BioE 332 Lecture 8:
Programming Neurogrid -- the python way

Peiran Gao

Spring 2011

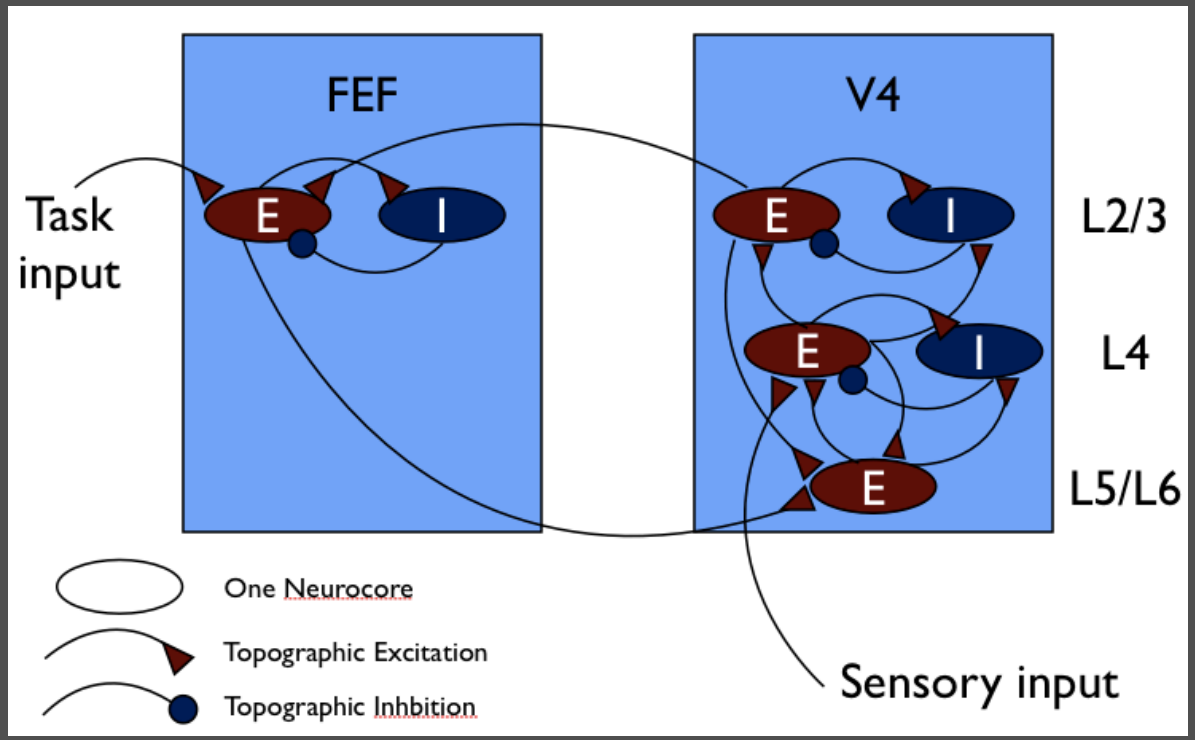
Updated by Kwabena Boahen

Spring 2012

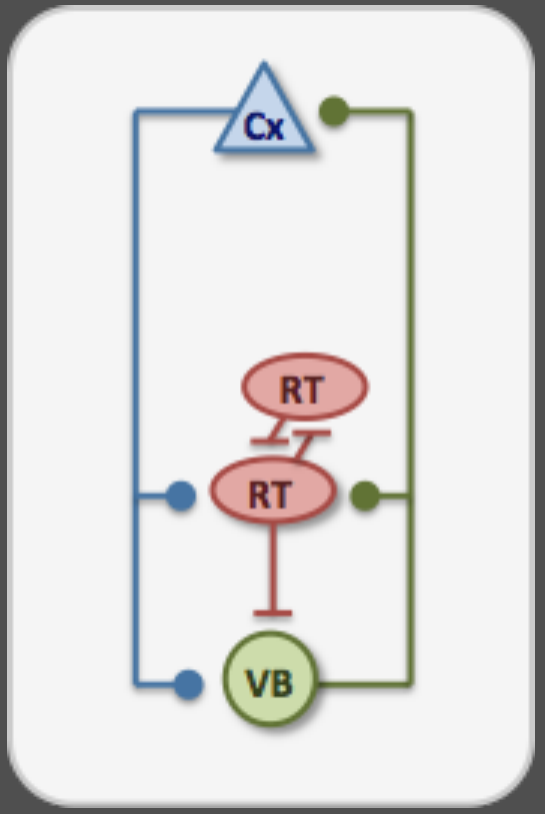
Ben Varkey Benjamin

Spring 2013

Neurogrid is Designed for Hierarchical and Modular Networks



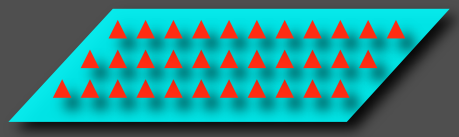
Nick Steinmetz



Christine Lee

Network Structures

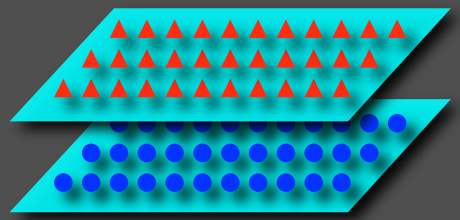
Pool



- Neurons of the same model
- Integer width by height
- Each neuron identified by unique integer coordinate
- Disable neurons to get custom shapes and/or densities

```
>>> p = Pool(nrn_quad, width = 65,
height = 65)
```

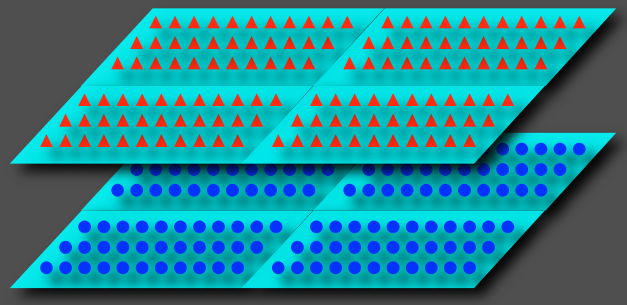
Group



- A collection of network structures (pools, groups, arrays)

```
>>> t = Group()
>>> t.AddChild(p)
```

Array

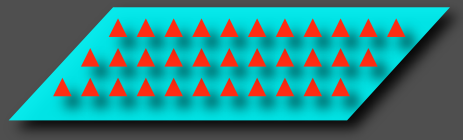


- An array of a base network structure (pools, groups, arrays) with specified repetitions in width and height

```
>>> a = Array(base = t, w = 2, h = 2)
```

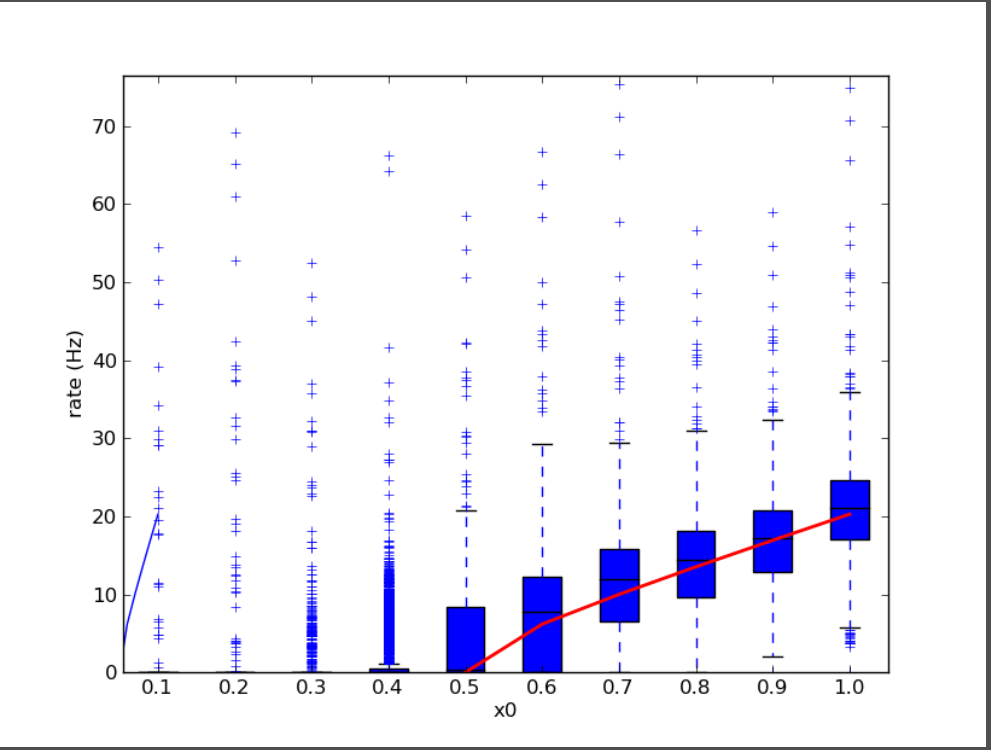
FI Curve with a Pool of Neuron

Pool

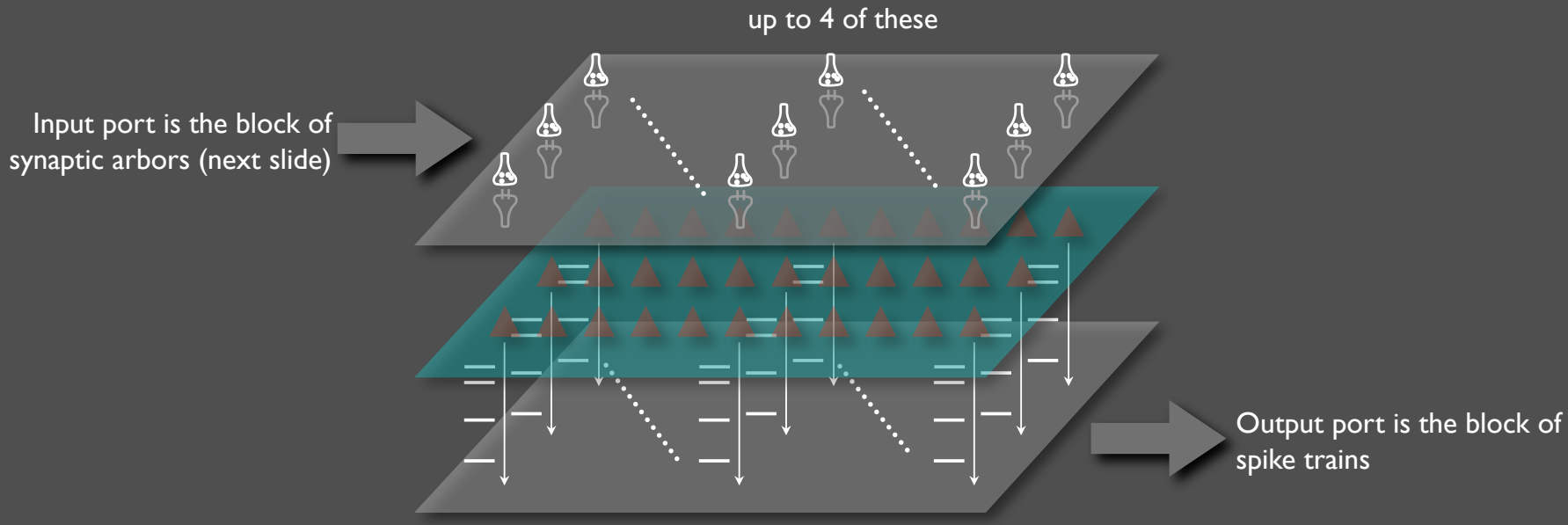


- Neurons of the same model
- Integer width by height
- Each neuron identified by unique integer coordinate
- Disable neurons to get custom shapes and/or densities

```
>>> # Set the input value
>>> x0 = .5
>>>
>>> # Create the quadratic neuron model
>>> som_quad = Soma("quadratic", {"tau": 10e-3})
>>> nrn_quad = Neuron("quad_neuron", som_quad)
>>>
>>> # Make a Pool of 64 by 64 quadratic neurons
>>> p = Pool(nrn_quad, 64, 64)
>>>
>>> # Map the Pool to hardware
>>> MapNetwork(p)
```



Connecting Pools Through Ports

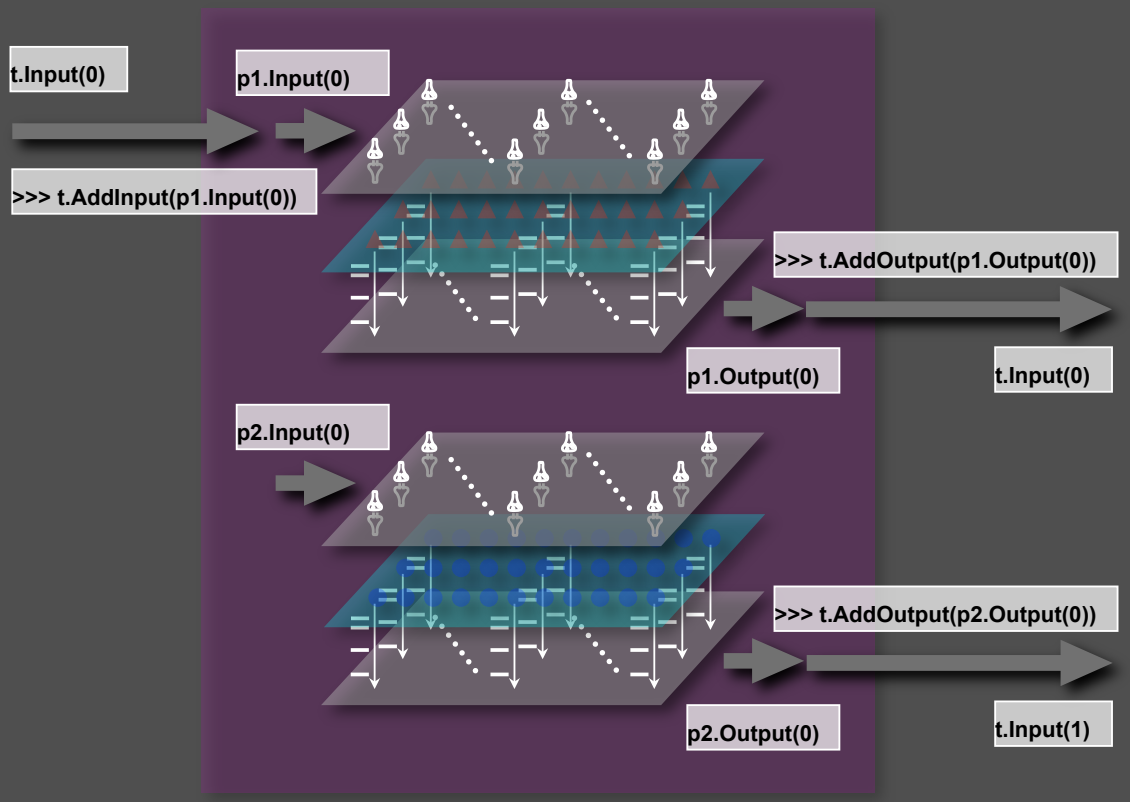


- Connect output port to input port = Projection (later slides)
- Ports are attached to pools, therefore, they inherit width and height of the pool
- A port's arbors and spike trains have the same coordinate system as its pool's neurons

Taking Ports to the Next Level

Create a group:

```
>>> t = Group()
>>> t.AddChild(p1)
>>> t.AddChild(p2)
```



Vertical Projections

- Sends spike train at (x, y) in the output port to arbor at (x, y) in the input port (topographic)
- Input/output ports must have same widths and heights, but don't have to have the same number of neurons

```
>>> # Making a Group
>>> g = Group("2pop")
>>> g.AddChild(p1)
>>> g.AddChild(p2)
>>>
>>> # Mutually connect
>>> g.VerticalProject(p1.Output(0), p2.Input(0))
>>> g.VerticalProject(p2.Output(0), p1.Input(1))
```

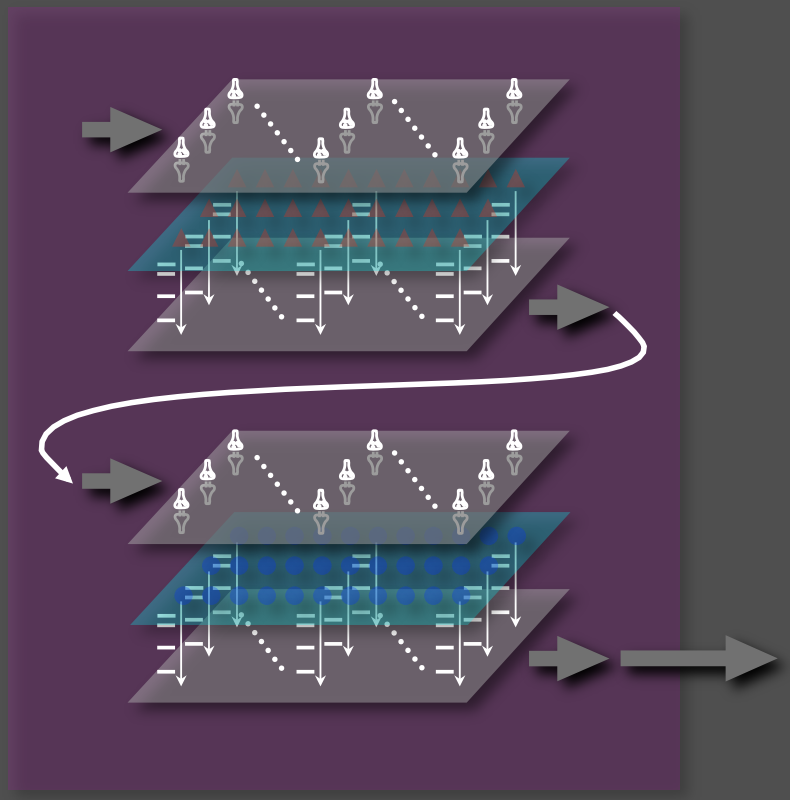
Horizontal Projections

- Sends spike train at (x1, y1) in the output port to arbor at (x2, y2) in the input port (point-to-point)
- Input/output ports must have same widths and heights, but don't have to have the same number of neurons
- You can specify a weight w < 1 that specifies the probability that a spike is delivered
- Requires the daughterboard

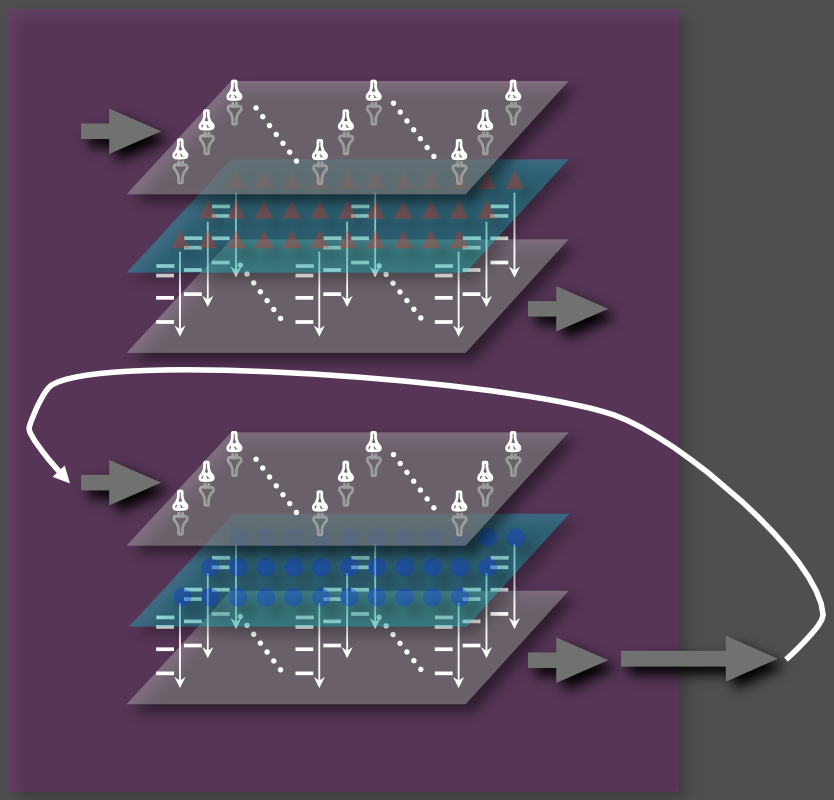
```
>>> # Making a Group
>>> g = Group("2pop")
>>> g.AddChild(p1)
>>> g.AddChild(p2)
>>>
>>> # Point-to-point connection
>>> g.HorizontalProject(p2.Output(0), x1, y1, p1.Input(0), x2 , y2 , 0.1)
```

Projections are owned by network structures, and are only to be made between ports at the same level of network hierarchy

OK



NOT OK



2 Population Interaction

```
>>> # Making model for first population
>>> syn1 = Synapse(“syn_generic”, {“erev”: .1, “tau_syn”: 10e-3, “lambda”: .6})
>>> som1 = Soma(“quadratic”, {“x0”: .7,’ “tau”: 10e-3})
>>> som1.AddSynapse(syn1)
>>> nrn1 = Neuron(“pop1_nrn”, som1)
>>> p1 = Pool(nrn1, 64, 64)
>>>
>>> # Making model for second population
>>> syn2 = Synapse(“syn_generic”, {“erev”: .1, “tau_syn”: 10e-3, “lambda”: .6})
>>> som2 = Soma(“quadratic”, {“x0”: .7, “tau”: 10e-3})
>>> som2.AddSynapse(syn2)
>>> nrn2 = Neuron(“pop2_nrn”, 64, 64)
>>> p2 = Pool(nrn2, 64, 64)
>>>
>>> # Making a Group
>>> g = Group(“2pop”)
>>> g.AddChild(p1)
>>> g.AddChild(p2)
>>>
>>> # Mutually connect
>>> g.VerticalProject(p1.Output(0), p2.Input(0))
>>> g.VerticalProject(p2.Output(0), p1.Input(1))
>>>
>>> # Map the network
>>> MapNetwork(g)
```