本一致,偏差可忽略不计,计算所耗费的时间仅一个时钟周期.

除此以外,由于 sigmoid 函数关于 $(0, 0.5)$ 对称,因此,只需计算 $x \in [0, +\infty)$ 区间的函数输出值. 在输入大于约 7.5 时,函数输出接近于 1,其与 sigmoid 函数理论值相差不大于 0.000 1 数量级.

4 激励函数实现方案

4.1 Matlab 拟合

Matlab 工具箱提供了 polyfit 函数,用户可用其进行插值拟合并限定最高阶的次数. 函数的返回值为按照自变量阶数降序排列的每一阶系数. 本文将正半轴分为 11 个子区间,各区间的拟合四阶多项式如表 1 所示.

表 1 四阶拟合 sigmoid 函数多项式

区间	拟合多项式
$[0, 0.7)$	$f(x) = 0.003\,27x^4 - 0.022\,754x^3 + 0.000\,486x^2 + 0.249\,953x + 0.500\,001$
$[0.7, 1)$	$f(x) = 0.005\,305x^4 - 0.027\,078x^3 + 0.003\,981x^2 + 0.248\,666x + 0.500\,185$
$[1, 1.7)$	$f(x) = 0.003\,836x^4 - 0.020\,322x^3 - 0.007\,728x^2 + 0.257\,716x + 0.497\,556$
$[1.7, 2)$	$f(x) = 0.001\,461x^4 - 0.005\,048x^3 - 0.044\,732x^2 + 0.297\,758x + 0.481\,224$
$[2, 2.7)$	$f(x) = -0.000\,104x^4 + 0.007\,847x^3 - 0.084\,711x^2 + 0.353\,002x + 0.452\,518$
$[2.7, 3)$	$f(x) = -0.000\,726x^4 + 0.014\,219x^3 - 0.109\,200x^2 + 0.394\,898x + 0.425\,596$
$[3, 3.7)$	$f(x) = -0.000\,768x^4 + 0.014\,682x^3 - 0.111\,101x^2 + 0.398\,323x + 0.423\,320$
$[3.7, 6)$	$f(x) = 0.001\,299x^3 - 0.023\,021x^2 + 0.139\,352x + 0.709\,786$
$[6, 7.5)$	$f(x) = -0.000\,049x^4 + 0.001\,523x^3 - 0.018\,025x^2 + 0.096\,6x + 0.801\,317$
$[7.5, 9)$	$f(x) = 0.000\,045x^3 - 0.001\,245x^2 + 0.011\,653x + 0.963\,177$
$[9, +\infty)$	$f(x) = 1$

当输入为负时,因 sigmoid 函数具有对称性,可得:
$$f(x) = 1 - f(-x), x < 0 \tag{5}$$

如图 4 所示,输入值在 0 到 9 范围内,采样点间隔为 0.000 1. 计算拟合多项式的函数值与 sigmoid 函数理论值的误差绝对值. 由图像可知,在子区间边界处出现误差极大值. 尽管如此,误差极大值数量级也仅达到 10^{-4} .

4.2 FPGA 工程实现

本文的实验平台为 xc7vx330t FPGA 平台实现神经网络浮点激励函数. 目前广泛使用的神经网络模型均涉及到大量浮点数运算^[6-9],而 Xilinx 提供了

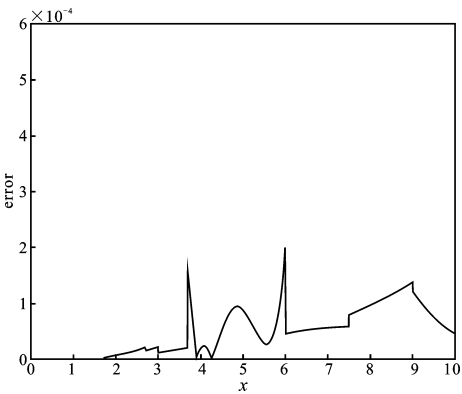


图 4 拟合函数误差曲线

性能出色的浮点运算 IP core. 其所使用的是 AXI4-Stream 总线协议及 IEEE-754 浮点数标准,能够进行高速流数据传输并对浮点运算单元实现分时复用,适用于神经网络运算.

单个二次多项式硬件实现方案如图 5 所示.

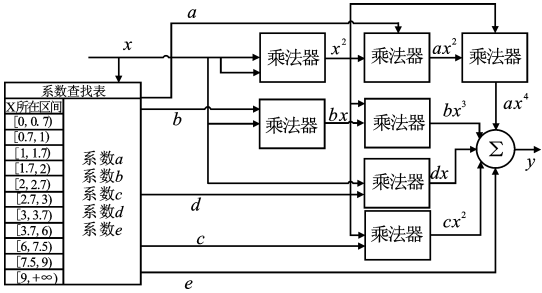


图 5 激励函数运算单元

其多项式表示为 $y = ax^4 + bx^3 + cx^2 + dx + e$. 激励信号 x 输入到系数查找表中进行区间判定并取得其对应区间的多项式系数,之后将信号 x 及其系数发送到总线上. 由三个二输入乘法器得到 ax^4 , 并且可以复用数据项 x^2 . 因此,经过两个乘法器便可得 bx^3 , 经过一个乘法器可得 cx^2 和 dx . 最后将每项分别输入到加法器中计算出最后 sigmoid 函数拟合值. 正半轴的 11 个子区间所对应的自变量系数由 LUT 查找表的方式存储. 因激励函数关于 $(0, 0.5)$ 对称,负半轴用式(5) 换算即可.

5 实验结果分析

本文基于 Xilinx Virtex-7 系列 xc7v×330t FPGA 板载晶振为 100 MHz 的实验平台实现浮点激励函数,并编写仿真测试文件. 使用 Xilinx Vivado 与 Matlab 对浮点激励函数进行联合仿真,以及编译运行业界广泛认可的 caffe 库的 sigmoid 激励函数进行测试.

由于集成了浮点运算单元,FPGA 浮点运算结果如表 2 所示.在保留小数点后 4 位的情况下,误差绝对值在 0.000 1 及其以下的数量级.而且,运算速度上 FPGA 具有较明显的优势.本文 CPU 采用 4 核心 8 线程的 Intel® Xeon® CPU E3-1231 v3 @ 3.40 GHz 作为测试平台.表 3 所示为 CPU 的 caffe 库输入 8 000 万条数据所需的运行时间与同数据集下的 FPGA 运行时间对比数据.

表 2 激励函数随机点的 FPGA 逼近结果

输入值	FPGA 计算值	CPU 理论值	误差绝对值
0.23	0.557 2	0.557 2	<0.000 1
0.65	0.657 0	0.657 0	<0.000 1
0.79	0.687 8	0.687 8	<0.000 1
0.95	0.721 1	0.721 1	<0.000 1
1.43	0.806 9	0.806 9	<0.000 1
1.67	0.841 6	0.841 5	0.000 1
2.28	0.907 2	0.907 2	<0.000 1
2.51	0.924 9	0.924 8	0.000 1
2.82	0.943 8	0.943 7	0.000 1
2.90	0.947 8	0.947 8	<0.000 1
3.01	0.953 0	0.953 0	<0.000 0
3.69	0.975 6	0.975 6	<0.000 1
4.83	0.992 1	0.992 1	<0.000 1
4.98	0.993 3	0.993 2	0.000 1
5.26	0.994 8	0.994 8	<0.000 1
5.78	0.996 9	0.996 9	<0.000 1
6.13	0.997 8	0.997 8	<0.000 1
7.39	0.999 3	0.999 4	0.000 1
7.66	1.000 0	0.999 5	0.000 5

表 3 CPU 与 FPGA 平台的处理效率对比

平台	数据量	运行时间/ μ s
Xeon®E3-1231 v3	800 000 00	306 301 4
Xilinx Virtex-7 FPGA	800 000 00	266 667

Vivado 仿真波形如图 6 所示.Xilinx 提供的单精度浮点运算 IP 性能十分出色并且实现了流水线计算,具有高速高效的并行运算特性.因 Verilog 语法所限,图中输入 x 为 2.82 的 IEEE-754 浮点十六进制数表示 40347AE1,其十进制信号 x_d 为 282.电路输出 result 在下一时钟周期得到,其值浮点十六进制表示为 3F719A9E,十进制信号 result_d 为 0.943 765 5 $\approx$ 0.943 8.

由图 7 所示的综合资源占用报告,可以发现 IO 资源占用较大.在调用浮点运算单元时,若实际应用没有很强的实时性要求,可适当调高运算器的延时,可减少浮点运算单元对片内 IO 资源的占用.

6 结束语

本文介绍了卷积神经网络中常用的几种激励函数的 FPGA 实现方案,并选择当前最广泛、普遍的 sigmoid 函数作为研究对象.在 FPGA 中采用分段逼近的方法,并设计了运算电路,使用与 CPU 版的 caffe 库相同的单精度浮点运算单元进行浮点数运算,电路性能达到一个时钟周期即可得到输出结果.通过 FPGA 的并行运算效率与当前普遍得到认可的 caffe 开源框架进行对比.实验结果说明,在相同数据集下 FPGA 在并行度较高的浮点运算方面使用 100 MHz 的时钟能够对 4 核 8 线程的 CPU 实现 11 倍以上的加速.因此,FPGA 在深度学习领域具有广阔的应用前景.

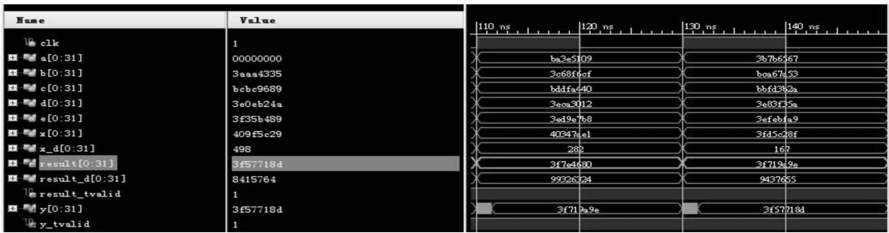


图 6 激励函数 FPGA 仿真

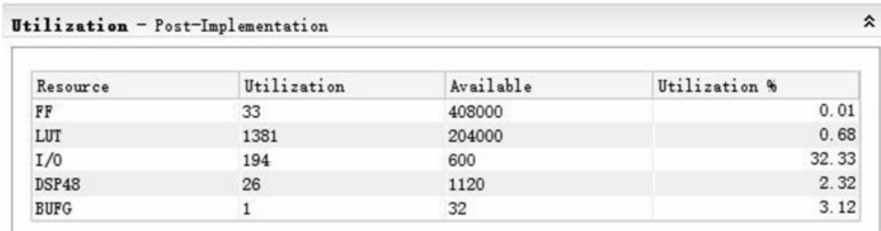


图 7 FPGA 并行浮点激励函数综合的资源

参考文献:

[1] LeCun Yann, Yoshua Bengio, Geoffrey Hinton. Deep learning[J]. Nature,2015,521(7553): 436-444.

[2] 李彦冬,郝宗波,雷航. 卷积神经网络研究综述[J]. 计算机应用,2016,36(9):2508-2515.

[3] Jia Yangqing, Evan Shelhamer, Jeff Donahue, et al. Caffe: Convolutional architecture for fast feature embedding[C]// Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014.

[4] Vishnu P Nambiar, Mohamed Khalil-Hani, Riadh Sahnoun, et al. Hardware implementation of evolvable block-based neural networks utilizing a cost efficient sigmoid-like activation function[J]. Neurocomputing, 2014(140): 228-241.

[5] Manish Panicker, BabuC. Efficient FPGA implementation of sigmoid and bipolar sigmoid activation functions for multilayer perceptrons[J]. IOSR Journal of Engineering, 2012, 2(6): 1352-1356.

[6] EMMANUEL ADETIBA, IBIKUNLE F A, DAR-AMOLA S A, et al. Implementation of efficient multi-layer perceptron ANN neurons on field programmable gate array chip[J]. International Journal of Engineering & Technology IJET-IJENS, 2014, 14 (1); 151-159.

[7] Li Bingzhe , Hassan Najafi M, David J Lilja. Using stochastic computing to reduce the hardware requirements for a restricted boltzmann machine classifier[C] // The 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. Monterey, California, USA,2016:36-41.

[8] Sahil Abrol, Rita Mahajan. Implementation of single artificial neuron using various activation functions and XOR gate on FPGA chip[C]//IEEE:2015 Second International Conference on Advances in Computing and Communication Engineering. IEEE,2015:118-123.

[9] Djalal Eddine, Khodja Stéphane Simard, Implementation of optimized approximate sigmoid Function on FPGA circuit to use in ANN for control and monitoring [J]. CONTROL ENGINEERING AND APPLIED INFORMATICS, 2015, 17(2): 64-72.

作者简介:
李施豪 男,(1992-),硕士研究生.研究方向为 VLSI 与人工智能应用.

应三丛(通讯作者) 男,(1975-),博士,副教授.研究方向为计算机应用. E-mail:yingsancong@163. com.

(上接第 104 页)

[10] 刘迎澍,杨旭辉. 面向智能家居的实时电能信息采集系统设计[J]. 电子设计工程,2016,24(14):122-125.

作者简介:

赵玉芹 女,(1983-),硕士研究生,讲师.研究方向为智能信息处理与模式识别. E-mail:18538216930@163. com.