Learning in large-scale spiking neural networks

Trevor Bekolay

Center for Theoretical Neuroscience
University of Waterloo

Master's thesis presentation
August 17, 2011

Outline

Unsupervised learning

Supervised learning

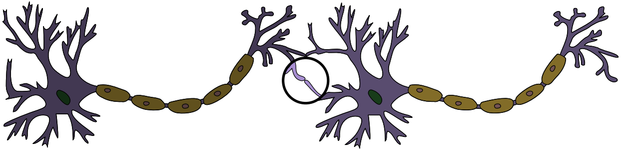
Reinforcement learning

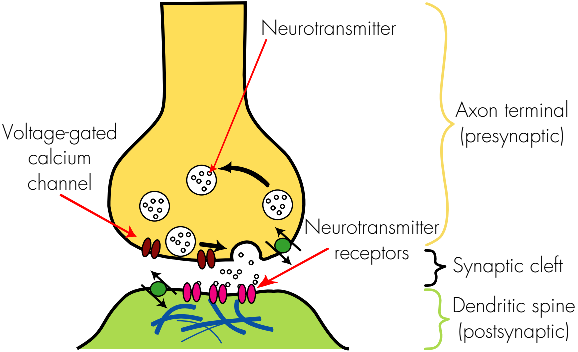
Unsupervised learning

The problem

How does the brain learn without feedback?

Neurons connect at synapses





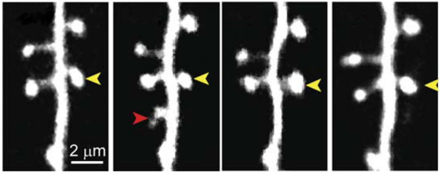
Synaptic strengths change

Definitions

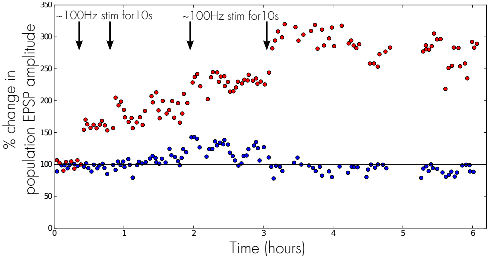
This is called synaptic plasticity.

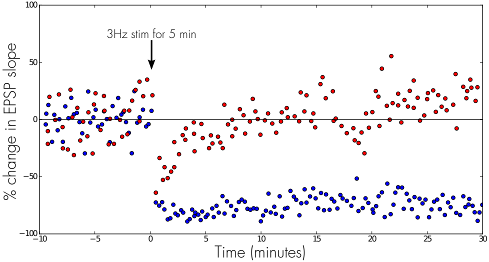
When a synapses gets stronger, it is potentiated.

When a synapse gets weaker, it is depressed.

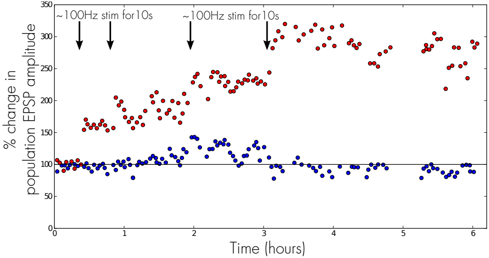


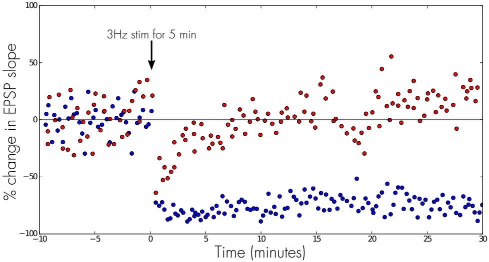
Modelling LTP/LTD





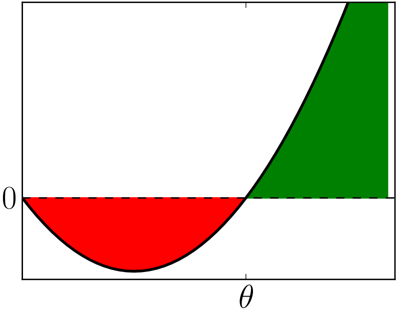
Modelling LTP/LTD



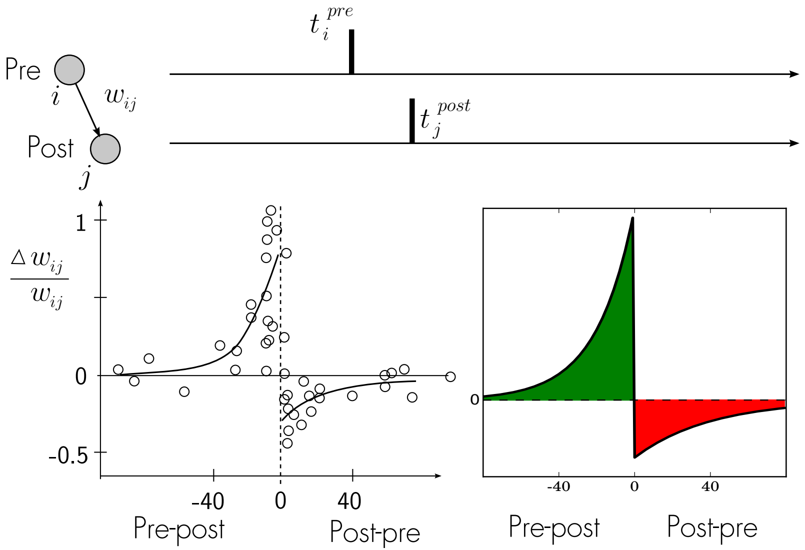


BCM rule

$$\Delta\omega_{ij} = \kappa a_i a_j (a_j - \theta)$$



Modelling LTP/LTD with timing



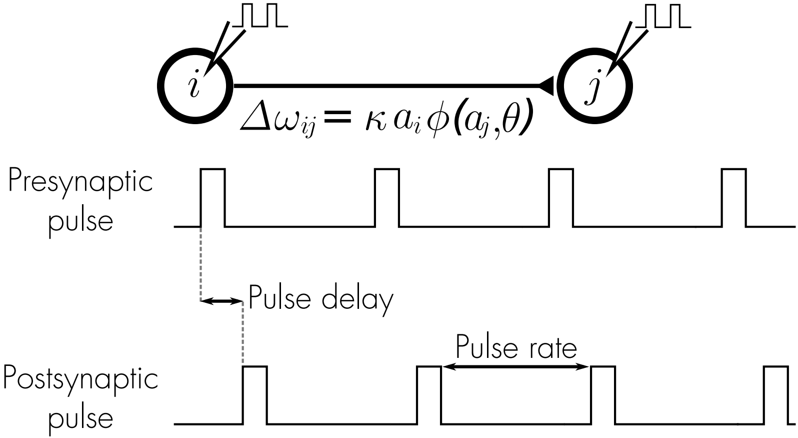
What’s the difference?

What's the difference?

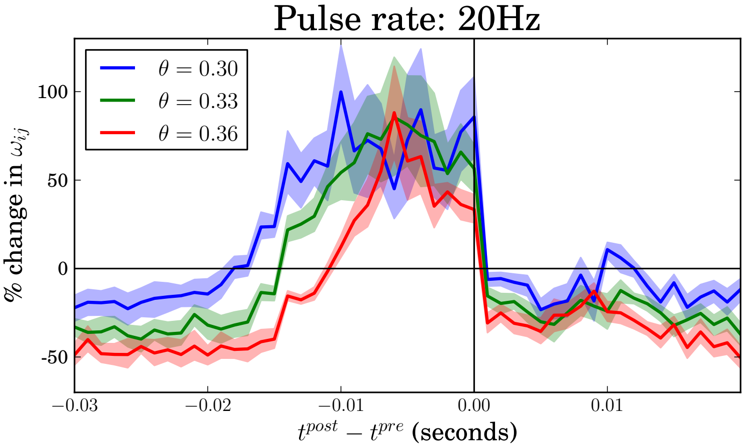
Theorem

If spike trains have Poisson statistics, STDP can be mapped onto a BCM rule (Izhikevich, 2003 and Pfister & Gerstner, 2006).

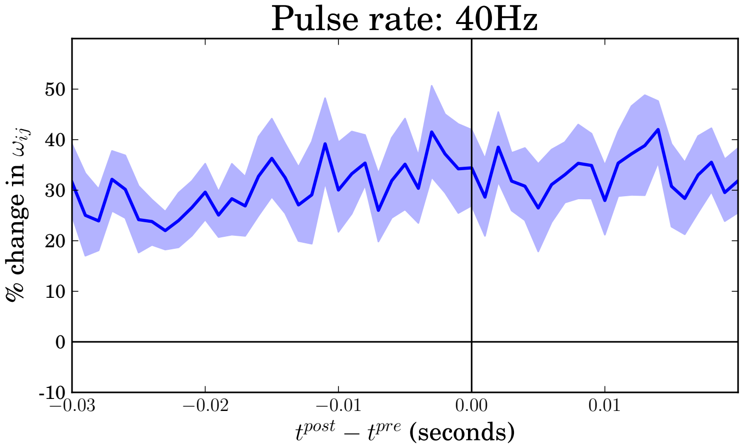
Test with a simulation



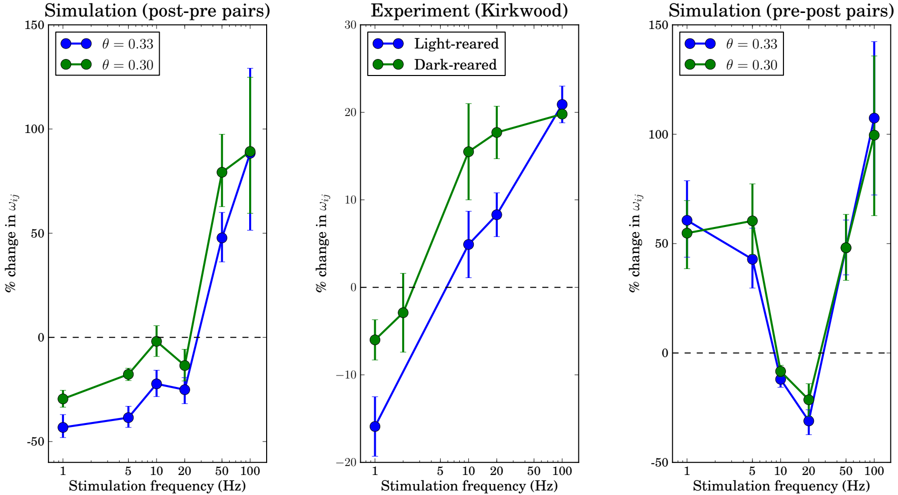
Recreated STDP curve



Exhibits frequency dependence



Frequency dependence details



Supervised learning

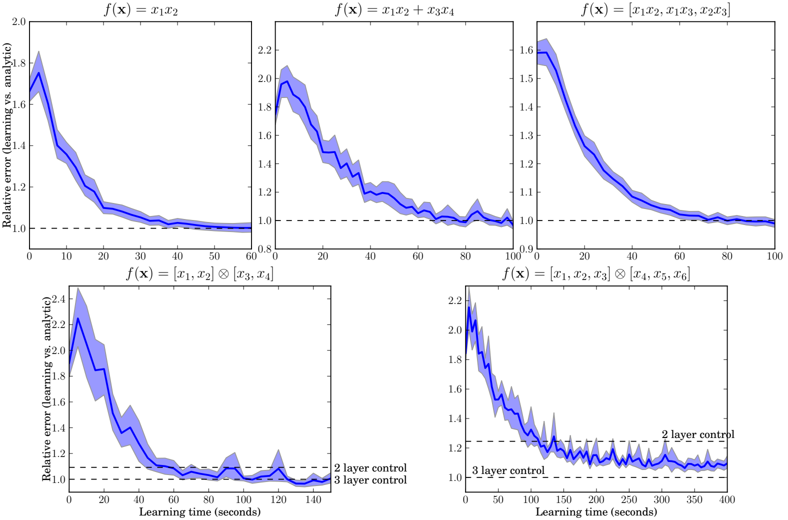
The problem

Given input $\mathbf{X}$ and output $\mathbf{Y}$, find $G(\mathbf{x}) = \mathbf{y}$

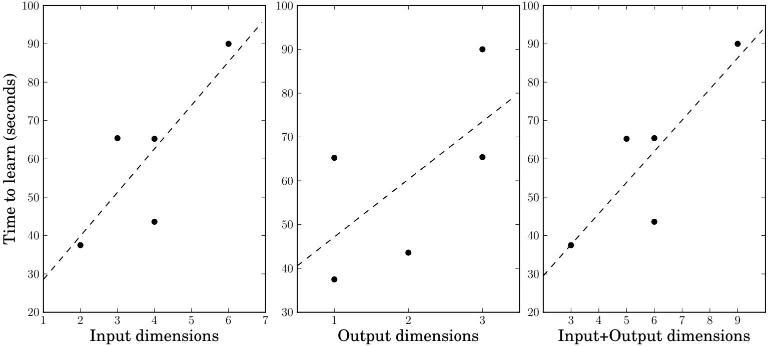
Learning nonlinear functions

Function	Input dimensions	Output dimensions	Neurons in learning network	Neurons in control network
$f(\mathbf{x}) = x_1 x_2$	2	1	420	420
$f(\mathbf{x}) = x_1 x_2 + x_3 x_4$	4	1	756	756
$f(\mathbf{x}) = [x_1 x_2, x_1 x_3, x_2 x_3]$	3	3	756	756
$f(\mathbf{x}) = [x_1, x_2] \otimes [x_3, x_4]$	4	2	700	704 (2-layer) 1500 (3-layer)
$f(\mathbf{x}) = [x_1, x_2, x_3] \otimes [x_4, x_5, x_6]$	6	3	800	804 (2-layer) 1700 (3-layer)

Simulation results



Simulation results



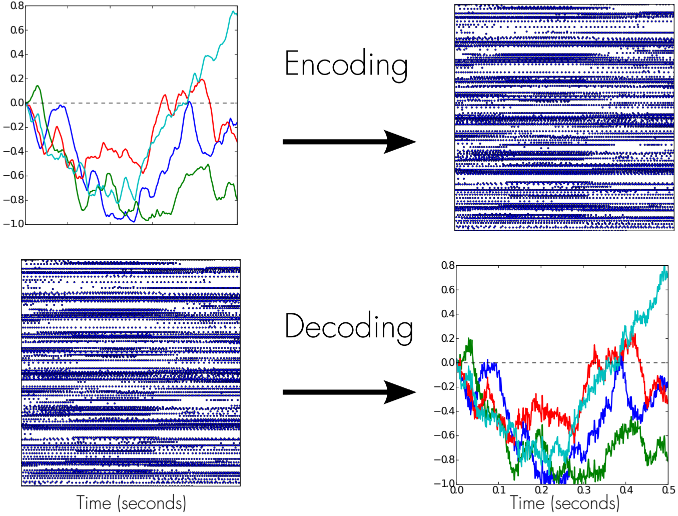
$\sim$ 2.25 hours to learn 500 $\rightarrow$ 500 D function

Large-scale models

Neural Engineering Framework

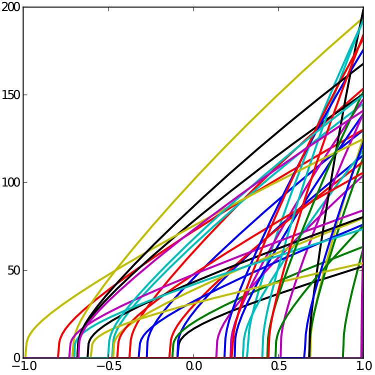
1. Representation (encoding and decoding)
2. Transformation

Representation

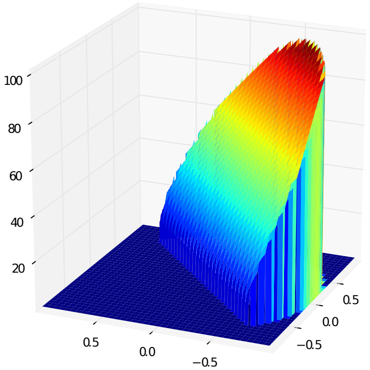


Encoding vectors represent neuron sensitivity

LIF tuning curves



Projected in 2-D space



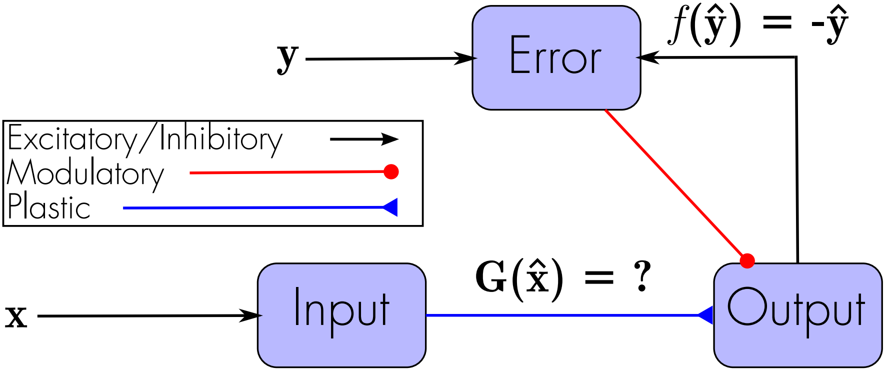
Plasticity in the NEF

$$\Delta\omega_{ij} = \kappa a_i \alpha_j \mathbf{e}_j \cdot \mathbf{E}$$

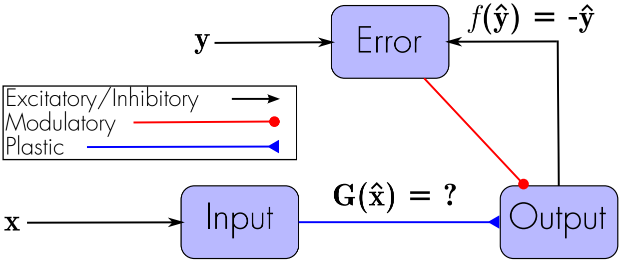
where

- $\mathbf{e}_j$ is the encoding vector and
- $\mathbf{E}$ is the error to be minimized.

Calculating $\mathbb{E}$ online



Our solution

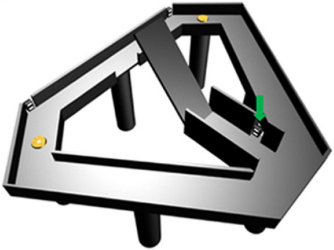


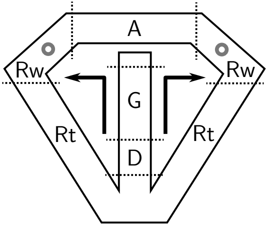
Network provides $\mathbf{E}$

+

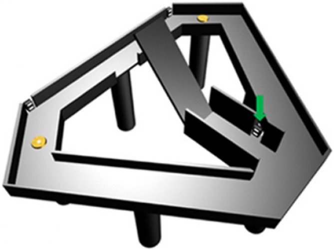
EM-rule $\Delta\omega_{ij} = \kappa a_i \alpha_j \mathbf{e}_j \cdot \mathbf{E}$

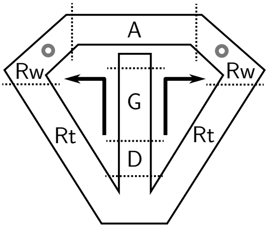
Reinforcement learning

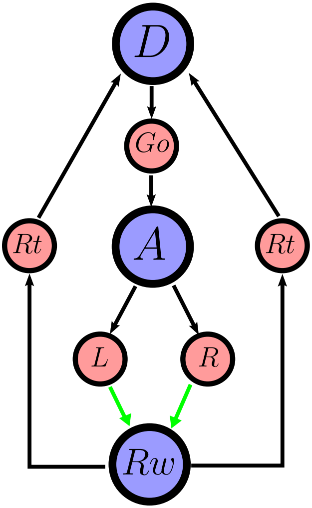




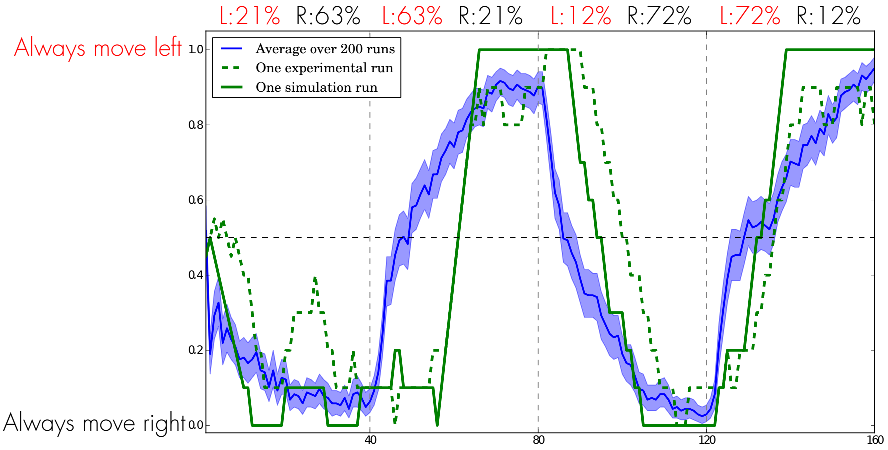
Reinforcement learning



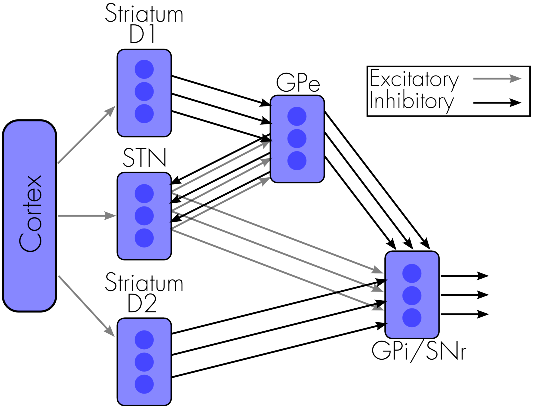




Behavioural results

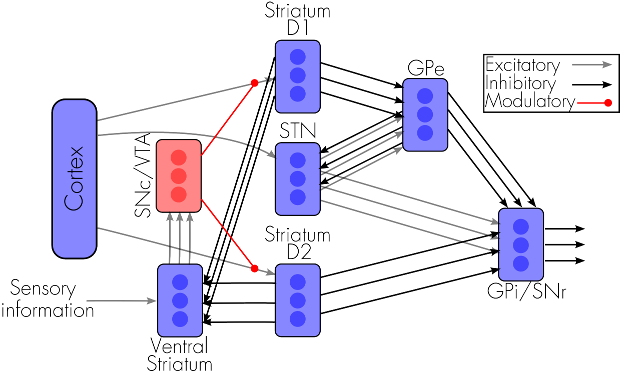


Action selection (basal ganglia)



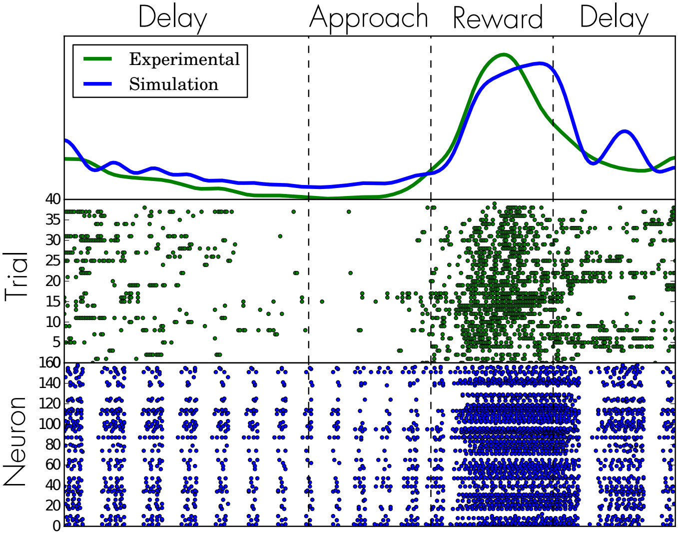
$$\pi(s) = \arg\max_a Q(s_{t+1}, a_{t+1})$$

Changing Q-values



Ventral striatum calculates reward prediction error

Neural results



Contributions

- Shown evidence that BCM captures qualitative features of STDP
- Proposed a biologically plausible solution to the supervised learning problem
- Created a model that solves stochastic reinforcement learning task and matches biology

A unifying learning rule

$$\Delta\omega_{ij} = \alpha_j a_i \left[\kappa_u a_j (a_j - \theta) + \kappa_s \mathbf{e}_j \cdot \mathbf{E} \right]$$

Acknowledgements

Thank you for listening and (possibly) reading!