Neural circuits for cognition

REINFORCE, synaptic learning from perturbations

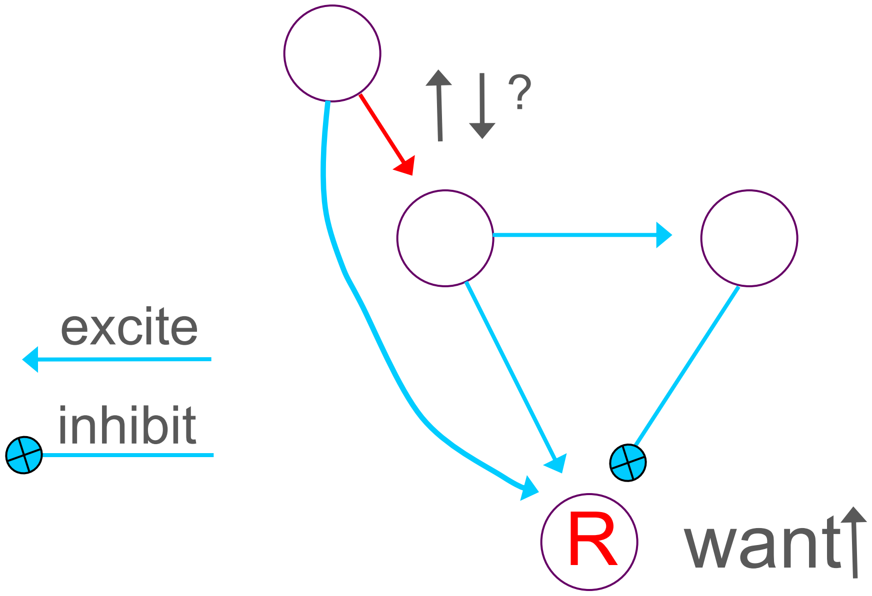
MIT Course 9.49/9.490

Instructor: Professor Ila Fiete

In-class journal club: delay-period activity

- **LAST TIME: Persistent states: Persistent Spiking Activity Underlies Working Memory.** Constantinidis C[1], Funahashi S[2,3], Lee D[4,5,6,7], Murray JD[5], Qi XL[8], Wang M[4], Arnsten AFT[4].
- **TODAY: Sequences: Choice-specific sequences in parietal cortex during a virtual-navigation decision task.** Harvey CD[1], Coen P, Tank DW.

Challenge: Activity-reward correlations (Hebbian learning) insufficient



$$\Delta W_{ij} = R x_i ?$$

Problem of (spatial) credit assignment

Spatial credit assignment for neurons

- How can one scalar reward signal signal be used to derive a signal for change in each synapse?
- If one could write down a model relating all synaptic weights to all neural responses to outputs to reward, and if the model were fully differentiable, one could use backpropagation.
- But plant dynamics might be unknown, there might be unmodeled noise, and all models are imperfect. Biases in the model can cause serious problems with learning.
- That’s why learning in software is seldom the same as learning in hardware.

REINFORCE

Williams 1992

Ω	state trajectory (all times, all variables) generated by system
$P_W(\Omega)$	probability of state trajectory (all times, all variables), parameterized by W
$R(\Omega)$	resulting rewards from world as consequence of state trajectory

$$\langle R(\Omega)\rangle = \sum_{\Omega} P_W(\Omega) R(\Omega)$$

Goal: adjust W to maximize expected reward.

REINFORCE

Williams 1992

Idea: derive gradient rule over expected reward; change weights to move along the gradient --

$$\frac{\partial \langle R(\Omega) \rangle}{\partial W} = \sum_{\Omega} \frac{\partial P_W(\Omega)}{\partial W} R(\Omega)$$

$$= \sum_{\Omega} P_W(\Omega) \frac{\partial \log P_W(\Omega)}{\partial W} R(\Omega)$$

$$= \left\langle \frac{\partial \log P_W(\Omega)}{\partial W} R(\Omega) \right\rangle$$

REINFORCE

Williams 1992

Idea: derive gradient rule over expected reward; change weights to move along the gradient --

$$\frac{\partial \langle R(\Omega) \rangle}{\partial W} = \sum_{\Omega} \frac{\partial P_W(\Omega)}{\partial W} R(\Omega)$$

$$= \sum_{\Omega} P_W(\Omega) \frac{\partial \log P_W(\Omega)}{\partial W} R(\Omega)$$

$$= \left\langle \frac{\partial \log P_W(\Omega)}{\partial W} R(\Omega) \right\rangle$$

$$\Delta W_{ij} \propto R(\Omega) \frac{\partial \log P_W(\Omega)}{\partial W}$$

→ sampling-based approximation of gradient

Interpretation of terms

$$\Delta W_{ij} \propto R(\Omega)\frac{\partial \log P_W(\Omega)}{\partial W}$$

reinforcement "characteristic eligibility"

The eligibility has zero mean:

$$\left\langle \frac{\partial \log P_W(\Omega)}{\partial W} \right\rangle = 0$$

(homework)

Learning rule is correlational/covariance-based: reward correlated with a term that has zero expectation on its own.
If reward uncorrelated with eligibility, then no change in W.

Example: recurrent network of Bernoulli-logistic units

Neuron model:

$x \in \{0, 1\}$

$$P(x) = px + (1-p)(1-x)$$

Bernoulli random variable

$$p_i = f(g_i) = \frac{1}{1 + e^{-g_i}}$$

Logistic function

Derivation of learning rule on board

REINFORCE and neural learning rules

- A class of algorithms that is less model-based: Model-free for how network output maps to reward, but model-based for neural activity.
- Obtaining explicit learning rule forms requires specific, differentiable models for neural activity.
- For specific stochastic neuron models (logistic-Bernoulli and more broadly the exponential family), we saw that the form of the synaptic learning rule can be simple.
- Provably gradient descent.

What about more complex neurons?

What about a rule that does not depend on/change for different neuron models?

Idea: perturbation-based learning (linearization)

- Totally model free for neurons and world: Do not require specific neural model: just that effect of perturbation on network dynamics is small.
- Each neuron estimates its own learning signal based on perturbation-outcome experiments.
- Covariance between experiments and outcome approximates the gradient signal.

	random variation →	consolidation
evolution	mutation recombination	replication
bacterial chemotaxis	tumbling	tumbling suppression
learning	node perturbation* weight perturbation*	synaptic strengthening

* Barto and Anandan 1985/Williams 1992; Xie and Seung 2004, Fiete and Seung 2006.

*Minsky 1954; Barto 1983; Williams 1992; Seung 2003.

Derivations and learning rules

- On the board

Summary: simple rules for gradient learning in neural neworks

- Derivation of 3-factor rules that involve reinforcement signal, presynaptic activity, and postsynaptic fluctuations for stochastic gradient learning.
- Rules replace gradients with covariance-based estimates of the gradient.
- An empirical approach: random trials correlated with changes in reinforcement.
- Model-free rules: the more model-free they are, the more generally they apply across neuron models. However, learning can be much slower.
- Next: Do animals use these kinds of learning rules, and what are the predictions? → Return to the songbird.
- Next after that: REINFORCE and where it sits within reinforcement learning.