Intro Lab

This lab is meant to expose workshop participants to examples of problems which can be applied to the SpiNNaker architecture.

Installation

The software installation instructions can be found here:
https://spinnakermanchester.github.io/latest/spynnaker_install.html
https://spinnakermanchester.github.io/latest/gfe_install.html

File download

All of these examples can be found here:
https://spinnakermanchester.github.io/latest/intro_lab.html

Please download and open a terminal at the top level of the folder.

Run Applications

Below is a list of applications with the corresponding folders and execution commands, please run each script as it currently stands, and attempt to understand what the application is doing.

1. Neural Network Synfire Chain
2. Conductive Material with Applied Heat
3. Sudoku Game Through Neural Network
4. Graphic Ray Tracer of an Environment
5. Simple Learning Network

Neural Network Synfire Chain

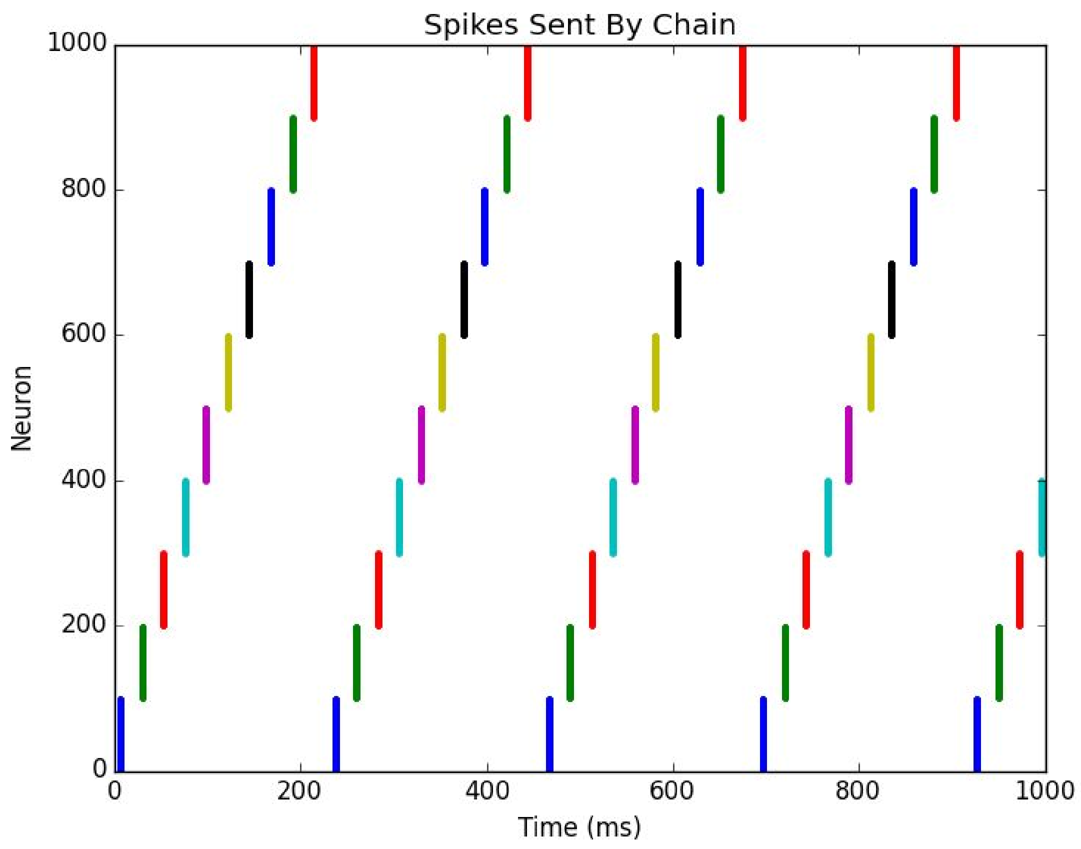


Figure 1: The output from a simple Synfire chain.

To run this example, from the top level of the folder type:

```
cd synfire
python synfire.py
```

A plot like the above should appear.

This example shows a PyNN Neural Network with a chain of 10 populations of 100 neurons each, where 10 neurons from each population excite all the neurons in the next population in the chain. The first population is then stimulated at the start of the simulation to start the chain running.

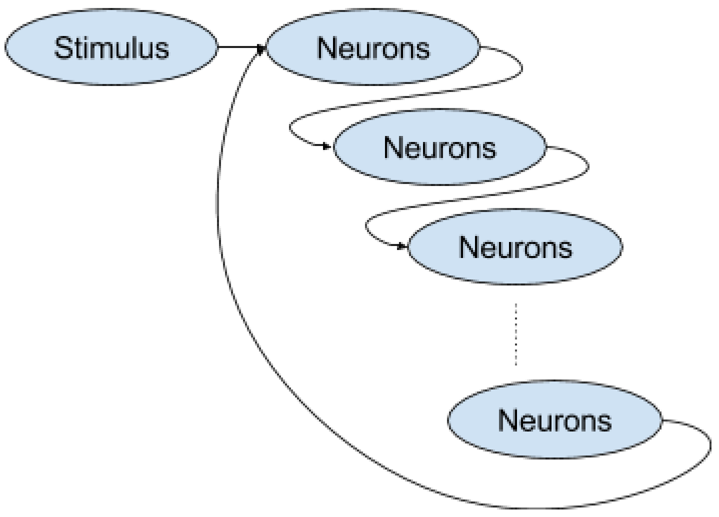


Figure 2: The Synfire Chain of Populations

Conductive Material with Applied Heat

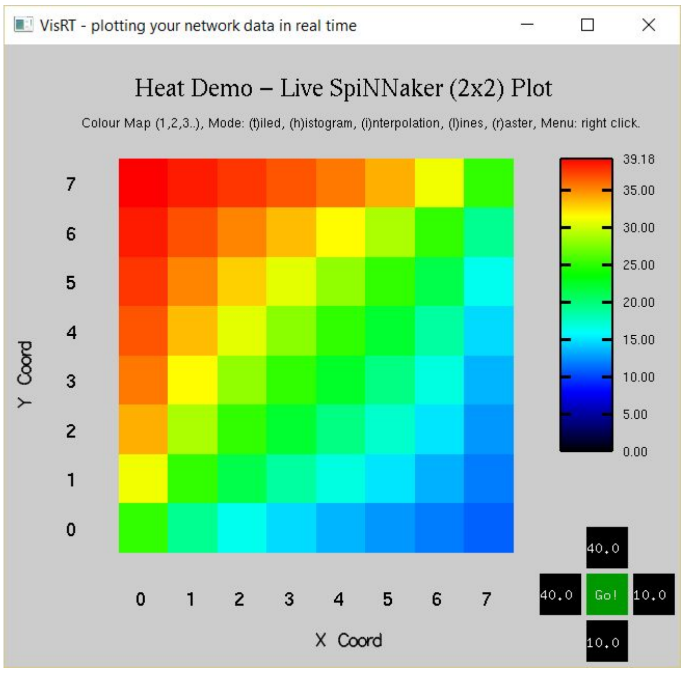

VisRT - plotting your network data in real time
Heat Demo – Live SpiNNaker (2x2) Plot
Colour Map (1,2,3..), Mode: (t)iled, (h)istogram, (i)nterpolation, (l)ines, (r)aster, Menu: right click.
Y Coord
X Coord
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
39.18
35.00
30.00
25.00
20.00
15.00
10.00
5.00
0.00
40.0
40.0
Go!
10.0
10.0

Figure 3: The output from the conductive material simulation

To run this example, from the top level of the folder type:

```
cd heat_demo
python heat_demo.py
```

A visualiser should appear here, as shown in Figure 3. You can press "9" to randomize the heat applied at each edge of the simulation, or press select a black square and press "+" to increase the temperature, or "-" to decrease it, followed by "g" to update it.

This example shows a piece of conductive material (e.g. a metal sheet) which is represented by a collections of cells which represent atoms of the material. Temperature is transferred between the atoms of the material in the simulation by sending packets over the SpiNNaker network. Figure 4 shows this application in graphical form.

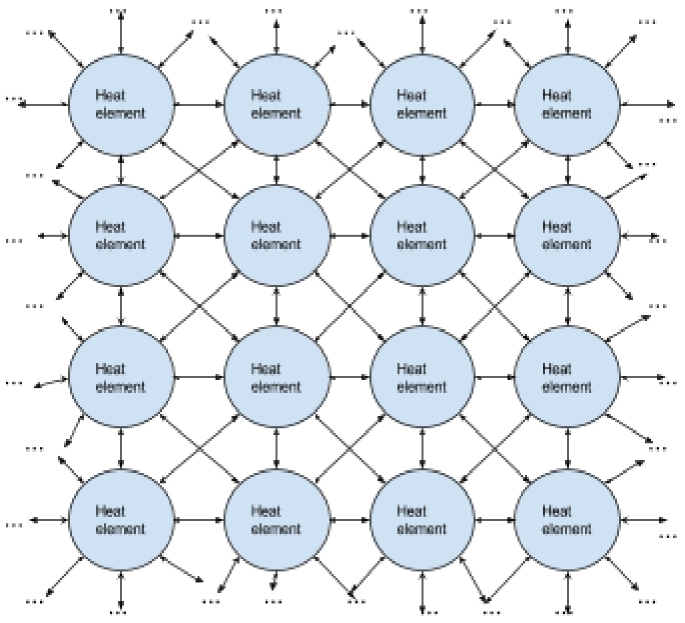
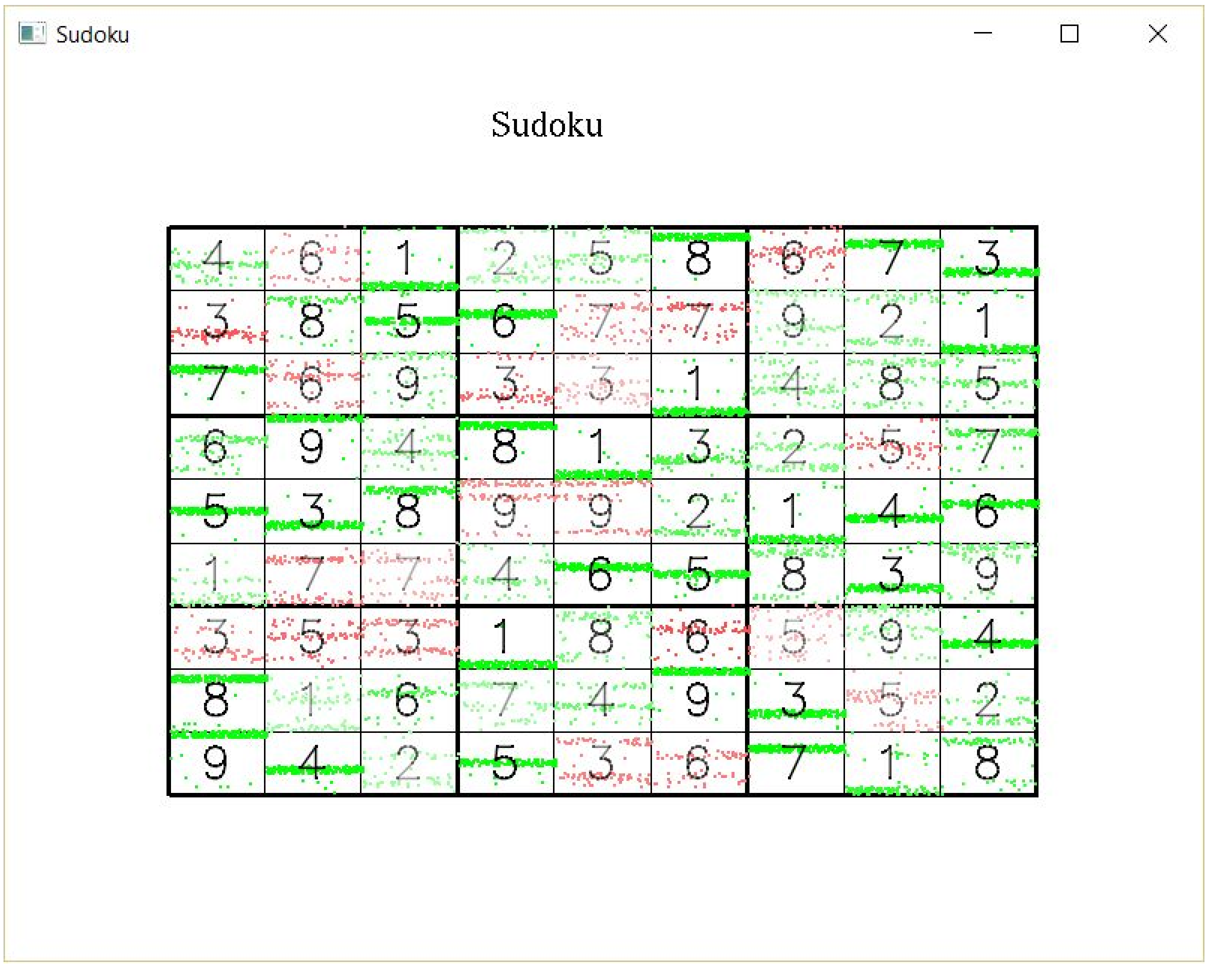


Figure 4: The conductive material application in graphical form

Sudoku Game Through Neural Network



4	6	1	2	5	8	6	7	3
3	8	5	6	7	7	9	2	1
7	6	9	3	3	1	4	8	5
6	9	4	8	1	3	2	5	7
5	3	8	9	9	2	1	4	6
1	7	7	4	6	5	8	3	9
3	5	3	1	8	6	5	9	4
8	1	6	7	4	9	3	5	2
9	4	2	5	3	6	7	1	8

Figure 5: The output from the Sudoku game application

To run this example, from the top level of the folder type:

```
cd sudoku
python sudoku.py
```

A visualiser will pop up, which is shown in Figure 5.

This example shows a PyNN neural network which describes a neural network for running sudoku problems. The spikes representing each cell are shown behind each number, with green output indicating that the value is valid according to the rules of Sudoku and red output indicating that the value is invalid. The problem to be solved is described near the top of the sudoku.py file, with 0s representing values to be computed. Note that on a small SpiNNaker board, the network is not always successful at solving the problem.

Graphic Ray Tracer of an Environment

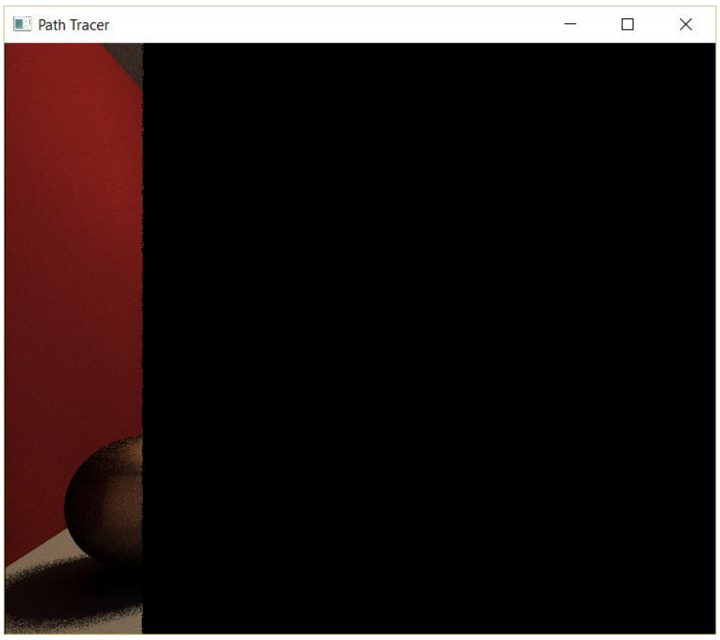


Figure 6: The output from the graphical ray tracer application

To run this example, from the top level of the folder type:

```
cd ray_trace
python ray_trace.py
```

A visualiser will pop up, which is shown in Figure 6.

This example shows a ray tracing application on SpiNNaker. This has been designed so that to operate in parallel; the more cores in use, the faster it completes. Note that it is still quite slow, and you may have to click on the window to force it to update.

Simple Learning Network

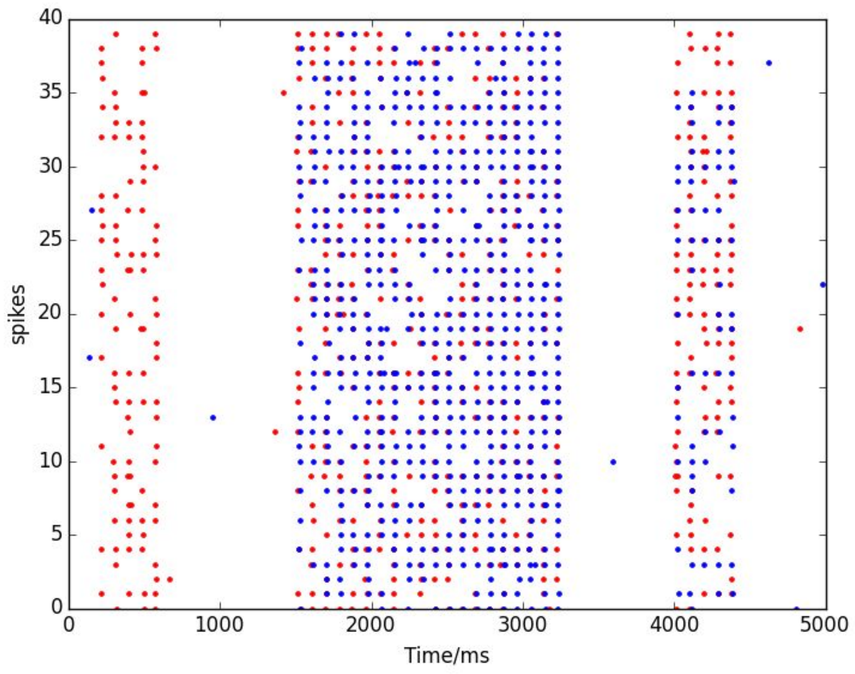


Figure 7: The output from the learning application

To run this example, from the top level folder type:

```
cd learning
python learning.py
```

A plot like the above should appear.

This example shows the spike outputs from two populations of neurons. At the start, only one of the populations spikes regularly. In the middle, some learning is done, and at the end, both populations spike regularly.