

darwin3_runtime_api 使用说明

Version: 4.0

! 最新更改

1. 支持从东西两侧进行神经元的配置（用户无需手动指定，仅需调用 `deploy_config` 即可）
2. 支持从西边发放脉冲给板卡，从西边或东边接受板卡的输出脉冲。`run_darwin3_withfile` 函数中可用 `outdir` 参数手动指定（默认为 `west`）

1 总体介绍

- 该 API 用于上位机对 Darwin 板卡进行初始化，配置时钟，神经元部署、激活，向神经元发放脉冲，接受神经元返回的脉冲，查询神经元状态等
- 上位机和 Darwin 板卡之间采取 **HTTP 协议**和 **TCP 协议**进行通信
- 该 API 使用 `setuptools` 进行打包操作，用户只需要安装打包好的 `whl` 格式文件即可直接使用

2 打包方法

该方法仅适用于无法正常安装预打包好的 `whl` 文件的用户以及需要对源代码进行调整后使用的用户。

1. 下载 `darwin3_runtime_api`，其中包含 `darwin3_runtime_api`、`darwin3_runtime_node`、`script` 三个文件夹以及 `setup.py` 脚本
2. 根据需求修改 `setup.py` 中的版本号（可选）
3. 运行 `python setup.py bdist_wheel`
4. 在 `dist` 文件中可以找到打包好的 `darwin3_runtime_api-$version$-py3-none-any.whl` 文件

3 安装方法

 注意：如果之前有安装过旧版本的 `darwin3_runtime_api`，请务必先卸载旧版本后再重新安装

- 安装方法

```
1 pip3 install PATH-TO-YOUR-whl_file
```

- 卸载方法

```
1 pip3 uninstall darwin3_runtime_api
```

4 使用方法

- 上位机 (PC)
 - 将模型文件放在指定目录（具体见 [5 目录格式](#)）
 - 使用 `from darwin3_runtime_api import darwin3_device`，在需要使用 `darwin3_runtime_api` 的文件中添加 `darwin3_device` 类
 - 对类进行实例化后即可使用相应方法（具体见 [6 API 可用方法](#)），实例化时需要关注如下参数：
 - **protocol (str)**
 - 所使用的通信协议。目前仅支持 `TCP` 和 `LOCAL`（使用 `LOCAL` 即意味着没有上位机的存在，所有的操作均在 Darwin3 板卡上完成）。
 - `default='TCP'`
 - **ip (list(str))**
 - Darwin3 开发板的 IP 地址。默认使用列表中的第一个 IP 进行通信（目前暂不支持使用第二个 IP 通信）
 - `default=['172.31.111.35']`
 - **port (list(int))**
 - 和 Darwin3 通信使用的端口号。当前版本中用户请勿对此项做修改，使用 `default` 即可
 - `default=[6000, 6001]`
 - **step_size (int)**
 - 每个时间步维持的 FPGA 时钟周期数, 对应一个时间步的时长为 $10\text{ns} \times \text{step_size} \times 2$ ，最小可配置的值为 512
 - `default=25000`
 - **app_path (str)**
 - 模型文件的存储目录。该目录中需要将配置文件按照一定的格式存储，具体见 [5 目录格式](#)
 - `default='../'`
 - 下位机 (Darwin3 板卡)
 - 运行 `darwin3_runtime_server.py`
 - 下位机会自动重置 DMA 和板卡状态，并开启 HTTP 服务，等待上位机连接

5 目录格式

上位机运行模型时，需要按照指定的目录格式进行配置文件的存放，具体如下所示

```
1  .
2  └─ apps (name user-defined upon instantiation)
3     └─ config_files
4        └─ spikes.dwno (generated by script, users don't need to care)
```

```

5      |      |      | 0-1-config.dwnc
6      |      |      | 0-1-ax.txt
7      |      |      | 0-1-de.txt
8      |      |      | 0-2-config.dwnc
9      |      |      | 0-2-ax.txt
10     |      |      | 0-2-de.txt
11     |      |      | 1-1-config.dwnc
12     |      |      | 1-1-ax.txt
13     |      |      | 1-1-de.txt
14     |      |      | input_neuron.json
15     |      |      | pop_h_1.json
16     |      |      | pop_h_2.json
17     |      |      | output_neuron_xxx.json
18     |      |      | ...
19     |      |  ── deploy_files (generated by script, users don't need to care)
20     |      |      |  ── deploy_input.dwnc
21     |      |      |  ── deploy_flitin.txt
22     |      |      |  └─ deploy_flitin.bin
23     |      |  ── input_files (generated by script, users don't need to care)
24     |      |      |  ── run_input.dwnc
25     |      |      |  ── run_flitin.txt
26     |      |      |  └─ run_flitin.bin
27     |      |  ── debug_files (generated by script, users don't need to care)
28     |      |      |  ── get_neuron_state_flitin.bin
29     |      |      |  ── get_neuron_state_flitin.txt
30     |      |      |  └─ ...
31     |      |  ── output_files (generated by script, results)
32     |      |      |  ── recv_run_flit.txt
33     |      |      |  └─ recv_run_flit.bin
34     |      |  ── model (other app flies, name user-defined)

```

用户应主要关注存放好 `config_files` 中的文件，其余文件均为 API 在使用时生成。

6 API 可用方法

```

1 class darwin3_device(object):
2     def __init__(
3         self, protocol="TCP", ip=['172.31.111.35'], port=[6000, 6001],
4         step_size=25000, app_path=".."
5     ):
6         """
7         Args:
8             protocol (str): 与 Darwin3 开发板通信使用的协议，默认 TCP，可选 LOCAL，暂不
9                             支持其它
10             ip (list(str)): Darwin3 板卡设备 ip 序列，单芯片开发板最多支持两个 ip
11                             默认使用 ip[0] 进行上下位机通信（暂不支持 ip[1] 的连接）
12             port (list(int)): 与 Darwin3 开发板通信使用的端口，默认为 6000 和 6001
13             step_size (int): 每个时间步维持的 FPGA 时钟周期数，对应时长为 10ns *
14                             step size * 2

```

```

12                                     (汇编工具介绍与上位机通信流程中有换算关系，对应run_input.dwno
中最开始的配置)
13         app_path (str):    模型文件的存储目录
14         """
15
16     def reset(self):
17         """
18         复位硬件接口相关逻辑和硬件系统(darwin3 芯片，DMA 等)
19         Args:
20             None
21         Returns:
22             None
23         """
24
25     def darwin3_init(self, freq=333):
26         """
27         按照指定频率配置 darwin3 芯片。
28         Args:
29             freq (int): 频率 (默认 333MHz，仅支持 20MHz 和 333MHz)
30         Returns:
31             None
32         """
33
34     def deploy_config(self):
35         """
36         在部署芯片上部署并使能相关核心，同时清除神经元的相关状态
37         Args:
38             None
39         Returns:
40             None
41         """
42
43     def run_darwin3_withoutfile(self, spike_neurons: list, outdir="west"):
44         """
45         接收应用给的 spike_neurons 作为输入，运行 len(spike_neurons) 个时间步
46         此函数不产生中间文件
47         Args:
48             outdir (str): 脉冲输出的方向，支持 east 和 west 方向的输出，默认 west
49             spike_neurons (list): sequence，本次应用输入给硬件的脉冲数据，
50             序列长度与时间步数量一致，没有脉冲的时间步给空值，格式如下所示：
51             隐含的时间步序号          输入神经元序号(逻辑序号)列表，这些神经元发放脉冲
52                                     [
53                                     1                [1,2,4]
54                                     2                [5,8]
55                                     3                []
56                                     ...              ...
57                                     ]
58
59         Returns:
60             result (list): 本次运行结束时硬件返回给应用的脉冲，格式如下所示：
61             隐含的时间步序号          输出神经元序号(逻辑序号)+输出层名称的列表，这些神经元
发放脉冲

```

```

62         [
63             ...
64             18             []
65             19             [('pop1',100), ('pop2',101)]
66             20             [('pop2',105), ('pop2',108),
('pop2',110)]
67             ...
68         ]
69     """
70
71     def run_darwin3_withfile(self, spike_neurons: list, outdir="west"):
72         """
73         接收应用给的 spike_neurons 作为输入，运行 len(spike_neurons) 个时间步
74         此函数会产生中间文件进行过渡
75         Args:
76             spike_neurons (list): sequence, 本次应用输入给硬件的脉冲数据，
77                                     序列长度与时间步数量一致，没有脉冲的时间步给空值
78         Returns:
79             result (list): 本次运行结束时硬件返回给应用的脉冲
80         """
81
82     def get_neuron_state(self, pop_name: str, state: list, offset=0):
83         """
84         从硬件获取神经元的状态，状态空间包括膜电位、权重和、推理参数、推理状态等，主要用于调试。
85         Args:
86             pop_name (str): 编译阶段的 Population Name
87             state (list): sequence[array], 包含一组或多组[神经元序号, [状态空间列表]]
88                 膜电位(vt) => "read", x, y, neuron_index+0x02000 按升序
89                 权重和(wgtsum) => "read", x, y, neuron_index+0x04000 按升序
90                 推理参数(inference_parameter) =>
91                 "read", x, y, neuron_index+0x1E000(按升序) 0x1EFFF-
neuron_index(按降序)
92                 推理状态(inference_status) =>
93                 "read", x, y, neuron_index+0x1F000(按升序) 0x1FFFF-
neuron_index(按降序)
94             offset (int): 用户指定的地址偏移
95         Returns:
96             None
97         """
98
99     def enable_neurons(self, dwnc_file="enable"):
100         """
101         将 config_files 中所有需要使用的神经元使能
102         Args:
103             dwnc_file (str): 生成的配置文件名称
104         Returns:
105             None
106         """
107
108     def disable_neurons(self, dwnc_file="disable"):
109         """
110         将 config_files 中所有需要使用的神经元取消使能

```

```

111         Args:
112             dwnc_file (str): 生成的配置文件名称
113         Returns:
114             None
115         """
116
117         def clear_neurons_states(self, ISC=False, LSC=False, clear=False,
dwnc_file="clear_states"):
118             """
119             清理 darwin3 芯片内部神经拟态核心的状态量
120         Args:
121             ISC (bool): inference status clear,
122                         推理状态中电流清零, 阈值和振荡电位复位, 1 有效
123                         相关配置寄存器: dedr_vth_keep, dedr_vth_gset,
124                         global_vth, dedr_res_keep, global_res
125             LSC (bool): learn status clear, 学习状态清零, 1 有效
126             clear (bool): 权重和清零, 膜电位复位, 1 有效
127                         相关配置寄存器: vt_rest
128
129             dwnc_file (str): 生成的配置文件名称
130         Returns:
131             None
132         """

```

7 示例测试代码

```

1  import os
2  import sys
3  from darwin3_runtime_api import darwin3_device
4  import time
5
6  test = darwin3_device.darwin3_device(ip=["192.168.1.89"], app_path='../',
step_size=399999)
7
8  test.reset()
9  test.darwin3_init(333)
10
11 test.deploy_config()
12
13 #####
14 # 这一部分只是为示例展示, 应用给的脉冲数据spike_neurons可直接作为参数传入到run_darwin3()
15 # 即print(test.run_darwin3(spike_neurons))
16 int_list = []
17 with open('../input_files/spikes_seq.txt', 'r') as f:
18     lines = f.readlines()
19     for line in lines:
20         item = line.split()
21         list_temp = [int(num) for num in item]
22         # print(list_temp)

```

```
23         int_list.append(list_temp)
24     print(len(int_list))
25     #####
26
27     print(test.run_darwin3_withfile(int_list))
28
29     # 这里是调试功能，用于获取神经元状态（膜电位、权重和等）
30     state = [[30,['vt']], [31,['vt']]]
31     test.get_neuron_state("pop_h_1", state)
32
```