Visualizing Network Connectivity with ConnPlotter

Hans Ekkehard Plesser & Eilen Nordlie

Norwegian University of Life Sciences
Simula Research Laboratory
RIKEN Brain Sciences Institute

8 October 2009
Outline

Happy Birthday, Neural Network Simulators!

Network Diagrams

Connectivity Pattern Tables

ConnPlotter

Perspectives
Happy Birthday, Neural Network Simulators!
Network Simulation: 55 years!
B. G. Farley & A. W. Clark, 1954

- Simulation of self-organized systems by digital computer
- MIT Memory test computer
  - 4096 16-bit words
  - 90,000 fetch/add per sec
- 64 leaky I&F neurons
- δ-synapses w/ delay
- exponentially decaying threshold
- Gaussian noise (LFG)
- 75% connectivity
- Hebbian learning
First Neuron Class: 40 years!

- Lars Walløe, J. K. S. Jansen, Kirsten Nygaard
- *A Computer Simulated Model of a Secondary Order Sensory Neuron*
- Model of neurons in dorsal spino-cerebellar tract
- Direct comparison to experimental data
- Implemented in Simula on a Univac 1107

```plaintext
process class spindle (N, freqrest, sens);
  integer N; real freqrest, sens;
  begin real amplit, del;
    del := 1000/(freqrest + sens * length);
    comment (this statement only serves to insert clarifying text in the program)
    The delay is now given its value.
    "length" is an external parameter representing muscle stretch;
    read (amplit);
    comment the value of "amplit" is fetched from some external source of information;
    hold (uniform (0, del));
    comment this statement is described below;
    impulse: if (time-tfire) < tblock then
      begin temp := time;
        comment this is the case of blocking.
        temp is updated, no other effect.
        "impulse" is a "label", giving a name to the subsequent statement.
        "go to impulse" brings us back to this statement;
        go to impulse
      end
    else if (backgrpot + spindlepot + amplit > barrier) then
      begin fire (N); temp := time;
        comment this is the case of firing
      end
      begin amplist := spindlepot + amplit;
        temp := time
        comment this is the case of a pulse building up the membrane potential without causing a firing;
      end;
    pause: hold (normal (del, A * del);
      go to impulse
    end;
```
Network Diagrams
What makes science science?

**Refutable hypotheses**
Hypotheses must be stated with sufficient detail and precision so that one can devise meaningful tests or counterexamples.

**Reproducible experiments**
Experiments must be described and performed so carefully, that others can *reproduce* them. Genuine failure to reproduce results invalidates original findings.

**Accumulation of knowledge**
Accumulation of knowledge through exchange, evolution and (sometimes) revolution of ideas.
What makes science science?

Refutable hypotheses
Hypotheses must be stated with sufficient detail and precision so that one can devise meaningful tests or counterexamples.

Reproducible experiments
Experiments must be described and performed so carefully, that others can reproduce them. Genuine failure to reproduce results invalidates original findings.

Accumulation of knowledge
Accumulation of knowledge through exchange, evolution and (sometimes) revolution of ideas.
What makes science science?

**Refutable hypotheses**

Hypotheses must be stated with sufficient detail and precision so that one can devise meaningful tests or counterexamples.

**Reproducible experiments**

Experiments must be described and performed so carefully, that others can *reproduce* them. Genuine failure to reproduce results invalidates original findings.

**Accumulation of knowledge**

Accumulation of knowledge through exchange, evolution and (sometimes) revolution of ideas.
What makes science science?

**Refutable hypotheses**
Hypotheses must be stated with sufficient detail and precision so that one can devise meaningful tests or counterexamples.

**Reproducible experiments**
Experiments must be described and performed so carefully, that others can *reproduce* them. Genuine failure to reproduce results invalidates original findings.

**Accumulation of knowledge**
Accumulation of knowledge through exchange, evolution and (sometimes) revolution of ideas.
What do we need?

- Reliable,
- Precise,
- Expressive,
- Easy-to-Use

means to visualize our models of neuronal networks.
What do we need?

- Reliable,
- Precise,
- Expressive,
- Easy-to-Use

means to visualize our models of neuronal networks.
What do we need?

- Reliable,
- Precise,
- Expressive,
- Easy-to-Use

means to visualize our models of neuronal networks.
What do we need?

- Reliable,
- Precise,
- Expressive,
- Easy-to-Use

means to visualize our models of neuronal networks.
What do we need?

- Reliable,
- Precise,
- Expressive,
- Easy-to-Use

means to visualize our models of neuronal networks.
What do we have?
What can we do?

► Develop standards for symbols (eg Kitano et al, Nature Biotechnol 2005)

► Draw network at different levels (from Nordlie et al, 2009)

► Problems:
  ► How to generate automagically?
  ► Confusing line crossings
Dot doesn’t help ...
Connectivity Pattern Tables
NEST Topology: Simple Layers
Real networks: Complex Layers

From Hill & Tononi, J Neurophysiol, 2005, 93, 1671–1698
NEST Topology: Composite Layer Elements

Each color represents a neuron model
Connections are made by specifying entire layer and model to connect to/from
Populations, Groups, Projections

Population  Homogeneous group of neurons with 2D-layout
Group       Collection of populations, e.g., a layer
Projection  Rule for connecting two populations

Mask        Only target population neurons inside mask are connected
Kernel      Probability of connection

Synapse model
Connectivity Pattern Table (CPT)

- Connectivity matrix showing kernels & masks
- Intensity = weight × probability
Aggregate CPTs

- Condense by combining across populations, synapse models, or both

![Diagram showing aggregate CPTs with labels A, B, and C.]
Different synapse types

- Different colors
- Co-occurring types placed side-by-side

<table>
<thead>
<tr>
<th></th>
<th>IG</th>
<th>RG</th>
</tr>
</thead>
<tbody>
<tr>
<td>IG</td>
<td><img src="image1" alt="IG" /></td>
<td><img src="image2" alt="RG" /></td>
</tr>
<tr>
<td>RG</td>
<td><img src="image3" alt="RG" /></td>
<td><img src="image4" alt="RG" /></td>
</tr>
</tbody>
</table>

- E: Excitatory
- I: Inhibitory
Aggregate with synapse types
...and as CPT

<table>
<thead>
<tr>
<th></th>
<th>Ret</th>
<th>Tp</th>
<th>Rp</th>
<th>Vp_h</th>
<th>Vp_v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ret</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vp_h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vp_v</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Labels:**
- Inter
- Relay
- L23in
- L23pyr
- L4in
- L4pyr
- L56in
- L56pyr

**Colors:**
- Red
- Blue
- Green
- Purple

**Legend:**
- Red: Connection
- Blue: Indirect Connection
- Green: High Connection
- Purple: Very High Connection
### Partially Aggregated CPTs

<table>
<thead>
<tr>
<th></th>
<th>Ret</th>
<th>Tp</th>
<th>Rp</th>
<th>Vp_h</th>
<th>Vp_v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ret</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vp_h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vp_v</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Connections
- Inter
- Relay
- L23in
- L23pyr
- L4in
- L4pyr
- L56in
- L56pyr
- L23in
- L23pyr
- L4in
- L4pyr
- L56in
- L56pyr
Fully Aggregated CPTs

<table>
<thead>
<tr>
<th></th>
<th>Ret</th>
<th>Tp</th>
<th>Rp</th>
<th>Vp_h</th>
<th>Vp_v</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ret</td>
<td></td>
<td>Tp</td>
<td>Rp</td>
<td>Vp_h</td>
<td>Vp_v</td>
</tr>
<tr>
<td>Tp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vp_h</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vp_v</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ConnPlotter
ConnPlotter: AutoMagic CPTs

- Python package
- Flexible generation of CPTs
- CPTs built from NEST Topology network specifications
- Use same code to build and draw models!
Example: Simple network

```
modelList = [('poisson_generator', 'P', {'rate': 10.0}),
             ('iaf_neuron', 'E', {'C_m': 200.0}),
             ('iaf_neuron', 'I', {'C_m': 150.0})]

layerList = [('IG', {'columns': 40, 'rows': 40,
                      'extent': [1.0, 1.0],
                      'elements': 'P'}),
             ('RG', {'columns': 40, ..., 'elements': ['E']})]

connectList = [('IG', 'RG',
                modCopy(common, {'connection_type': 'divergent',
                                 'synapse_model': 'static_synapse',
                                 'targets': {'model': 'E'},
                                 'mask': {'circular': {'radius': 0.8,
                                                      'weights': 2.0,
                                                      'delays': 1.0}}}),
                ... ]
```
import ConnPlotter as cpl
s_cp = cpl.ConnectionPattern(layerList, connectList)
s_cp.plot()
s_cp.plot(normalize=True)
s_cp.plot(mode='layer')
s_cp.plot(mode='totals')
s_cp.plot(mode='totals', normalize=True)
s_cp.plot(file='mycpt.eps')

cpt = cpl.ConnectionPattern(layerList, connectList,
    synTypes = ( ( cpl.SynType('AMPA', 1, 'red' ),
    cpl.SynType('NMDA', 1, 'green' ) )
    ( cpl.SynType('Dopa', 0.5, 'orange'),
    cpl.SynType('Sero', 0.2, 'brown' ) )
)
Creating the network

for model in modelList:
    nest.CopyModel(model[0], model[1], model[2])

for layer in layerList:
    exec '%s = topo.CreateLayer(layer[1])' % layer[0]

for conn in connectList:
    exec 'topo.ConnectLayer(%s,%s,conn[2])' % (conn[0], conn[1])
Perspectives
Perspectives

- Not all kernels (even in NEST Topology) supported right now
- Non-square populations don’t work 100% yet
- Non-centered projections not implemented
- Ignores boundary conditions
- Must become compatible with PyNN
- Do you like CPTs?
Collaborators

Eilen Nordlie

Marc-Oliver Gewaltig