

基于FPGA的移动机器人SNNs走廊场景分类器

王睿轶^{1,2,3}, 王秀青^{1,2,3}, 刘万明⁴, 王永吉⁵, 叶晓雅^{1,2,3}

(1. 河北师范大学 计算机与网络空间安全学院, 河北 石家庄 050024;

2. 河北省网络与信息安全重点实验室, 河北 石家庄 050024;

3. 河北省供应链大数据分析与安全工程研究中心, 河北 石家庄 050024;

4. 河北师范大学 中燃工学院, 河北 石家庄 050024;

5. 中国科学院 软件研究所, 北京 100190)

摘要:神经形态芯片是类脑计算的重要研究内容之一,神经网络的硬件实现是神经形态芯片实现的基础。具有生物似真性的脉冲神经网络(Spiking Neural Networks, SNNs),通过尖脉冲(Spikes)传递时空信息,更适于用硬件实现,是实现类脑计算的主要工具之一。该文提出一种基于FPGA的移动机器人SNNs走廊场景分类器:将移动机器人超声传感器信息进行脉冲编码后输入到SNNs走廊场景分类器中,通过FPGA分类器的脉冲输出模式来判断机器人所处的走廊场景,从而提高机器人的环境感知能力和自主性。详细讨论了脉冲积分点火神经元模型的FPGA实现原理,以及基于此神经元模型的SNNs走廊场景分类器的硬件实现方案,仿真及实验结果证明了所提基于FPGA的移动机器人SNNs走廊场景分类器的有效性。所提走廊场景分类器不受光照条件的影响,需要的传感器测量信息少,FPGA硬件资源占有率低(LE的利用率仅10%),分类速度快、准确率高,适于实际应用。该研究不仅可以提高移动机器人的环境感知能力和自主性,而且为硬件实现SNNs提供了有益参考。

关键词:脉冲神经网络;积分点火神经元模型;脉冲编码;现场可编程门阵列;移动机器人;超声传感器

中图分类号:TP183

文献标识码:A

文章编号:1673-629X(2023)12-0032-09

doi:10.3969/j.issn.1673-629X.2023.12.005

Mobile Robots' SNNs Corridor-scene-classifier Based on FPGA

WANG Rui-yi^{1,2,3}, WANG Xiu-qing^{1,2,3}, LIU Wan-ming⁴, WANG Yong-ji⁵, YE Xiao-ya^{1,2,3}

(1. School of Computer and Cyber Security, Hebei Normal University, Shijiazhuang 050024, China;

2. Hebei Provincial Key Laboratory of Network & Information Security, Shijiazhuang 050024, China;

3. Hebei Provincial Engineering Research Center for Supply Chain Big Data Analytics & Data Security, Shijiazhuang 050024, China;

4. China Institute of Gas Engineering, Hebei Normal University, Shijiazhuang 050024, China;

5. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: The neuromorphic-chip is one of the important research aspects of brain-inspired computing, and the hardware implementation of neural networks (NNs) is the basis of neuromorphic-chip. Spiking neural networks (SNNs) with biological plausibility, which convey temporal and spatial information by spikes, are suitable to be implemented with hardware, and are also one of the main tools for brain-inspired computing. A novel SNNs based mobile robots' corridor-scene-classifier implemented by FPGA is proposed. The ultrasonic sensor information of the mobile robot is encoded and input into the SNNs corridor scene classifier, and the corridor scene of the robot is judged by the pulse output mode of the FPGA classifier, so as to improve the environment perception ability and autonomy of the robot. The principle of the Approximation-Spiking IAF neuron model and the implementation of the SNNs corridor-scene-classifier based on Approximation-Spiking IAF by FPGA are discussed in detail. The simulation and experimental results validate the effectiveness

收稿日期:2023-01-31

修回日期:2023-05-31

基金项目:国家自然科学基金面上项目(61673160, 61175059);河北省自然科学基金资助项目(F2018205102);河北省高等学校科学技术研究重点项目(ZD2021063);河北师范大学重点基金(L2019Z11);河北师范大学在读研究生创新能力培养资助项目(CXZZSS2022073);河北师范大学2021年大学生课外学术科技创新项目(CG2021412204634)

作者简介:王睿轶(1998-),男,硕士研究生,研究方向为脉冲神经网络、深度强化学习;通讯作者:王秀青(1970-),女,教授,研究方向为先进机器人技术、脉冲神经网络。

of the proposed mobile robots' corridor-scene-classifier based on FPGA and SNNs. Besides the fast processing speed, the classification results of the proposed method are accurate and not influenced by lighting conditions, the needed amount of sensor data is small, the FPGA resource-conquer-rate is low (the utilization rate of LE is only 10%), which is suitable for practical application. Moreover, the proposed corridor classifier can also improve mobile robots' ability of environmental perception and autonomy, and provides a valuable input for SNNs implemented by hardware.

Key words: Spiking neural networks; integrated-and-fired neuron model; Spiking encoding; field programmable gate array; mobile robot; sonar sensor

0 引言

第三代神经网络——脉冲神经网络 (Spiking Neural Networks, SNNs)^[1-3]通过独立的尖脉冲 (Spikes) 传递信息,将时空信息同时融于网络中,不仅比前两代神经网络具有更好的生物似真性,而且更适于用硬件实现。近年来,作为重要科技前沿的脑科学和类脑智能技术引起了国际学术界的广泛关注,各国投入巨资启动了脑研究计划。类脑计算是脑研究计划的重要方面之一,SNNs 以其独有的生物似真性成为类脑计算的主要工具。神经形态芯片是类脑计算的重要研究内容之一,神经网络的硬件实现是神经形态芯片实现的基础,也是各种神经网络算法工程实用化的关键环节,进行 SNNs 硬件实现研究具有重要意义。基于硬件实现的 SNNs 解决方案,具有信息处理速度快、能耗低,适于实际应用等特点,成为当今研究的热点。

国外在 SNNs 硬件实现方面的研究成果显著。2013 年,瑞士苏黎世大学与联邦理工学院成功研制出基于 SNNs 的脑神经形态芯片^[4]。2014 年,IBM 公司推出运行功耗极低的 TrueNorth 芯片^[5],英国曼彻斯特大学开发了 SpiNNaker 芯片^[6],美国斯坦福大学研制出了神经栅格芯片系统^[7]。2020 年,英特尔发布了神经形态计算系统—Pohoiki Springs,该系统可提供 1 亿个脉冲神经元进行任务处理。2021 年,Nguyen 等人^[8]在芯片上实现了基于脉冲时间依赖可塑性 (Spike-Timing-Dependent Plasticity, STDP) 的深度 SNNs 学习方法,较基于软件的基线 SNNs 学习方案速度提升 2.1 倍、能耗减少 64%。2022 年, Panchapakesan 等人^[9]提出了基于率编码的 SyncNN 方法,该方法相较传统卷积神经网络 (Convolutional Neural Networks, CNNs) 在 MNIST 数据集上的训练速度提高 2.26 倍。

国内虽然对 SNNs 硬件实现的研究起步较晚,但也取得了一定成果。2016 年,浙江大学顾宗华团队设计出国内首款支持 SNNs 的类脑芯片——达尔文芯片^[10],并成功应用于类脑计算和模式识别等任务中。2017 年, Sun 等人^[11]提出了一种用于视觉信息特征提取的脉冲神经网络,并在 FPGA 上进行了实现。2019 年,清华大学类脑计算研究中心施路平团队提出了符合脑科学基本规律、可同时支持 ANNs 和 SNNs 的新型类脑计算架构——异构融合的天机类脑计算芯片架

构^[12]。2020 年, Zhang 等人^[13]提出一种基于 FPGA 应用于图像分类的低成本、高性能的 SNNs 实现方案,与软件实现相比运行速度提高了 908 578 倍。2021 年, Wang 等人^[14]利用加速体系结构提出一种基于 FPGA 的 SNNs,并采用逼近式加法器实现该架构,与前期相关工作相比 ALUT 降低 28%、能耗减少 29%。

SNNs 能同时融入时空信息的特点,使其成功应用于机器人环境感知^[15-16]和控制^[17-21]等领域,如:2018 年, Garcia 等人^[15]基于 STDP 展开了人形机器人视觉注意力的研究。2020 年, Tang 等人^[17]提出一种基于 SNNs 的移动机器人导航控制方法。2021 年, Lu 等人^[18]提出一种基于 STDP 的移动机器人避障策略。2022 年, Azimirad 等人^[19]提出一种基于 CNNs 和 SNNs 的机器人环境感知和控制系统。虽然类脑芯片已取得了一定进展,但 SNNs 神经形态芯片在机器人领域的应用仍有待进一步发展,很多面对机器人的类脑计算方法距离实际工程应用还有较大差距。例如,文献[19]中控制策略算法需要在上位 PC 机中完成,再将控制命令发送给机器人执行。神经形态芯片和集成在芯片中的神经控制器的使用,除提高机器人自主性、加速机器人系统的信息处理和执行速度外,还使整个机器人系统轻量化,更便于实际使用。

现场可编程门阵列 (Field Programmable Gate Array, FPGA) 可以通过数据并行处理和运算加速来提高实际信息处理的能力^[22-24]。该文提出一种基于 FPGA 的逼近式脉冲积分点火 (Integrate-and-Fire, IAF) 神经元模型硬件实现方案,基于 IAF 神经元模型实现了基于 FPGA 的移动机器人 SNNs 走廊场景分类器。此分类器可安置于机器人本体,使用的硬件资源少,结构简单,使用方便。所提基于 FPGA 的 SNNs 走廊场景分类器利用机器人超声传感器采集环境信息,受外界因素影响小,即使在光照不足或黑暗的走廊场景中依然能够准确、快速地进行走廊场景识别。仿真及实验结果验证了所提基于 FPGA 的移动机器人 SNNs 走廊场景分类器的有效性。

1 脉冲神经网络

1.1 积分点火模型

积分点火神经元模型是脉冲神经元模型中广为使

用的一种阈值点火模型,是脉冲响应模型(Spike Response Model, SRM)^[1]的特例。IAF 神经元模型原理如图1和公式(1)所示^[1]。电路中输入电流(I)分为两部分:一部分流向电阻(R);另一部分给电容(C)充电。当电容(C)两侧电压超过阈值(ϑ)时,神经元发放脉冲。

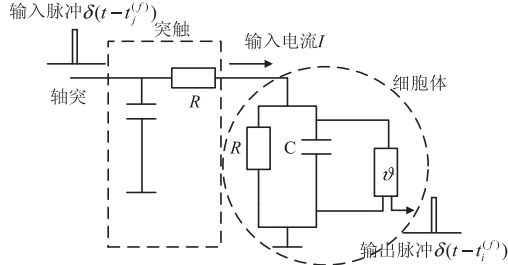


图1 脉冲 IAF 神经元模型原理

$$\tau_m \frac{du_i}{dt} = -u_i(t) + RI(t) \quad (1)$$

其中, u_i 表示神经元 i 膜电压值, $\tau_m = RC$ 是神经元膜时间常数。由公式(1), IAF 模型亦可如式(2)所示^[1]。

$$u_i(t) = \sum_{t_i^f \in F_i} \eta_i(t - t_i^f) + \sum_{j \in \Gamma_i} \sum_{t_j^f \in F_j} \omega_{ij} \varepsilon_{ij}(t - t_j^f) \quad (2)$$

其中, η_i 是神经元对自身输出脉冲的响应; ω_{ij} 是神经网络中突触前神经元 j 对突触后神经元 i 的连接权值; ε_{ij} 是突触神经元 i 接收来自突触前神经元 $j \in \Gamma_i$ 时的脉冲响应; t_i^f 和 t_j^f 是第 i 和第 j 个脉冲神经元输出脉冲的时刻; F_i 和 F_j 分别是突触后神经元 i 和突触前神经元 j 输出脉冲的时间集合; Γ_i 表示与第 i 个突触后神经元相连的突触前神经元集合。公式(2)中 η_i 和 ε_{ij} 核具体形式如公式(3)和(4)所示^[1]:

$$\eta_i(s) = -(\vartheta - u^{\text{rst}}) \exp(-\frac{s}{\tau_m}) H(s) \quad (3)$$

$$\varepsilon_{ij}(s) = \int_0^s \exp(-\frac{s'}{\tau_m}) \frac{1}{\tau_s} \exp(-\frac{s-s'}{\tau_s}) H(s') ds' \quad (4)$$

其中, ϑ 为点火阈值, u^{rst} 为静息期电压, τ_m , τ_s 为时间常数, $H(s)$ 为阶跃函数。

1.2 脉冲时延编码

时延编码(Latency Coding)通过不同的脉冲时延表示输入神经元的刺激信号的强弱^[1]:刺激信号越弱,神经元发放脉冲时刻越晚、时延越长;刺激信号越强,神经元输出脉冲时刻越早、脉冲时延越短。

2 移动机器人 SNNs 走廊场景分类器的 FPGA 实现

2.1 基于 SNNs 的移动机器人走廊场景分类器的基本原理

走廊环境是室内移动机器人常见的结构化场景,

准确感知走廊环境,对实现移动机器人的自主性具有重要意义。文献[21]针对7种常见的结构化走廊场景,提出基于超声传感信息的移动机器人 SNNs 走廊场景分类器(超声传感器工作原理及16个超声传感器在移动机器人本体中的分布见文献[21])。7种常见的结构化走廊场景如图2所示,图2(a)~(g)分别代表走廊场景1~7。SNNs 拓扑结构如图3所示,具体工作原理及相应 FPGA 实现方案见2.2.1~2.2.3。

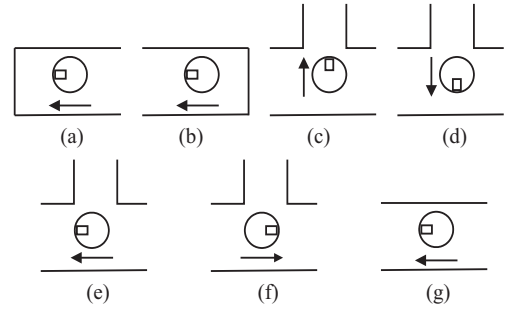


图2 常见的7种走廊场景

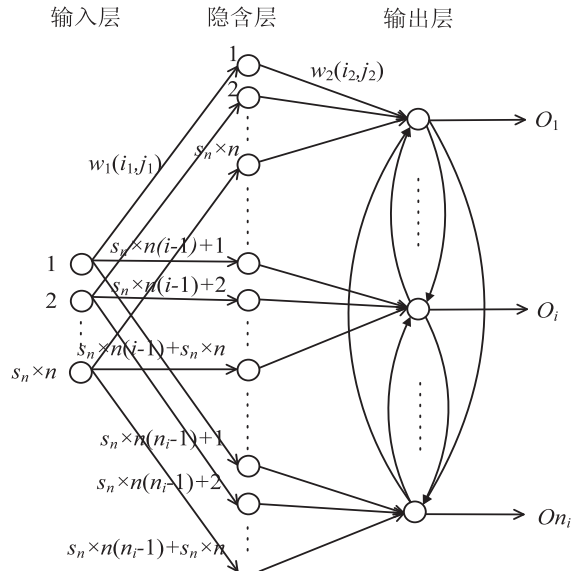


图3 移动机器人走廊场景分类器中 SNNs 拓扑结构

第1层为输入层,由 $s_n \times n$ 个输入神经元组成。其中 s_n 为机器人本体超声传感器环中的超声传感器个数, n 为在某走廊场景中 n 个连续时刻采集的 n 组超声传感器信息。

第2层为隐含层,有 $s_n \times n \times n_i$ 个神经元。第1层神经元到第2层神经元的连接权值均为 $w_1(i_1, j_1) = 1$, 其中 j_1 为输入层神经元序号, $j_1 = 1, 2, \dots, s_n \times n$, i_1 为隐含层神经元序号, $i_1 = 1, 2, \dots, s_n \times n \times n_i$, n_i 为待分类走廊场景个数。

第3层为输出层,有 n_i 个输出神经元。第2层神经元到第3层神经元的连接权值均为 $w_2(i_2, j_2) = 1$, 其中 j_2 为隐含层神经元序号, $j_2 = 1, 2, \dots, s_n \times n \times n_i$, i_2 为输出层神经元序号, $i_2 = 1, 2, \dots, n_i$ 。输出层神经元之间为侧向抑制联结,采用“赢者通吃(Winner-Takes-

All, WTA) ”的原则。

2.2 移动机器人 SNNs 走廊场景分类器的 FPGA 实现原理

利用 FPGA 对文献[21]中移动机器人 SNNs 走廊场景分类器进行硬件实现,具体原理如图 4 所示。将移动机器人采集的超声传感器信息进行脉冲编码后输入所设计的基于 FPGA 的 SNNs 走廊场景分类器中,通过 SNNs 的脉冲输出模式判断移动机器人所处的走廊场景,采用硬件描述语言(Verilog HDL)设计并实现了图 4 各模块的相应功能。

(1) SNNs 输入层硬件实现由输入时延编码模块(input_layer_latency_coding)完成(详见 2.2.1 节);

(2) SNNs 隐含层的硬件实现由均值与方差运算

模块(mean_variance)、浮点开方模块(floating_square_root)和隐含时延编码模块(hidden_layer_latency_coding)3 个功能模块实现(详见 2.2.2 节);

(3) 输出层由逼近式 IAF 神经元模块(approximate_iaf_neuron)实现。该模块实现了基于 FPGA 的逼近式脉冲积分点火神经元模型(详见 2.2.3 节)并输出相应于不同走廊场景的脉冲模式,通过输出脉冲模式识别走廊场景类别;

(4) FPGA 将所得走廊场景的识别结果提供给移动机器人控制器进行下一步任务决策的同时,由场景类别 LCD 输出模块(scene_classification_lcd_output)通过 LCD 屏显示当前机器人所处走廊场景信息。

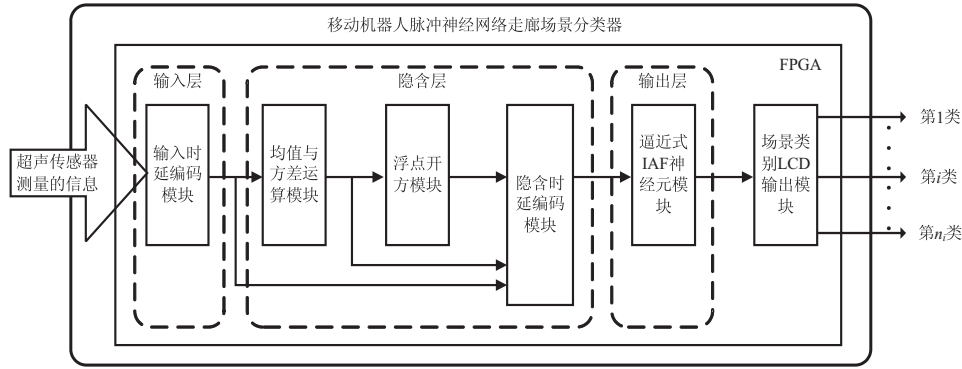


图 4 基于 FPGA 的移动机器人 SNNs 走廊场景分类器的实现原理

2.2.1 输入层的 FPGA 实现

输入时延编码模块原理:

将移动机器人在走廊场景中连续测得的 n 组超声传感信息经过脉冲时延编码后构造融合向量,作为输入层脉冲神经元的输入。

移动机器人超声传感信息编码成脉冲时延编码的具体过程如下:

(1) 超声传感器信息的脉冲时延编码为:

$$D_{ij}(m) = T_{win} \times (1 - \frac{x_{ij}(m)}{\max(x)}) \quad (5)$$

其中, T_{win} 为脉冲编码的时间窗,在此设为 20 ms。 $x_{ij}(m)$ 为超声传感器在第 i 个走廊场景的第 $j(j=1,2,\dots,14)$ 组测量信息中第 $m(m=1,2,\dots,16)$ 个超声传感器的测量值。 $\max(x)$ 为超声传感器的最大测量范围,取值为 4 000 mm。 $D_{ij}(m)$ 为输出的脉冲时延编码。

(2) SNNs 输入层融合输入向量为 D_{ik} 。

$$D_{ik} = [d_{ik}, d_{i(k+1)}, \dots, d_{i(k+n-1)}]^T \quad (6)$$

其中, i 为待分类走廊场景($i=1,2,\dots,7$), k 为某走廊场景中输入数据的组数($k=1,2,\dots,5$), d_{ik} 为第 i 个走廊场景中第 k 个时刻的 16 个超声传感器测量信息的脉冲时延编码($d_{ik} = [D_{ik}(1), D_{ik}(2), \dots, D_{ik}(m)]$), n 为在走廊场景中采用 n 个连续时刻采集

的超声传感器信息构造融合向量的组数($n=3$),融合向量的构造见文献[21]。

输入时延编码模块的 FPGA 实现:

在对 SNNs 进行硬件实现时,采用硬件描述语言编程:为减少 FPGA 硬件使用资源、提高运算速度和计算精度,算术运算器均采用 32 位单精度浮点数,其中整数部分用高 12 位二进制数代表,小数部分用低 10 位二进制数代表。对超声传感器测量信息进行预处理后用 Verilog HDL 编程实现公式(5),具体实现如算法 1 所示:

算法 1:输入时延编码模块的 FPGA 实现算法

输入:整个输入时延编码模块的仿真时间($T_{simulation}$)

输入层神经元数量(N_{neuron})

超声传感器测量信息($x_{ij}(m)$)

输出:时延编码($D_{ij}(m)$)

变量:时延编码的时间窗长(T_{win})

超声传感器的最大测量范围($\max(x)$)

(1) FOR t_{sim} FROM 0 TO $T_{simulation}$

(2) FOR n_{neuron} FROM 0 TO N_{neuron}

(3) IF($x_{ij}(m) = \max(x)$)

(4) $D_{ij}(m) = 0$

(5) ELSE

(6) $D_{ij}(m) = T_{win} * (1 - x_{ij}(m) * 2^{10}/\max(x) \% 1024)$

为保证运算精度,提出数据扩大再缩小的运算方法,如步骤 6 中首先把 $x_{ij}(m)$ 扩大 2^{10} , 然后除以 $\max(x)$, 再模除 1 024 进行缩放。取低 10 位二进制数表示小数部分可满足使用要求。

2.2.2 隐含层的 FPGA 实现

隐含层工作原理:

由于第 1 层神经元到第 2 层神经元的连接权值均为 $w_1(i, j_1) = 1$, 输入层的脉冲时延编码经过联结突触输入到隐含层。隐含层有着将输入数据向聚类中心汇集的作用。

在第 i 个走廊场景中分别采集 l 组超声传感器测量信息, 计算该组脉冲编码的均值 (μ_i) 和方差 (σ_i)。

$$\mu_i = \frac{1}{l} \sum_{j=1}^l d_{ij} \quad (7)$$

$$\sigma_i = \left(\frac{1}{l} \sum_{j=1}^l (d_{ij} - \mu_i)^2 \right)^{1/2} \quad (8)$$

其中, d_{ij} 为第 i 个走廊场景中第 j 组超声传感器测量信息的脉冲时延编码, l 为超声传感器测量组数。

隐含层中神经元的输出如式(9)所示:

$$O_i^2 = \left\| \frac{d_{ik} - \mu_i}{\sigma_i} \right\| \quad (9)$$

令 $O^2(16 \times n \times (i-1) + k)$ 为隐含层第 $(16 \times n \times (i-1) + k)$ 个神经元的输出, 则 $O^2(16 \times n \times (i-1) + k) = O_i^2(k)$ 。

令

$$t_d(j) = O^2(16 \times n \times (i-1) + k) \quad (10)$$

其中, $t_d(j)$ ($j = 1, 2, \dots, s_n \times n \times n_i$) 为隐含层第 j 个脉冲神经元的时延编码。

均值与方差运算模块的 FPGA 实现:

利用 FPGA, 采用硬件描述语言编程实现公式(7)的均值计算: 对第 i 个走廊场景中在连续时刻采集的 l 组超声传感器测量信息 (d_{ij}) 进行连加运算, 然后利用除法器对求和结果与 l 相除取其平均, 实现均值运算的硬件实现。

对公式(8)需进行方差的硬件实现, 因为开方运算在硬件实现中难以操作需转换为平方。首先, 将式(8)转换成式(11), 对第 i 个走廊场景中分别采集的 l 组超声传感器测量信息 (d_{ij}) 减去均值取平方; 接着, 对运算出的结果相加求和再除以 l 取均值, 运算 σ_i^2 作为浮点开方模块的输入信号。

$$\sigma_i^2 = \frac{1}{l} \sum_{j=1}^l (d_{ij} - \mu_i)^2 \quad (11)$$

浮点开方模块的 FPGA 实现:

浮点开方模块利用逐位循环开方算法的基本原理进行硬件实现。逐位循环 (Digit-Recurrence) 开方算

法^[25-26]是一种类似于试平方根的手算方法, 其原理为: 通过时钟周期不断迭代, 利用折半遍历比较, 计算差值不断去逼近真实值, 最终判断最接近的根即为开方结果。

隐含时延编码模块的 FPGA 实现:

通过输入时延编码模块、隐含层均值与方差运算模块和隐含层浮点开方模块分别计算出输入时延编码、均值和方差的结果, 采用硬件描述语言编程来完成隐含时延编码模块的 FPGA 硬件实现。公式(9)计算结果有 2 种情况, 如公式(12)所示。

$$O_i^2 = \begin{cases} \frac{d_{ik} - \mu_i}{\sigma_i}, & \text{if } d_{ik} \geq \mu_i \\ \frac{\mu_i - d_{ik}}{\sigma_i}, & \text{otherwise} \end{cases} \quad (12)$$

为保证计算精度问题, 隐含层的输出由 2 部分组成:

(1) 整数部分: 因运算出的隐含时延编码的高 10 位均为 0, 所以可利用移位运算符左移 10 位, 也就是被移位的数据高 10 位 0 被丢弃, 而低 10 位固定补 0 作为小数部分;

(2) 小数部分: 输入时延编码与各个均值用减法器连接后扩大 2^{10} , 再除以方差, 运算结果用 1 024 取余作为小数部分。

最终, 将其 2 部分直接相加即可得到隐含层神经元输出的脉冲时延编码值。

隐含时延编码模块 FPGA 实现算法如下:

算法 2: 隐含时延编码模块的 FPGA 实现算法

输入: 整个隐含时延编码模块的仿真时间 ($T_{\text{simulation}}$)

隐含层神经元数量 (N_{neuron})

超声传感器测量信息 ($x_{ij}(m)$)

输入时延编码 (d_{ik})

隐含层均值 (μ_i)

隐含层方差 (σ_i)

输出: 隐含时延编码 (t_d)

变量: 超声传感器的最大测量范围 ($\max(x)$)

(1) FOR t_{sim} FROM 0 TO $T_{\text{simulation}}$

(2) FOR n_{neuron} FROM 0 TO N_{neuron}

(3) IF ($x_{ij}(m) = \max(x)$)

(4) $t_d = (\mu_i / \sigma_i < 10) + (\mu_i * 2^{10} / \sigma_i \% 1024)$

(5) ELSE IF ($d_{ik} < \mu_i$)

(6) $t_d = ((d_{ik} - \mu_i) / \sigma_i < 10) + ((d_{ik} - \mu_i) * 2^{10} / \sigma_i \% 1024)$

(7) ELSE

(8) $t_d = ((\mu_i - d_{ik}) / \sigma_i < 10) + ((\mu_i - d_{ik}) * 2^{10} / \sigma_i \% 1024)$

2.2.3 输出层的 FPGA 硬件实现

逼近式 IAF 神经元模块的硬件实现原理:

输出层神经元为 IAF 神经元。隐含层神经元的输

出脉冲通过连接权值为 $w_2(i_2, j_2)$ 的联结突触将激活膜潜能($u_i(t)$)输入到输出层 IAF 神经元。为了便于 FPGA 实现脉冲 IAF 神经元模型(见公式(2)),提出由公式(13)~(15)实现的逼近式 IAF 神经元模型来近似实现复杂 IAF 神经元模型:

$$u_i(t) = w_2(i_2, j_2) \varepsilon(t - t_d(j)) \quad (13)$$

式中, t 为输出层的时间窗长,核函数(ε)近似为:

$$\varepsilon(t - t_d(j)) = \frac{t - t_d(j)}{\tau} \quad (14)$$

其中, $t_d(j)$ 为隐含层输出脉冲的时延编码,膜时间常数 $\tau = 800$ ms。

输出层逼近式 IAF 神经元的输出脉冲为:

$$O_i(t) = \begin{cases} 1, & \text{if } u_i(t) \geq \vartheta \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

其中, ϑ 为脉冲神经元的点火阈值。

逼近式 IAF 神经元模块的 FPGA 实现:

输出层的逼近式 IAF 神经元模块实现了膜潜能计算和脉冲 IAF 神经元模型的相应功能,该模块的核心环节是核函数(ε)的计算。使用减法器 and 除法器分别计算逼近式 IAF 神经元模块中由隐含层神经元输出脉冲(该输出脉冲采用脉冲时延编码,详见 2.2.2)提供的激活膜潜能值,该激活膜潜能值与逼近式 IAF 神经元点火阈值进行比较,来判断是否发放脉冲。该研究针对 7 个走廊场景进行分类,输出层设有对应的 7 个输出 IAF 神经元。如有一输出神经元先发放脉冲,因最先点火的其他神经元对其他神经元有侧向抑制作用,使得其他输出神经元不再点火。

逼近式 IAF 神经元模块的 FPGA 实现如下:

算法 3:逼近式 IAF 神经元模块的 FPGA 实现算法

输入:整个逼近式 IAF 神经元模块的仿真时间($T_{\text{simulation}}$)

输出层神经元数量(N_{neuron})

隐含层的时延编码($t_d(j)$)

输出:膜潜能($u_i(t)$)

脉冲输出(O_i)

变量:膜时间常数(τ)

点火阈值(ϑ)

- (1) FOR t_{sim} FROM 0 TO $T_{\text{simulation}}$
- (2) FOR n_{neuron} FROM 0 TO N_{neuron}
- (3) $u_i(t) = (t - t_d(j)) / \tau$
- (4) IF ($u_i(t) \leq \vartheta$)
- (5) $O_i = 1$
- (6) ELSE
- (7) $O_i = 0$

3 实验及结果分析

所提基于 FPGA 的移动机器人 SNNs 走廊场景分类器选用 Altera 公司 Cyclone IV EP4CE10F17C8 型号

FPGA 芯片,仿真环境为 Modelsim,开发环境为 Quartus II 13.1。使用硬件描述语言(Verilog HDL)设计并实现了分类器中输入时延编码、均值与方差运算、浮点开方、隐含时延编码、逼近式 IAF 神经元和场景类别 LCD 输出模块的相应功能。下面对上述功能模块的时序仿真结果进行分析和讨论,其中 clk 和 clr 分别为系统时钟和复位信号, cnt 为波形信号计数器。

实验中采用走廊场景分类准确率、分类器中 FPGA 所实现模块的计算精度,以及 FPGA 的资源利用率等指标对所提方法进行评价:(1)分类准确率:用于验证所提走廊场景分类器的有效性;(2)计算精度:SNNs 走廊场景分类器 FPGA 实现的核心环节之一是用逼近式 IAF 神经元模型(见公式(13)~(15))来近似实现复杂脉冲 IAF 神经元模型。其中实现逼近式 IAF 神经元模型的 FPGA 相应模块的计算精度直接决定了最终 IAF 模型的实现精度。实验中采用绝对误差和相对误差作为评价指标,反映各模块的计算准确程度;(3)资源利用率:指示所设计系统硬件资源使用量与可用量之比。利用率越低,意味着硬件系统的可拓展性资源越多,越便于未来系统功能的进一步升级和扩展。

3.1 输入时延编码模块的时序仿真及结果分析

输入时延编码模块的仿真结果如图 5 所示。超声传感器测量信息($x_{ij}(m)$)(见式(5))为输入信号(x_{in}),时延编码($d_{ij}(m)$)(见式(5))作为输出信号($T_{\text{delay_out}}$)。图 5 实验结果表明:超声传感测量信息(x_{in})为 4 000 mm 时,时延编码($T_{\text{delay_out}}$)为 0 ms;超声传感测量信息(x_{in})为 830 mm, 800 mm 时,时延编码($T_{\text{delay_out}}$)分别为 15.859 4 s, 16.015 6 ms。实验结果的误差分析见表 1,验证了 FPGA 实现脉冲时延编码方案的正确性。

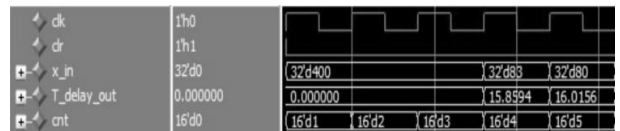
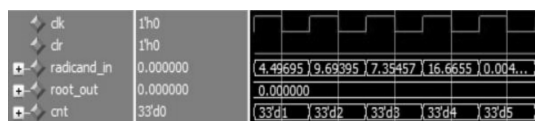


图 5 输入时延编码模块的仿真波形

3.2 浮点开方模块的时序仿真及结果分析

图 6(a)被开方数仿真波形中,被开方数(σ_i^2 , 见式(11))作为输入信号(radicand_in),图 6(b)开方数仿真波形中,开方数(σ_i , 见式(8))作为输出信号(root_out)。图 6(a)中被开方数(radicand_in)为 4.496 95, 9.693 95, 7.354 57, 16.665 5 等值时,图 6(b)中输出开方数(root_out)为 2.120 12, 3.113 28, 2.711 91, 4.082 03 等值,所得开方数运算结果(见表 1)表明基于 FPGA 的浮点开方模块运算在误差许可范围内,能够满足使用要求,所提浮点开方模块的计算方

案正确。



(a) 被开方数的输入仿真波形



(b) 开方数输出仿真波形

图 6 浮点开方模块的仿真波形

3.3 隐含时延编码模块的时序仿真及结果分析

隐含时延编码模块的仿真结果如图 7 所示。图中输入信号有: x_{in} , T_{delay_in} (分别代表超声传感器测量信息 ($x_y(m)$) 和时延编码 ($D_y(m)$), 见式(5)), $mean_in$ (代表均值 (μ_i), 见式(7)) 和 $variance_in$ (代表方差 (σ_i), 见式(8))。输出信号 (t_{d_out}) 代表隐含层的时延编码 ($t_d(j)$), 见式(10))。图 7 中超声传感器测量信息 (x_{in}) 为 4 000 mm, 时延编码 (T_{delay_in}) 为 0 ms, 均值 ($mean_in$) 为 3.637 7 和方差 ($variance_in$) 为 2.120 12 时, 隐含层时延编码 (t_{d_out}) 输出为 1.714 84 ms, 时延编码模块的其他实验结果 (见表 1) 亦在误差允许范围内。仿真实验结果表明基于 FPGA 的隐含时延编码方案正确有效。

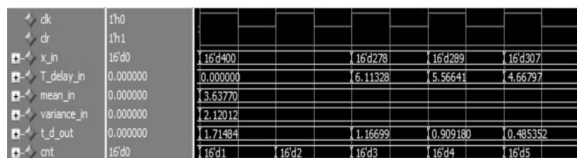
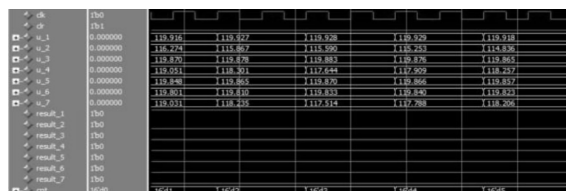


图 7 隐含时延编码模块的仿真波形

3.4 逼近式 IAF 神经元模块的时序仿真及结果分析

在逼近式 IAF 神经元模块的仿真实验中, 设定计算时间轴长 10 000 ms, 包含 $m = 5$ 个时间窗, 每一时间窗口 (T) 均为 2 000 ms, 点火阈值设置为 119.908。时序仿真时输入第 1 类至第 7 类走廊场景采集的超声数据到所设计的 FPGA 走廊场景分类器中, 得到 7 个输出神经元的膜潜能和输出脉冲模式, 根据输出脉冲模式得出机器人当前所处走廊场景。当机器人所处第 i 个走廊场景时, 输出层第 i 个神经元 (O_i) 的输入激活膜潜能最先达到点火阈值发放脉冲时, 因为 SNNs 采用侧向抑制联结, 此时输出层神经元仅有 O_i 点火, 机器人所处走廊场景为 1。因篇幅所限, 在此仅给出了第 1, 7 类 2 个走廊场景下的仿真结果 (如图 8 所示)。逼近式 IAF 神经元模块的输出信号包括代表膜潜能 ($u_i(t)$, 见式(13)) 的信号 (u_1-u_7) 和代表脉冲输出 ($O_i(t)$, 见式(15)) 信号的 $result_1-result_7$ 。当图 8(a) 中输入信号为第 1 类走廊场景所采集的超声数据时, 可以看到图中输出信号 ($result_1$) 为高电

平, 而信号 ($result_2-result_7$) 均为低电平, 走廊场景分类器识别机器人位于走廊场景 1; 图 8(b) 中输入信号为第 7 类走廊场景所采集的超声数据, $result_7$ 为高电平, $result_1-result_6$ 为低电平, 走廊场景分类器识别机器人位于走廊场景 7。



(a) 输入第 1 类走廊场景的超声数据时, 输出神经元的膜潜能和输出脉冲



(b) 输入第 7 类走廊场景的超声数据时, 输出神经元的膜潜能和输出脉冲

图 8 逼近式 IAF 神经元模块的仿真波形

图 8 实验结果表明, 当机器人位于第 1 和 7 类走廊场景时, 将采集到的相应走廊场景超声传感器数据输入到所设计的基于 FPGA 的走廊场景分类器中, 均能正确分类。实验结果证明, 采用脉冲时延编码, 结合所提逼近式脉冲 IAF 神经元模型, 实现了走廊场景的识别。

3.5 FPGA 中各模块运算误差分析

在利用 FPGA 实现走廊场景分类器功能时, 需要 FPGA 实现相应的计算均值、方差、开方等各种运算。为保证有效实现所设计 SNNs 的相应功能, 需要基于 FPGA 的运算模块达到所需的计算精度。图 4 中输入层、隐含层、输出层各模块采用 FPGA 实现时, 产生的误差有:

(1) 输入信号转换产生的误差。在进行 FPGA 实现之前, 需将超声传感器信息转换成定点型数据, 这使得实际数据中小数部分被截断而带来误差;

(2) 运算结果溢出产生的误差。基于 FPGA 的 SNNs 在进行基本数据运算时, 数据可能因位宽不足导致溢出, 计算出的数据需要截断, 截断数据会引起相应的计算误差。

为了减少运算误差, 提出了基于 SNNs 的数据扩大再缩小的运算方法 (见 2.2.1 节), 以保证 FPGA 实现的运算精度。各模块运算误差如表 1 所示, 其中 δ 和 E_r 分别表示绝对误差和相对误差。通过表 1 可知, 各模块计算的绝对误差最大在 10^{-4} 数量级, 相对误差在 10^{-3} 数量级以下。实验结果证明了所提方法能够满足分类器所需的精度要求。

表 1 各模块运算误差

输入时延编码模块					
组数	超声信息 (x_{in})/mm	实验值 (T_{delay_out})	理论值	δ	Er/%
1	4 000	0	0	0	0
2	830	15.859 4	15.850 0	9.40 E-03	0.06
3	800	16.015 6	16.000 0	1.56 E-02	0.1
浮点开方模块					
组数	被开方数 ($radicand_{in}$)	实验值 ($root_{out}$)	理论值	δ	Er/%
1	4.496 95	2.120 12	2.120 60	4.81 E-04	0.02
2	9.693 95	3.113 28	3.113 51	2.31 E-04	0.01
3	7.354 57	2.711 91	2.711 93	2.10 E-05	0
隐含时延编码模块					
组数	超声信息 (x_{in}) (mm)	实验值 (t_{d_out})	理论值	δ	Er/%
1	4 000	1.714 84	1.715 80	9.59 E-04	0
2	2 780	1.166 99	1.167 66	6.70 E-04	0.06
3	2 890	0.909 18	0.909 71	5.37 E-04	0.06

3.6 走廊场景分类实验结果

图 9 为 FPGA 通过 RGB TFT-LCD 接口在机器人本体上的 LCD 液晶屏中显示走廊场景分类器输出的移动机器人当前所处走廊场景类别及所对应的走廊场景示意图,走廊场景示意图具体如图 2 所示。

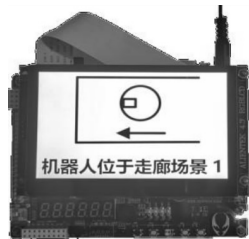


图 9 基于 FPGA 的 SNNs 走廊场景分类器实验结果

由于脉冲神经网络的使用,基于 FPGA 的移动机器人走廊场景分类器具有占用芯片资源少、可扩展性好、计算效率高的优点,其中 LE 的占用率仅 10% (1 067/10 320)。

所提基于 FPGA 的移动机器人 SNNs 走廊场景分类器的分类准确率为 92.86%。对于同样的走廊场景识别任务,采用相同的超声传感信息,文献[21]中基于 BP 神经网络的软件实现方法分类准确率为 82%,文献[16]中基于阶次编码和 NeuCube 模型的 SNNs 软件实现方法的分类准确率为 92.86%,文献[21]中基于时延编码和 IAF 神经元模型的 SNNs 方法的分类准确率为 94%。所提方法是文献[21]SNNs 软件方法的

硬件实现,二者分类准确率仅相差 1.14%。实验结果表明,所提 SNNs 硬件实现方案可以保证 SNNs 走廊场景分类的有效性,且基于 FPGA 的 SNNs 走廊场景分类器更适于实际使用,该研究有助于 SNNs 神经形态芯片在机器人领域的应用。

4 结束语

针对移动机器人 SNNs 走廊场景分类器的硬件实现进行研究,提出了基于 FPGA 的 SNNs 走廊场景分类器的实现方案:(1)提出逼近脉冲 IAF 神经元模型的近似处理方法,在保留脉冲 IAF 神经元模型的神经计算特征和动力学特征的同时,利用 FPGA 实现了该神经元模型的相应逻辑和功能;(2)利用 FPGA 完成实现了分类器中 SNNs 隐含层均值、方差模块的运算:将输入减去均值除以方差,实现人工神经网络中的批规范化运算,从而加速了神经网络的训练和模型的收敛;(3)提出了基于脉冲 IAF 神经元模型的数据运算方法,采用先扩大、再缩小 2^i 的方法保证了神经网络硬件实现所需精度。

实验部分 FPGA 实现的输入时延编码、均值与方差运算、浮点开方、隐含时延编码和逼近式 IAF 神经元模块的时序仿真结果及相应误差分析表明,各模块计算的绝对误差最大在 10^{-4} 数量级,相对误差在 10^{-3} 数量级以下,满足了分类器所需的运算精度要求,利用 FPGA 有效地实现了脉冲 IAF 神经元模型。此外,基于 FPGA 的移动机器人 SNNs 走廊场景分类器的分类准确率为 92.86%,验证了所提方法对于走廊场景识别的有效性。在 FPGA 硬件资源使用方面,所提走廊场景分类器的 LE 利用率仅 10%,说明此分类器具有良好的扩展性能。

所设计的走廊场景分类器不受光照条件的影响,需要的传感器信息少、结构简单易于实施。该研究可提高移动机器人的环境感知能力,为脉冲神经网络的实用化提供了解决思路,为类脑神经形态芯片的发展提供了有益参考。信息处理速度快、能耗低、轻量化的神经机器人将是机器人技术未来发展方向之一,该研究亦为未来实现神经机器人奠定了基础。

参考文献:

- [1] WOLFGANG M, CHRISTOPHER M. Pulsed neural networks [M]. Massachusetts: MIT Press, 1999.
- [2] GERSTNER W, KISTLER W. Spiking neuron models [M]. Cambridge: Cambridge University Press, 2002.
- [3] WOLFGANG M. Networks of spiking neurons: the third generation of neural network models [J]. Neural Networks, 1997, 10 (9): 1659-1671.
- [4] EMRE N, JONATHAN B, UELI R, et al. Synthesizing cogni-

- tion in neuromorphic electronic systems[J]. Proceedings of the National Academy of Sciences of the United States of America, 2013, 110(37): E3468–E3476.
- [5] MEROLLA P A, ARTHUR J V, ALVAREZ-ICAZA R, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface[J]. Science, 2014, 345(6197): 668–673.
- [6] STEVE F, FRANCESCO G, STEVE T, et al. The spinnaker project[J]. Proceedings of the IEEE, 2014, 102(5): 652–665.
- [7] BEN V B, PEIRAN G, EMMETT M, et al. Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations[J]. Proceedings of the IEEE, 2014, 102(5): 699–716.
- [8] NGUYEN T N, VEERAVALLI B, FONG X Y. Connection pruning for deep spiking neural networks with on-chip learning[C]//2021 ACM/SIGDA international symposium on field-programmable gate arrays. New York: ACM, 2021: 1–8.
- [9] PANCHAPAKESAN S, FANG Z M, LI J. SyncNN: evaluating and accelerating spiking neural networks on FPGAs[J]. ACM Transactions on Reconfigurable Technology and Systems, 2022, 15(48): 1–27.
- [10] SHEN J, MA D, GU Z, et al. Darwin: a neuromorphic hardware co-processor based on spiking neural networks[J]. Science China Information Sciences, 2016, 59(2): 1–5.
- [11] SUN Q, WU Q, WANG X, et al. A spiking neural network for extraction of features in colour opponent visual pathways and FPGA implementation[J]. Neurocomputing, 2017, 228: 119–132.
- [12] PEI J, DENG L, SONG S, et al. Towards artificial general intelligence with hybrid tianjic chip architecture[J]. Nature, 2019, 572(7767): 106–111.
- [13] ZHANG G, LI B, WU J, et al. A low-cost and high-speed hardware implementation of spiking neural network[J]. Neurocomputing, 2020, 382: 106–115.
- [14] WANG Y, ZHANG H, OH K I, et al. Energy efficient spiking neural network processing using approximate arithmetic units and variable precision weights[J]. Journal of Parallel and Distributed Computing, 2021, 158: 164–175.
- [15] GARCIA D H, ADAMS S, RAST A, et al. Visual attention and object naming in humanoid robots using a bio-inspired spiking neural network[J]. Robotics and Autonomous Systems, 2018, 104: 56–71.
- [16] 王秀青, 侯增广, 潘世英, 等. 基于多超声传感器信息和 NeuCube 的移动机器人走廊场景识别[J]. 计算机应用, 2015, 35(10): 2833–2837.
- [17] TANGG Z, KUMAR N, MICHMIZOS K P. Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware[C]//2020 IEEE/RSJ international conference on intelligent robots and systems. Las Vegas: IEEE, 2020: 6090–6097.
- [18] LU H, LIU J X, LUO Y L, et al. An autonomous learning mobile robot using biological reward modulate STDP[J]. Neurocomputing, 2021, 458: 308–318.
- [19] AZIMIRAD V, RAMEZANLOU M T, SOTUBAD S V, et al. A consecutive hybrid spiking-convolutional (CHSC) neural controller for sequential decision making in robots[J]. Neurocomputing, 2022, 490: 319–336.
- [20] WANG X Q, HOU Z G, LV F, et al. Mobile robots' modular navigation controller using spiking neural networks[J]. Neurocomputing, 2014, 134: 230–238.
- [21] 王秀青. 基于 Spiking 神经网络的移动机器人环境感知及行为控制的研究[D]. 北京: 中国科学院研究生院, 2007.
- [22] CHEN Y H, WANG C, GUO H X, et al. Accelerating spiking neural networks using quantum algorithm with high success probability and high calculation accuracy[J]. Neurocomputing, 2022, 493: 435–444.
- [23] 潘坤榕, 夏福源, 李瑞民, 等. 基于 FPGA 的高效卷积神经网络设计[J]. 计算机技术与发展, 2022, 32(7): 105–110.
- [24] 许杰, 张子恒, 王新宇, 等. 一种基于 Zynq 的 CNN 加速器设计与实现[J]. 计算机技术与发展, 2021, 31(11): 108–113.
- [25] 夏宏, 李笑盈. 浮点开方运算单元的电路设计[J]. 计算机工程与应用, 2001, 37(11): 39–41.
- [26] CHMIEL M, KULISZ J, CZERWINSKI R, et al. An IEC 61131-3-based PLC implemented by means of an FPGA[J]. Microprocessors and Microsystems, 2016, 44: 28–37.