

This file demonstrates a bug with PartitionInterchip related to small networks and the n_chips parameter

In [1]:

```
from tensorflow.keras.layers import *
from nengo_loihi.hardware.allocators import PartitionInterchip
import tensorflow as tf
import nengo_dl
import nengo
import numpy as np
import nengo_loihi
```

```
2022-01-11 06:37:28.602630: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dLError: libcudart.so.11.0: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /slurm/intel-archi/lib
2022-01-11 06:37:28.602693: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dLError if you do not have a GPU set up on your machine.
2022-01-11 06:37:33.670790: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcudart.so.11.0'; dLError: libcudart.so.11.0: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /slurm/intel-archi/lib
2022-01-11 06:37:35.932628: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dLError: libcuda.so.1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /slurm/intel-archi/lib
2022-01-11 06:37:35.932672: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
```

In [2]:

```
# define a small convolutional network
def simple_neural_network():
    inputs = Input(shape=(64, 64, 3))
    spiking_input = Activation(tf.nn.elu)(inputs) # this will be replaced by SpikingRectified
    conv_out1 = Conv2D(32, (5, 5), strides=(2, 2), padding='valid', activation=tf.nn.relu,
    conv_out2 = Conv2D(64, (3, 3), strides=(2, 2), padding='valid', activation=tf.nn.relu,
    conv_out3 = Conv2D(128, (2, 2), strides=(2, 2), padding='valid', activation=tf.nn.relu
    conv_out4 = Conv2D(256, (2, 2), strides=(2, 2), padding='valid', activation=tf.nn.relu

    flat_out = Flatten()(conv_out4)
    output = Dense(4, activation=None, name="dense", use_bias = False)(flat_out)

    model = tf.keras.Model(inputs=inputs, outputs=output)
    return model
```

In [3]:

```
# Convert into spiking model
ann_model = simple_neural_network()
nengo_converter = nengo_dl.Converter(
    ann_model,
    swap_activations={tf.nn.relu: nengo_loihi.LoihiSpikingRectifiedLinear(), # this is ou
                    tf.nn.elu: nengo.SpikingRectifiedLinear()},
)
```

```
2022-01-11 06:37:37.469108: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'libcuda.so.1'; dLError: libcuda.so.1: cannot open shared object file: No such file or directory; LD_LIBRARY_PATH: /slurm/intel-archi/lib
2022-01-11 06:37:37.469157: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2022-01-11 06:37:37.469186: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] ker
```

nel driver does not appear to be running on this host (ncl-edu): /proc/driver/nvidia/versi
on does not exist

```
In [4]: # Specify first layer as running off chip
with nengo_converter.net as net:
    nengo_loihi.add_params(net) # allow on_chip to be set
    net.config[nengo_converter.layers[ann_model.layers[1]].ensemble].on_chip = False
```

```
In [5]: # Define our input to the SNN
nengo_input = nengo_converter.inputs[ann_model.layers[0]]
with nengo_converter.net as net:
    nengo_input.output = nengo.processes.PresentInput(
        np.random.random((12, 64, 64, 3)), presentation_time=.02
    )
```

```
In [6]: # Assign block shapes
block_sizes = [None, (16, 16, 4), (8, 8, 16), None]
conv_layers = [layer for layer in ann_model.layers if "conv" in str(layer).lower()]
for layer, block_size, layer_idx in zip(conv_layers, block_sizes, range(len(conv_layers))):
    if block_size == None: #
        continue
    output_shape = tuple(layer.output.shape[1:])
    with nengo_converter.net as net:
        net.config[net.ensembles[layer_idx+1]].block_shape = nengo_loihi.BlockShape(block
```

The neural network will fail trying to allocate 0 num_cores.

```
In [7]: # Try to run the network on loihi
with nengo_loihi.Simulator(
    nengo_converter.net,
    hardware_options=dict(n_chips = 3, allocator = PartitionInterchip(), snip_max_spikes_
    precompute = False,
) as sim:
    model_utilization = sim.model.utilization_summary() # print model utilization
    print("\n".join(model_utilization))
    quit()
```

```
/homes/mjurado3/miniconda3/envs/loihi_vishal/lib/python3.9/site-packages/nengo_loihi/simul
ator.py:159: UserWarning: Model is precomputable. Setting precompute=False may slow execut
ion.
```

```
warnings.warn(
```

```
1
```

```
76
```

```
0
```

```
-----
AssertionError                                Traceback (most recent call last)
```

```
/tmp/ipykernel_1111267/4268878930.py in <module>
```

```
1 # Try to run the network on loihi
```

```
----> 2 with nengo_loihi.Simulator(
```

```
3     nengo_converter.net,
```

```
4     hardware_options=dict(n_chips = 3, allocator = PartitionInterchip(), snip_max_
spikes_per_step = 3500),
```

```
5     precompute = False,
```

```
~/miniconda3/envs/loihi_vishal/lib/python3.9/site-packages/nengo_loihi/simulator.py in __i
nit__(self, network, dt, seed, model, precompute, target, progress_bar, remove_passthroug
h, hardware_options)
```

```
196         assert HAS_NXSDK, "Must have NxSDK installed to use Loihi hardware"
```

```

197         use_snips = not self.precompute and self.sims.get("host", None) is not
None
--> 198         self.sims["loihi"] = HardwareInterface(
199             self.model, use_snips=use_snips, seed=seed, **hardware_options
200         )

~/miniconda3/envs/loihi_vishal/lib/python3.9/site-packages/nengo_loihi/hardware/interface.py in __init__(self, model, use_snips, seed, snip_max_spikes_per_step, n_chips, allocator)
92
93     # --- build
--> 94     self.nxsdk_board = build_board(
95         self.board, use_snips=self.use_snips, seed=self.seed
96     )

~/miniconda3/envs/loihi_vishal/lib/python3.9/site-packages/nengo_loihi/hardware/builder.py in build_board(board, use_snips, seed)
30     n_cores_per_chip = board.n_cores_per_chip
31     n_synapses_per_core = board.n_synapses_per_core
--> 32     nxsdk_board = NxsdkBoard(
33         board.board_id, n_chips, n_cores_per_chip, n_synapses_per_core
34     )

~/miniconda3/envs/loihi_vishal/lib/python3.9/site-packages/nxsdk/arch/n2a/n2board.py in __init__(self, id, numChips, numCores, initNumSynapses, options, lmtOptions, numCoresPerChip, generation)
58     self.generation = generation
59
--> 60     super(N2Board, self).__init__(id,
61                                   numChips=numChips,
62                                   numCores=numCores,

~/miniconda3/envs/loihi_vishal/lib/python3.9/site-packages/nxsdk/arch/base/nxboard.py in __init__(self, id, numChips, numCores, initNumSynapses, options, lmtOptions, numCoresPerChip, chipClass)
95
96     if numChips > 0:
--> 97         self.allocateCoresByChip(numChips, numCores, initNumSynapses)
98
99     self._sync = False

~/miniconda3/envs/loihi_vishal/lib/python3.9/site-packages/nxsdk/arch/base/nxboard.py in allocateCoresByChip(self, numChips, numCores, numSynapses)
174     newCores = []
175     for i, chip in enumerate(self.allocateChips(numChips)):
--> 176         c = chip.allocateCores(numCoresPerChip[i], numSynPerCorePerChip[i])
177         newCores.extend(c)
178

~/miniconda3/envs/loihi_vishal/lib/python3.9/site-packages/nxsdk/arch/base/nxchip.py in allocateCores(self, numCores, numSynapses)
188     print(numCores)
189     # Validate numCores
--> 190     assert (isinstance(numCores, int) or isinstance(numCores, np.integer)
191             ) and numCores > 0, "<numCores> must be a positive integer."
192     numCores = int(numCores)

```

AssertionError: <numCores> must be a positive integer.