
# *welcome to Week 7*

Computation and the Brain

Fall 2019

# Wrapping up dynamical systems: Hopfield nets

Wherever you start, the system will converge

”if unhappy, flip”

So, it can be used as memory

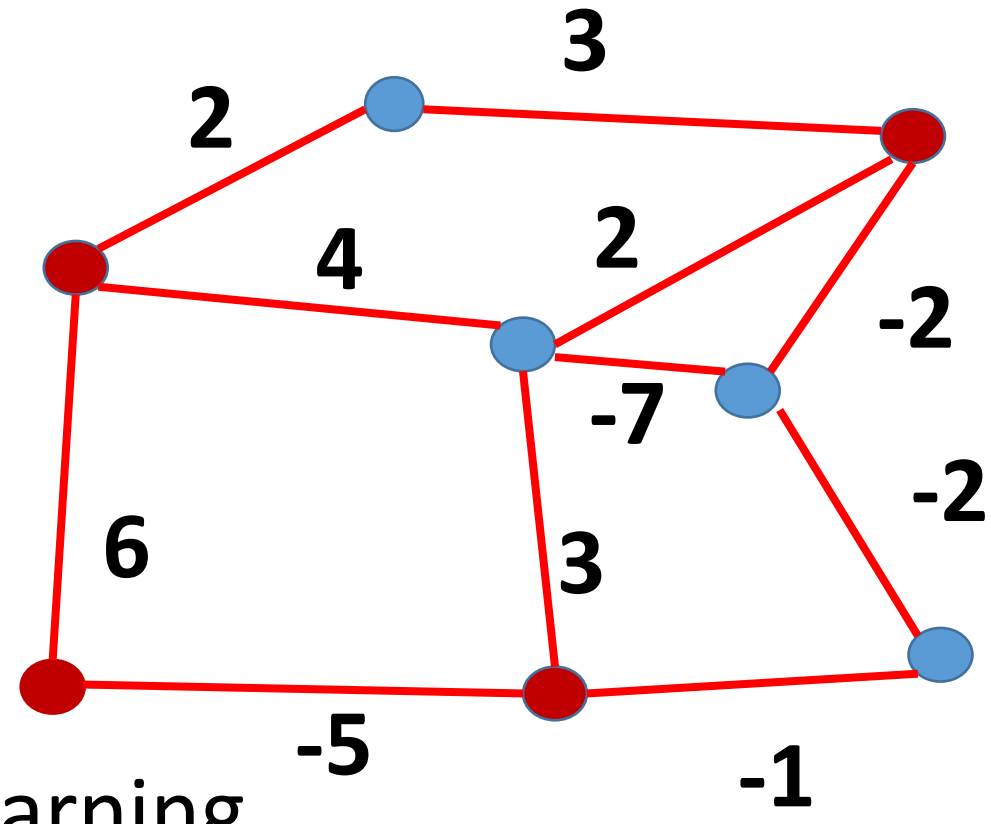
You can train it to remember

a set of patterns

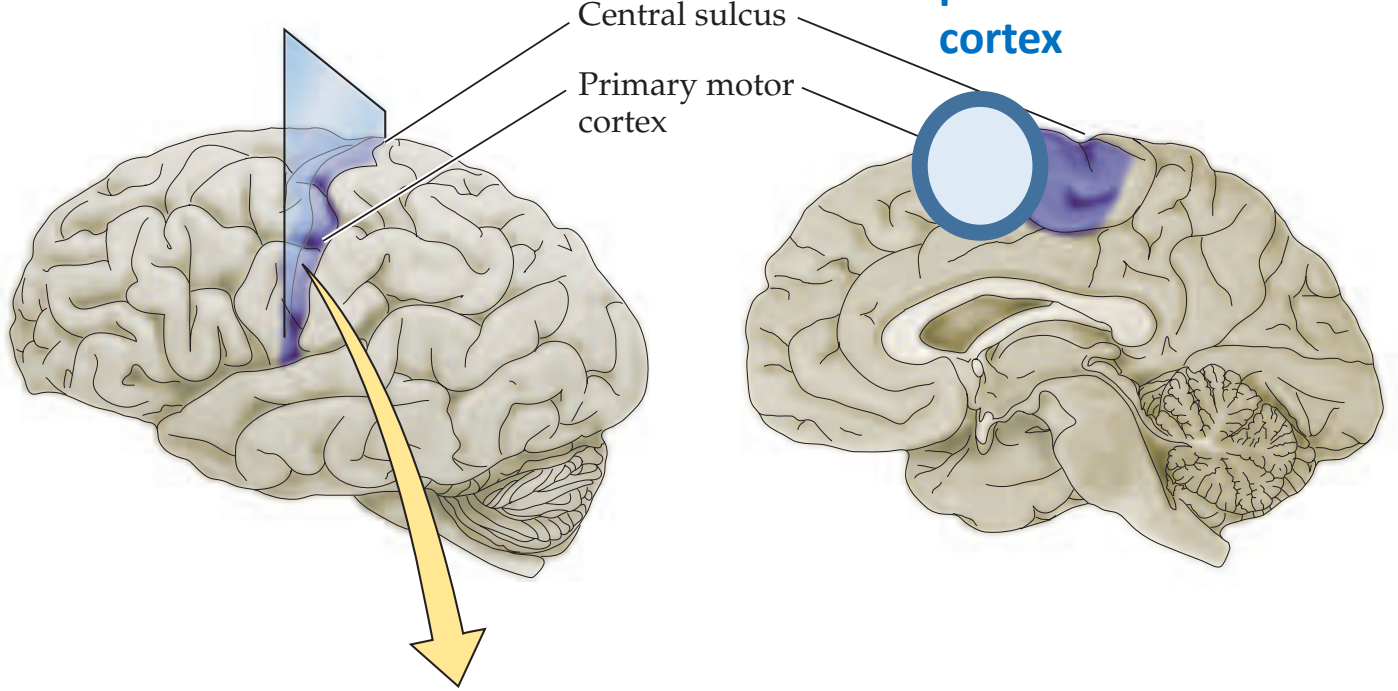
Capacity: ~ 0.135n

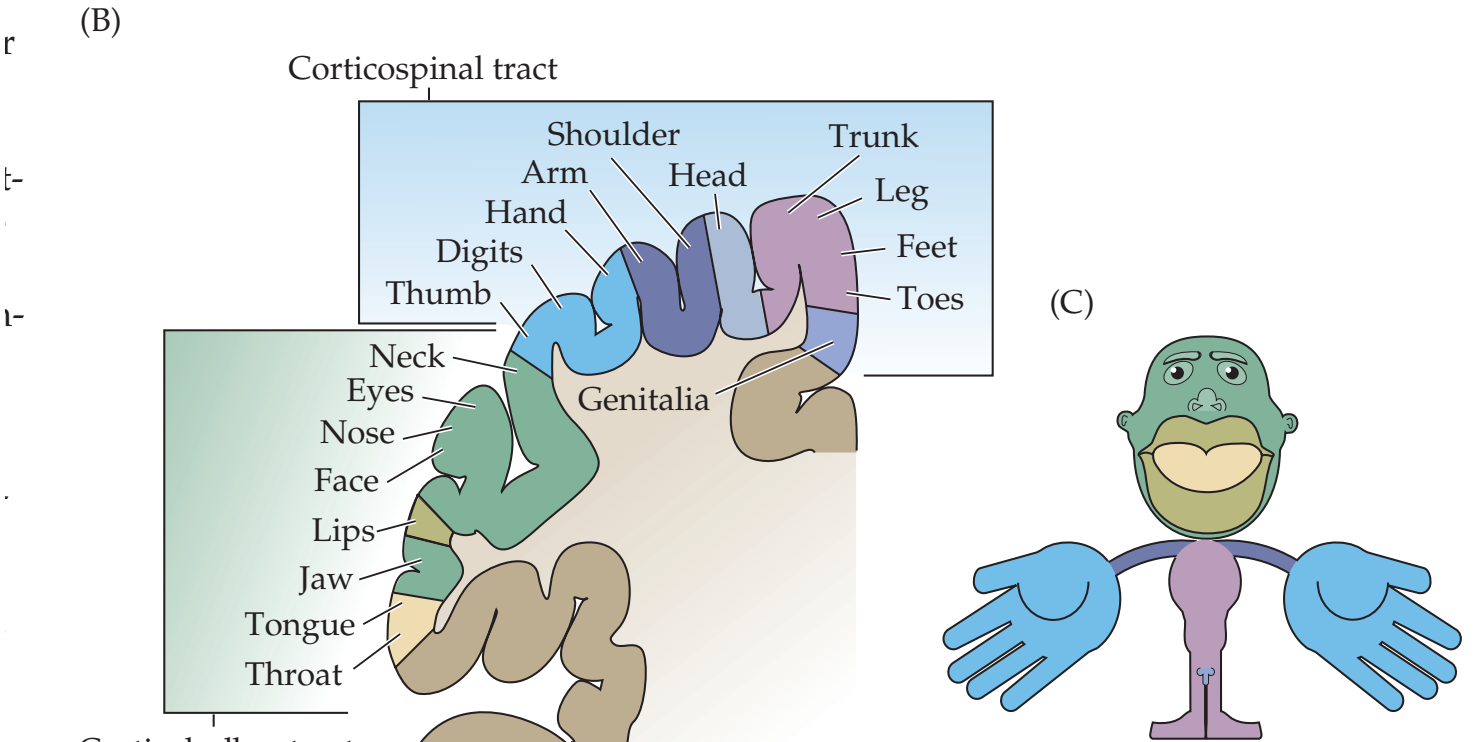
Boltzmann machines

Part of the prehistory of deep learning…

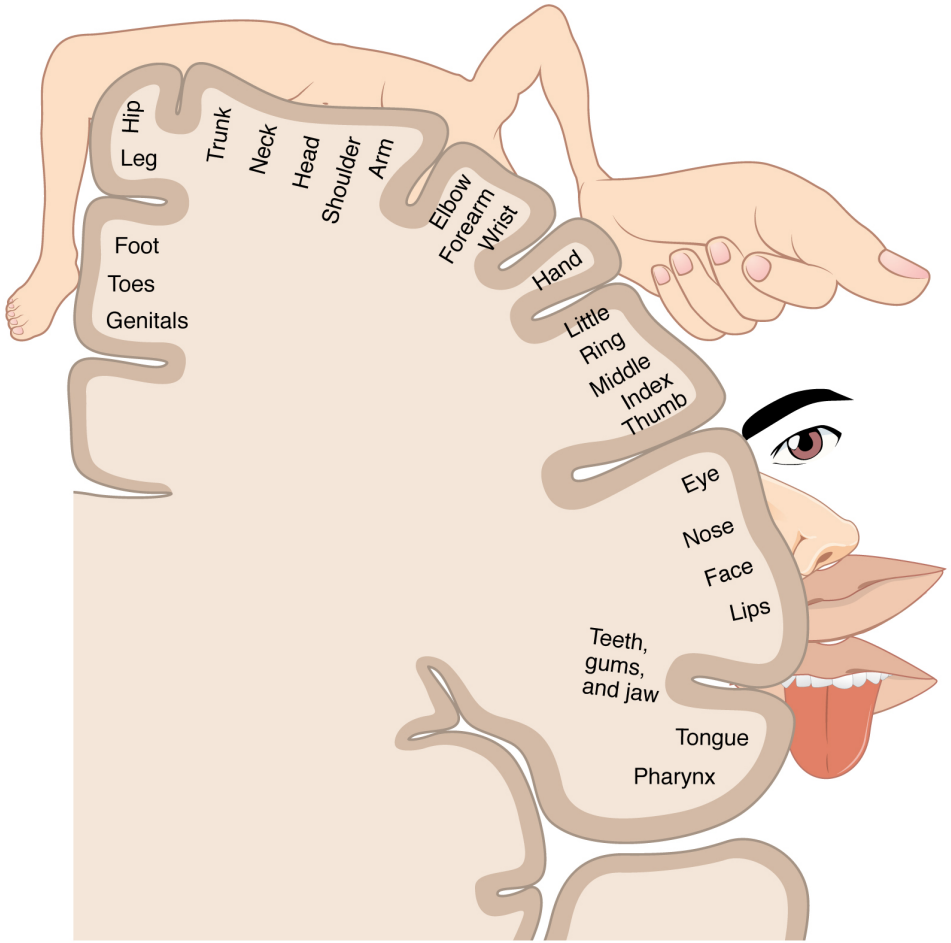


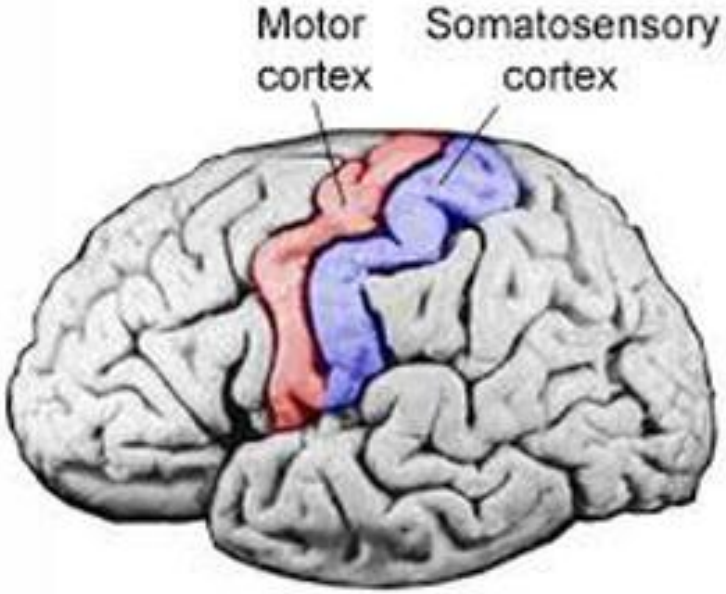
# The Motor Cortex



# Compare with: the somatosensory humunculus

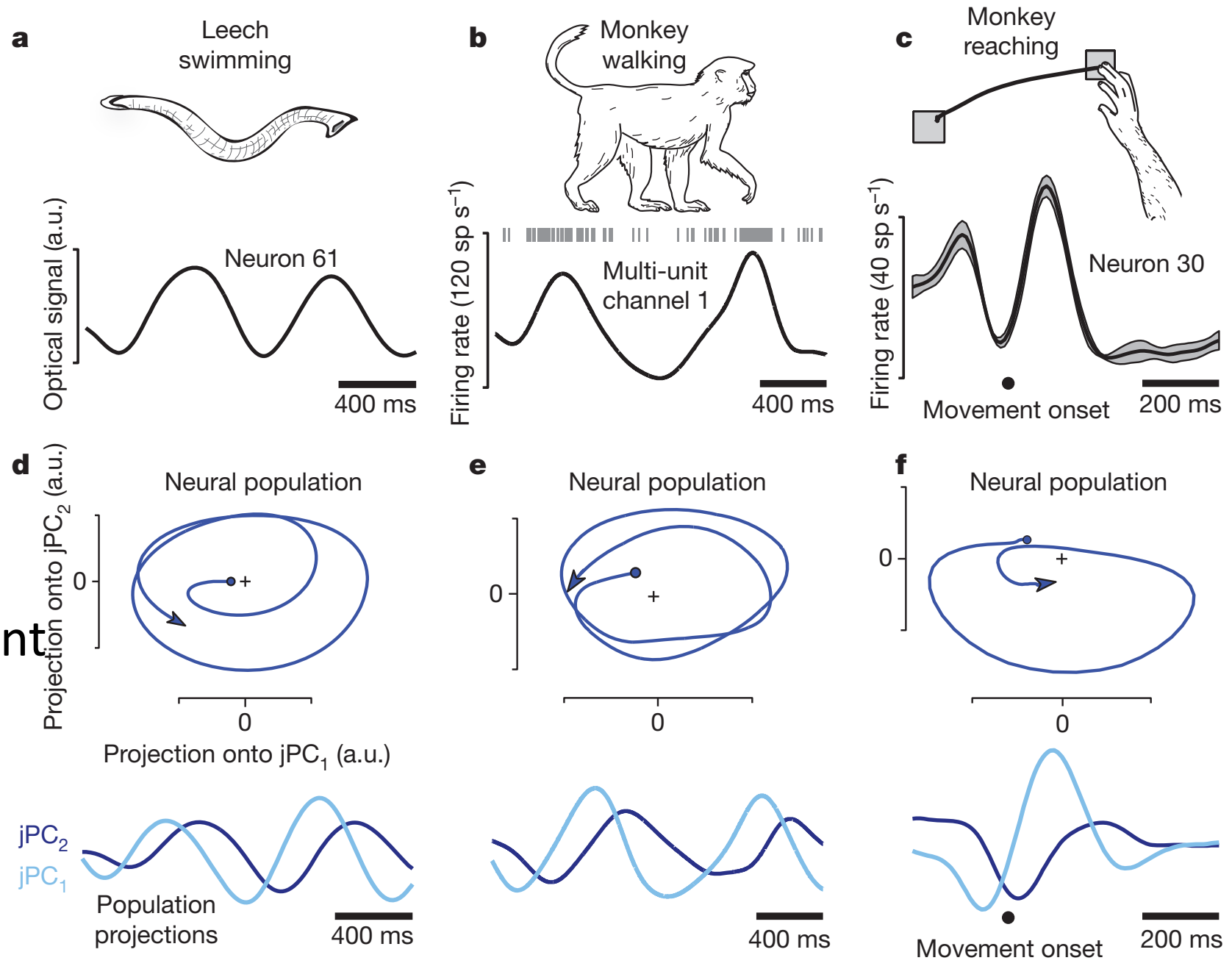



# Traditional view: M1, M2, M4, etc
# Modern view [eg Churchland et al. 2012]

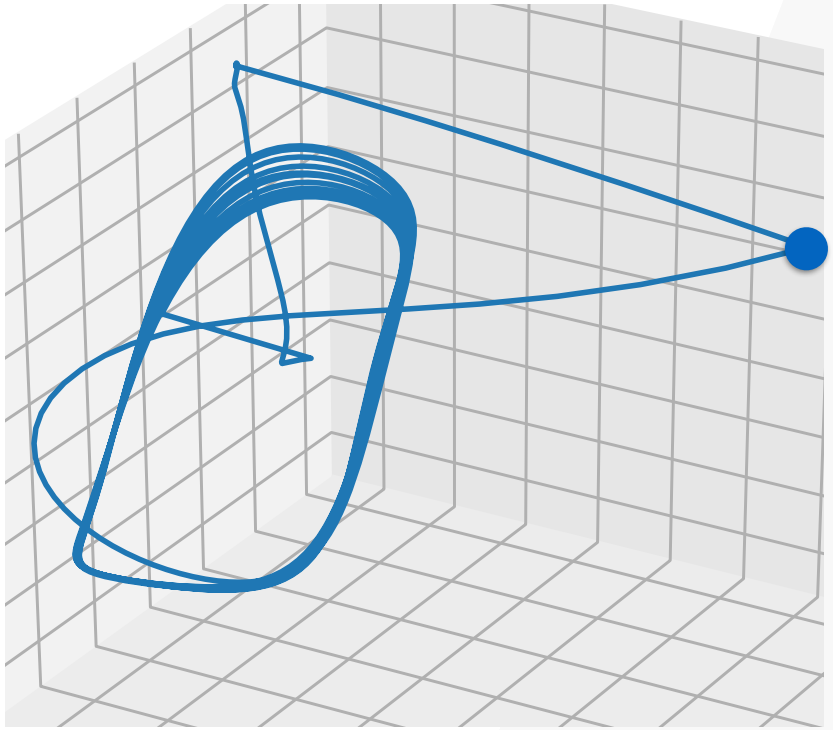
- A neuron in the motor cortex is a row (eqn) of a dynamical system, whereby the motor cortex generates and controls movement: $\dot{\mathbf{r}}^t = \mathbf{f}(\mathbf{r}^t, \mathbf{u}^t)$
- So, neural responses reflect underlying dynamics, and encode stimuli only incidentally
- Hypothesis: ”movement generation across the animal kingdom involves rhythmic, oscillatory activity”
- **BUT** does the neural state rotate during all motion (even non-rhythmic movement like reaching)?

- Two animals
- Three motions
- Two of the motions are rhythmic, the third is not
- The all have significant cyclic projections of neural activity
- ***jPCA***



# Precise Estimates of Single-Trial Dynamics in Motor Cortex using Deep Learning Techniques

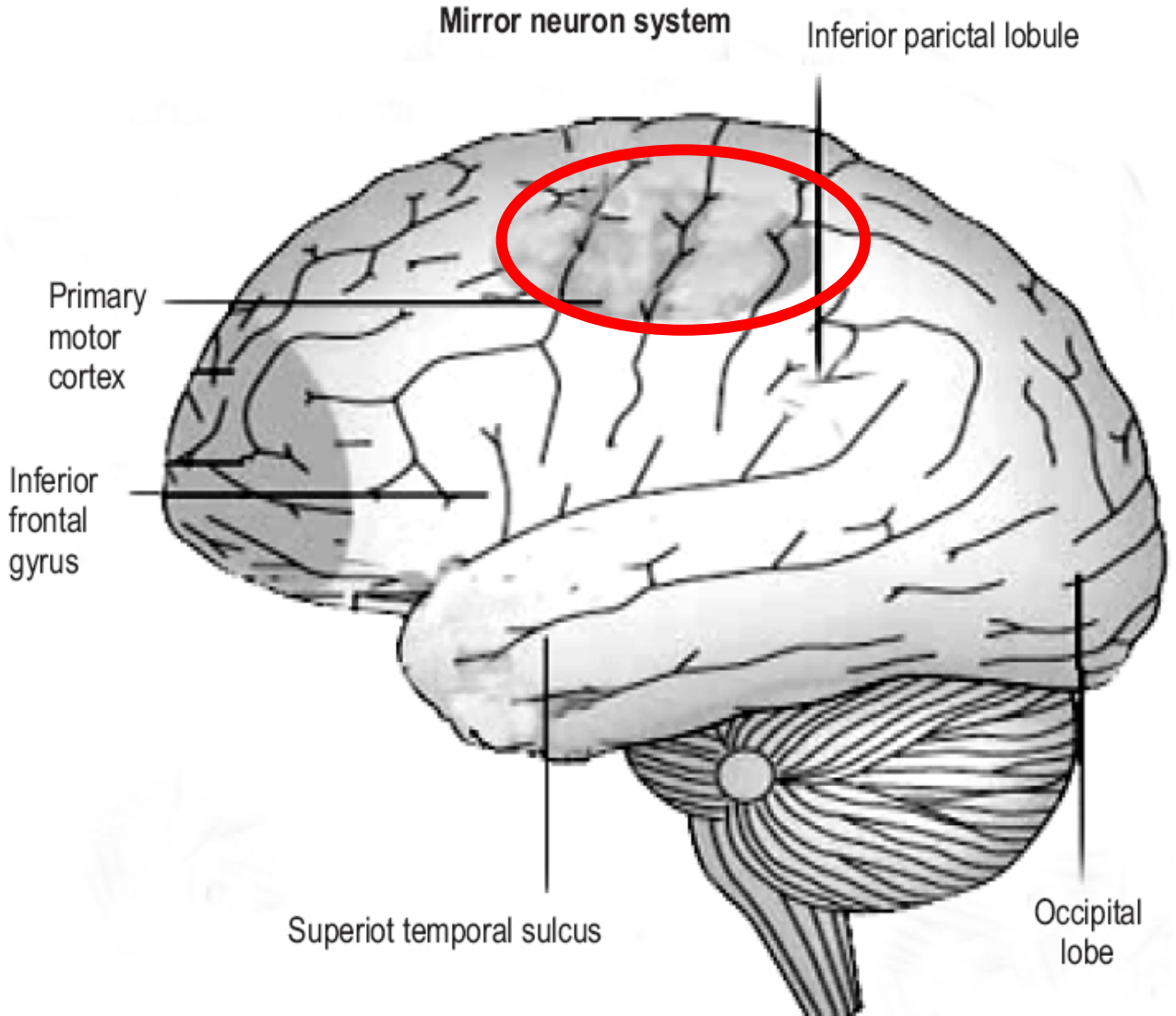
…Mark Churchland, ***Larry Abbott****, …



**his slides*

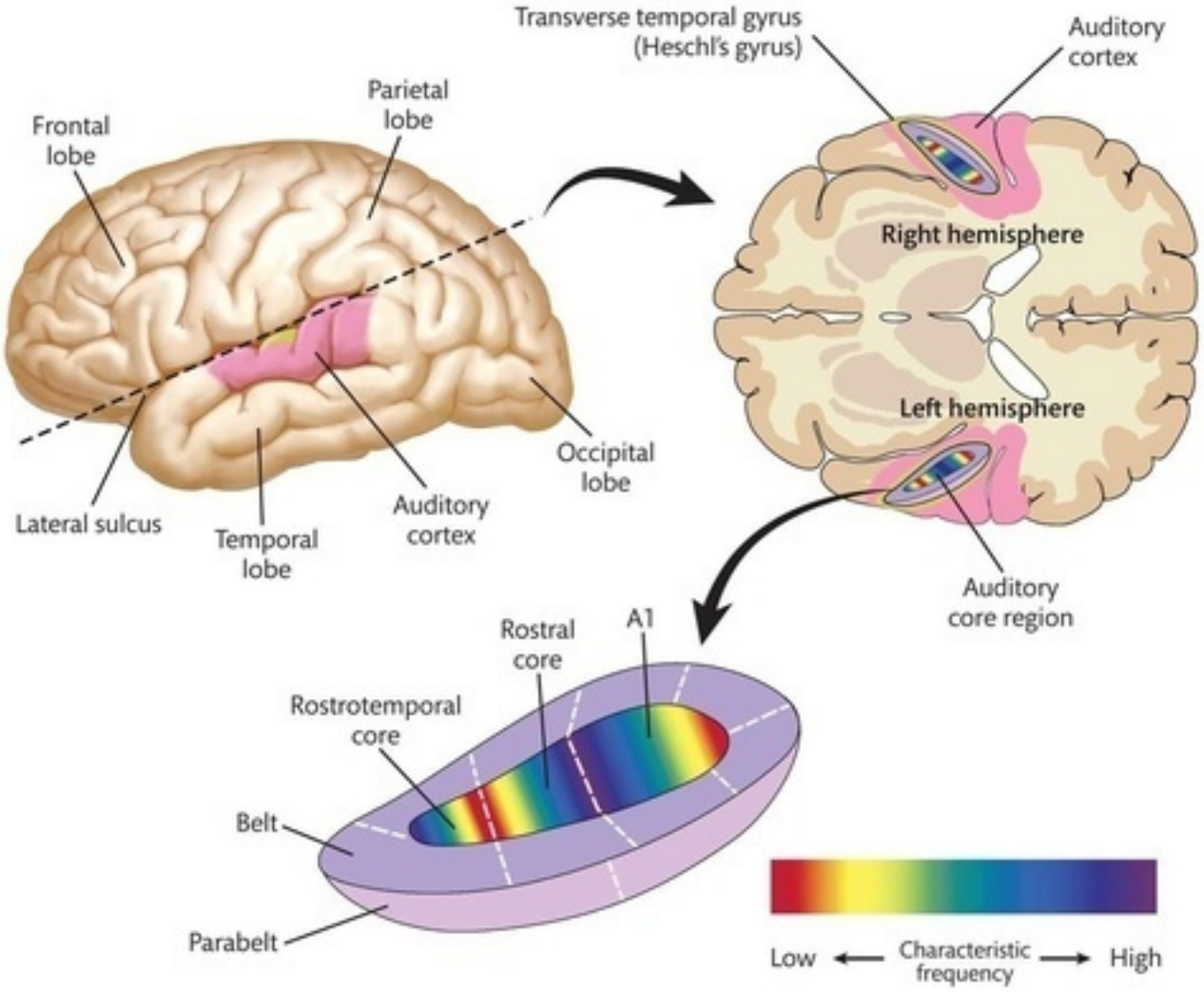
# Also in the motor cortex: the mirror neuron system (MNS)…

…connections to language…



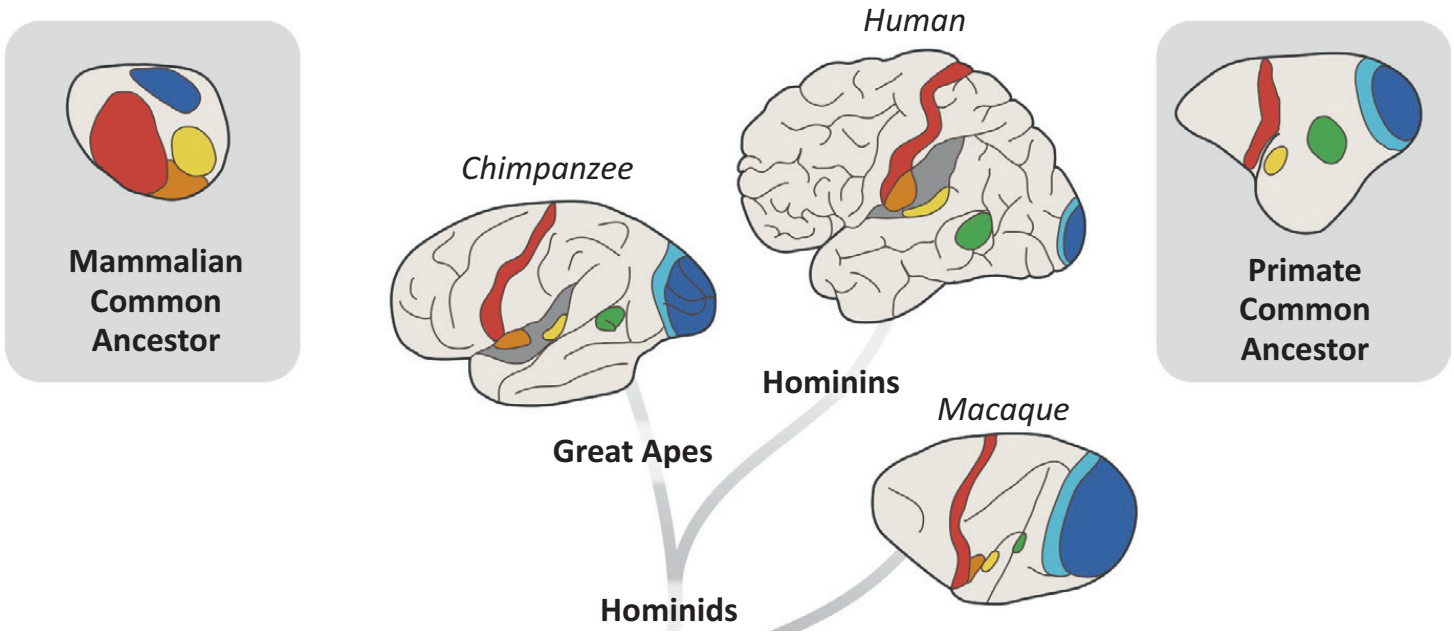
# Btw auditory cortex

- Near the ear
- A1, A2, belt, parabelt
- Still cochlea signal goes through thalamus (MGN)
- Specializes in frequencies
- Right: tonal, music
- Left: temporal aspects, rhythm, ***speech***



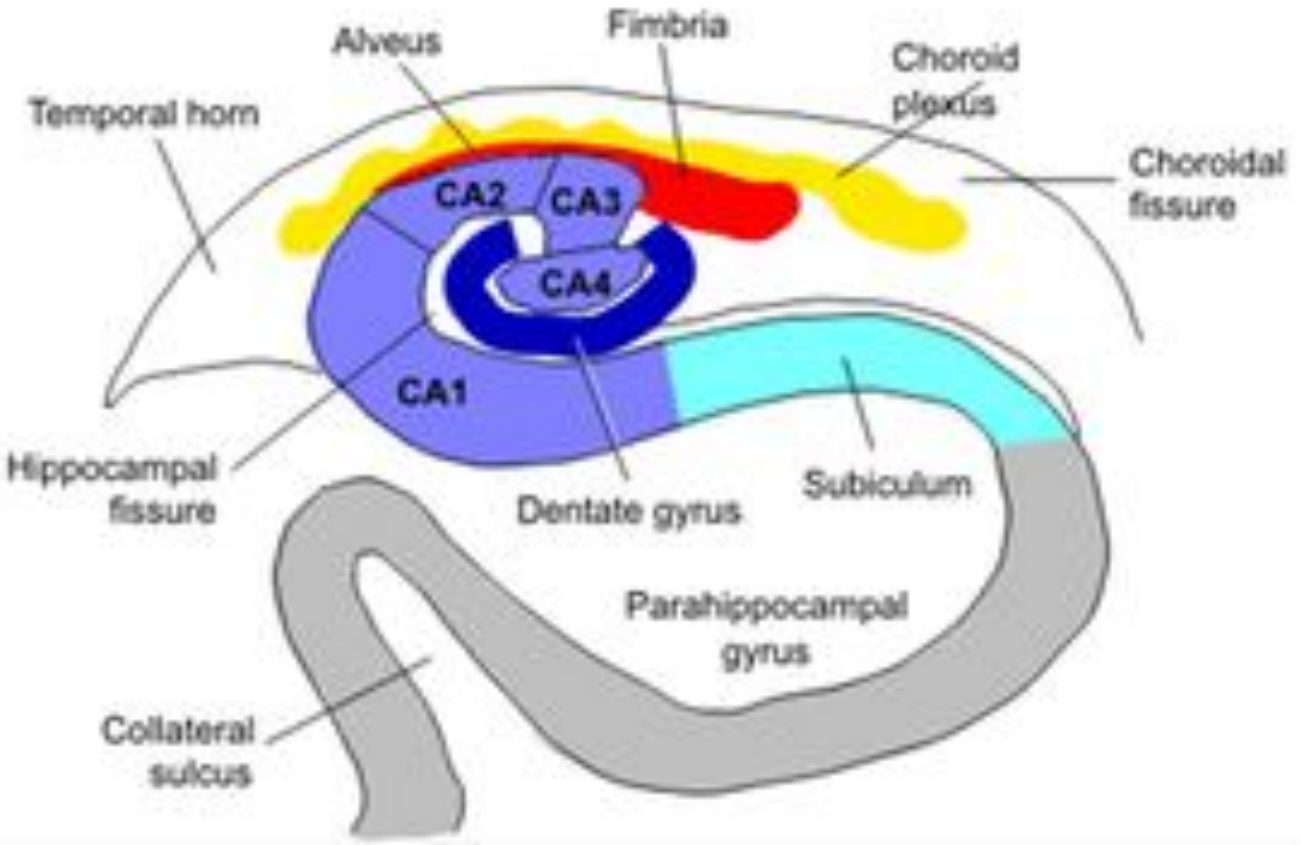
# Soooo…  we have talked about:

- Vision
- Olfaction
- Somatosensory
- Audition
- Motion
- (I/O)
- ***Where else in the Brain does computation happen?***

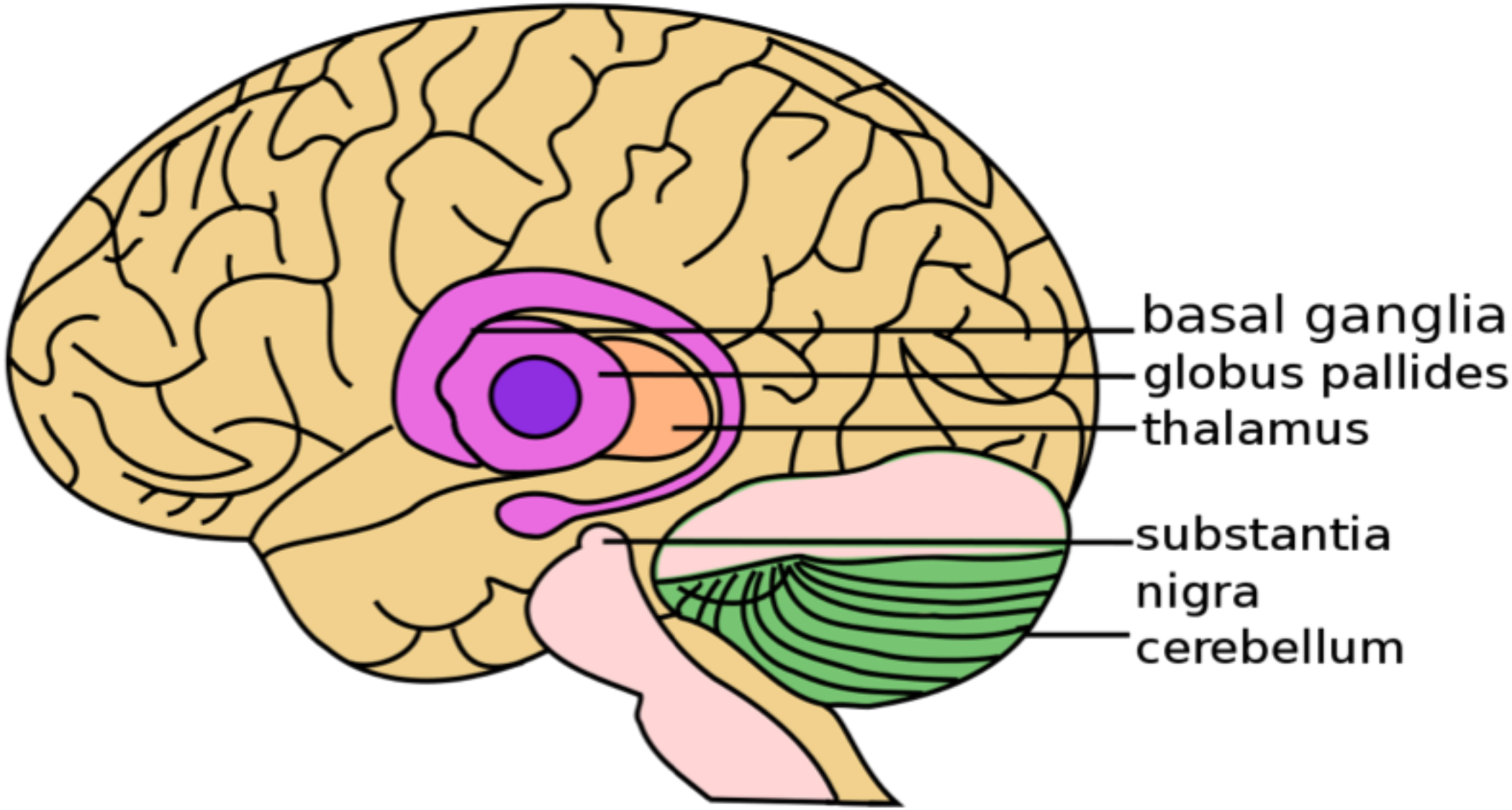



# The rise of the *association cortex*
(= neocortex excluding sensory and motor areas, and the DMN)

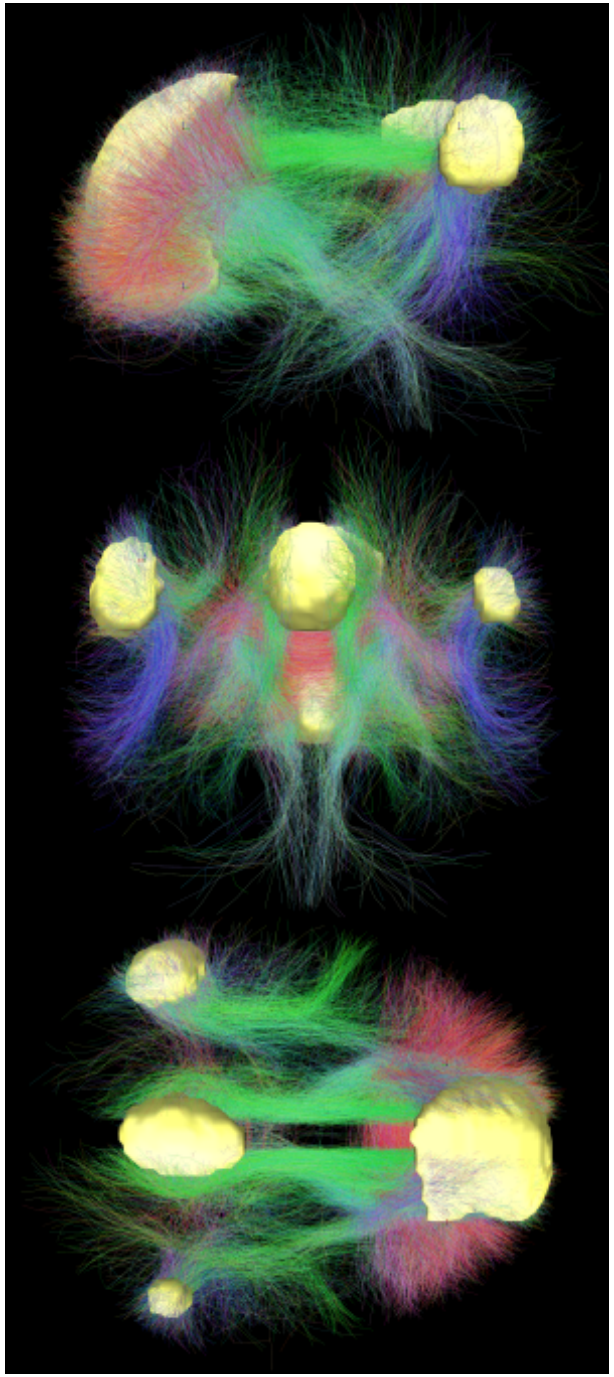
# Hippocampus: the gateway of the AC



# Basal Ganglia and Related Structures of the Brain



…and the default mode network (DMN)…



# Oscillations in the Brain

***frequency bands***

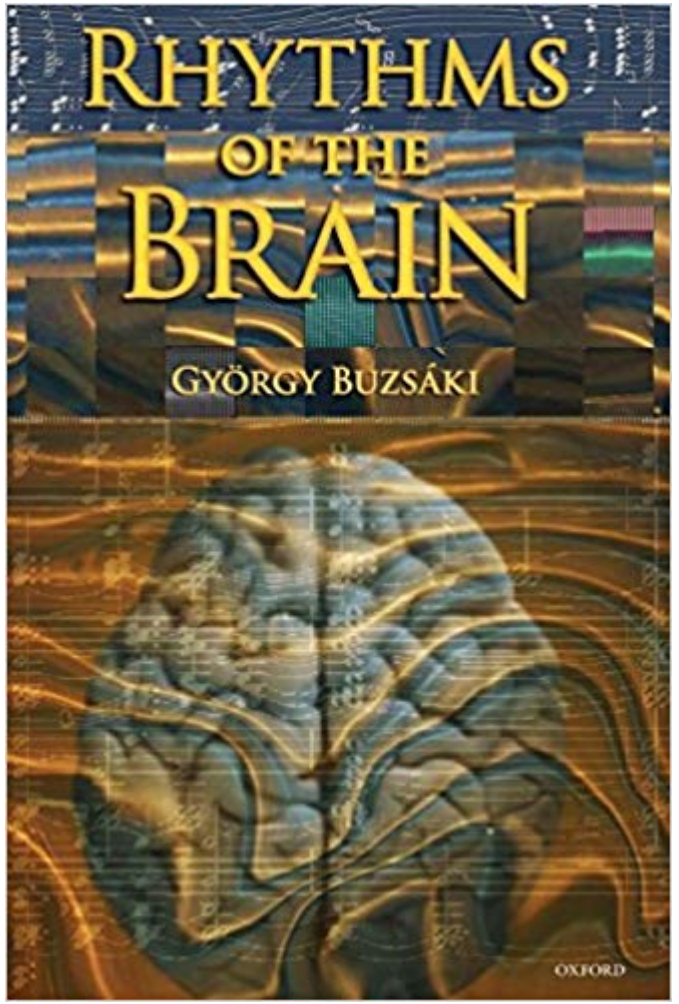
*delta* (2–4 Hz)

*theta* (4–8 Hz)

low *gamma* (40–70 Hz)

high *gamma* (70–150 Hz)

***default mode: 0.01 hz***



- If you use **imperfect inverses in DTP (Difference Target Propagation),** wouldn't the error margin get bigger and bigger? How does the reconstruction Lillicrap mentioned help with
- How does the result in Lillicrap's paper **inform future research for finding biologically plausible weight-update mechanisms**?
- Is it possible that the **success of backprop** is in large part due to the ease of efficient implementation for large datasets? **Perhaps if somebody was able to implement an equally fast and scalable LBFGS, it would work even better?**
- Lillicrap talked a lot about how backpropagation isn't biologically accurate **but not a lot about how the brain actually updates weights or activations**. **Does the brain use a form of non-precise credit assignment** (weight updating) and how are these weights learned?
- **how is possible to know that complex language is unique to humans?**

- The **anatomic regions responsible for language are** mostly in the neocortex which is **present in all mammals** to different extents, however, non-human animals do not have the ”genetic program” to learn language. **What do these areas do in other animals** and how could we use that to understand the mechanisms used by humans for language?
- Are there any **NLP techniques** inspired by how the brain does language?
- What evidence do Chomsky, Berwick, etc. use to argue that language is its own mental organ/set of finite computational mechanisms and **not "epiphenomenon of the human capacity** to share intentions in the use of communication"? (HW1).
- Are there feasible hypotheses for why the brain might have the **grey matter vs. white matter** distinction?
- Is there a **standard in terms of locating and observing different parts of the brain?** While the larger structures of the brain are consistent, the subsections seem much more complex and could be misidentified. What about variation between brains? Are the folds of the brain unique to people or is there a consistent pattern?

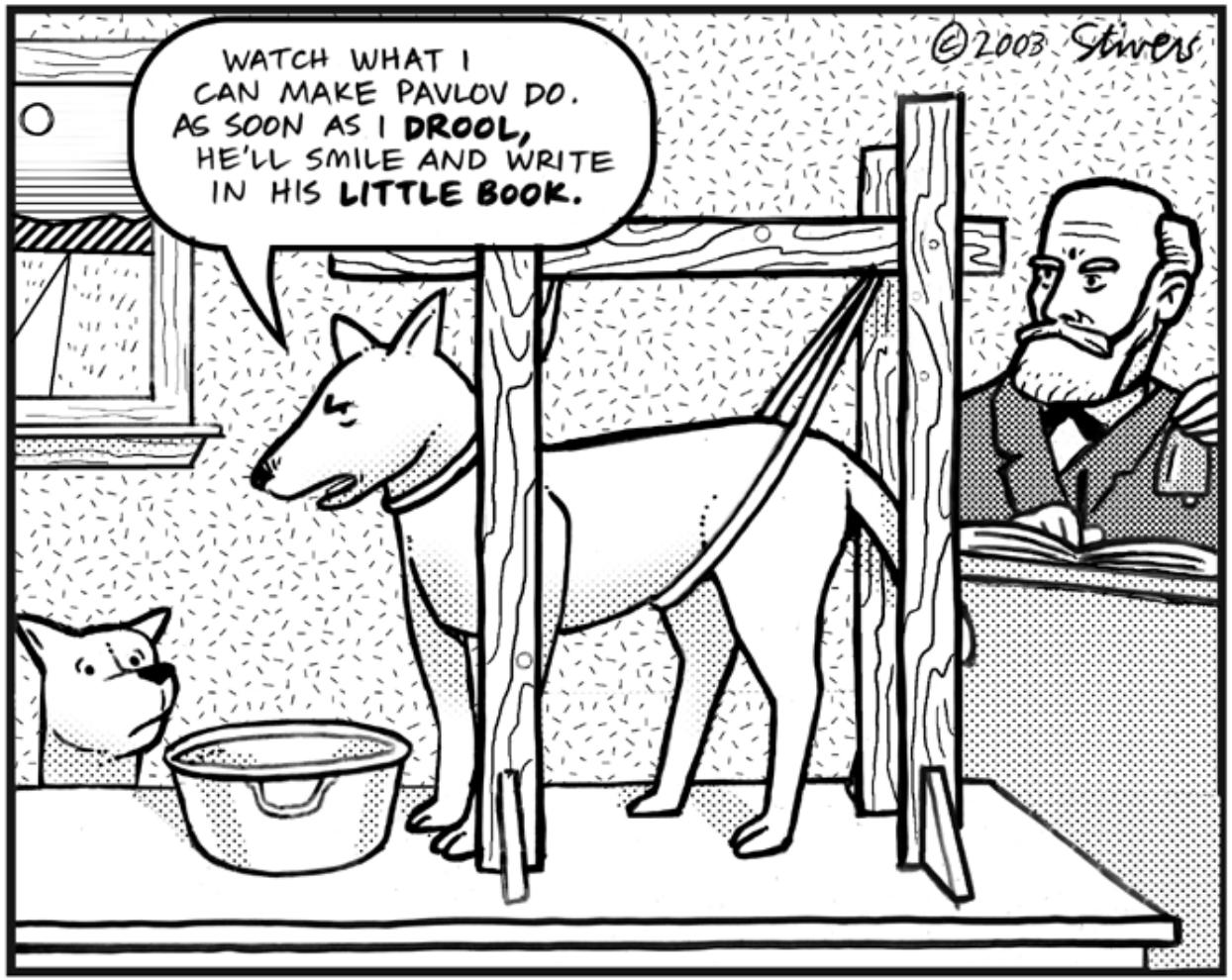
- **Is evolution done developing the human brain?** If yes, why, and are humans at fault for trying to interfere with evolution? If no, what else could evolution develop the brain to do?
- The distinction that Freiderici makes between the cognitive system and neuroscience echoes Marr's 3 levels - **is this a meaningful distinction or just a useful tool for conceptual analysis?**
- Why does Friederici say that one group of researchers views **language as distinct from communication?** For what other purpose might language have evolved?
- When playing the piano, our hands know where to go without us having to consciously think about the movement. Similarly, when typing, we might make mistakes because we auto-complete with an incorrect but very common sequence of characters. I imagine this effect happens in language and reasoning as well. **Is this something that is understood or localized in the brain?** Could this be a core feature enabling language?
- Friderici talks about two interfaces of language: the external that perceives and produces language and the internal that actually understands the concepts and intentions of language. Is currently NLP/computational language research moving towards bridging **the gap between the external and internal interface?**
- Is current research on semantics based deep learning **actually learning concepts** and intentions or just learning another way to represent language?

- Can we safely conclude that the **cognitive system is distinct (in the sense of being independent) from neuroscience** (as it relates to the relationship between composition, structure organisation and function)?
- It seems the MERGE operation is very abstract - clearly every language *has* some sort of ‘merge’, **but what are the properties of the machine that accepts valid merges?**
- **What has the division of language into subcomponents led to in ML?** In particular dividing it into semantics and syntax. I suppose current state of the art NLP (ie BERT- LSTMs / Transformers) don’t particularly enforce syntax and the system has to learn that, and then of course it uses the words in a somewhat intelligible way so it clearly has achieved some meaning. The subdivision doesn’t come up in a clear way as far as I can tell, except that it’s **much easier for the systems to learn syntax than semantics**.
- Do you think that a system taking **purely language input** can **build up a conception of the world** (ie some sort of mental model for what the world is) **without having ever experienced it** in any capacity?
- Is **language's definition tied to the nature of the task** or the computational mechanism by which it is rendered possible in the world?
- **Can we understand meaning without language?** Do we understand meaning in the form of language?
- Is and has the **definition of language been entirely agreed upon** in the past? If we were to expand the definition, and thus possibly include forms of communication that animals take part in, would it help our understanding and knowledge of this phenomenon?

- When it comes to fields like **natural language processing/linguistics and computational neuroscience,** is there much overlap or **collaboration between the two?**
- (Not directly related to the reading) **What would be the properties of a computational model consisting of random graphs?** I.e., the model cannot rely on any a priori connectivity to perform computation, but must set its 'weights' to solve some task. Can we prove bounds on the minimum graph size, average degree, training time needed, etc. required to solve various tasks?
- I was wondering **how supported the research of Jeff Hawkins is**. I definitely agree with the idea that the job of a large part of the brain is prediction, but the idea of the column being the key to unlocking the understanding of the brain seemed controversial during the talk. **Is this sort of research agreed upon in the field?**

# Today

- Reinforcement learning
- Conditioning vs
  Temporal Differences
- RL in the Brain
- The math of RL
- Deep RL and Alpha Go

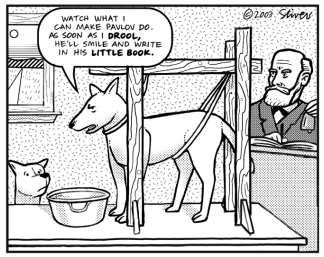


# Next

- Language (about 4 weeks)
- Preliminary, ***volunteer!***) project presentations
- Evolution and Development of the Brain
- ***Dec 4:  Project presentations!***

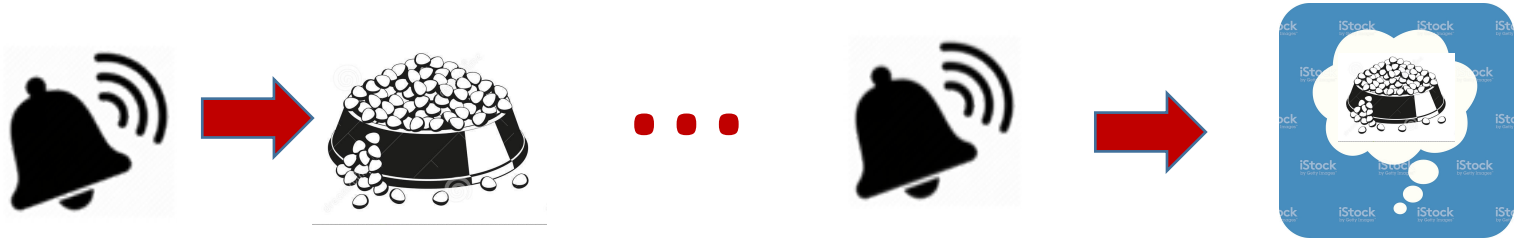
# Reinforcement learning in the Brain

- Learning about the world, or about your actions, exclusively through rewards
- Rewards: positive or negative: food vs. work
- Conditioning, pre-1980s:  how do past rewards affect animal expectations?
- Reinforcement learning: how do rewards affect animal behavior?



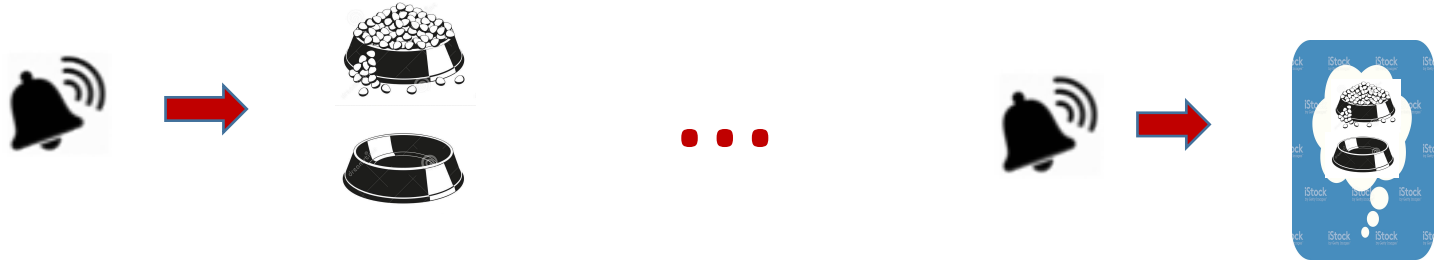

# Classical conditioning

- Pavlovian

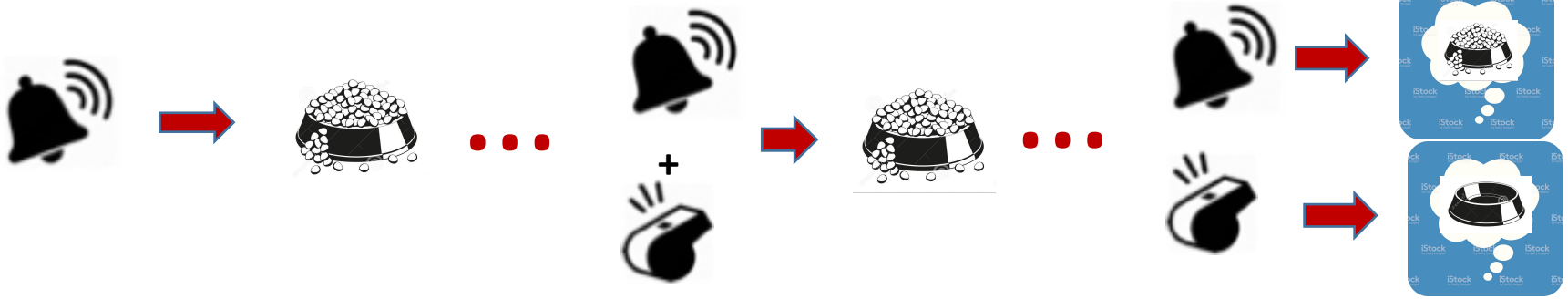


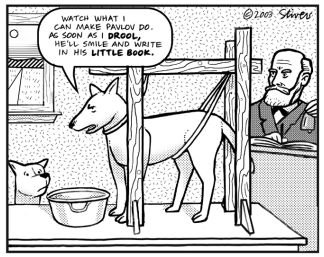
- Extinction

- Partial



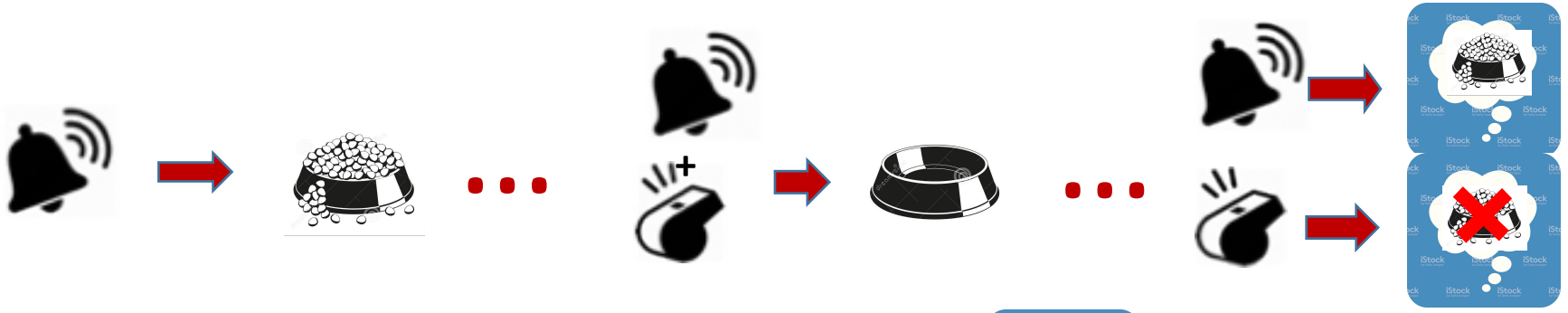
- Blocking



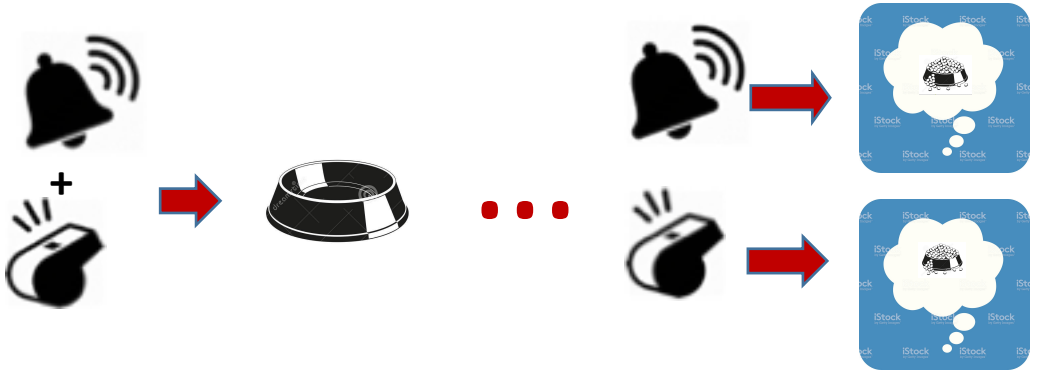



# Classical conditioning (cont.)

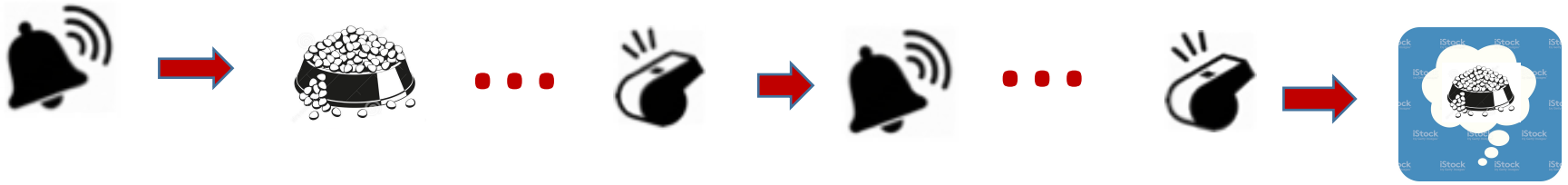
- Inhibition



- Overshadow



- Aristotelian



# Explanation (1960s): Rescola-Wagner update (delta rule)

Stimulus u, prediction x, weights w

(w, u possibly vectors)

Reward R

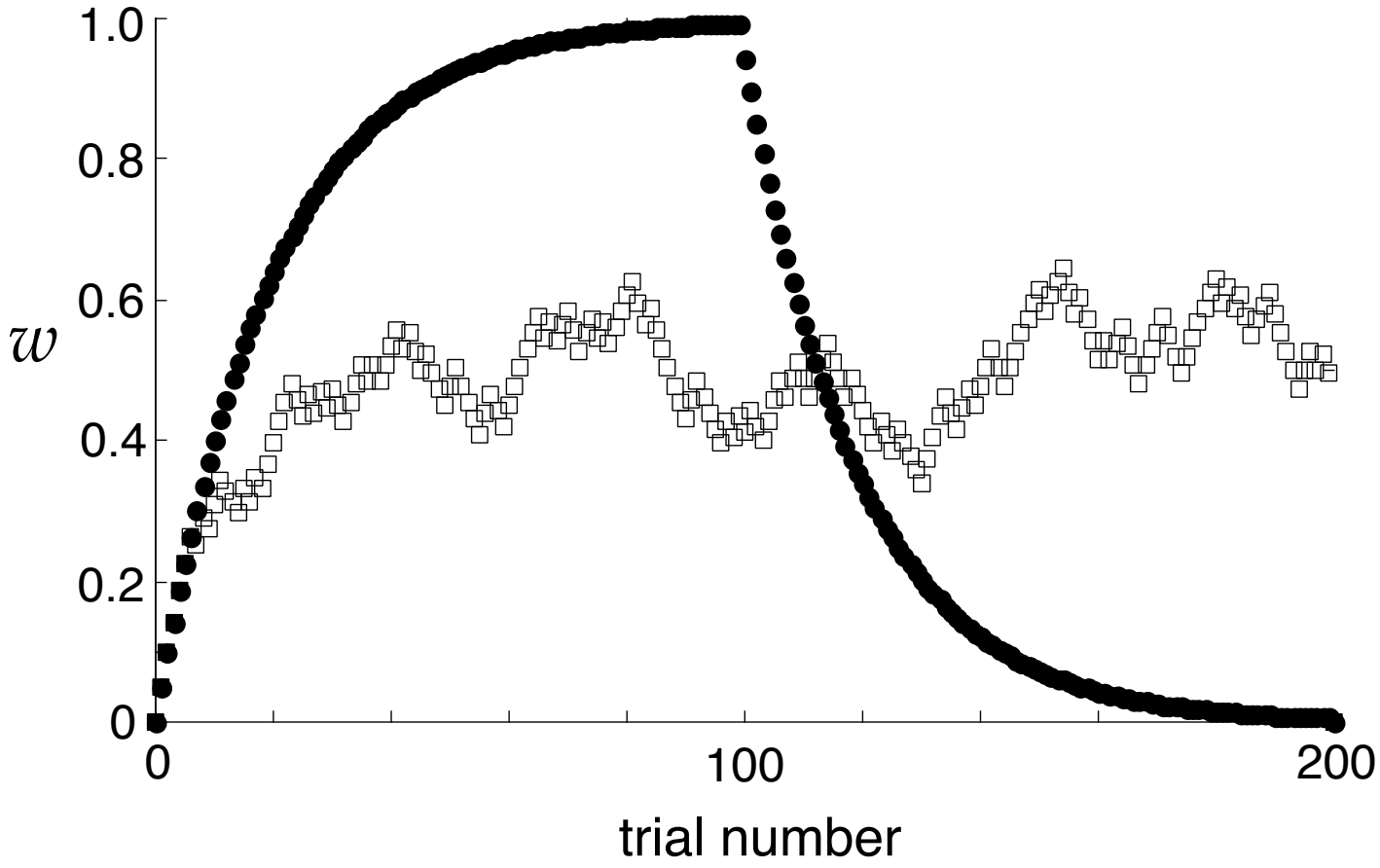
$x = u \cdot w$

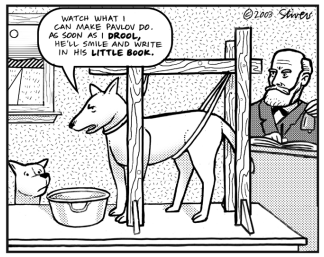
Rescola-Wagner plasticity: $\mathbf{w \rightarrow w + \varepsilon \cdot (R - x) \cdot w}$ ($\delta$ = $R - x$)

Note that this is gradient descent minimizing $\boldsymbol{\delta^2}$

# It explains classical conditioning well enough

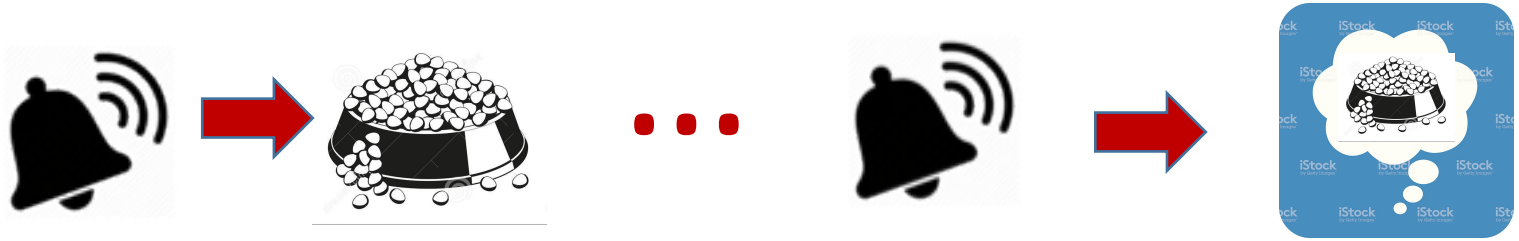
100 x (bell, food) + 100 x (bell, no food)
vs 200 x (bell, random)





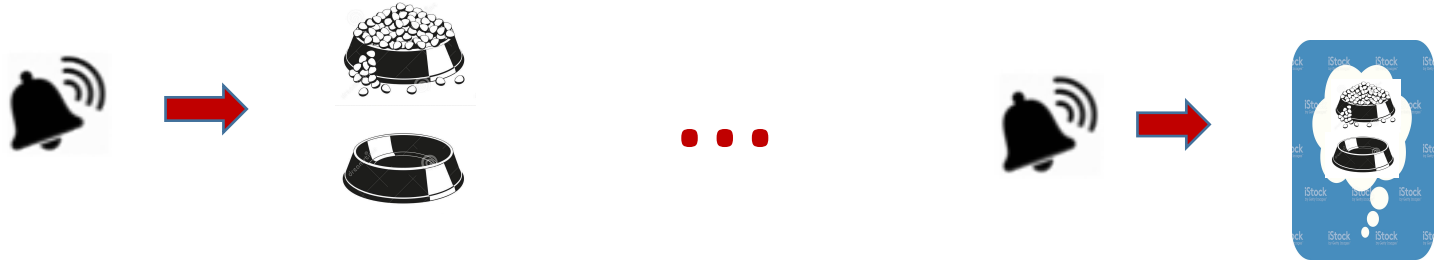

# Classical conditioning

- Pavlovian

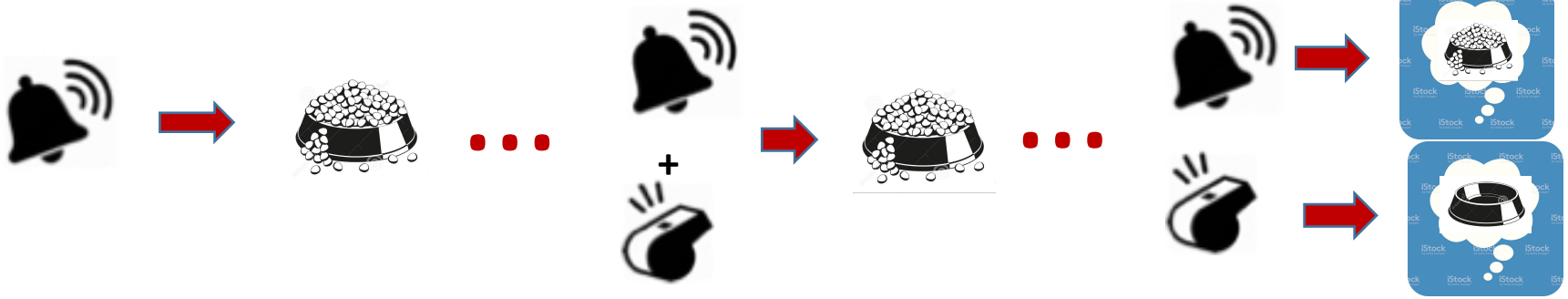


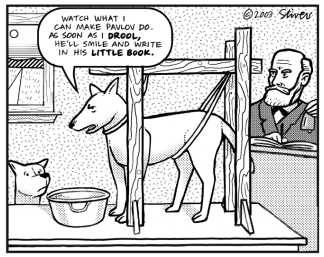
- Extinction

- Partial



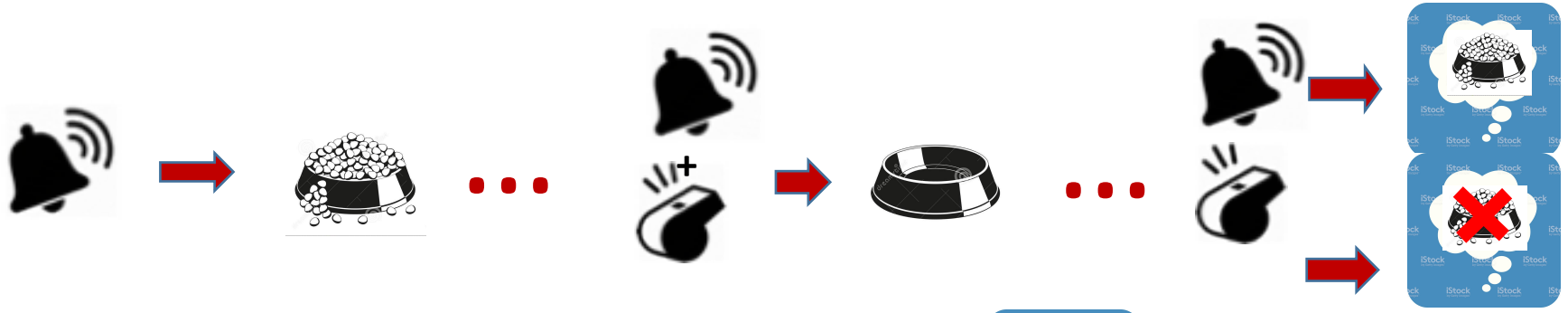
- Blocking



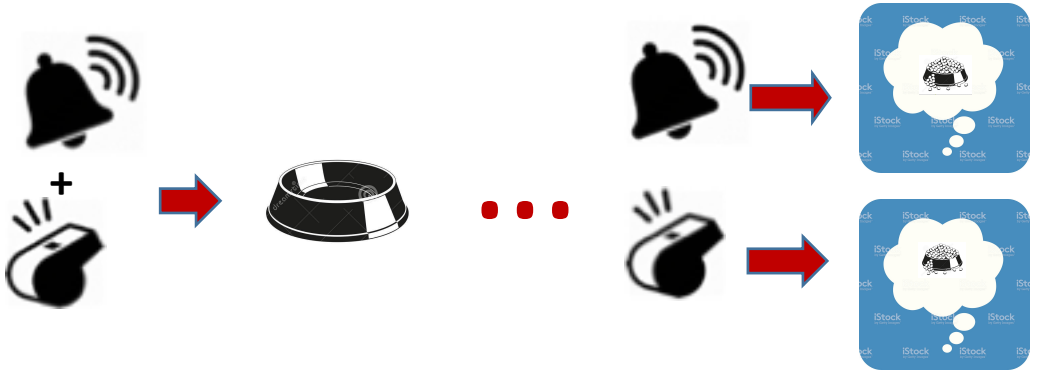



# Classical conditioning (cont.)

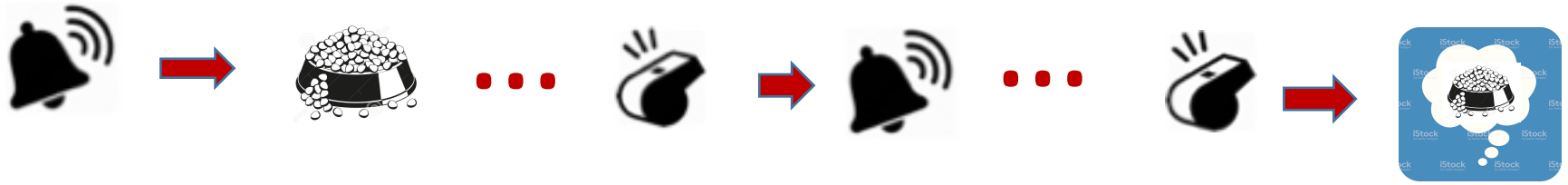
- Inhibition

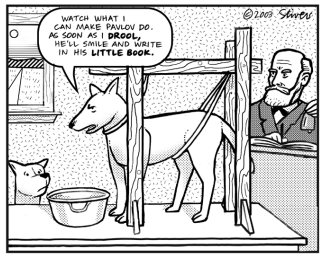


- Overshadow



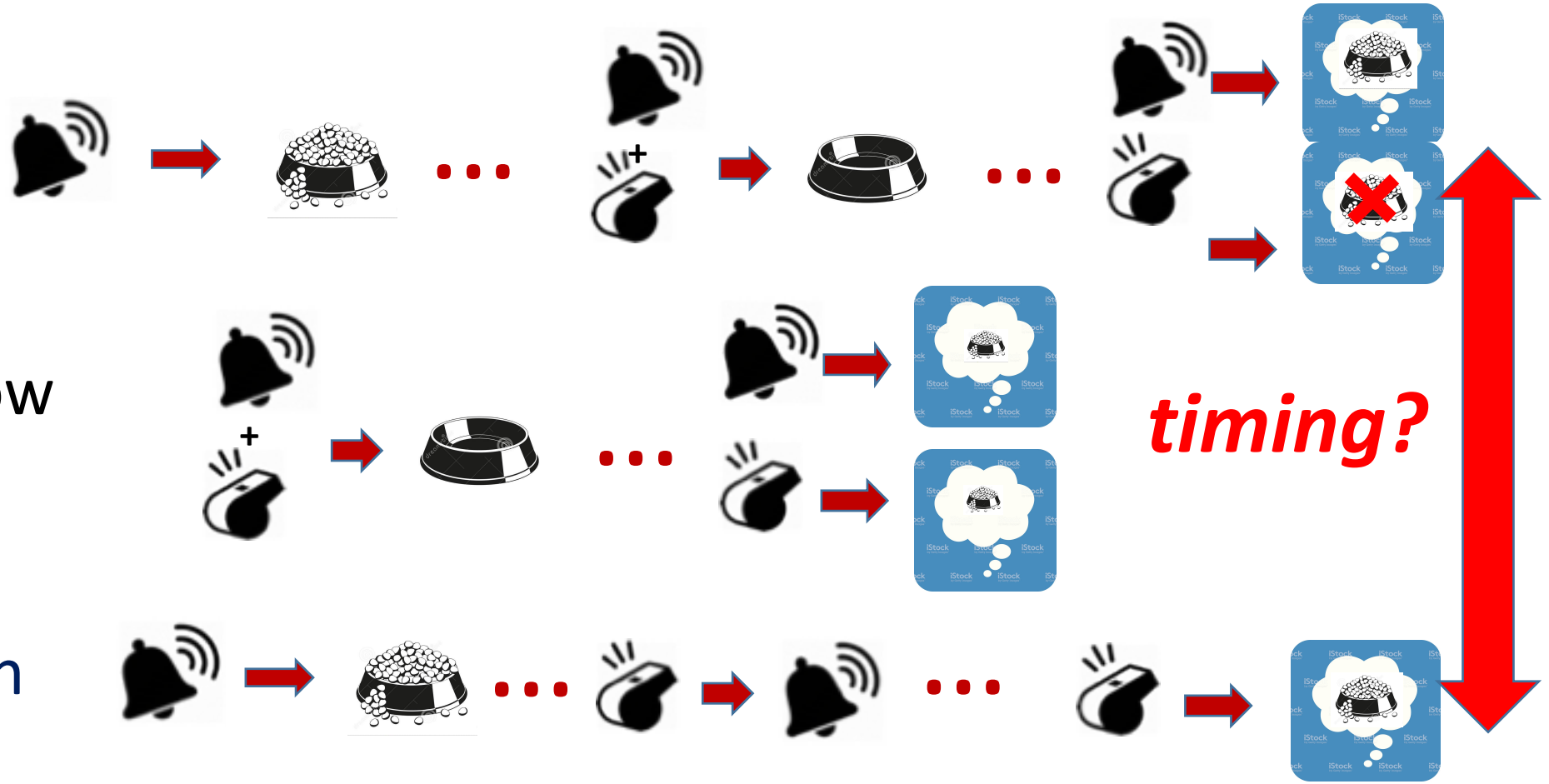
- Aristotelian






# Classical conditioning (cont.)

- Inhibition
- Overshadow
- Aristotelian



# Classical conditioning underestimates the Brain: no foresight

Animals choose actions
by looking beyond
the present reward…

# Reinforcement learning and temporal differences: [Sutton and Barto 1980s]

Stimulus u(t), prediction x(t), weights w(t)

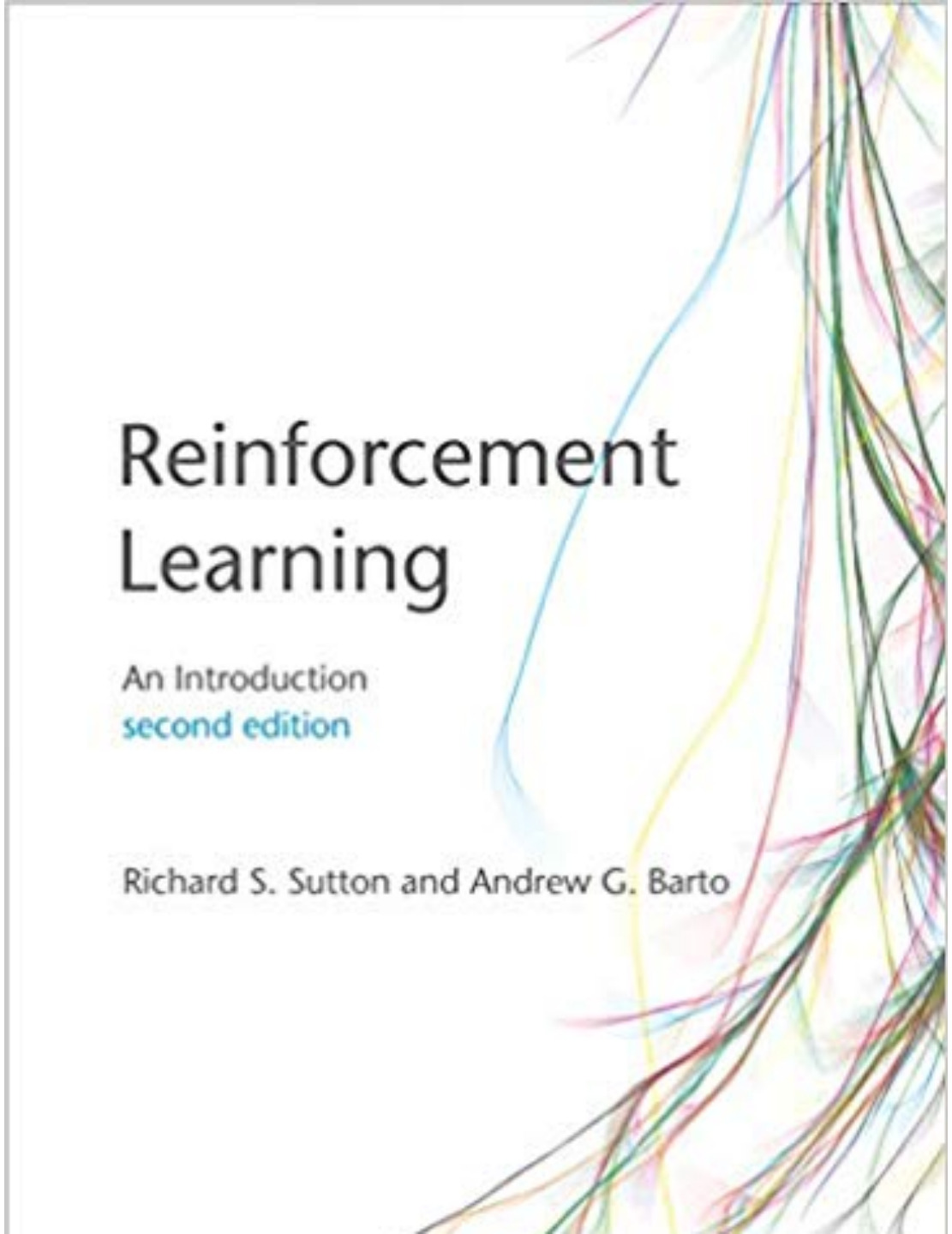
*x: predicted value of all future rewards…*

$$x(t) = u(t) \cdot w(t)$$

$$x(t) = R(t+1) + \gamma \cdot R(t+2) + \gamma^2 \cdot R(t+3) + \gamma^3 \cdot R(t+4) + \ldots$$

*Discount rate* *"Robot, carpe diem"*

# Second edition!



# Familiar algebra:

$\mathbf{x(t+1) = R(t) + \gamma\, x(t)}$

$\mathbf{\delta = R(t) + \gamma\, x(t) - x(t+1)}$ prediction error

(Note: classical conditioning is the $\gamma = 0$ case)

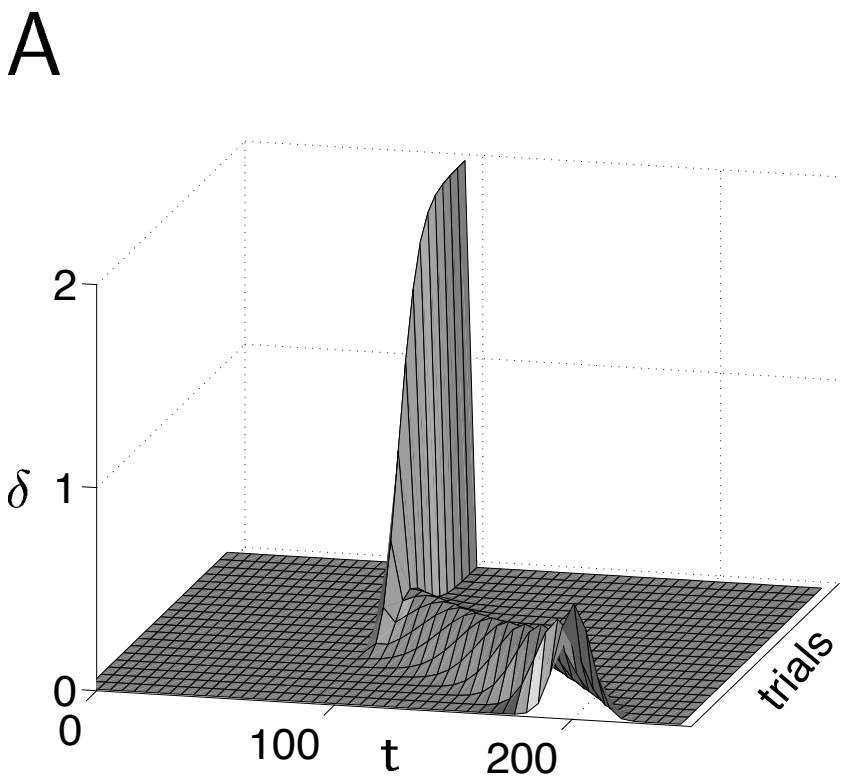
Minimize $\delta^2$ via gradient descent

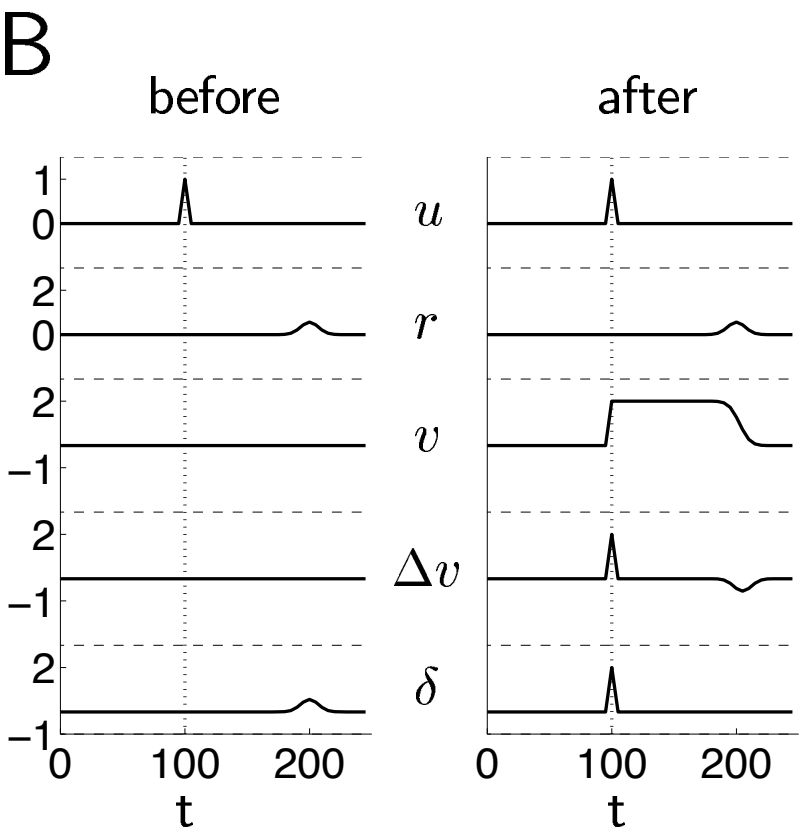
This is the Temporal Difference algorithm

Allows the neural network to predict future rewards…

# It works!

Stimulus at t = 100, reward at t = 200, $\boldsymbol{\gamma} = 1$




u(t)

R(t)

x(t)

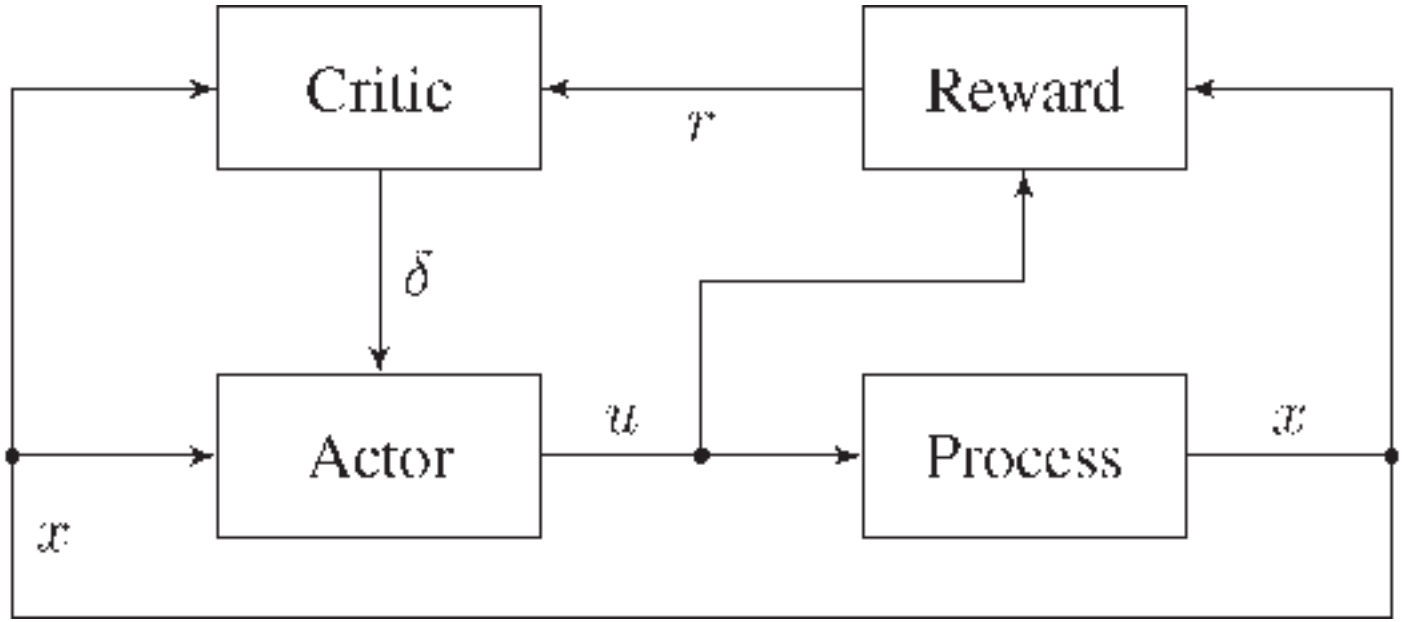
x(t) – x(t-1)

δ

# It works!  [Tesauro 1992] backgammon

before
there was
AlphaGo,
there was
TD-Gammon

# Reinforcement learning:
# The actor – critic formulation



# Reinforcement Learning in the brain

**Spot:**

The δ calculator

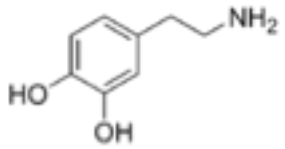
The weights w of the stimuli

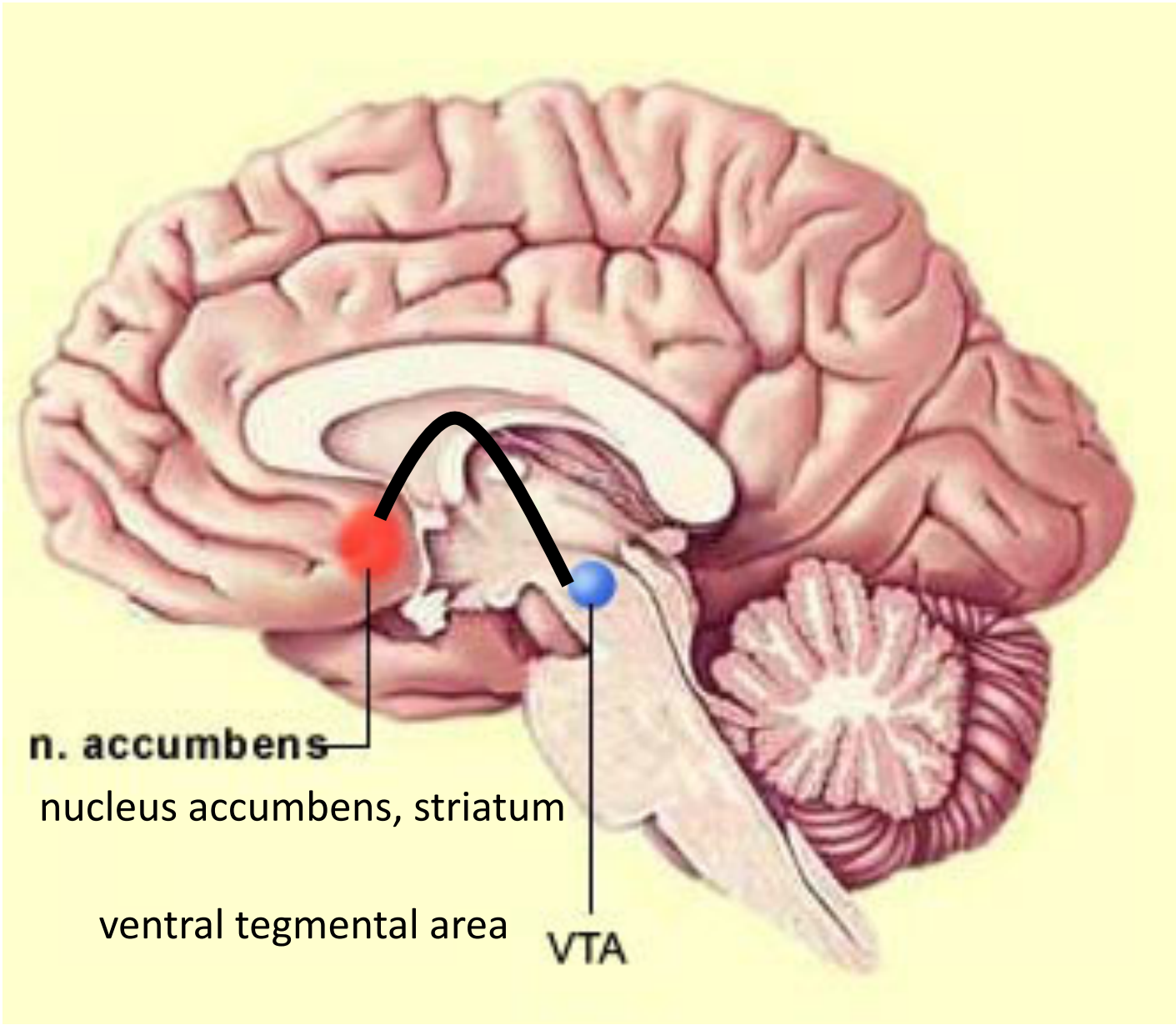
The reward delivery system

The reward circuit

Btw: the reward is

***dopamine***

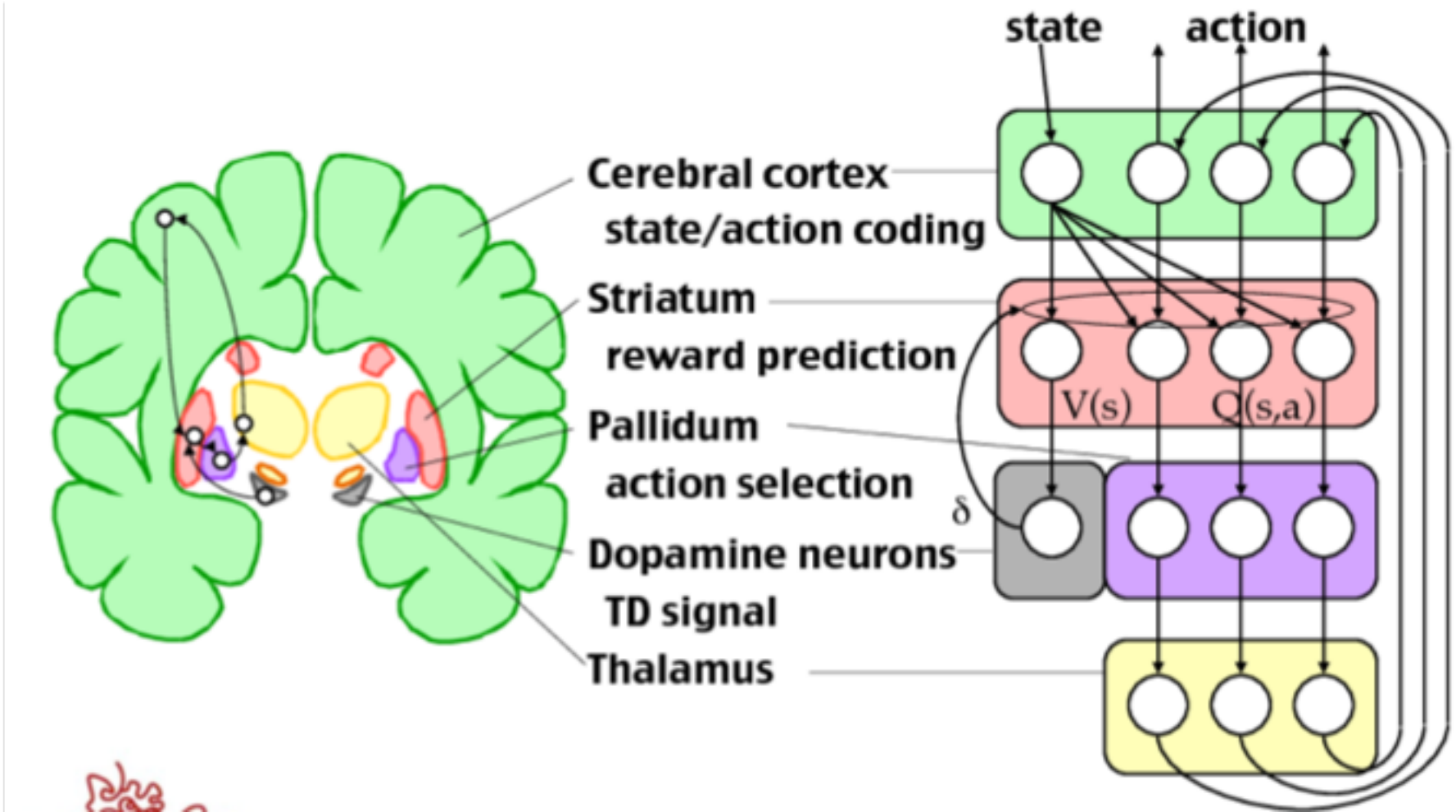


# Btw: is the brain optimizing something?

- Energy?
- The free-energy principle cf: Karl Friston?
- Fitness with the planet and the times?
- Social interaction and success?
- Number of offspring?
- Prediction accuracy?
- ***Dopamine levels!***

# Reinforcement learning in the brain: we are still guessing the details…



# A parenthesis: yet another biologically plausible deep net idea

- Dopamine release seems to affect synaptic weights!
- Yagishita et al, *Science* 2014: Synapses (from the cortex to the striatum) are enhanced if dopamine is released within 0.5 – 2 secs of the synapse's firing

# DENNs: DopaminErgic neural nets

If a link ab fired

ie, if both a and b fired in this minibatch

then its strength is increased by $\varepsilon \cdot (¼ - \text{error}^2)$

- Experiments: performance compares favorably to signSGD
- (and to DNNEvolution)

# RL: Summary

- In Pavlovian learning and its relatives, you learn from the stimuli
- In Reinforcement learning, you learn from your own actions
- In the brain dopamine and basal ganglia, striatum, VTA, nA, etc. seem to play a role in reinforcement learning
- (But we still do not know)
- Next: The math and deep learning of RL

# Important model in RL: multi-armed bandits

m actions a = 1…m

Each has an unknown reward distribution $D_a$

You are stuck at the casino for a very long time T

How would you play?

# Warning: a highly dopaminergic problem

The multi armed bandits problem was formulated during the war, and efforts to solve it so sapped the energies and minds of Allied scientists that the suggestion was made that the problem be dropped over Germany, as the ultimate instrument of intellectual sabotage.
– P. Whittle

# Exploration vs. exploitation

- One machine has the best expectation, $M^*$
- Best thing to do is play this forever – but you don't know which…
- So, at time t you choose A(t)
- Every machine i has a (unknown) gap $G_a = M^* - M_a$
- You want to minimize regret: Regret(A) = $\Sigma_t G_{A(t)}$

# Exploration vs. exploitation

- You somehow start trying machines, evaluating their mean by sampling
- Greedy: Always stick to the best estimate, never explore → regret is linear in T
- Noisy greedy: 10% of the time try a new machine, otherwise greedy → regret still linear in T

# Exploration vs. exploitation: is linear regret inevitable?

- Lower bound: regret is at least log T · $(\Sigma_a G_a/KL(a, a^*))$

# Log regret is possible!

- Natural idea: if an action has good sample mean but few samples, punish it a little.
- But how much?
- Recall that the probability that sample mean is b away from true mean is about

$$\exp(-\text{samplesize} \cdot b^2)$$ (Hoeffding bound)

# Algorithm UCB1 [Auer, Cesa-Bianchi, and Fischer 2002]

Estimate the expectation of every action as

current mean + $\sqrt{(\log t/\text{sample size})}$,

- Note: the added term encourages exploration of poorly sampled actions

- **Theorem**: UCB1 has regret at most $10 \log T \cdot (\Sigma_a G_a^{-1})$

# An even better algorithm: Posterior sampling

- Maintain a parametrized model (perhaps a deep net…) of the reward distributions: $p(r \mid a) = F(r, \theta_a)$
- We sample rewards according to the current parameters
- Treat this sample as the truth and pick the best action Update the parameters ($\theta_1$, $\theta_2$, $\theta_3$, …) according to the result
- This algorithm (Thompson's algorithm ***1933***) is not only logarithmic, but it achieves the lower bound [Agrawal 2011]

# Model with prior: time-dependent bandits

Suppose now the machines are known Markov chains with a reward at each transition

We know the current states

Discounted rewards

# Gittins index of a Markov chain

- Suppose you had a choice between a Markov chain M and the one-state Markov chain R (= reward)
- You are given the option to play M for a while and then R forever. For how long would you play M?
- Obviously, the larger R, the shorter.
- For large enough R, you will not touch M
- Gittins index of M: the smallest R for which you would not touch M

# Gittins index of a Markov chain

**Gittins Theorem:** The best thing to do is always play the machine with the highest Gittins index

# Markov decision processes (MDPs)

- Gittins allows you to switch Markov chains
- In MDPs the Markov chains all share the same states
- You have a choice of action at each state
- Your choice changes the transition probabilities from this state, and the reward distribution of this state
- Strategy A: You choose, once and for all, what to do at each state, and then follow the resulting Markov chain
- What is the best strategy?

# Example: the taxicab problem (From Ron Howard's book)

A taxi serves three adjacent towns: A, B, and C.

Each time the taxi discharges a passenger, the driver must

choose from three possible actions:

(1) "Cruise" the streets looking for a passenger.
(2) Go to the nearest taxi stand (hotel, train station, etc.)
(3) Wait for a radio call from the dispatcher with instructions (but not possible in town B because of distance and poor reception).

**MDP model:**

**States**: {A, B, C}

**Action sets**:

$K_A = \{1,2,3\}$, $K_B = \{1,2,3\}$, $K_C = \{1,2\}$

**Transition probability matrices**

| Cruising streets | Waiting at taxi stand | Waiting for dispatch |
|---|---|---|
| $P^1 = \begin{bmatrix} 1/2 & 1/4 & 1/4 \\ 1/2 & 0 & 1/2 \\ 1/4 & 1/4 & 1/2 \end{bmatrix}$ | $P^2 = \begin{bmatrix} 1/16 & 3/4 & 3/16 \\ 1/16 & 7/8 & 1/16 \\ 1/8 & 3/4 & 1/8 \end{bmatrix}$ | $P^3 = \begin{bmatrix} 1/4 & 1/8 & 5/8 \\ 0 & 1 & 0 \\ 3/4 & 1/16 & 3/16 \end{bmatrix}$ |

$$R^1 = \begin{bmatrix} 10 & 4 & 8 \\ 14 & 0 & 18 \\ 10 & 2 & 8 \end{bmatrix} \qquad R^2 = \begin{bmatrix} 8 & 2 & 4 \\ 8 & 16 & 8 \\ 6 & 4 & 2 \end{bmatrix} \qquad R^3 = \begin{bmatrix} 4 & 6 & 4 \\ 0 & 0 & 0 \\ 4 & 0 & 8 \end{bmatrix}$$

Rewards

Problem: What is the best action at each city?

# Value of a state: Bellman's equation

$$\mathbf{V[state] = \max_A [R(state,A) + \gamma \cdot E_A [V[next\ state])}$$

Note:

It's a linear program…

***And*** you can iterate it

# Another algorithm: Policy iteration

Start with any policy A

Calculate the values of each state

Policy improvement: update A using one-step look-ahead

(Can converge faster)

# So, what's the problem?

- Problem is, Chess has $10^{50}$ states, Go has $10^{170}$ , while an automated driver may have more…
- Policy A is a parametrized function of the state
- Action at state s is $A(s,\theta)$
- E.g., parameter $\theta$ is the set of weights in trained Alpha Go
- $J(\theta)$ = the expected reward of $A(s,\theta)$

# How do you maximize J(θ)?

- By policy iteration
- Which means, stochastic gradient ascent: calculate ∇θ by sampling paths of the Markov chain (roll-outs)
- Problems:

  Large variance, many samples needed

  Fixed policy, samples may be repeating

  Slow convergence, small learning constant

# So, what to do?

Large variance, many samples needed

Solution: subtract from total reward of each sample a baseline equal to an estimate of the reward

Alternative: use value iteration…

Fixed policy, samples may be repeating

Use two different policies, one for evaluation and one for sampling paths

Slow convergence, small learning constant

More tricks