# Architectures of Intelligence

Lecture 10

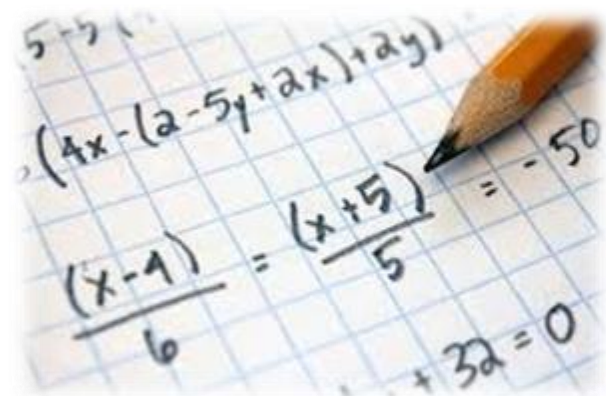
**Neural Networks: from Spikes to Symbols**

Jelmer Borst

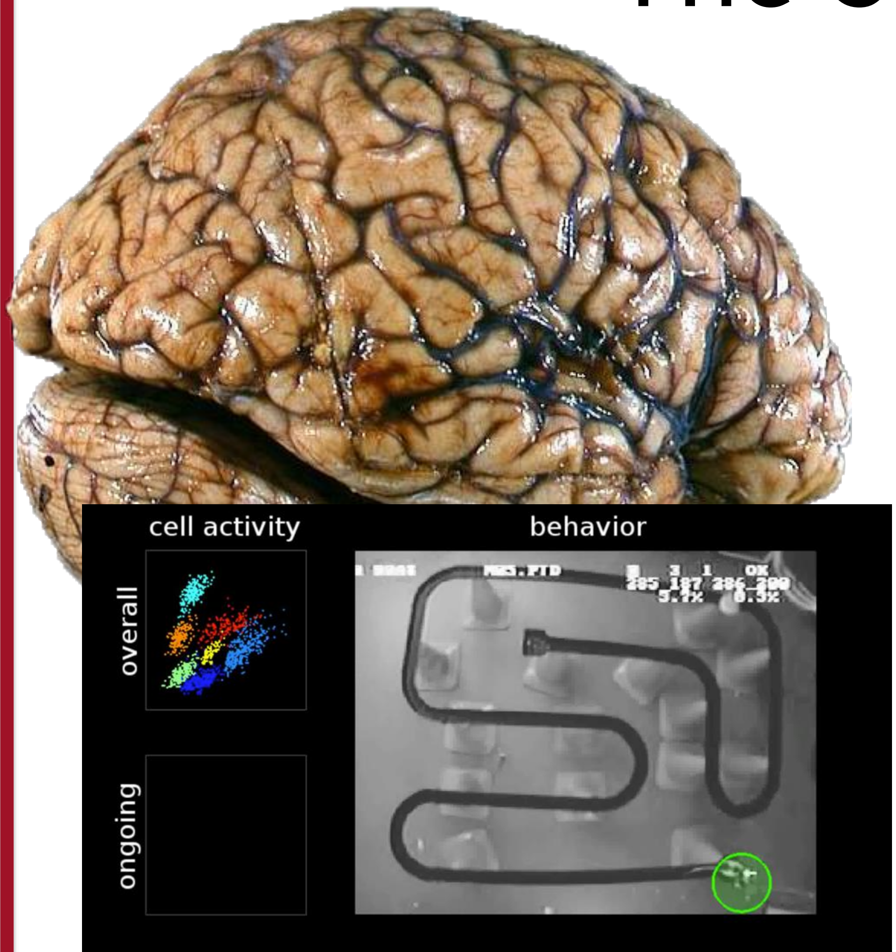
university of groningen

# Today

- Neural Networks II: from spikes to symbols
  - Spikes & Symbols
  - Symbol manipulation
  - Cognitive Control
  - Learning in Nengo
- Zbrodoff in Nengo

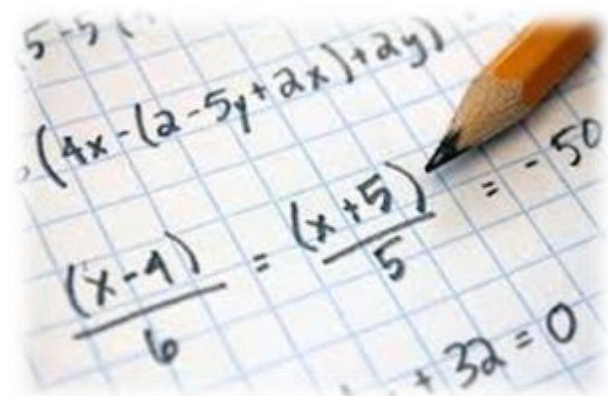


# The Challenge

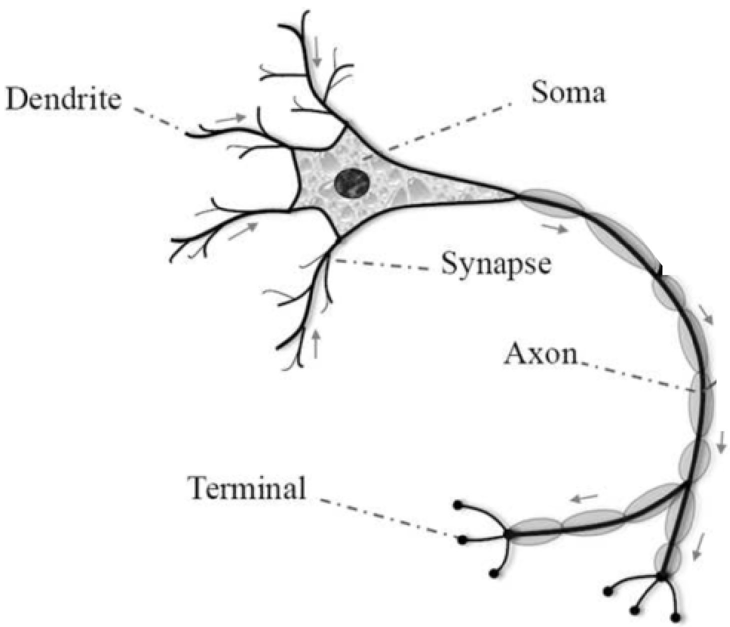



86 billion spiking
neurons

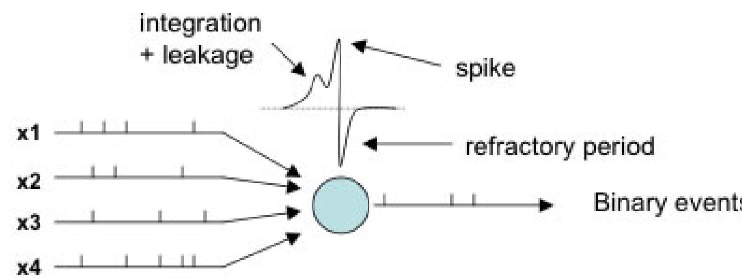
Higher-level
cognitive functions



# Recap: Spiking Neurons & Values

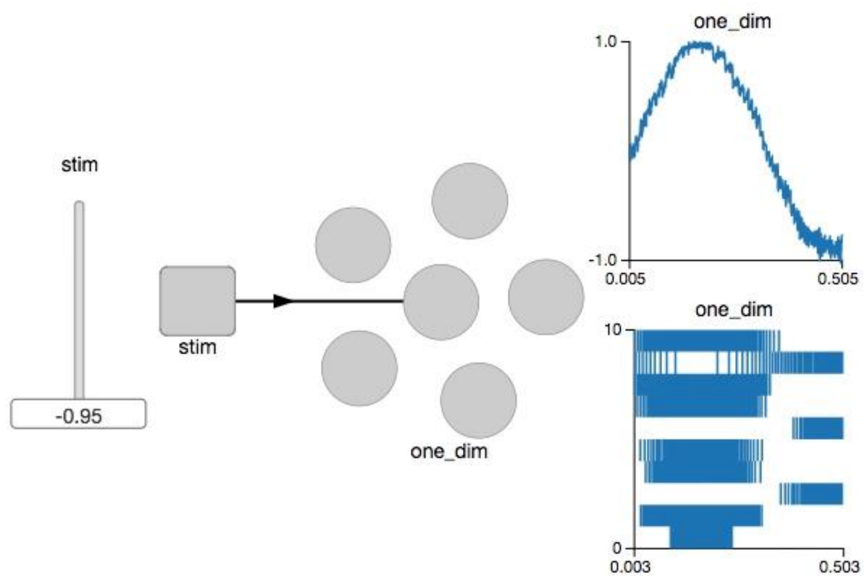



**Artificial spiking neurons**
(leaky-integrate-and-fire neurons)



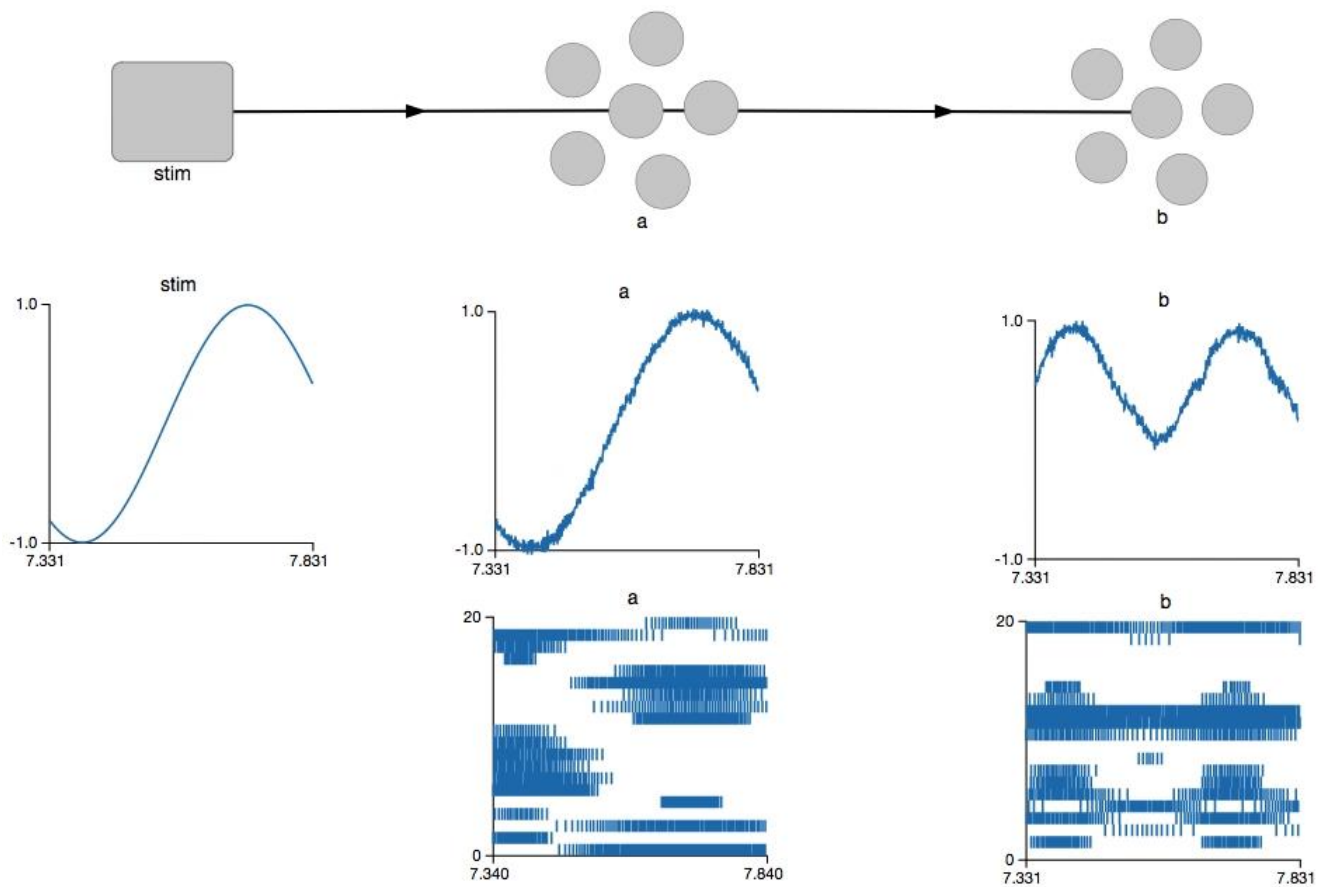
# Representing values

- Ensembles of neurons represent values



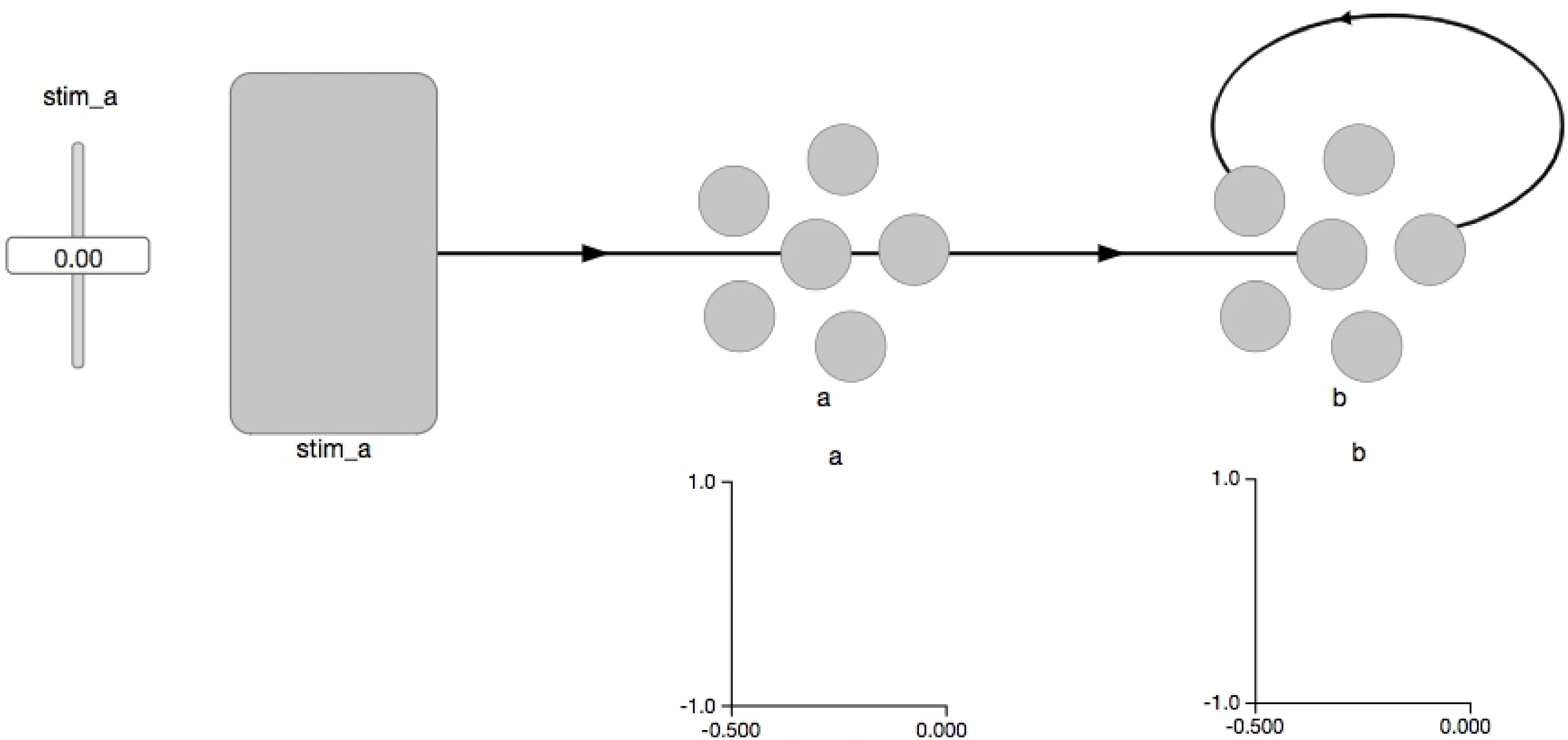
# Representing values

- Values can be communicated and transformed

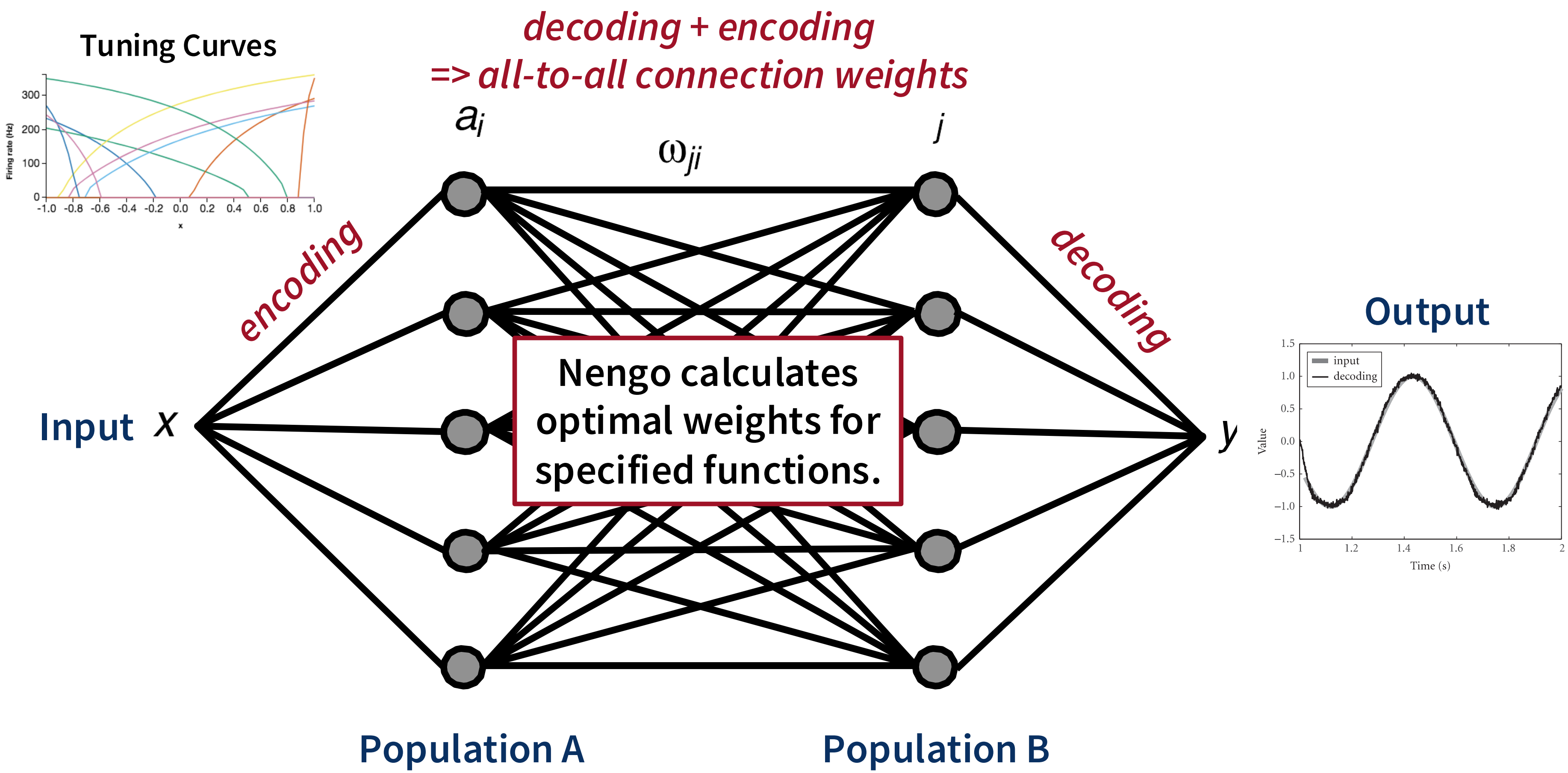


# Representing values

- and stored through recurrent connections



# Encoding and Decoding



?

# Challenges for (Spiking) Neurons

- How to represent values?
- How to communicate values?
- How can we store values?

- How to represent symbols?
- How to ground symbols in the world?

# Challenges for (Spiking) Neurons

- How to represent values?
- How to communicate values?
- How can we store values?

- How to represent symbols?
- How to ground symbols in the world?

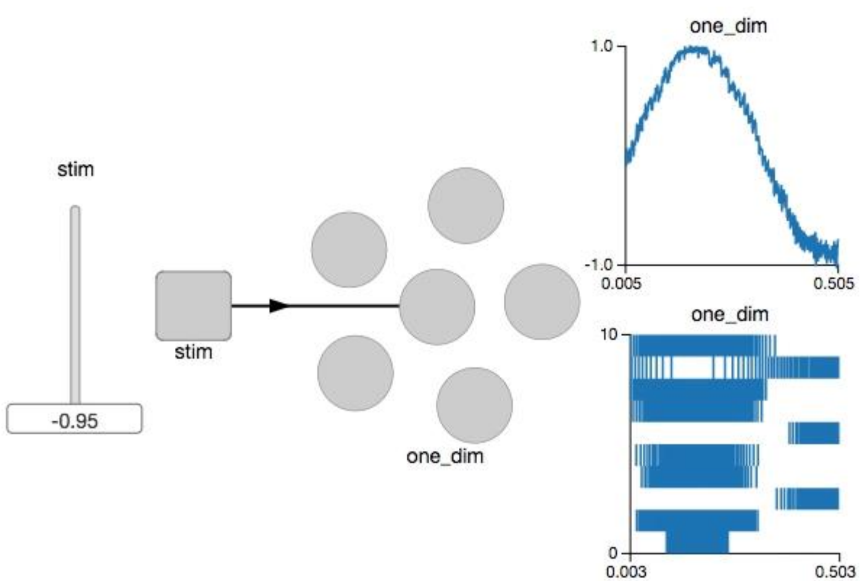
**What are symbols and why do we need them?**

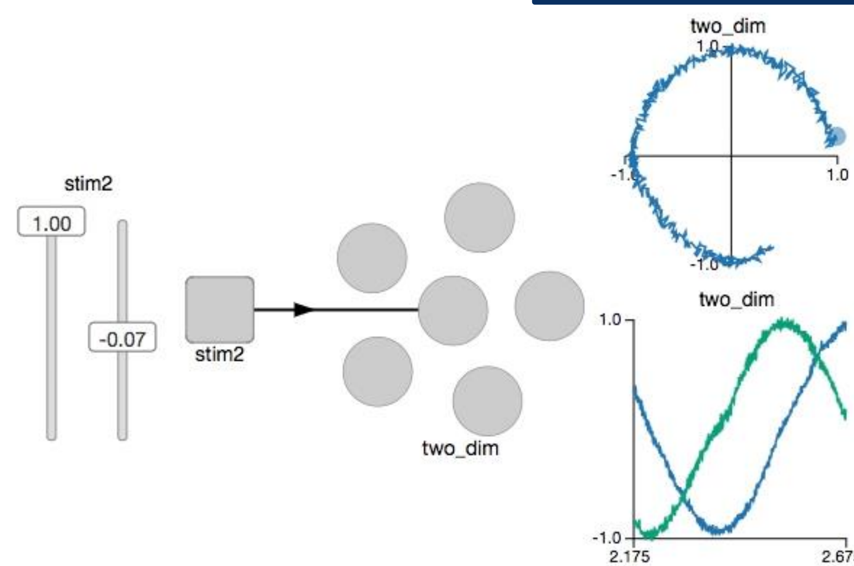
# Symbols in Spiking Neurons

- Symbols are different from values
- Easy to represent in multiple dimensions – same thing in neurons!

… or are they?
1 = chair
2 = table

[1,0] = chair
[0,-1] = table
[1, 1] = couch

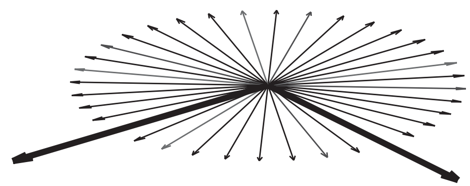


# Symbols in Spiking Neurons

- Symbols are different from values
- Easy to represent in multiple dimensions – same thing in neurons!
- Two dimensions are insufficient

A. 2-D: 36 vectors



# Symbols in Spiking Neurons

- Symbols are different from values
- Easy to represent in multiple dimensions – same thing in neurons!
- Two dimensions are insufficient
- With ~700 dimensions we can store the whole human vocabulary of ~100,000 concepts in ~40,000 neurons (= .000047% of the brain)
- Symbols are referred to as Semantic Pointers

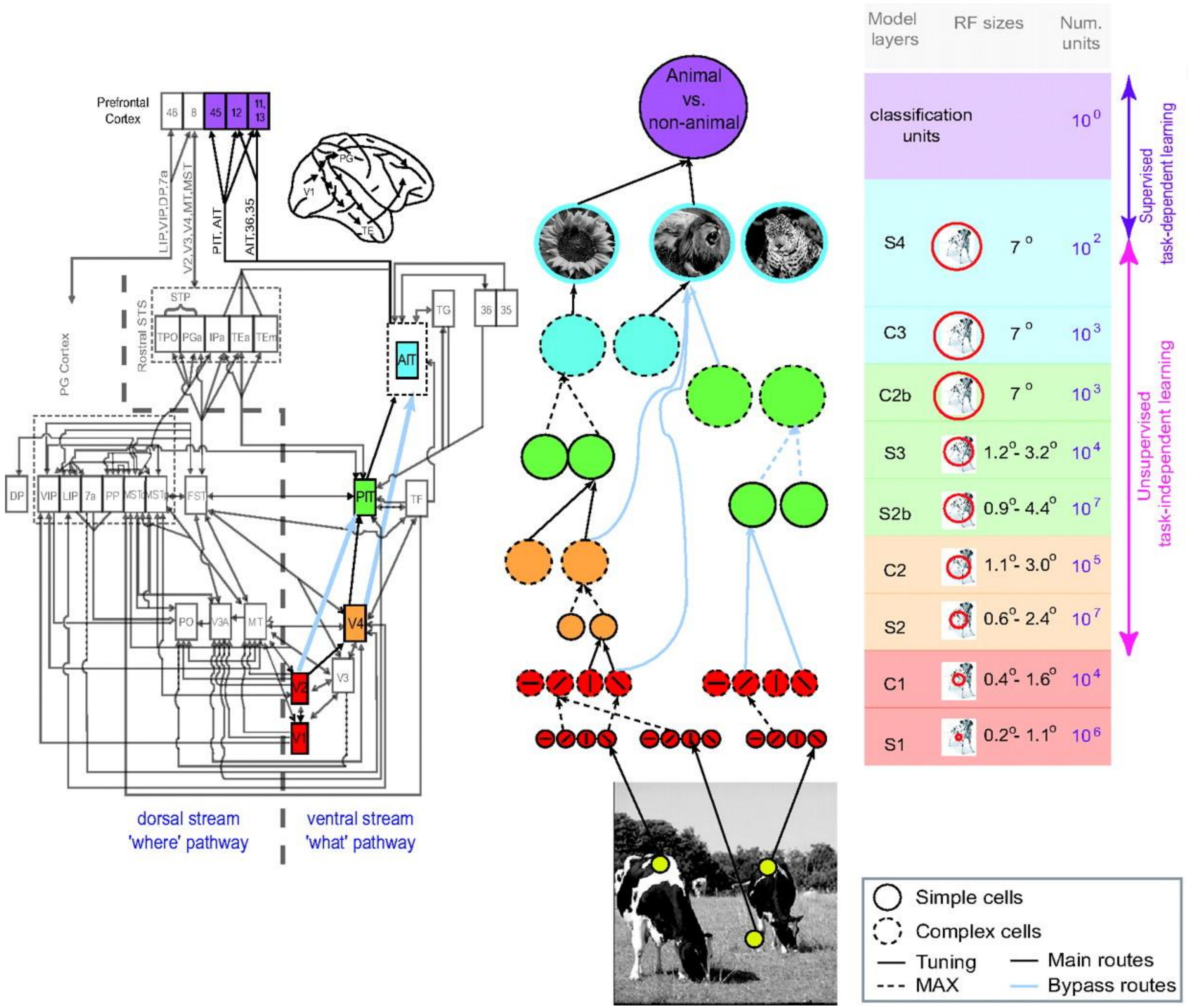
**Demo!**

# Challenges for (Spiking) Neurons

- How to represent values?
- How to communicate values?
- How can we store values?

- How to represent symbols?
- How to ground symbols in the world?

# SPA: Semantic Pointer Architecture




Serre et al., 2007 PNAS

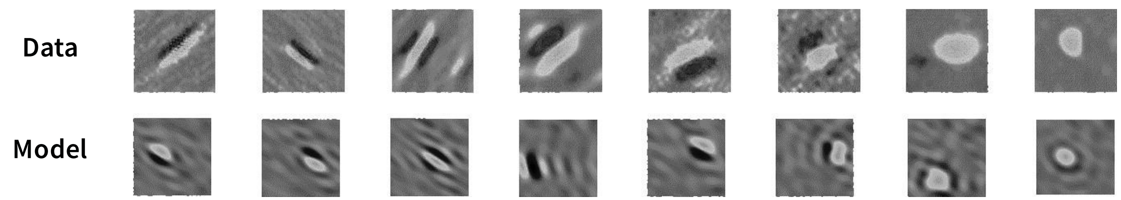
# SPA: Semantic Pointer Architecture

- High-dimensional vectors are referred to as semantic pointers (=symbols)
- The pointer is the activity of the top level of a standard visual processing hierarchy
- The pointer is a compressed representation of the full percept (≈ jpeg)
- The pointer can be used for ‘symbol manipulation’ (≈ jpeg file)
- It can also be used to reactivate a full visual representation (lossy ‘dec

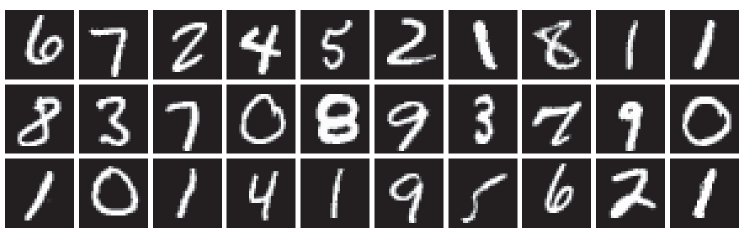
Visual = Perceptual

# Semantic Pointers applied to Numbers

**Tuning Curves in V1**



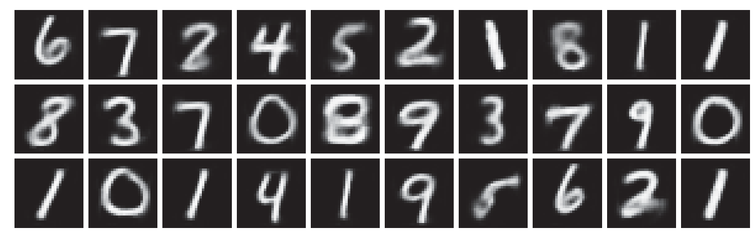

**Input**



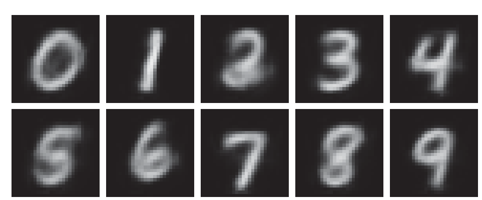
28x28 pixels = 784 dimensions

**compressed to 50 dimensional SPs**

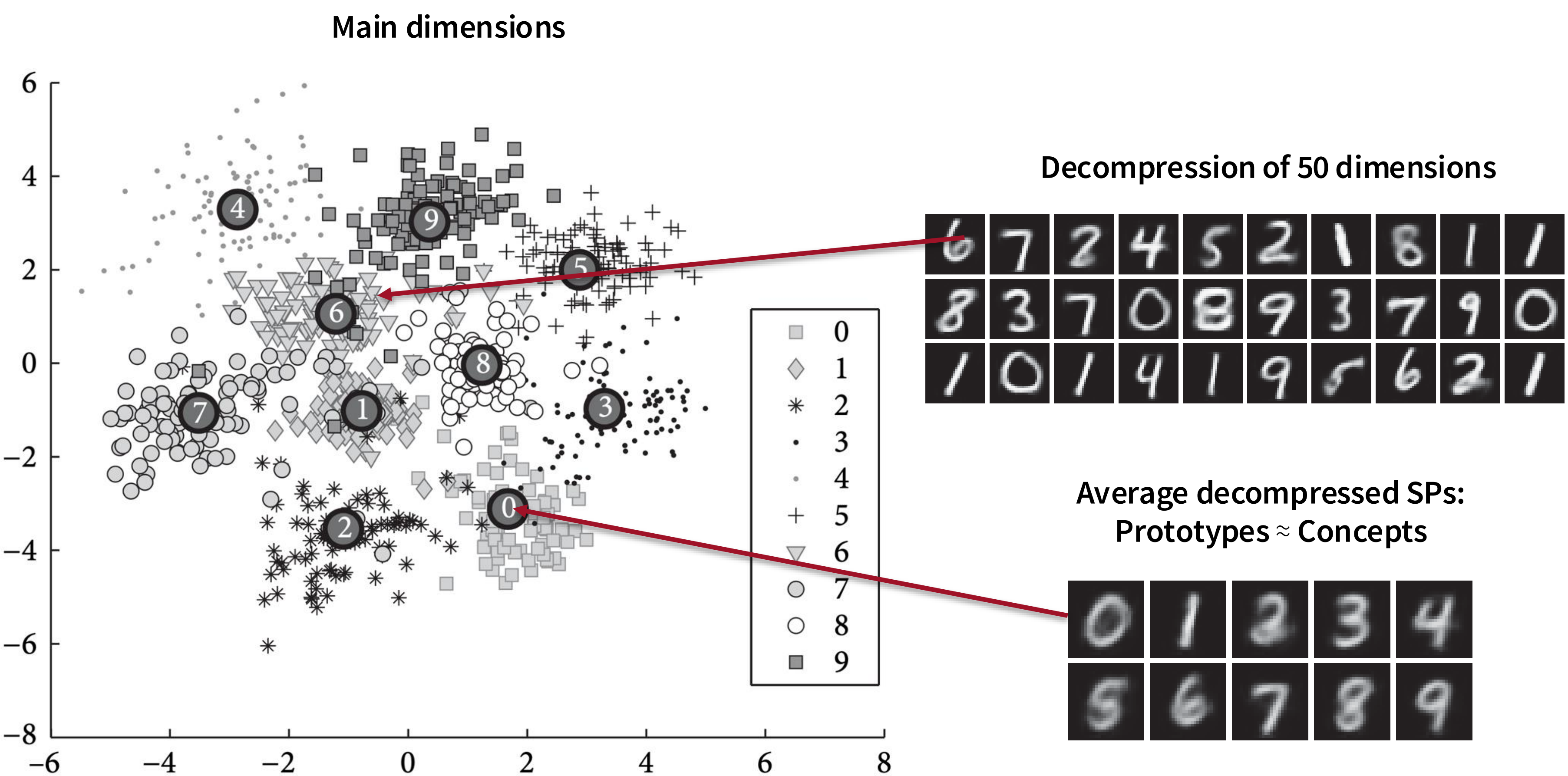
**Decompression of 50 dimensions**



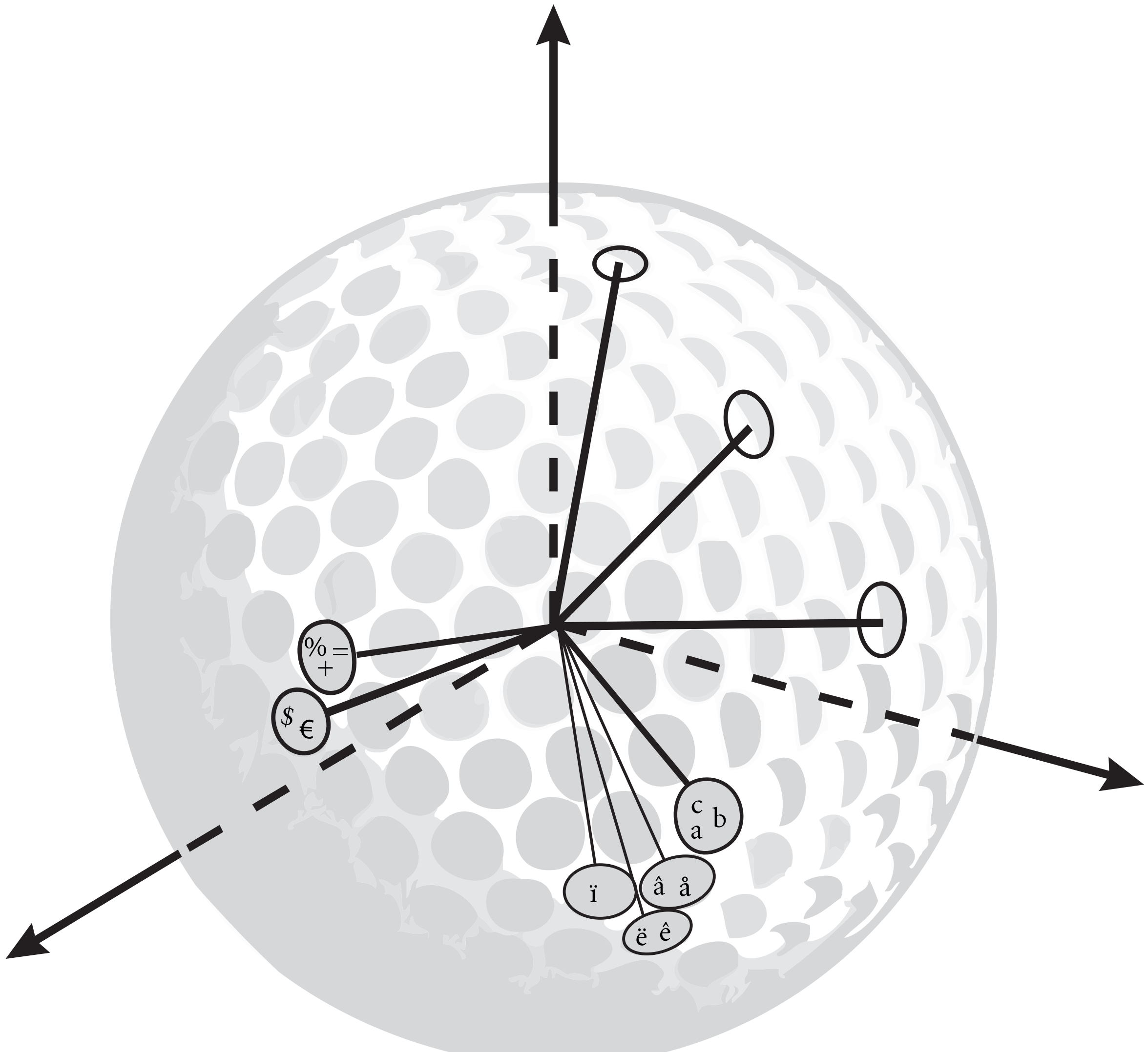
**Average decompressed SPs: Prototypes ≈ Concepts**



# Semantic Pointers applied to Numbers



# Semantic Pointer Similarity



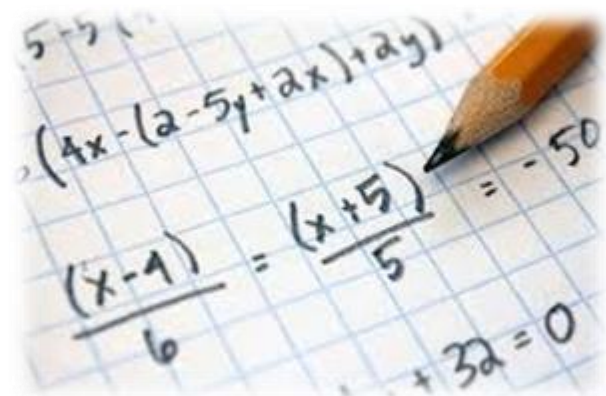
# Summary

- Symbols are represented by high-dimensional vectors called Semantic Pointers
- Semantic Pointers ultimately point at concepts in the world => they are grounded

# Today

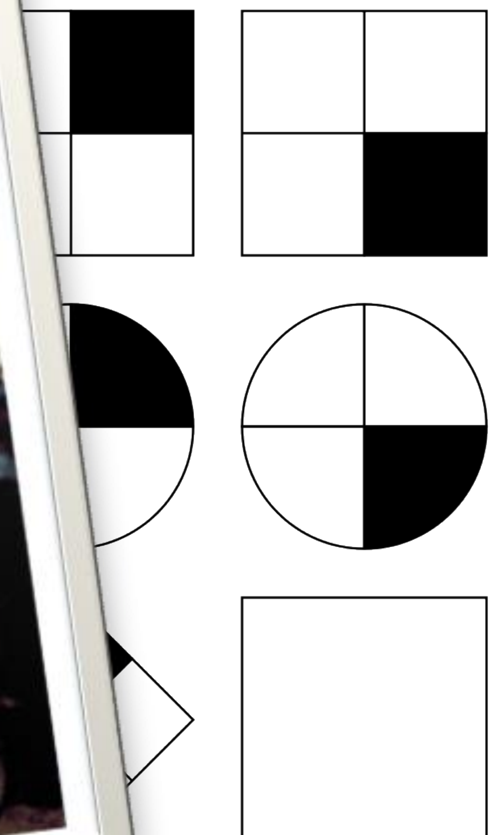
- Neural Networks II: from spikes to symbols
  - Spikes & Symbols
  - Symbol manipulation
  - Cognitive Control
  - Learning in Nengo

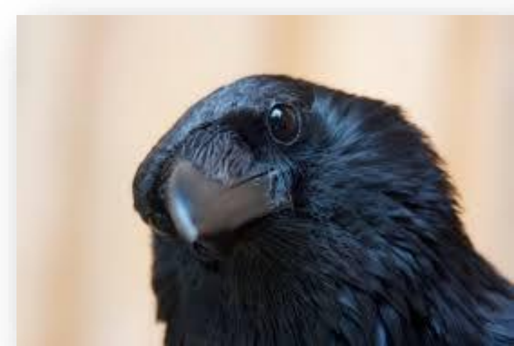
- Zbrodoff in Nengo



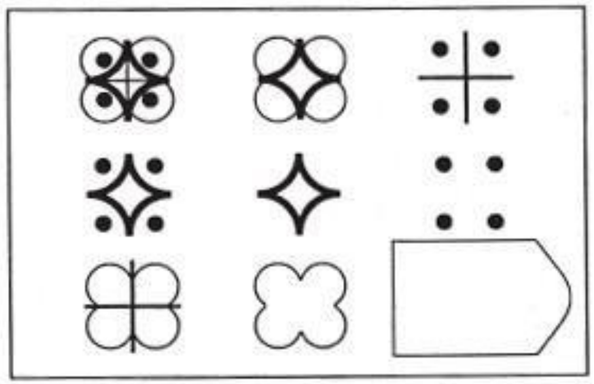
# Raven’s Progressive Matrices

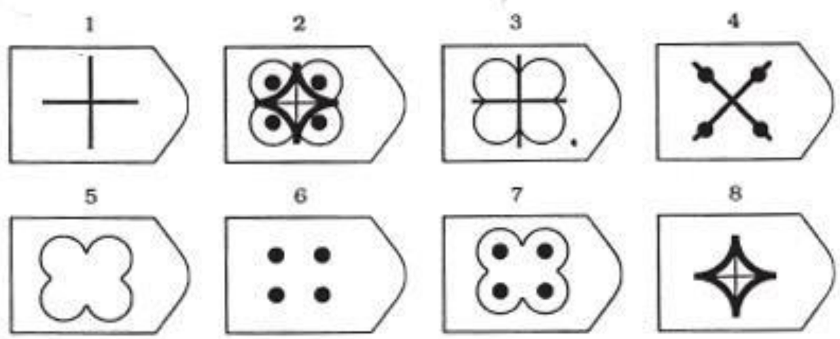
John C. Raven (1902-1970)



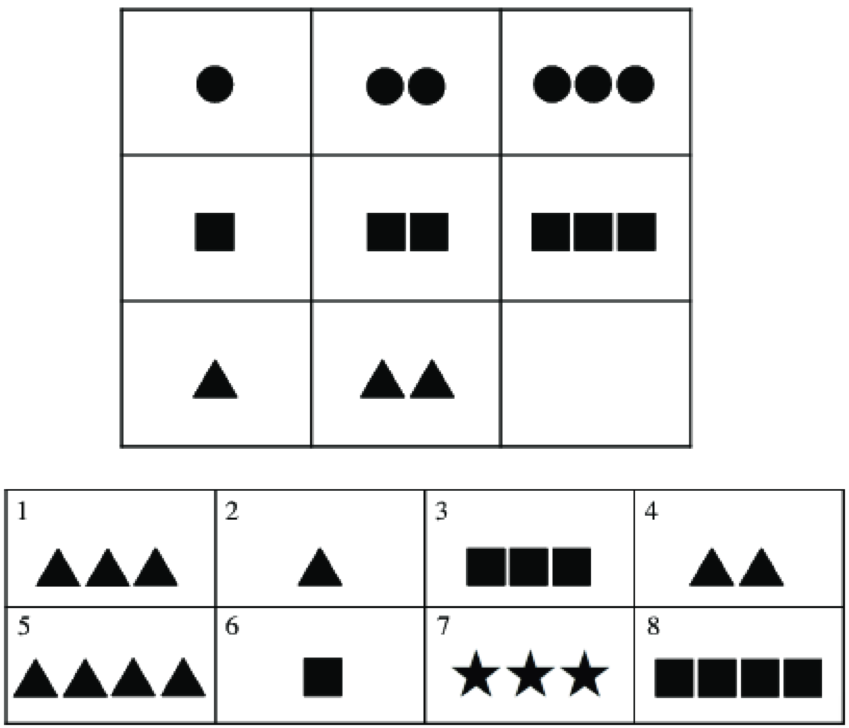


# Raven’s Progressive Matrices

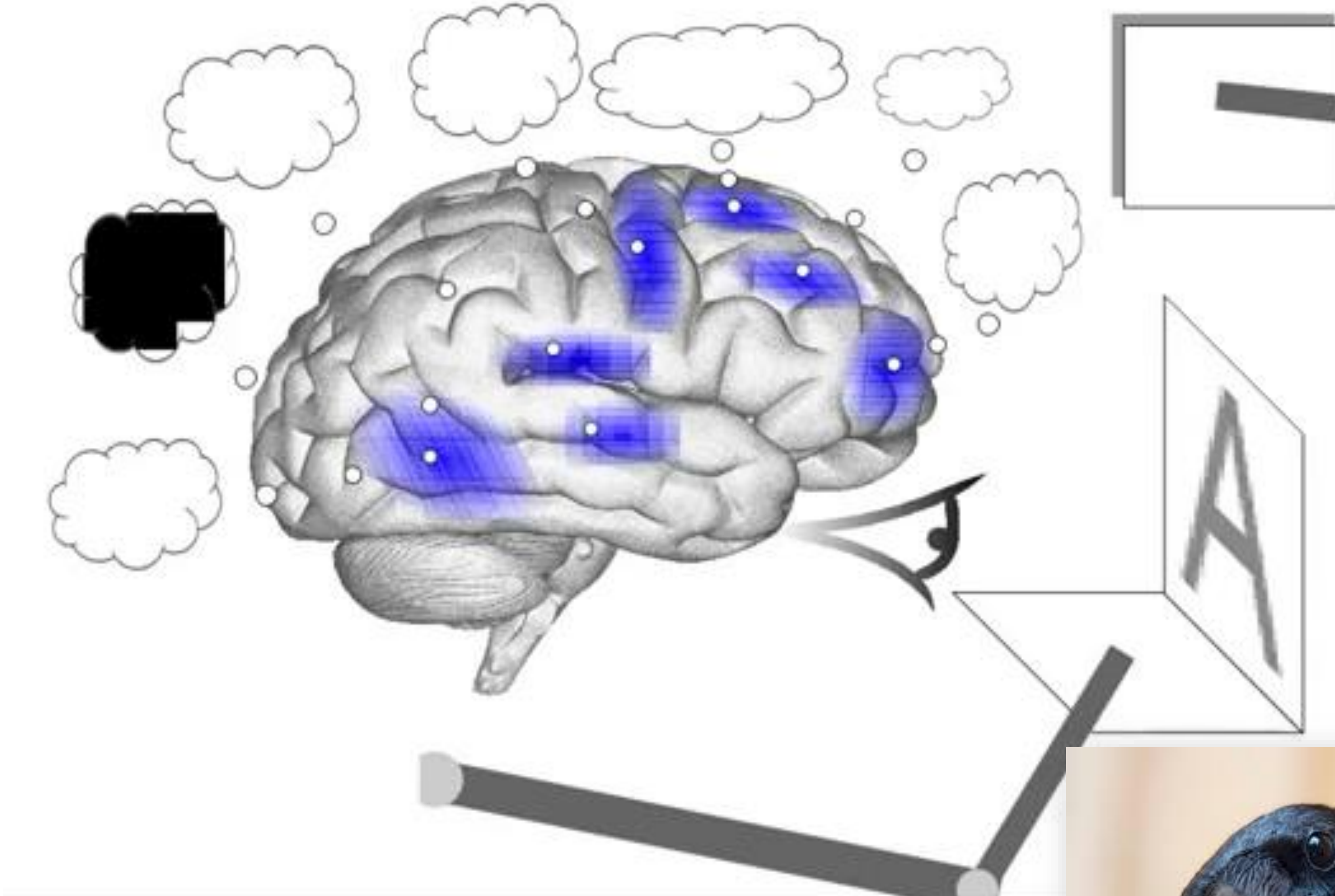


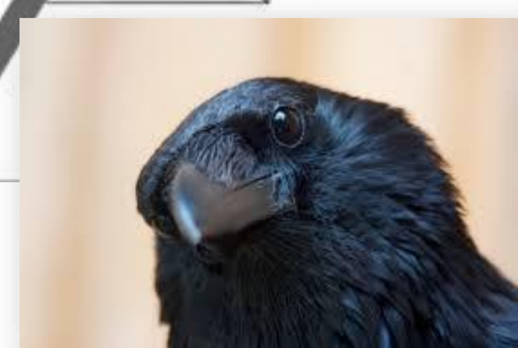



# Raven’s Progressive Matrices




# Raven’s Progressive Matrices



# Symbol mani

- Combining symbols:
  RED SQUARE and
- Role tagging (b
  DOG CHASE
  vs
  CAT CHASE

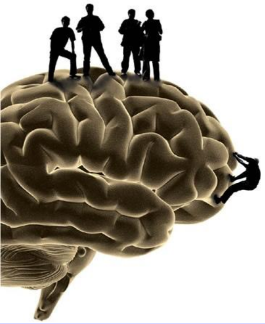
# Binding in Nengo

- Each symbol is represented by a random vector
- Binding through circular convolution:
  - red circle = RED $\circledast$ CIRCLE = new vector
  - Compare to: $2 \times 4 = 8$
- Unbinding through inverse:
  - What is red?
    - What was 2 multiplied with?
    - $(2 \times 4) / 2 = (2 \times 4) \times 2^{-1} = 4$
  - (RED $\circledast$ CIRCLE) $\circledast$ RED$^{-1}$ = CIRCLE
    - Deconvolution through convolution with inverse

# Composition in Nengo

- Red circle and blue square =
  RED ⊛ CIRCLE + BLUE ⊛ SQUARE
  - Why not RED + CIRCLE + BLUE + SQUARE?
- Dog chases cat =
  SUBJECT ⊛ DOG + VERB ⊛ CHASES + OBJECT ⊛ CAT

- We can unbind directly from
  - What is the object?
  - (SUBJECT ⊛ DOG + VERB ⊛ CHASE
    $\text{OBJECT}^{-1}$ = CAT + noise



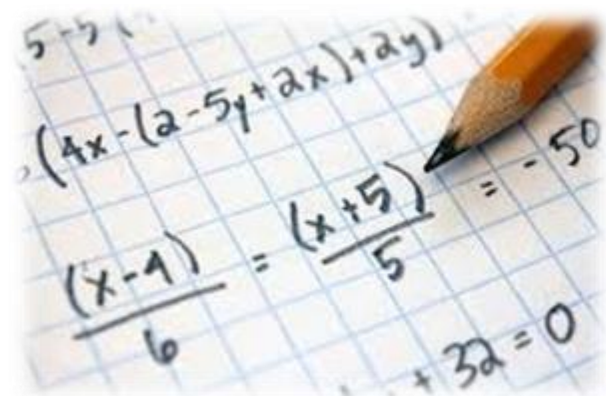
# Pattern Completion

| | | |
|---|---|---|
| 1 | 11 | 111 |
| 4 | 44 | 444 |
| 5 | 55 | ? |

# Today

- Neural Networks II: from spikes to symbols
  - Spikes & Symbols
  - Symbol manipulation
  - Cognitive Control
  - Learning in Nengo

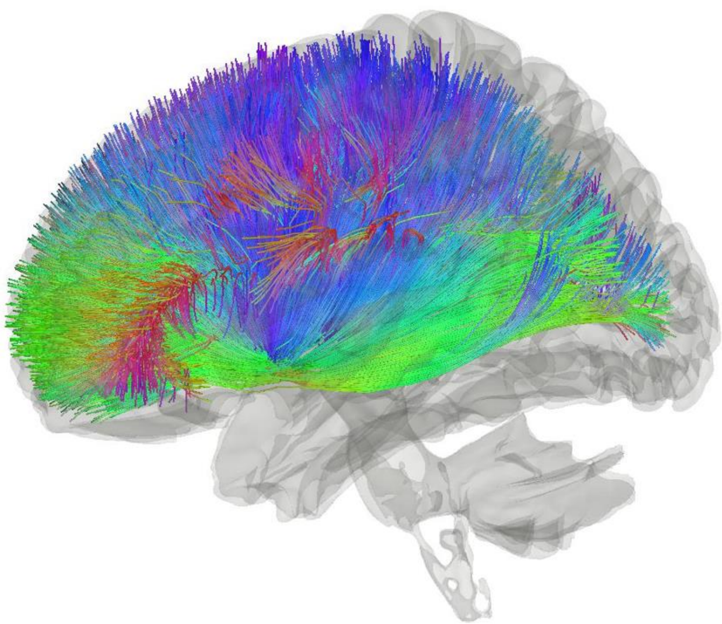
- Zbrodoff in Nengo

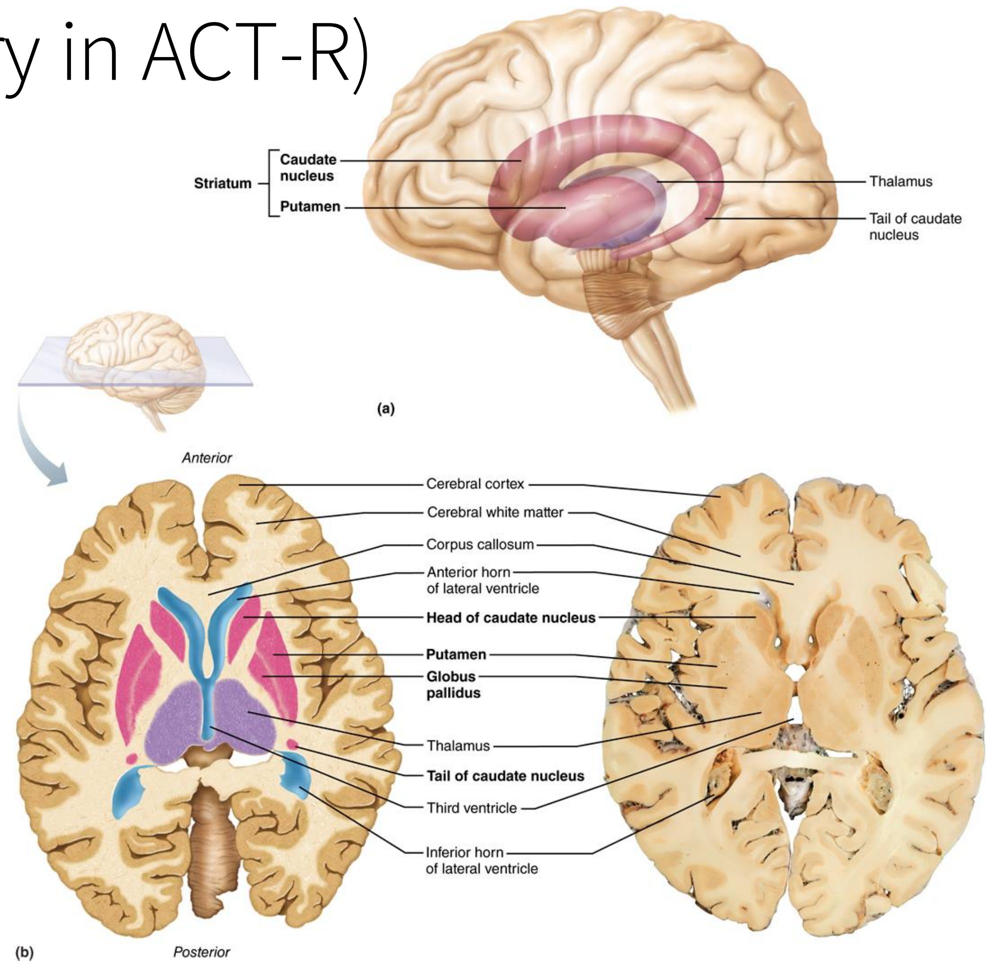


# Basal Ganglia

- Action-selection mechanism
  (~ procedural memory in ACT-R)

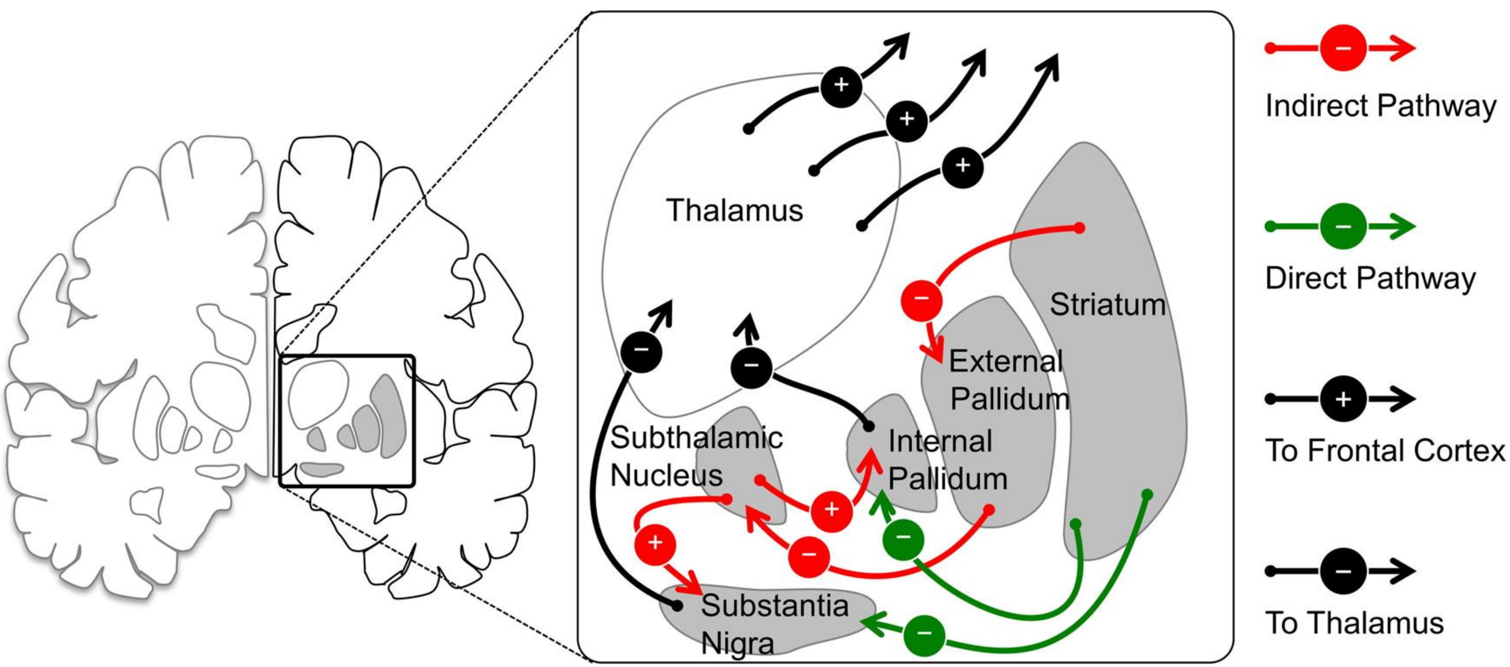
**Connectivity to Striatum**





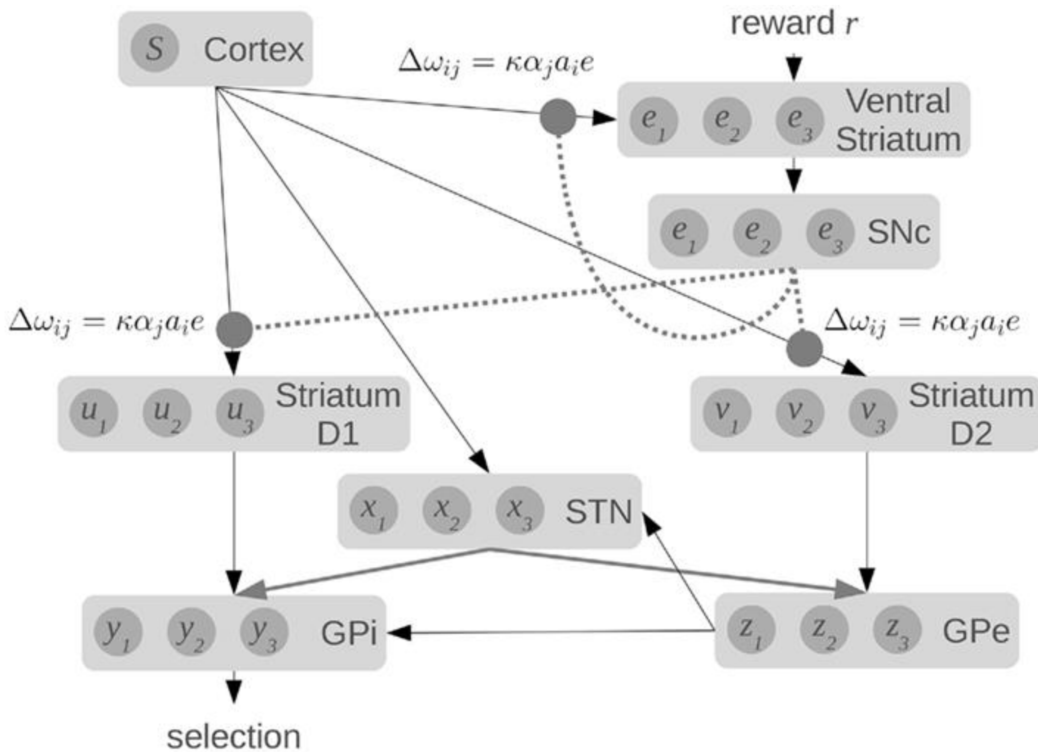
# Basal Ganglia

- Action-selection mechanism (~ procedural memory in ACT-R)



# Basal Ganglia

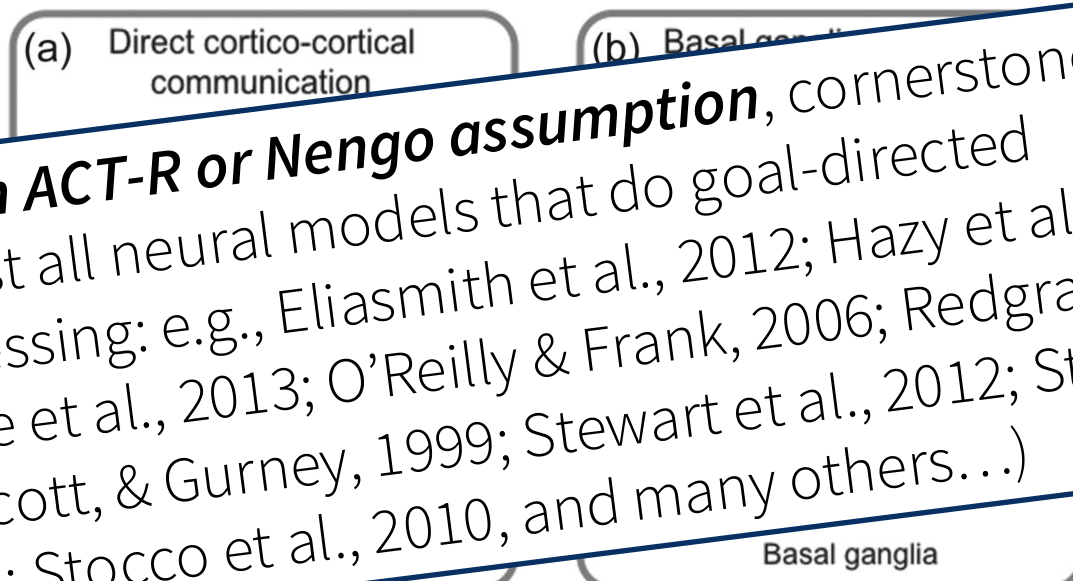
- Action-selection mechanism (~ procedural memory in ACT-R)




Stewart, Bekolay, & Eliasmith, 2012

# Basal Ganglia

- Action-selection mechanism
  (~ procedural memory in ACT-R)




***Not an ACT-R or Nengo assumption***, cornerstone of almost all neural models that do goal-directed processing: e.g., Eliasmith et al., 2012; Hazy et al., 2007; Kriete et al., 2013; O'Reilly & Frank, 2006; Redgrave, Prescott, & Gurney, 1999; Stewart et al., 2012; Stocco, 2017; Stocco et al., 2010, and many others…)

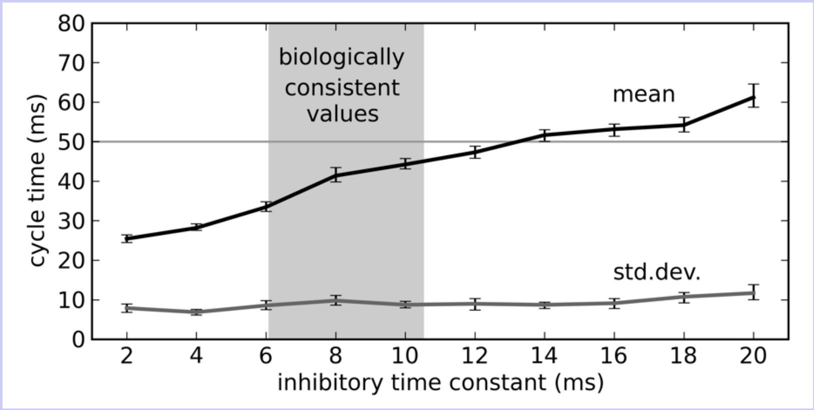
**Demo!**

# Connecting Psychology and Neuroscience

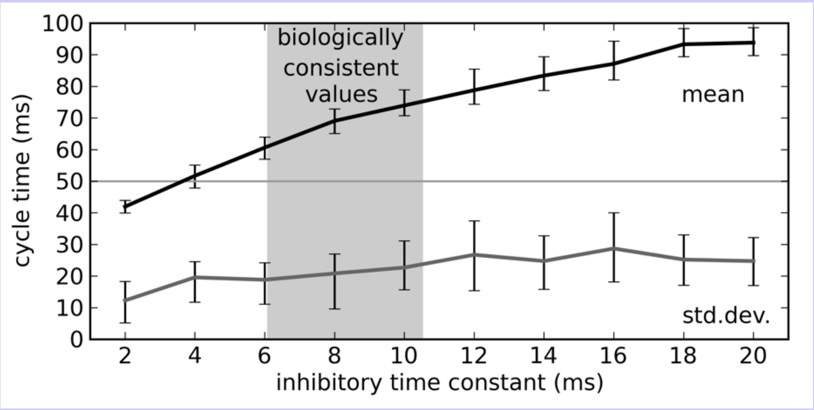
Terry Stewart

- Example
  - Standard model-fitting in psychology (e.g. ACT-R) says it takes 50 milliseconds to select and execute an action
  - Set neural parameters from neuroscience

Simple actions




Complex actions

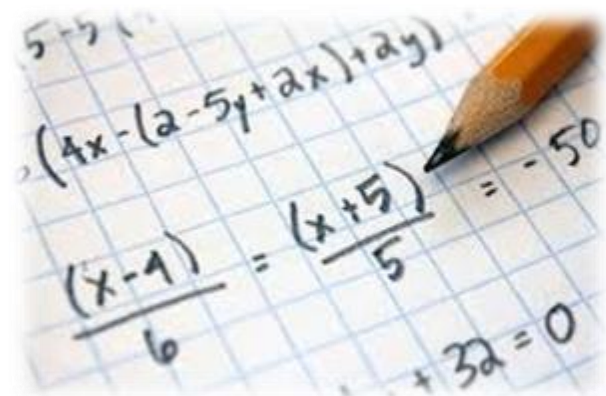



  - Average: 50 milliseconds (no fitting!)
  - 40 for some, 70 for others (new theory!)

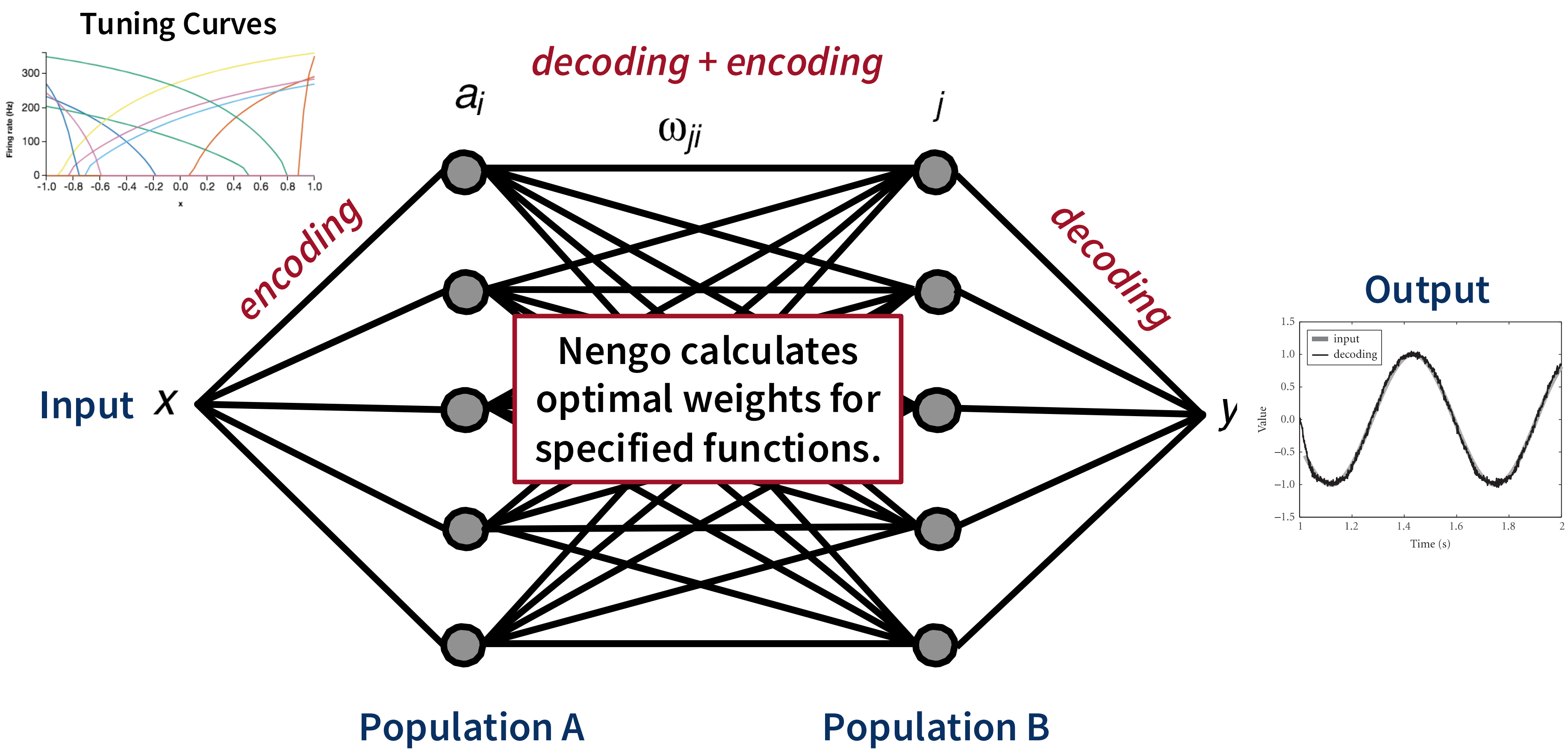
# Today

- Neural Networks II: from spikes to symbols
  - Spikes & Symbols
  - Symbol manipulation
  - Cognitive Control
  - Learning in Nengo

- Zbrodoff in Nengo



# Learning


Tuning Curves
300
200
100
0
Firing rate (Hz)
-1.0 -0.8 -0.6 -0.4 -0.2 0.0 0.2 0.4 0.6 0.8 1.0
x
decoding + encoding
$a_i$
$\omega_{ji}$
$j$
encoding
decoding
Input $x$
$y$
Nengo calculates optimal weights for specified functions.
Output
input
decoding
Value
Time (s)
Population A
Population B

# Learning

- Learning in neurons: changing connection weights
- Nengo advantage over standard NNs:
  it calculates optimal encoders and decoders
  (= connection weights)

- So why do we need learning?
  - Explain how we got the connection weights in the first place (i.e. no intelligent design)
  - Explain how online learning works
  - Sometimes it is difficult to pre-specify weights (especially in basal ganglia), or they might not be optimal

# Learning in ACT-R

- Instance-based learning
  - Repetition yields stronger memories
  - Nengo:
    - Hebbian learning (unsupervised)
    - PES (supervised)

- Utility learning
  - Reinforcement learning: rewards result in higher utilities
  - Nengo: Q-learning

# Hebbian learning

- Hebbian learning:
  - Neurons that fire together, wire together
  - Unsupervised

- Nengo: BCM rule (Bienenstock, Cooper, & Munro, 1982)

$$Dw_{ij} = k a_i a_j (a_j - q)$$

**learning rate**

**activity neurons *i* and *j***

**modulation depending on activity *j***

# PES learning

- ~ Delta rule in classical NNs
  - Supervised learning
  - Requires knowing the error between what you represent and what you want to represent

- Nengo: PES rule (Prescribed Error Sensitivity)

$$\Delta\omega_{ij} = \kappa a_i \underbrace{(\alpha_j E \cdot e_j)}_{\textbf{error given properties of neuron } \boldsymbol{j}}$$

**learning rate**

**activity neuron *i***

**Demo!**

# Learning in ACT-R

- Instance-based learning
  - Repetition yields stronger memories
  - Nengo:
    - Hebbian learning (unsupervised)
    - PES (supervised)

- Utility learning
  - Reinforcement learning: rewards result in higher utilities
  - Nengo: Q-learning

# Utility learning

- Learning how to act (vs new knowledge)
- Reinforcement learning (Q-learning)
  - requires a reward signal

- Basal Ganglia
- Dopamine indicates (unexpected) reward

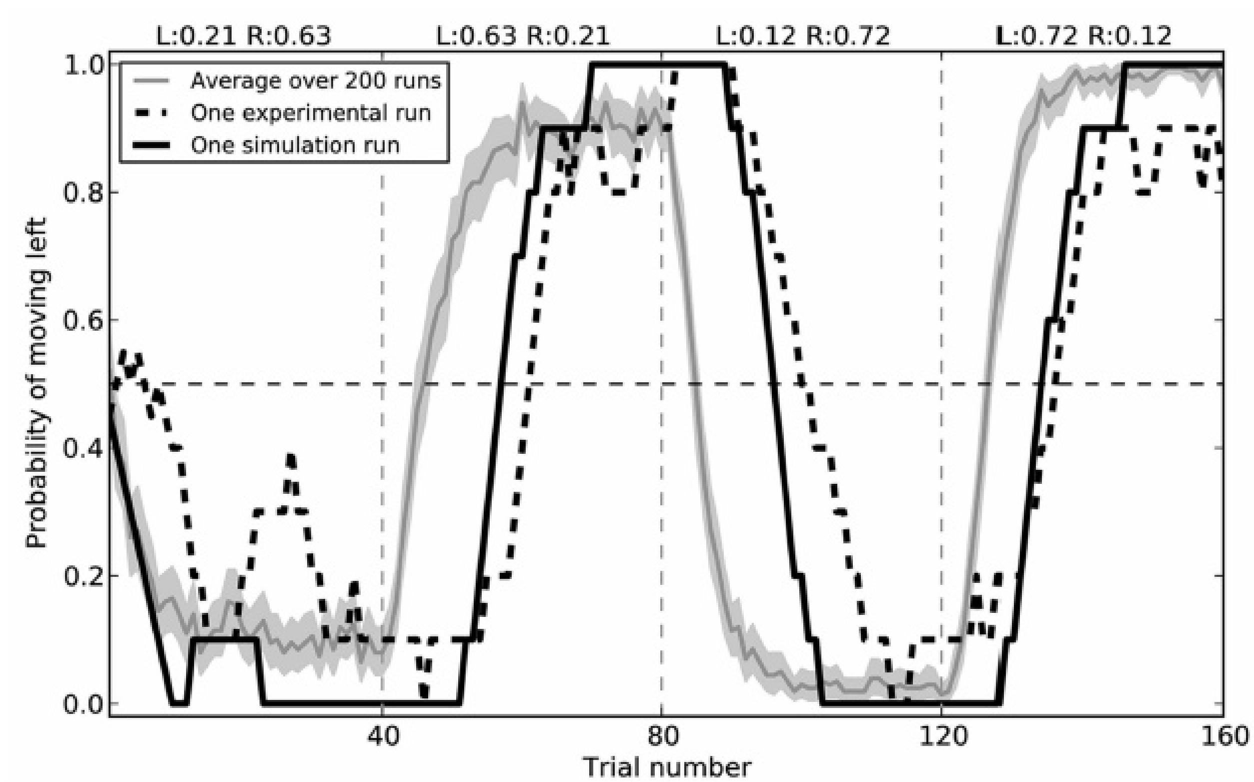
- Example: probability learning in rats

# Choose a button

A

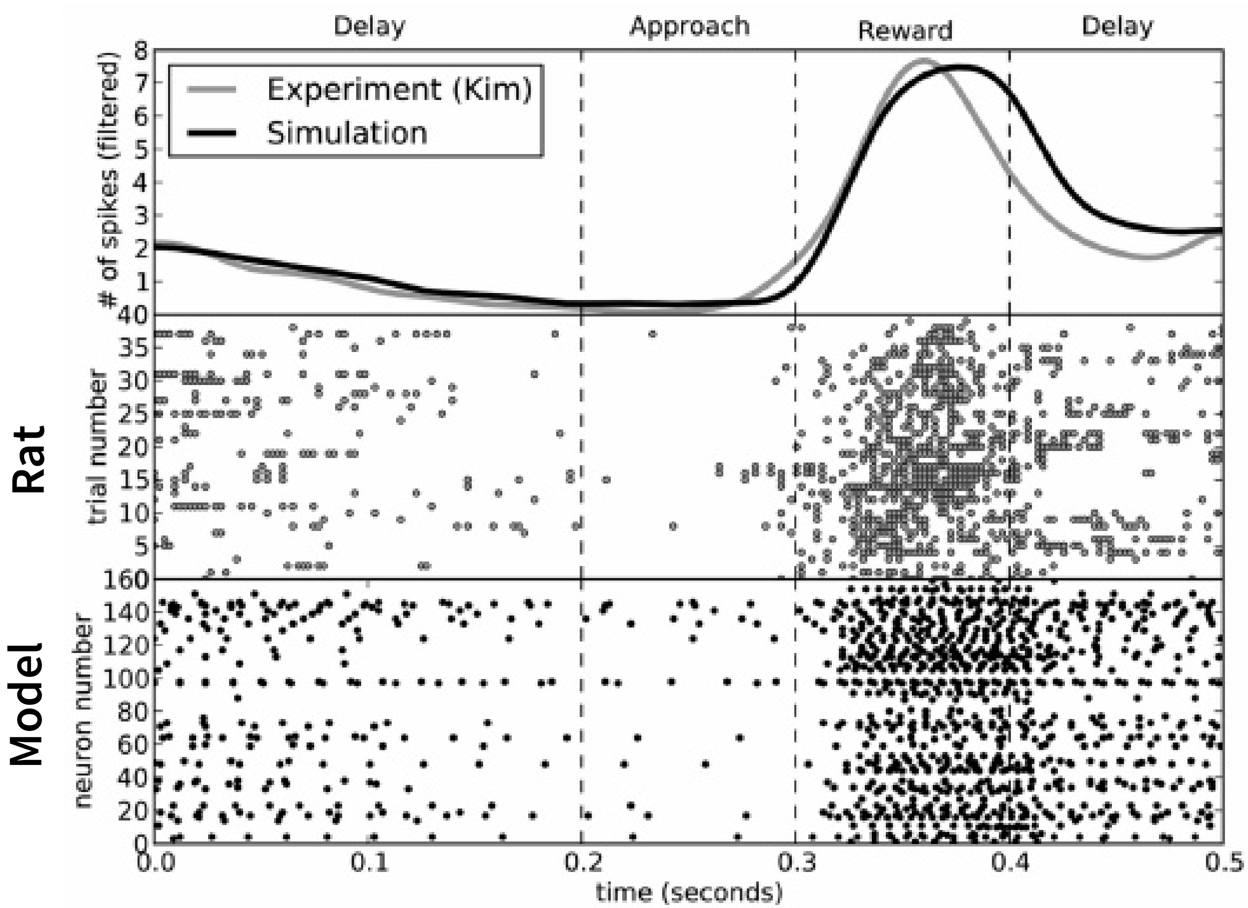
B

# Probability learning




Demo!

# Utility learning: Firing Rates in Striatum



# What about memory?

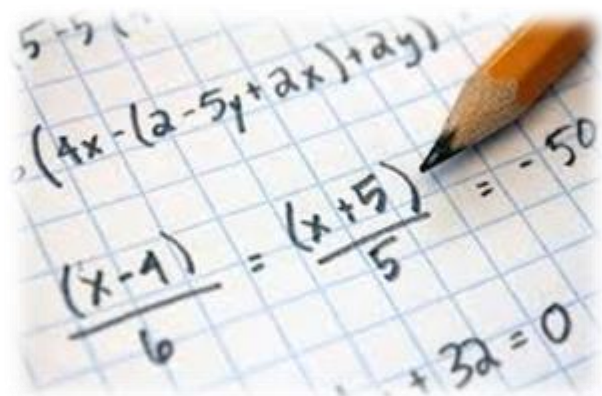
- Build-in memory system (without learning)
  - Tries to match input
    - cf. 
```
+retrieval>
     isa          addition-fact
     addend1      2
     addend2      4
```
  - Could be learned through PES learning
    => associative recognition example

- However, some things are missing:
  - No activation levels
  - No different retrieval times
  - No online learning:
    no recency or frequency effects

# Today

- Neural Networks II: from spikes to symbols
  - Spikes & Symbols
  - Symbol manipulation
  - Cognitive Control
  - Learning in Nengo

- Zbrodoff in Nengo



# Nengo model of mathematical development

- Task: learning addition (4 + 5 = 9)
- Two learning phases:
  - Counting
  - Retrieval

≈ Zbrodoff model in ACT-R



TOPICS
TOPICS IN COGNITIVE SCIENCE

Topics in Cognitive Science (2016) 1–15

ISSN:1756-8757 print / 1756-8765 online
DOI: 10.1111/tops.12242

This article is part of the topic "Best of Papers From the Cognitive Science Societ
Annual Conference," Wayne D. Gray (Topic Editor). For a full listing of topic papers, se
http://onlinelibrary.wiley.com/journal/10.1111/(ISSN)1756-8765/earlyview

Improving With Practice: A Neural Model of Mathematical Development

Sean Aubin, Aaron R. Voelker, Chris Eliasmith
*Centre for Theoretical Neuroscience, University of Waterloo*

Received 7 October 2016; accepted 19 October 2016

**Abstract**

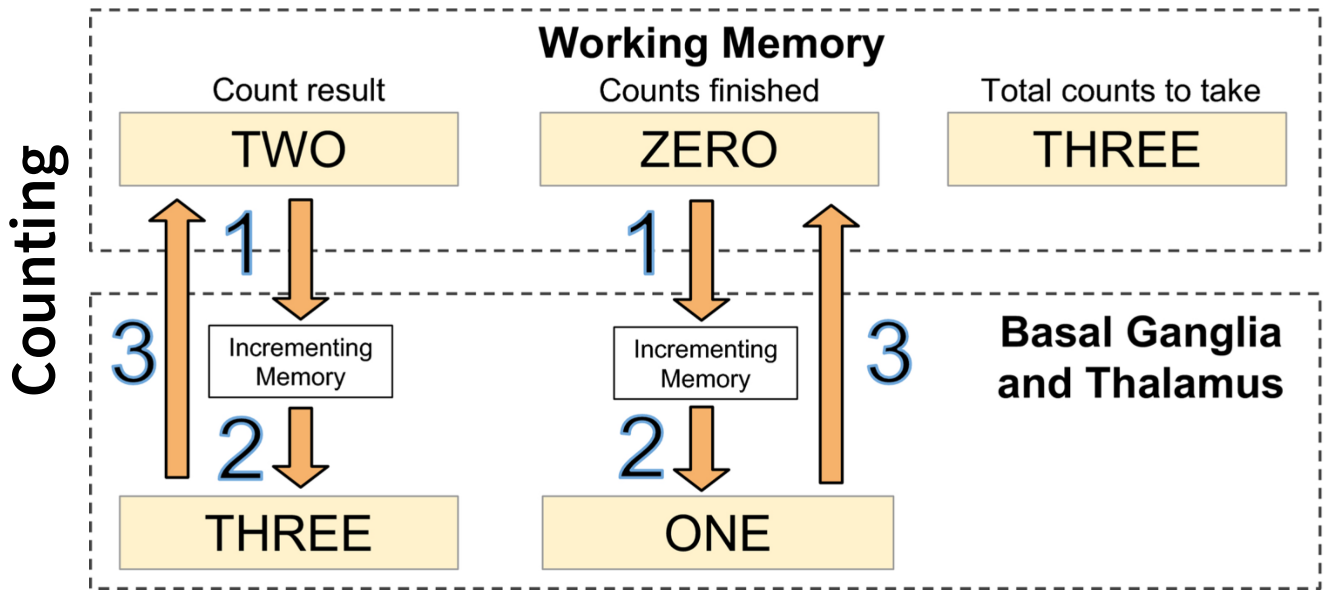
The ability to improve in speed and accuracy as a result of repeating some task is an im
tant hallmark of intelligent biological systems. Although gradual behavioral improvements f
practice have been modeled in spiking neural networks, few such models have attempted
explain cognitive development of a task as complex as addition. In this work, we model the p
gression from a counting-based strategy for addition to a recall-based strategy. The model c
sists of two networks working in parallel: a slower basal ganglia loop and a faster
network. The slow network methodically computes the cou
responding to the addition of

# The model

- Two networks:
  - Slow counting
  - Fast retrieval

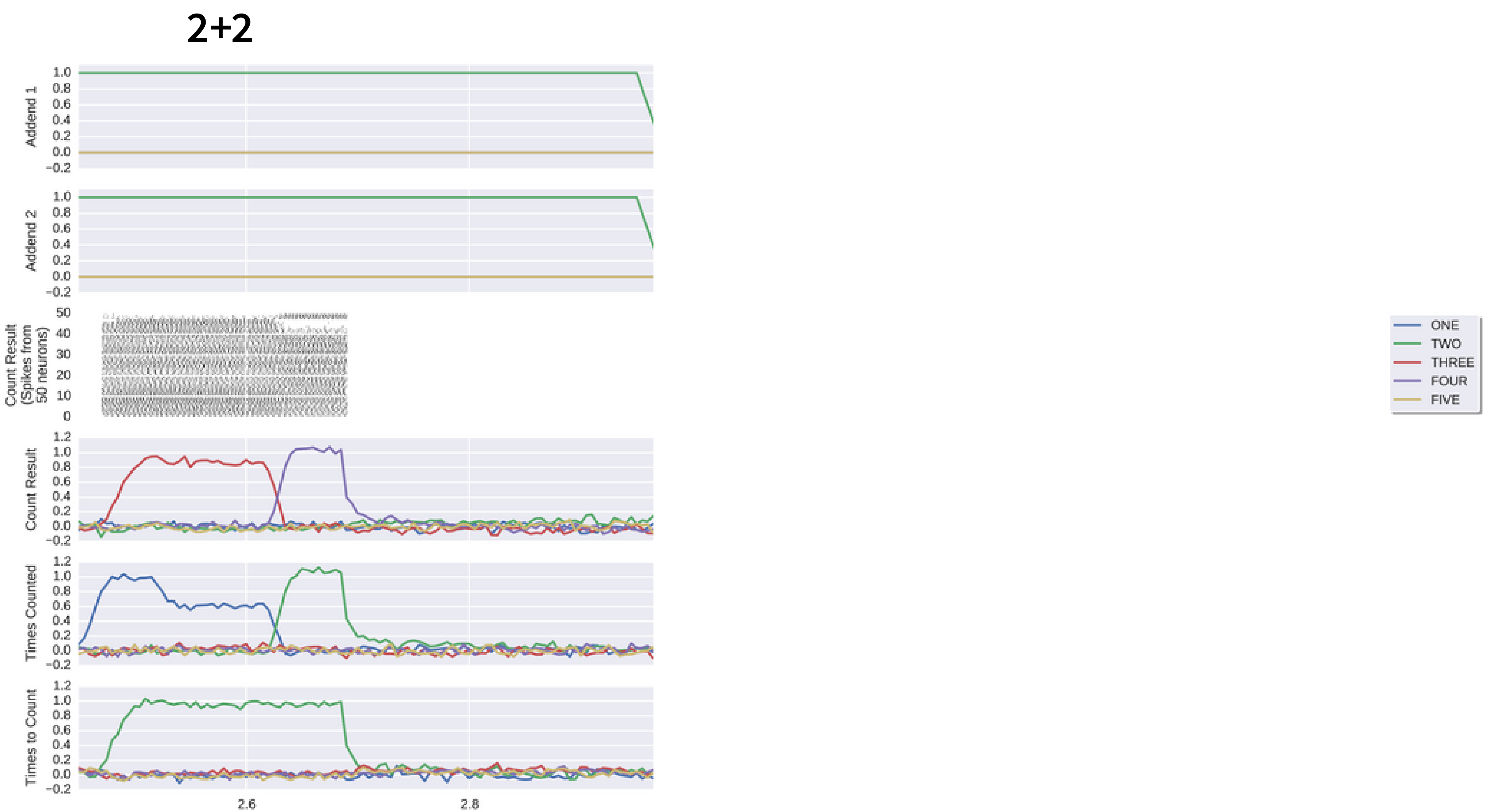
Aubin, Voelker, & Eliasmith, 2016

# The model: Counting

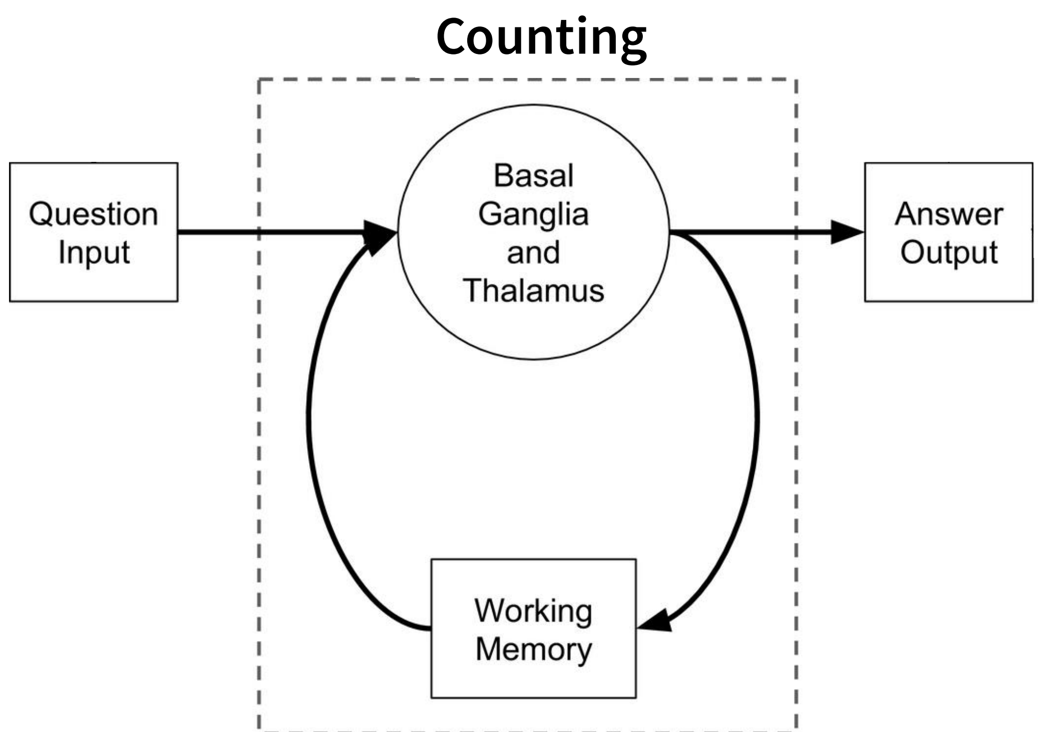



Aubin, Voelker, & Eliasmith, 2016

# Results: Counting



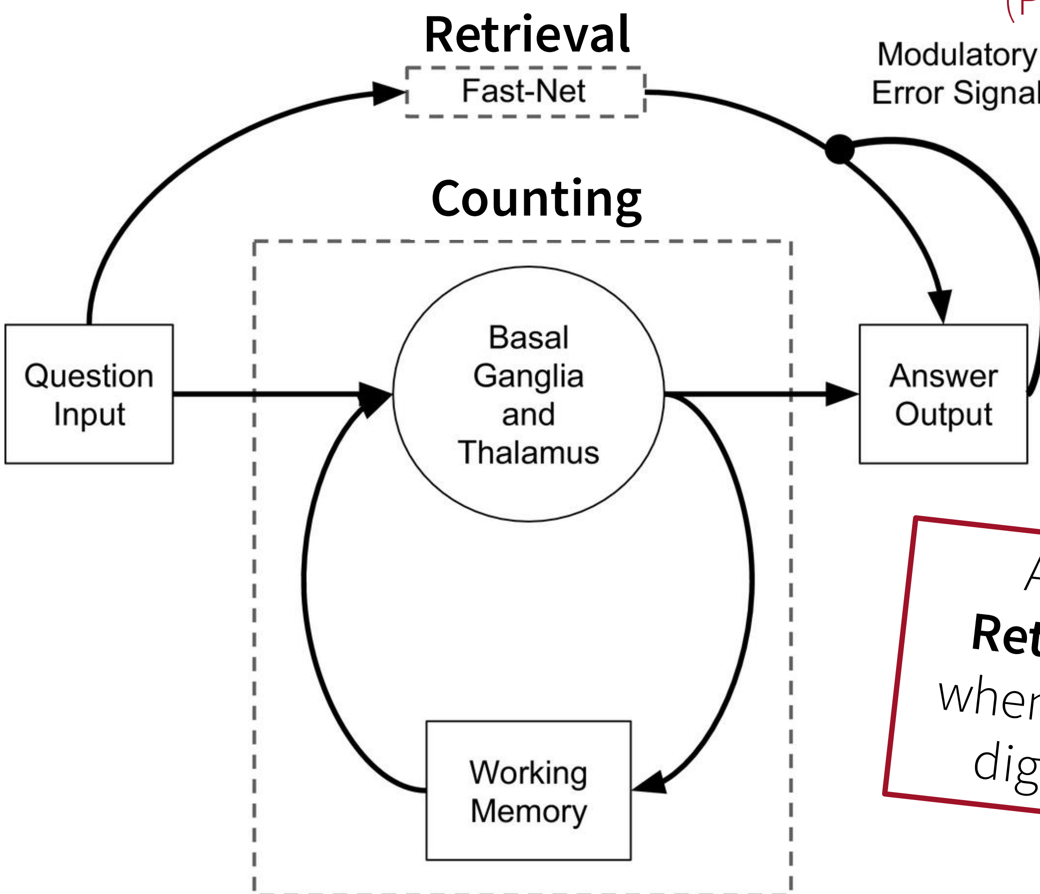
# The model




Aubin, Voelker, & Eliasmith, 2016

# The model

= supervised learning
(PES rule)



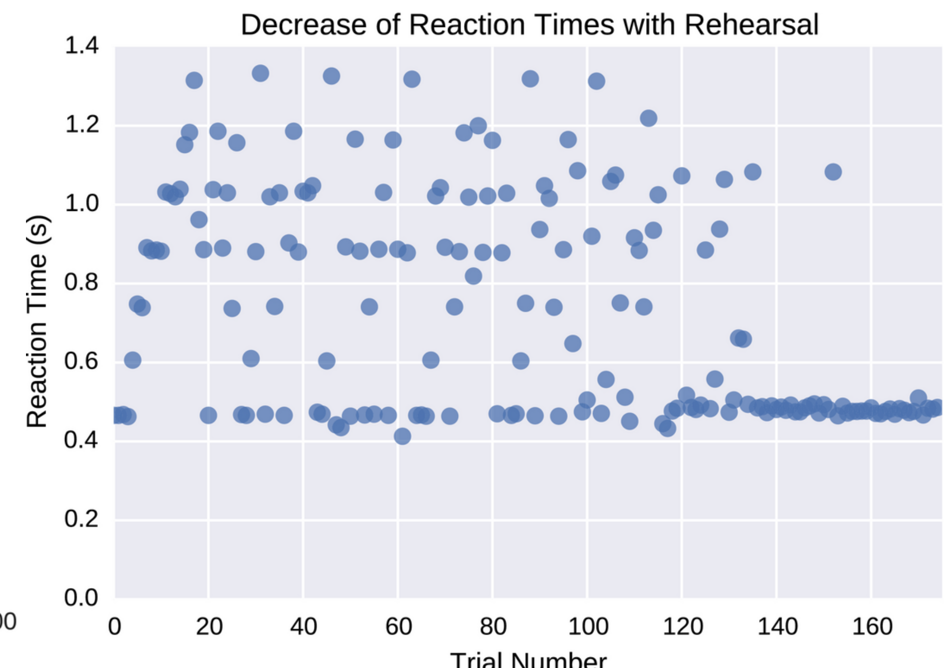

Answer from **Retrieval** is used when its output is a digit-like vector

Aubin, Voelker, & Eliasmith, 2016

# Results: Retrieval

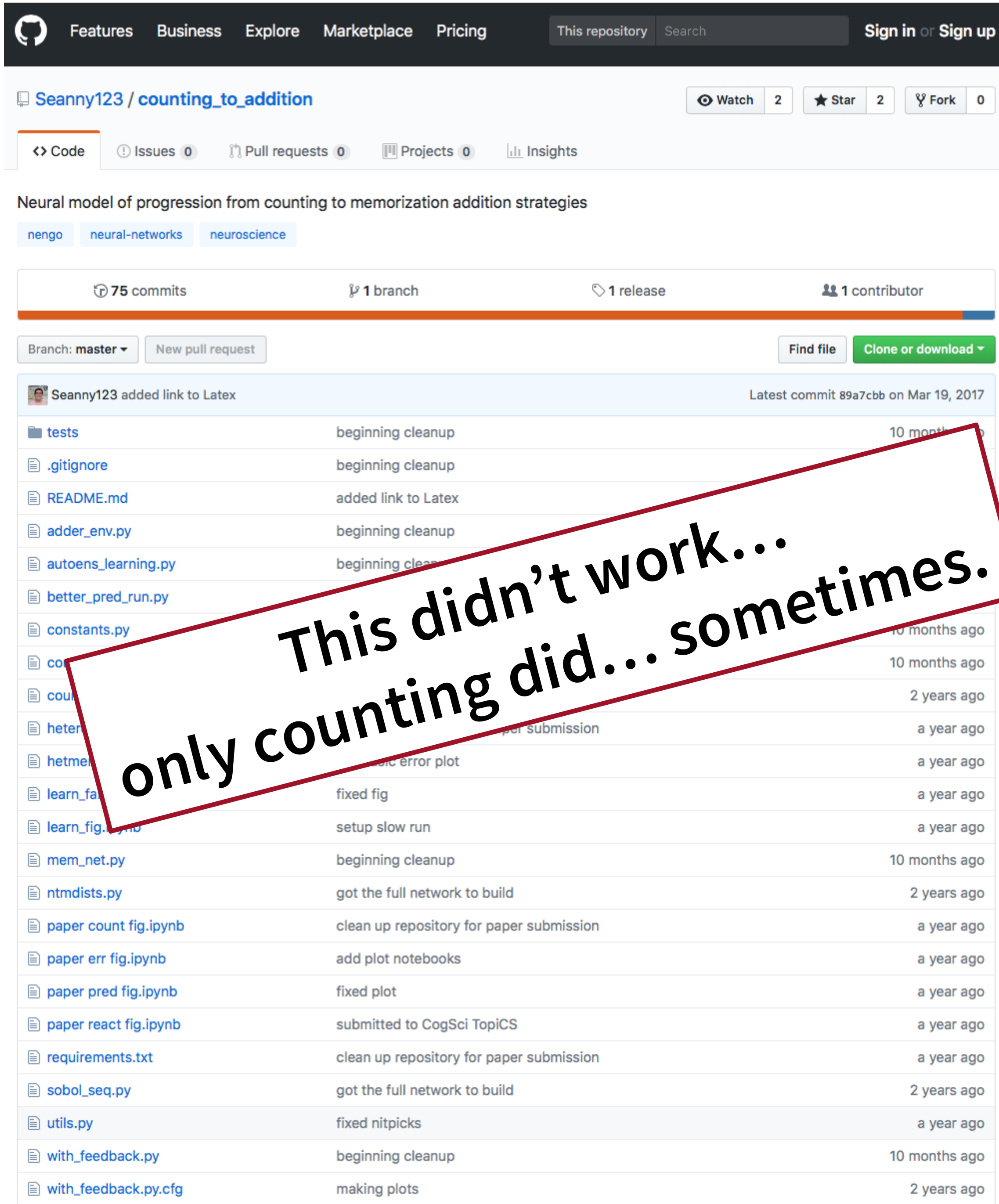
**Error in memory**


**Decrease in RT**




**What is missing here?**

# The actual model




**This didn't work…**
**only counting did… sometimes.**

# Addition in ACT-R

## Declarative Memory

```
(clear-all)

(define-model addition

(sgp :esc t :lf .05)

(chunk-type count-order first second)
(chunk-type add arg1 arg2 sum count)

(add-dm
   (a ISA count-order first 0 second 1)
   (b ISA count-order first 1 second 2)
   (c ISA count-order first 2 second 3)
   (d ISA count-order first 3 second 4)
   (e ISA count-order first 4 second 5)
   (f ISA count-order first 5 second 6)
   (g ISA count-order first 6 second 7)
   (h ISA count-order first 7 second 8)
   (i ISA count-order first 8 second 9)
   (j ISA count-order first 9 second 10)
   (second-goal ISA add arg1 5 arg2 2))
```

## Procedural Memory

```
(P initialize-addition
   =goal>
      ISA         add
      arg1        =num1
      arg2        =num2
      sum         nil
  ==>
   =goal>
      ISA         add
      sum         =num1
      count       0
   +retrieval>
      ISA         count-order
      first       =num1
)

(P terminate-addition
   =goal>
      ISA         add
      count       =num
      arg2        =num
      sum         =answer
  ==>
   =goal>
      ISA         add
      count       nil
   !output!       =answer
)

(P increment-count
   =goal>
      ISA         add
      sum         =sum
      count       =count
   =retrieval>
      ISA         count-order
      first       =count
      second      =newcount
  ==>
   =goal>
      ISA         add
      count       =newcount
   +retrieval>
      ISA         count-order
      first       =sum
)

(P increment-sum
   =goal>
      ISA         add
      sum         =sum
      count       =count
    - arg2        =count
   =retrieval>
      ISA         count-order
      first       =sum
      second      =newsum
  ==>
   =goal>
      ISA         add
      sum         =newsum
   +retrieval>
      ISA         count-order
      first       =count
)

(goal-focus second-goal)
)
```

# Intermediate Conclusions

- Cognitive models based on spiking neurons:
  - Very complex
  - Hard to control
  - Hard to match behavior (at least it is not shown)
  - Explains how the brain might do cognition

- Cognitive models based on symbolic processing:
  - Less complex
  - Easy to control
  - Good at explaining RTs and accuracy
  - Can predict brain data, but does not explain

# Remainder

- No lecture this Thursday
- No lecture Monday Jan 6
- Final lecture on Thursday Jan 9:
  - Large-scale Nengo model
  - ACT-R vs. Nengo
  - Exam preps
  - Blackjack tournament results

# Happy holidays!