Which Features Matter How Much When
Memory Patterns and Learning in NeurOS

Lee Scheffler, Cognitivity
What This Is All About

Features

Patterns

Learning, recognition

Generation/reification

Pattern recombination

Cascades/layers/meshes of patterns

One-shot, continuous

Sets, sequences, temporal
Approach

(Braitenberg: "...downhill invention...")

Build, run and improve cognitive functions *by interconnecting* reusable biologically inspired components

*Open, integrative, NOT one-size-fits-all*
Talk Overview

• Brief tour of NeurOS/NeuroBlocks
• **Memory patterns**
  – Sets, Sequences, Temporal
  – Reification
• Example cognitive functions
  – unsupervised learning
  – concurrent exemplars & stereotypes
  – prediction
  – layers of patterns
  – labels as synonyms (supervised learning)
  – imagination
  – context disambiguation
  – attention

© Copyright 2016 Lee Scheffler
NeurOS Designer IDE

- design canvas
- module
- link
- neural circuit: directed graph, event flow, loops
- drag & drop
- reusable modules

© Copyright 2016 Lee Scheffler
# Basics (1)

**neural circuit (graph):**

<table>
<thead>
<tr>
<th>module</th>
<th>Group/layer of neurons with similar function; state</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>Multiplexed event signal path: axons of multiple neurons</td>
</tr>
<tr>
<td>neural circuit</td>
<td>Directed signal flow graph, loops, nestable subgraphs</td>
</tr>
<tr>
<td>event</td>
<td>New spiking rate of a neuron</td>
</tr>
</tbody>
</table>
neural circuit (graph):

- **module**: Group/layer of neurons with similar function; state
- **link**: Multiplexed event signal path: axons of multiple neurons
- **neural circuit**: Directed signal flow graph, loops, nestable subgraphs
- **event**: New spiking rate of a neuron
neural circuit (graph):

<table>
<thead>
<tr>
<th>module</th>
<th>Group/layer of neurons with similar function; state</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>Multiplexed event signal path: axons of multiple neurons</td>
</tr>
<tr>
<td>neural circuit</td>
<td>Directed signal flow graph, loops, nestable sub-graphs</td>
</tr>
<tr>
<td>event</td>
<td>New spiking rate of a neuron</td>
</tr>
</tbody>
</table>
neural circuit (graph):

<table>
<thead>
<tr>
<th>module</th>
<th>Group/layer of neurons with similar function; state</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>Multiplexed event signal path: axons of multiple neurons</td>
</tr>
<tr>
<td>neural circuit</td>
<td>Directed signal flow graph, loops, nestable sub-graphs</td>
</tr>
<tr>
<td>event</td>
<td>New spiking rate of a neuron</td>
</tr>
</tbody>
</table>
neural circuit (graph):

-events--->

links

neuron spiking

Time (milliseconds)

relative neuron spike rate

event code:
(0, id,1.0)  (25, id,0)  (50, id, 0.5)  (75, id,0.33)  (120, id,0)
# Module Types

<table>
<thead>
<tr>
<th>inputs (senses)</th>
<th>keyboard</th>
<th>file in</th>
<th>temporal pattern in</th>
<th>grid in</th>
<th>data gen</th>
<th>stream in</th>
<th>MIDI in</th>
</tr>
</thead>
<tbody>
<tr>
<td>processing</td>
<td>filter</td>
<td>transformer</td>
<td>Max Avg</td>
<td>group ops</td>
<td>wrapper</td>
<td>subgraph</td>
<td></td>
</tr>
<tr>
<td>memory</td>
<td>working memory</td>
<td>set patterns</td>
<td>sequence patterns</td>
<td>temporal patterns</td>
<td>pattern reify</td>
<td></td>
<td></td>
</tr>
<tr>
<td>outputs (actions, effectors)</td>
<td>print</td>
<td>plot</td>
<td>cloud</td>
<td>grid out</td>
<td>stream out</td>
<td>log</td>
<td>MIDI out</td>
</tr>
</tbody>
</table>
Long-Term Memory: Feature Patterns

• **Feature**: any distinct concept at any level of abstraction
  – any distinct neural (axonal) signal
  – spiking rate of a neuron

• **Pattern**: collection of weighted features
  – like a neuron or neuron assembly
  – optional expected value, time distributions
  – *no predefinition of input feature space*

• **Matching (recognition)**: sum of weight-value products
  – normalized, difference/error tolerant
  – output is a relative spiking rate reflecting match confidence

• **Pattern space**: collection of patterns
  – managed/accessed by one or more Modules

• **Cascaded/layered patterns of patterns**
  – a pattern is a feature

*NOT neural networks!*
## Memory Pattern/Module Types

<table>
<thead>
<tr>
<th>Sets</th>
<th>Concurrent feature collections in any order</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Semantic range: any/OR, a few, some, many, most, all/AND</td>
</tr>
<tr>
<td>Sequences</td>
<td>Time-independent sequences of features</td>
</tr>
<tr>
<td></td>
<td>Parameters for non-exact sequence matching</td>
</tr>
<tr>
<td>Temporal</td>
<td>Time-relative sequences of multiple features</td>
</tr>
<tr>
<td>Sequences</td>
<td>Parameters for non-exact matching, speed range</td>
</tr>
<tr>
<td>Reify</td>
<td>Inverse: generate pattern features</td>
</tr>
<tr>
<td></td>
<td>e.g., prediction, feedback, imagination</td>
</tr>
</tbody>
</table>

© Copyright 2016 Lee Scheffler
Learning Rules

• Compute and emit match scores for all (relevant) patterns in pattern space
  – these patterns are features too!
• If some pattern match score exceeds novelty threshold, update best matching pattern:
  – adjust feature weights; anneal over repetition
  – add/remove features
• Otherwise, create a new pattern
• Forget patterns which rarely/never match
Cognitive Function Examples

• unsupervised learning
• concurrent exemplars & stereotypes
• prediction
• layers of patterns
• labels as synonyms (supervised learning)
• imagination
• context disambiguation
• attention
Unlabeled Data

fur, snout, barks, big, 4 feet, brown, tail
hair, snout, whines, medium, 4 feet, black, tail, friendly, slobbers
fur, yips, small, 4 feet, white, nasty
fur, tail, small, black, 4 feet, pointy ears, meows, whiskers
fur, tail, small, grey, 4 feet, pointy ears, meows, whiskers
tail, calico, purrs, 4 feet, whiskers, claws

Small dataset, one-shot/continuous incremental learning
Unsupervised Learning
Pattern space
Learning controls
Match score normalization (some/many semantic)
Novelty threshold
<table>
<thead>
<tr>
<th>pattern id:</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>stereotype_1</th>
<th>stereotype_2</th>
<th>group_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 feet</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>barks</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.69</td>
<td></td>
<td>0.14</td>
</tr>
<tr>
<td>big</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.69</td>
<td></td>
<td>0.14</td>
</tr>
<tr>
<td>black</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.31</td>
<td>0.18</td>
<td>0.22</td>
</tr>
<tr>
<td>brown</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.69</td>
<td></td>
<td>0.14</td>
</tr>
<tr>
<td>calico</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.23</td>
<td></td>
<td>0.33</td>
</tr>
<tr>
<td>claws</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.23</td>
<td></td>
<td>0.33</td>
</tr>
<tr>
<td>friendly</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fur</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.69</td>
<td>0.77</td>
<td>0.59</td>
</tr>
<tr>
<td>grey</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0.26</td>
<td></td>
<td>0.22</td>
</tr>
<tr>
<td>hair</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>medium</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meows</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0.43</td>
<td></td>
<td>0.37</td>
</tr>
<tr>
<td>nasty</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pointy ears</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>0.43</td>
<td></td>
<td>0.37</td>
</tr>
<tr>
<td>purrs</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td>0.23</td>
<td></td>
<td>0.33</td>
</tr>
<tr>
<td>slobbers</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>small</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>0.77</td>
<td></td>
<td>0.45</td>
</tr>
<tr>
<td>snout</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td>0.22</td>
</tr>
<tr>
<td>tail</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.67</td>
<td>0.92</td>
</tr>
<tr>
<td>whines</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>whiskers</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>0.67</td>
<td></td>
<td>0.7</td>
</tr>
<tr>
<td>white</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>yips</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Pattern Match Scores

<table>
<thead>
<tr>
<th>curve min-max:</th>
<th>pattern id:</th>
<th>0.25 - 1</th>
<th>0.5 - 0.0</th>
<th>0.0 - 0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#1</td>
<td>#2</td>
<td>#3</td>
<td>#4</td>
</tr>
<tr>
<td>fur, barks, tail</td>
<td>0.24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>small, meows, whiskers</td>
<td></td>
<td>0.17</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>small, meows, whiskers, black</td>
<td></td>
<td>0.33</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>big, brown, barks, tail</td>
<td>0.43</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Concurrent Exemplars & Stereotypes
Prediction

Partial features ➔ pattern recognition ➔ regeneration of missing features
Layers of Patterns

F = \{a, e, f, g1\}
Layers of Patterns

\( F = \{a, e, f, g1\} \)

<table>
<thead>
<tr>
<th>letter</th>
<th>A</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>N</th>
<th>O</th>
<th>R</th>
<th>S</th>
<th>U</th>
<th>V</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g2</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>g1</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>i</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>h</td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>m</td>
<td></td>
<td>1.0</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>k</td>
<td></td>
<td>1.0</td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>j</td>
<td></td>
<td></td>
<td>1.0</td>
<td></td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Layers of Pattern Recombination
(including bi-directionality)

pixels → motion

edges → curves

corners → objects

letters → ngrams → words ↔ phrases

syllables ↔ phonemes

intervals → durations

chords → musical phrases → melodies

sentences ↔ captions

songs

© Copyright 2016 Lee Scheffler
Labeled Data

dog = fur, snout, barks, big, 4 feet, brown, tail

dog = hair, snout, whines, medium, 4 feet, black, tail, friendly, slobbers

dog = fur, yips, small, 4 feet, white, nasty

cat = fur, tail, small, black, 4 feet, pointy ears, meows, whiskers

cat = fur, tail, small, grey, 4 feet, pointy ears, meows, whiskers

cat = tail, calico, purrs, 4 feet, whiskers, claws
<table>
<thead>
<tr>
<th>Type</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dog</td>
<td>fur, snout, barks, big, 4 feet, brown, tail</td>
</tr>
<tr>
<td>Dog</td>
<td>hair, snout, whines, medium, 4 feet, black, tail, friendly, slobbers</td>
</tr>
<tr>
<td>Dog</td>
<td>fur, yips, small, 4 feet, white, nasty</td>
</tr>
<tr>
<td>Cat</td>
<td>fur, tail, small, black, 4 feet, pointy ears, meows, whiskers</td>
</tr>
<tr>
<td>Cat</td>
<td>fur, tail, small, grey, 4 feet, pointy ears, meows, whiskers</td>
</tr>
<tr>
<td>Cat</td>
<td>tail, calico, purrs, 4 feet, whiskers, claws</td>
</tr>
</tbody>
</table>
Labels as Synonyms
(Supervised Learning)
Context Disambiguation

sports_context = batter, hit, ground, throw, ball, strike, out, safe, pitch, beat, score, single, double, triple
music_context = conductor, orchestra, chorus, baton, pitch, beat, score, arpeggio, chord, key, note
cooking_context = chef, kitchen, eggs, flour, spatula, beat, pan, pot, batter
Context Disambiguation

sports_context = batter, hit, ground, throw, ball, strike, out, safe, pitch, beat, score, single, double, triple

music_context = conductor, orchestra, chorus, baton, pitch, beat, score, arpeggio, chord, key, note

cooking_context = chef, kitchen, eggs, flour, spatula, beat, pan, pot, batter

The chef broke the eggs and used a whisk to beat them.
The conductor waved his baton at the orchestra to keep the beat.
The batter hit the pitch on the ground and beat the throw to first.
Context Disambiguation

sports_context = batter, hit, ground, throw, ball, strike, out, safe, pitch, beat, score, single, double, triple
music_context = conductor, orchestra, chorus, baton, pitch, beat, score, arpeggio, chord, key, note
cooking_context = chef, kitchen, eggs, flour, spatula, beat, pan, pot, batter

The chef broke the eggs and used a whisk to beat them

The conductor waved his baton at the orchestra to keep the beat

The batter hit the pitch on the ground and beat the throw to first
Attention: Global Variables

- roaring: 1.0
- hissing: 1.0
- thunder: 1.0
- fire: 1.0
- lightning: 1.0
- snake: 1.0
- tiger: 1.0
- yelling: 1.0

Graph parameter updates:

G.DANGER = IN.value

- sense inputs
  - sense inputs
  - apply gain: \( \min(1, 0.5 + G.DANGER) \)
  - plot

- working mem
- danger patterns
- update DANGER
Attention: Global Variables

- roaring: 1.0
- hissing: 1.0
- thunder: 1.0
- fire: 1.0
- lightning: 1.0
- snake: 1.0
- tiger: 1.0
- yelling: 1.0

Graph parameter updates:

```
G.DANGER = IN.value
```

- Working mem
- Danger patterns
- Update DANGER

Sense inputs: keyboard input

```
tree grass tiger tree grass roaring tree grass snake grass grass grass grass grass grass grass
```

Plot

© Copyright 2016 Lee Scheffler
Summary and Conclusions

Features → Patterns

learning, recognition

generation/reification

pattern recombination

cascades/layers/meshes of patterns

one-shot, continuous

sets, sequences, temporal

© Copyright 2016 Lee Scheffler
Layers of Pattern Recombination (including bi-directionality)

- pixels → motion
- curves
- objects
- letters → ngrams → words ↔ phrases
- scenes
- captions
- sentences
- phonemes ↔ syllables
- intervals
- durations
- musical phrases
- melodies
- songs
- tones
- durations
- chords
- instrument voices

© Copyright 2016 Lee Scheffler
NeurOS/NeuroBlocks Futures

development kit: GUI, modules, run-time

cognitive functions/systems: more, diverse, complex "cognitive Olympics"

modules: inputs, outputs, memory, processing external technology wrappers, custom modules

application areas: robotics, Internet of Things, ...

tools: improvements

platforms: ports, distributed systems, custom hardware

sharing: open ecosystem, modules, assemblies, apps
Thank You

www.Cognitivity.Technology