Code synfire chain (chaîne synchronisée de neurones, activité qui se propage dans le réseau)

Actuellement, SpiNNaker utilise la version PyNN 0.7.

Import des bibliothèques externes : PyNN pour SpiNNaker, NumPy pour la génération de nombres aléatoires, et matplotlib pour le traçage de graphique.

```
import numpy.random

import matplotlib.pyplot as plt

import pyNN.spiNNaker as sim
```

Définition des paramètres des neurones : nombre et taille des populations neuronales (11 populations de 8 neurones), propriétés des neurones, poids synaptiques et retards, etc.

```
n_populations = 11
population_size = 8
neuron_parameters = {
    'cm': 0.2,
    'v_reset': -70,
    'v_rest': -70,
    'v_thresh': -47,
    'e_rev_I': -70,
    'e_rev_E': 0.0,
}
weight_exc_exc = 0.005
weight_exc_inh = 0.005
weight_inh_exc = 0.5
delay = 3.0
rng_seed = 42
stimulus_onset = 25.0
stimulus_sigma = 0.5
runtime = 150.0
```

Initialisation du simulateur

```
sim.setup(timestep=0.1)
```

Création de 11 populations de neurones excitateurs intégrer-et-tirer (FI) et 11 populations de neurones IF inhibiteurs.

```
populations = {'exc': [], 'inh': []}
for syn_type in ('exc', 'inh'):
      populations[syn_type]=[sim.Population(population_size, sim.IF_cond_exp,
                                            neuron_parameters)
                             for i in range(n_populations)]
```

Connexion de chaque population excitatrice à la paire suivante de populations excitatrices et inhibitrices, et chaque population inhibitrice à la population excitatrice dans la même paire.

```
connector_exc_exc = sim.AllToAllConnector(weights=weight_exc_exc, delays=delay)
connector_exc_inh = sim.AllToAllConnector(weights=weight_exc_inh, delays=delay)
connector_inh_exc = sim.AllToAllConnector(weights=weight_inh_exc, delays=delay)

for i in range(n_populations):
    j = (i + 1) % n_populations
```

```
    prj_exc_exc = sim.Projection(populations['exc'][i], populations['exc'][j],
                                 connector_exc_exc, target='excitatory')
    prj_exc_inh = sim.Projection(populations['exc'][i], populations['inh'][j],
                                 connector_exc_inh, target='excitatory')
    prj_inh_exc = sim.Projection(populations['inh'][i], populations['exc'][i],
                                 connector_exc_exc, target='inhibitory')
```

La première paire de populations est stimulée par des spikes.

```
numpy.random.seed(rng_seed)
stim_spikes = numpy.random.normal(loc=stimulus_onset,
                                  scale=stimulus_sigma,
                                  size=population_size)
stim_spikes.sort()
stimulus = sim.Population(1, sim.SpikeSourceArray, {'spike_times': stim_spikes})

prj_stim_exc = sim.Projection(stimulus, populations['exc'][0],
                              connector_exc_exc, target='excitatory')
prj_stim_inh = sim.Projection(stimulus, populations['inh'][0],
                              connector_exc_inh, target='excitatory')
```

Enregistrement des spikes de toutes les populations.

```
for syn_type in ('exc', 'inh'):
      for population in populations[syn_type]:
            population.record()
```

Exécution de la simulation

```
sim.run(runtime)
```

Parcours des populations, récupération des spikes, traçage d'un graphique (spike time vs neuron index)

```
colours = {'exc': 'r', 'inh': 'b'}

id_offset = 0
for syn_type in ['exc', 'inh']:
      for population in populations[syn_type]:
            spikes = population.getSpikes()
            colour = colours[syn_type]
            plt.plot(spikes[:,1], spikes[:,0] + id_offset, ls='', marker='o', ms=1,
                        c=colour, mec=colour)
            id_offset += population.size

plt.xlim((0, runtime))
plt.ylim((-0.5, 2* n_populations * population_size + 0.5))
plt.xlabel('time (t)')
plt.ylabel('neuron index')
plt.savefig("synfire_chain.png")
```

Conseils d'utilisation de la plateforme :

- séparer exécution et enregistrement des spikes dans un fichier (code submit sur la plateforme) de l'analyse et de la visualisation des résultats sur graphique(des outils sont disponibles sur la plateforme?)

- pour exécuter une même simulation avec plusieurs simulateurs différents, il est pratique de modifer simplement le nom du simulateur PyNN à importer sans toucher au code. Fournir le nom du simulateur en argument de ligne de commande : `python run.py spiNNaker`
Puis dans le code :

```
from pyNN.utility import get_script_args
simulator_name = get_script_args(1)[0]
exec("import pyNN.%s as sim" % simulator_name)
```

La pile logicielle SpiNNaker essaie de partitionner automatiquement les populations définies dans le script en blocs de neurones de la taille d'un cœur (la plus petite taille atomique de ressource pour une machine). Ces blocs de neurones sont ensuite placés sur la machine de manière optimale par rapport aux ressources disponibles. Cependant, il est possible d'influencer manuellement le partitionnement et le placement.

Une contrainte de partitionnement peut être ajoutée à la population, ce qui peut limiter le nombre de neurones que chaque cœur contiendra au maximum :

```
populations[syn_type]=sim.Population(population_size, sim.IF_cond_exp, neuron_parameters)
populations[syn_type].add_constraint(sim.PartitionerMaximumSizeConstraint(200))
```

Source: https://electronicvisions.github.io/hbp-sp9-guidebook/building_models.html

Exécution d'un job sur la plateforme

Job Manager → New job
→ Choisir la plateforme (SpiNNaker, BrainScaleS, …)
→ Entrer le code PyNN ou l'url d'un repo Git ou l'url d'un zip avec le code.
→ Optionnel : entrer la commande pour run le code en spécifiant arguments si nécessaire
→ Optionnel : spécifier une configuration pour le hardware et des fichiers d'entrée
→ Submit

On reçoit alors un mail indiquant que le job a été soumis et le job est marqué comme « submitted ». En cas de fonctionnement normal de la plateforme, en quelques minutes le job est exécuté et on reçoit alors un nouveau mail indiquant le résultat de l'exécution (finished ou error). En cas d'erreur, on peut consulter un log indiquant la ligne fautive et le type d'erreur rencontré. En cas de succès, on peut consulter plus de détails sur l'exécution du job.

On peut voir sa date de soumission et sa date de fin ainsi que l'auteur de la soumission. Si le job produit des fichiers de sortie, par exemple un graphique, on peut le visualiser et le récupérer sur le stockage de la collab (sinon il n'est plus consultable au bout de trois mois). On a accès au code exécuté, à la configuration spécifiée (si aucune configuration n'a été renseignée, on peut juste voir `{"resource_allocation_id":98}`).
Enfin, nous avons accès à une partie « Log » qui contient la trace d'exécution du job.

```
2017-01-23 12:03:47 INFO: Read config files: /home/spinnaker/.spynnaker.cfg,
spynnaker.cfg, /opt/git/sPyNNaker/spynnaker/spynnaker.cfg
2017-01-23 12:03:47 INFO: sPyNNaker (c) 2016 APT Group, University of Manchester
2017-01-23 12:03:47 INFO: Release version 3.0.0() - September 2016. Installed in folder
/opt/git/sPyNNaker
2017-01-23 12:03:47 INFO: Will search these locations for binaries:
/opt/git/SpiNNFrontEndCommon/spinn_front_end_common/common_model_binaries :
/opt/git/sPyNNaker/spynnaker/pyNN/model_binaries
2017-01-23 12:03:47 INFO: Setting appID to 30.
2017-01-23 12:03:47 WARNING: A timestep was entered that has forced sPyNNaker to
automatically slow the simulation down from real time by a factor of 10.0. To remove
this automatic behaviour, please enter a timescaleFactor value in your .spynnaker.cfg
2017-01-23 12:03:47 INFO: Setting time scale factor to 10.0.
2017-01-23 12:03:47 INFO: Setting machine time step to 100.0 micro-seconds.
2017-01-23 12:03:47 INFO: Starting execution process
2017-01-23 12:03:49 INFO: Starting new HTTP connection (1): jabez.cs.man.ac.uk
2017-01-23 12:03:49 INFO: Time 0:00:00.061481 taken by
FrontEndCommonHBPMaxMachineGenerator
Generating a virtual machine
|0                          50%                         100%|
 ============================================================
2017-01-23 12:03:49 INFO: Time 0:00:00.384202 taken by
FrontEndCommonVirtualMachineGenerator
Allocating virtual identifiers
|0                          50%                         100%|
 ============================================================
2017-01-23 12:03:49 INFO: Time 0:00:00.015256 taken by MallocBasedChipIDAllocator
Partitioning graph vertices
|0                          50%                         100%|
 ============================================================
Partitioning graph edges
|0                          50%                         100%|
 ============================================================
2017-01-23 12:03:49 INFO: Time 0:00:00.047971 taken by PartitionAndPlacePartitioner
2017-01-23 12:03:49 INFO: Starting new HTTP connection (1): jabez.cs.man.ac.uk
2017-01-23 12:03:54 INFO: Time 0:00:05.039528 taken by FrontEndCommonHBPAllocator
2017-01-23 12:03:54 INFO: Starting new HTTP connection (1): jabez.cs.man.ac.uk
2017-01-23 12:03:55 INFO: Creating transceiver for 10.2.225.105
2017-01-23 12:03:55 INFO: Working out if machine is booted
2017-01-23 12:03:57 INFO: Attempting to boot machine
2017-01-23 12:04:01 INFO: Attempting to boot machine
2017-01-23 12:04:03 INFO: Found board with version [Version: SC&MP 3.0.1 at
SpiNNaker:0:0:0 (built Wed Jul 20 10:07:23 2016)]
2017-01-23 12:04:03 INFO: Machine communication successful
2017-01-23 12:04:03 INFO: Detected a machine on ip address 10.2.225.105 which has 856
cores and 120 links
2017-01-23 12:04:03 INFO: Time 0:00:08.787303 taken by FrontEndCommonMachineGenerator
Allocating virtual identifiers
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.002366 taken by MallocBasedChipIDAllocator
Generating partitioner report
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.003131 taken by PartitionerReport
2017-01-23 12:04:03 INFO: Time 0:00:00.002081 taken by
FrontEndCommonApplicationGraphNetworkSpecificationReport
Filtering edges
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.010191 taken by GraphEdgeFilter
Placing graph vertices
```

```
|0                          50%                         100%|

================================================================================
==============================
2017-01-23 12:04:03 INFO: Time 0:00:00.022417 taken by OneToOnePlacer
Generating placement report
|0                          50%                         100%|
 ============================================================
Generating placement by core report
|0                          50%                         100%|
 ============================================================
Generating SDRAM usage report
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.007935 taken by PlacerReportWithApplicationGraph
Routing
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.008062 taken by RigRoute
Allocating tags
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.004171 taken by BasicTagAllocator
Reporting Tags
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.001513 taken by TagReport
Getting the number of keys required by each edge using theapplication graph
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.009419 taken by FrontEndCommonEdgeToNKeysMapper
Allocating routing keys
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.046751 taken by MallocBasedRoutingInfoAllocator
Generating Routing info report
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.003030 taken by routingInfoReports
Generating routing tables
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.002452 taken by BasicRoutingTableGenerator
Generating Router table report
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.002024 taken by unCompressedRoutingTableReports
Compressing routing Tables
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.010309 taken by MundyRouterCompressor
Generating compressed router table report
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.001907 taken by compressedRoutingTableReports
Generating comparison of router table report
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:03 INFO: Time 0:00:00.001248 taken by comparisonOfRoutingTablesReport
Generating sPyNNaker data specifications
|0                          50%                         100%|
 ============================================================
```

```
2017-01-23 12:04:04 INFO: Time 0:00:00.500901 taken by SpynnakerDataSpecificationWriter
Loading routing data onto the machine
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:04 INFO: Time 0:00:00.014264 taken by FrontEndCommonRoutingTableLoader
Clearing tags
|0                          50%                         100%|
 ============================================================
Loading Tags
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:04 INFO: Time 0:00:00.007574 taken by FrontEndCommonTagsLoader
Executing data specifications and loading data
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:04 INFO: Time 0:00:00.240546 taken by
FrontEndCommonHostExecuteDataSpecification
Finding binaries
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:04 INFO: Time 0:00:00.005139 taken by
FrontEndCommonGraphBinaryGatherer
Loading executables onto the machine
|0                          50%                         100%|
 2017-01-23 12:04:04 INFO: Starting new HTTP connection (1): jabez.cs.man.ac.uk
============================================================
2017-01-23 12:04:07 INFO: Time 0:00:02.765911 taken by
FrontEndCommonLoadExecutableImages
2017-01-23 12:04:07 INFO: Running for 1 steps for a total of 150.0 ms
2017-01-23 12:04:07 INFO: Run 1 of 1
Initialising buffers
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:07 INFO: Time 0:00:00.037594 taken by
FrontEndCommonBufferManagerCreator
Updating run time
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:07 INFO: Time 0:00:00.048147 taken by FrontEndCommonChipRuntimeUpdater
2017-01-23 12:04:07 INFO: Time 0:00:00.031850 taken by FrontEndCommonDatabaseInterface
2017-01-23 12:04:07 INFO: Time 0:00:00.018824 taken by
FrontEndCommonNotificationProtocol
2017-01-23 12:04:07 INFO: *** Running simulation... ***
Loading buffers (64 bytes)
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:07 INFO: *** Awaiting for a response from an external source to state
its ready for the simulation to start ***
2017-01-23 12:04:07 INFO: Starting application (SCPSignal.SYNC0)
2017-01-23 12:04:07 INFO: Checking that the application has started
2017-01-23 12:04:07 INFO: *** Awaiting for a response from an external source to state
its ready for the simulation to start ***
2017-01-23 12:04:07 INFO: Application started - waiting 1.6 seconds for it to stop
2017-01-23 12:04:09 INFO: Application has run to completion
2017-01-23 12:04:09 INFO: Time 0:00:01.767999 taken by FrontEndCommonApplicationRunner
Getting spikes for Population 0
|0                          50%                         100%|
 ============================================================
Getting spikes for Population 1
|0                          50%                         100%|
 ============================================================
Getting spikes for Population 2
```

```
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 3
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 4
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 5
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 6
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 7
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 8
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 9
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 10
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 11
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 12
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 13
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 14
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 15
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 16
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 17
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 18
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 19
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 20
|0                          50%                          100%|
 ============================================================
Getting spikes for Population 21
|0                          50%                          100%|
 ============================================================
```

A chaque étape, indication du temps pris par la tâche.

1) Infos sur le job
2) Partition
3) Attente machine disponible

```
2017-01-23 12:04:03 INFO: Found board with version [Version: SC&MP 3.0.1 at
SpiNNaker:0:0:0 (built Wed Jul 20 10:07:23 2016)]
2017-01-23 12:04:03 INFO: Detected a machine on ip address 10.2.225.105 which has 856
cores and 120 links
```

4) placement des populations et génération de rapports (consultables comme pour exécution virtuelle ?)
5) routage
6) chargement de l'exécutable sur la machine

```
2017-01-23 12:04:07 INFO: *** Running simulation... ***
Loading buffers (64 bytes)
|0                          50%                         100%|
 ============================================================
2017-01-23 12:04:07 INFO: *** Awaiting for a response from an external source to state
its ready for the simulation to start ***
2017-01-23 12:04:07 INFO: Starting application (SCPSignal.SYNC0)
2017-01-23 12:04:07 INFO: Checking that the application has started
2017-01-23 12:04:07 INFO: *** Awaiting for a response from an external source to state
its ready for the simulation to start ***
2017-01-23 12:04:07 INFO: Application started - waiting 1.6 seconds for it to stop
2017-01-23 12:04:09 INFO: Application has run to completion
2017-01-23 12:04:09 INFO: Time 0:00:01.767999 taken by FrontEndCommonApplicationRunner
Getting spikes for Population 0
|0                          50%                         100%|
 ============================================================
Getting spikes for Population 1
|0                          50%                         100%|
 ============================================================

                           ……

Getting spikes for Population 20
|0                          50%                         100%|
 ============================================================
Getting spikes for Population 21
|0                          50%                         100%|
 ============================================================
```

Plus d'informations doivent être disponibles dans les rapports générés lors de l'exécution. Nous avons accès à ces rapports dans le cas d'une exécution virtuelle (cf. compte-rendu du 16/01) mais est-ce le cas ici ? Envoi d'un mail aux admins pour demander plus d'informations.