Learning in Neural Networks: Three-factor learning rules

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Three-factor Learning Rules

- Reward-based learning needs three-factor learning rules
- 3-factor rules vs. 2-factor rules
- Neuromodulators act as 3rd factor
- Experiments supporting three-factor learning rules

Previous slide.

Since Hebbian learning rules are limited, we have to extend the framework and include a ‘third factor’ that could represent reward.

For most of the RL part, I also have videos on this page:
https://lcnwww.epfl.ch/gerstner/VideoLecturesRL-Gerstner.html

Video for this first section:

https://www.youtube.com/watch?v=jGj2sTdQLME
Which is part 4 of lecture:
Reinforcement Learning and the Brain: **3-factor rules and brain-style computing**
on
https://lcnwww.epfl.ch/gerstner/VideoLecturesRL-Gerstner.html

Review: Hebbian rules

Hebbian coactivation:
pre-post-post-post

Hebbian coactivation:
but no post-spikes

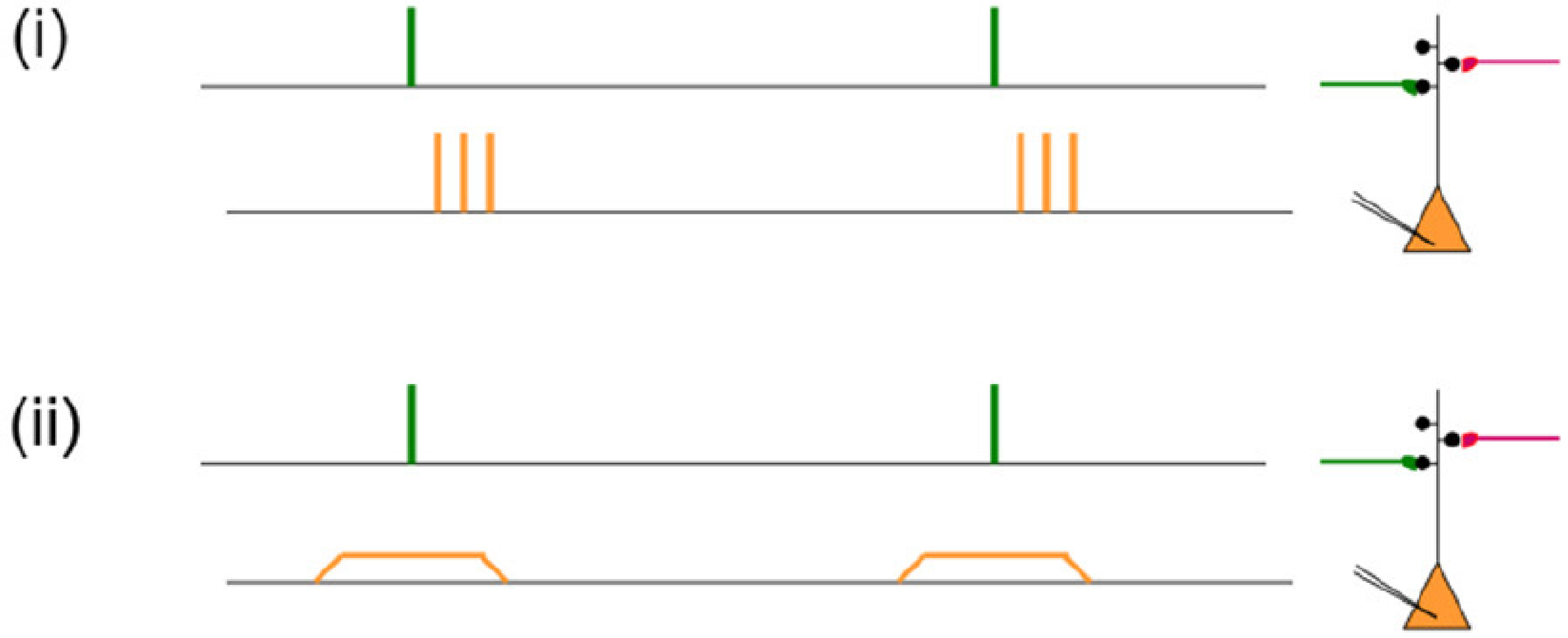


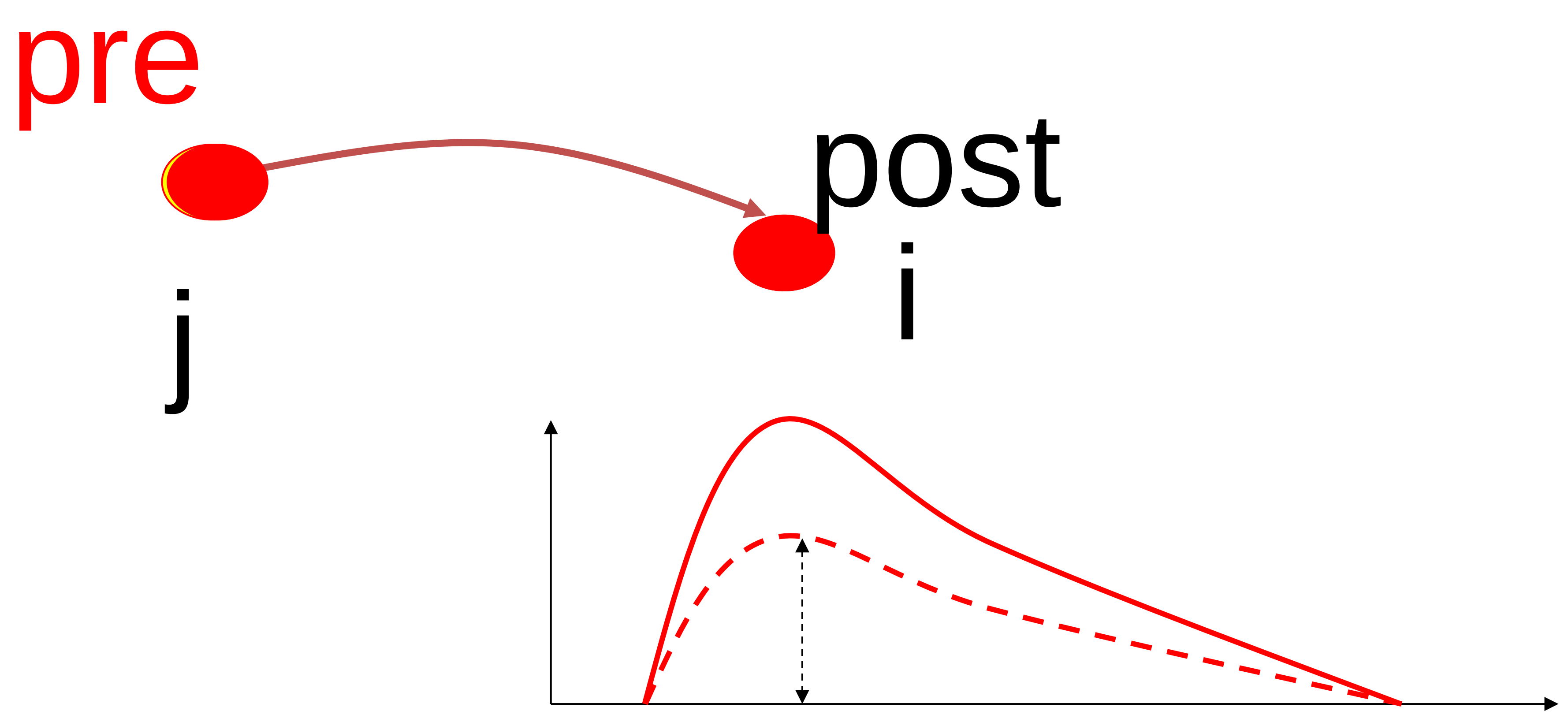
Image: Gerstner et al. (2018, review paper in Frontiers)

Previous slide.

Review: Hebb rules, but Hebbian learning rules are limited

Review: Hebb rule of synaptic changes

Hebbian Learning
= unsupervised learning



no notion of reward
or success.

$$\Delta w_{ij} = F\left(pre, post, w_{ij}\right)$$

Previous slide.

In standard Hebbian learning, the change of the synaptic weight depends only on presynaptic activity (pre) and the state of the postsynaptic neuron (post). The rule is local, and does not contain the notion of reward or success.

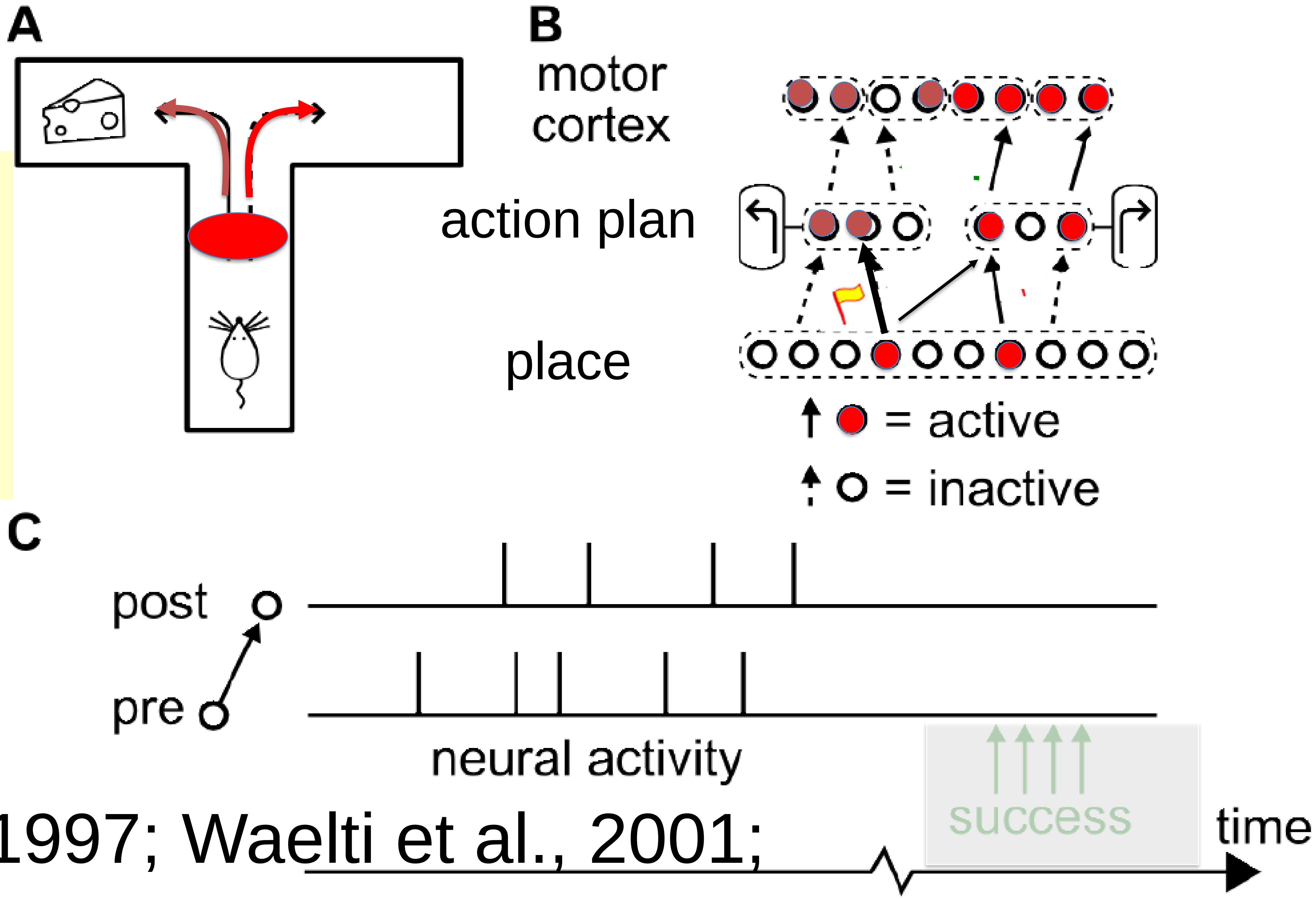
The value of the weight w_{ij} is measured by sending a test-pulse across the synapse. The change of the weight is a function of ‘pre’ and ‘post’ and the weight itself where ‘pre’ and ‘post’ are rather general variables.

Is Hebbian Learning sufficient? No! – We need a third factor!

Image: Fremaux and Gerstner, Front. Neur. Circ., 2016

Eligibility trace:
Synapse keeps memory of pre-post coincidences over a few seconds

Dopamine:
Reward/success



Schultz et al. 1997; Waelti et al., 2001;

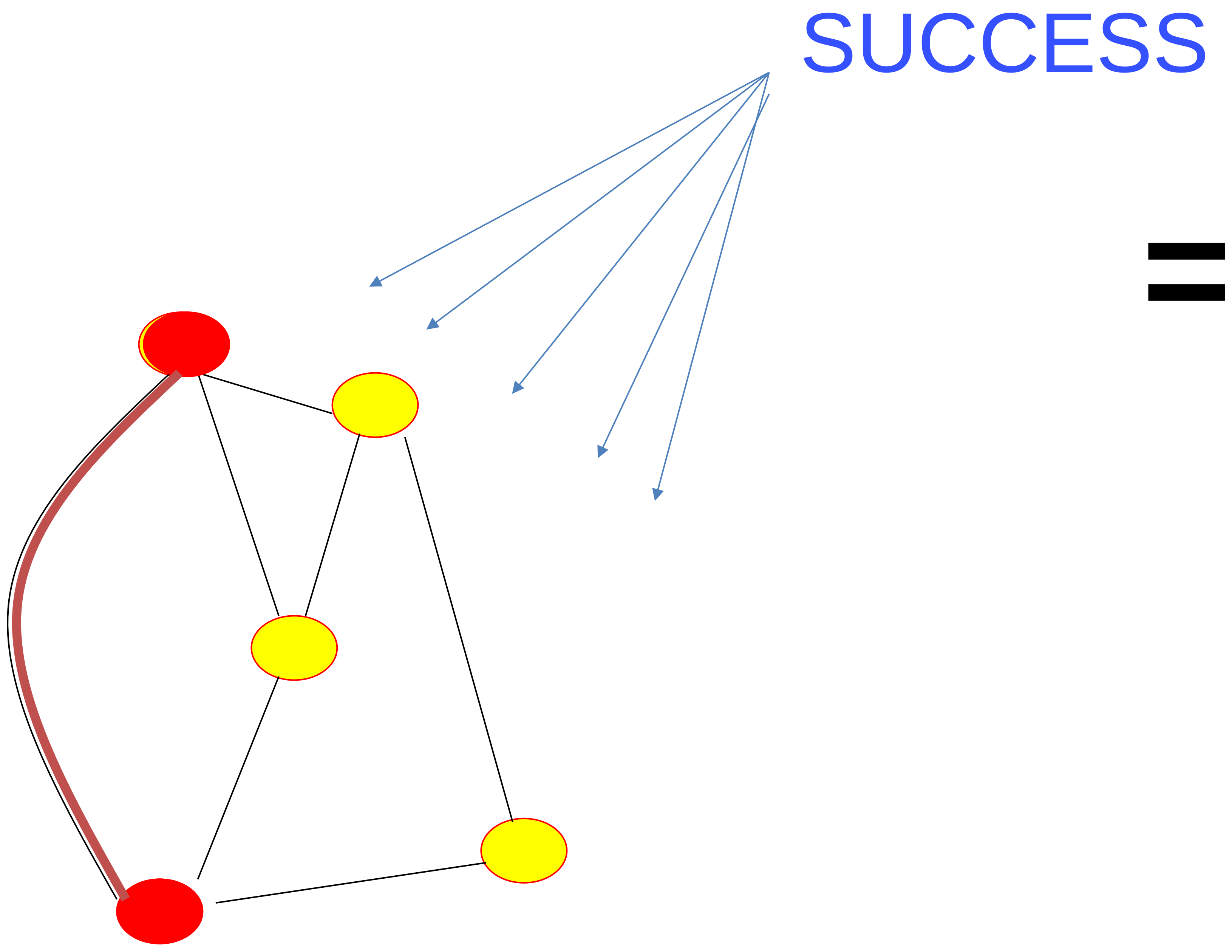
→ **Reinforcement learning: success = reward – (expected reward)**

TD-learning, SARSA, Policy gradient (book: Sutton and Barto, 2018)

Previous slide.
Hebbian learning as it stands is not sufficient to describe learning in a setting were rewards play a role. If joint activity of pre- and post causes stronger synapses, the rat is likely to repeat the same unrewarded action a second time. A three-factor rule adds the influence of a neuromodulator (e.g., dopamine): reward-modulate plasticity.

Hypothetical functional role of neuromodulated synaptic plasticity.
(A) Schematic reward-based learning experiment. An animal learns to perform a desired sequence of actions (e.g.,move straight,then turn left) in a T-maze through trial-and-error with rewards (cheese).
(B) The current position ("place") of the animal in the environment is represented by an assembly of active cells in the hippocampus.These cells connect to neurons (e.g.,in the dorsal striatum) which code for high-level actions at the decision point, e.g., "turn left" or "turn right." These neurons in turn project to motorcortex neurons, responsible for the detailed implementation of actions. Connections between neurons that are active together are marked (flag/eligibility trace).
(C) Neuromodulator timing. While spikes occur on the time scale of milliseconds, the success signal (green arrows/shaded) may come a few seconds later.

Classification of synaptic changes: Reinforcement Learning



Reinforcement Learning

= reward + Hebb

$$\Delta w_{ij} \propto F(pre, post, MOD; w_{ij})$$

local global

broadly diffused signal:
neuromodulator
(e.g., success)

Previous slide.

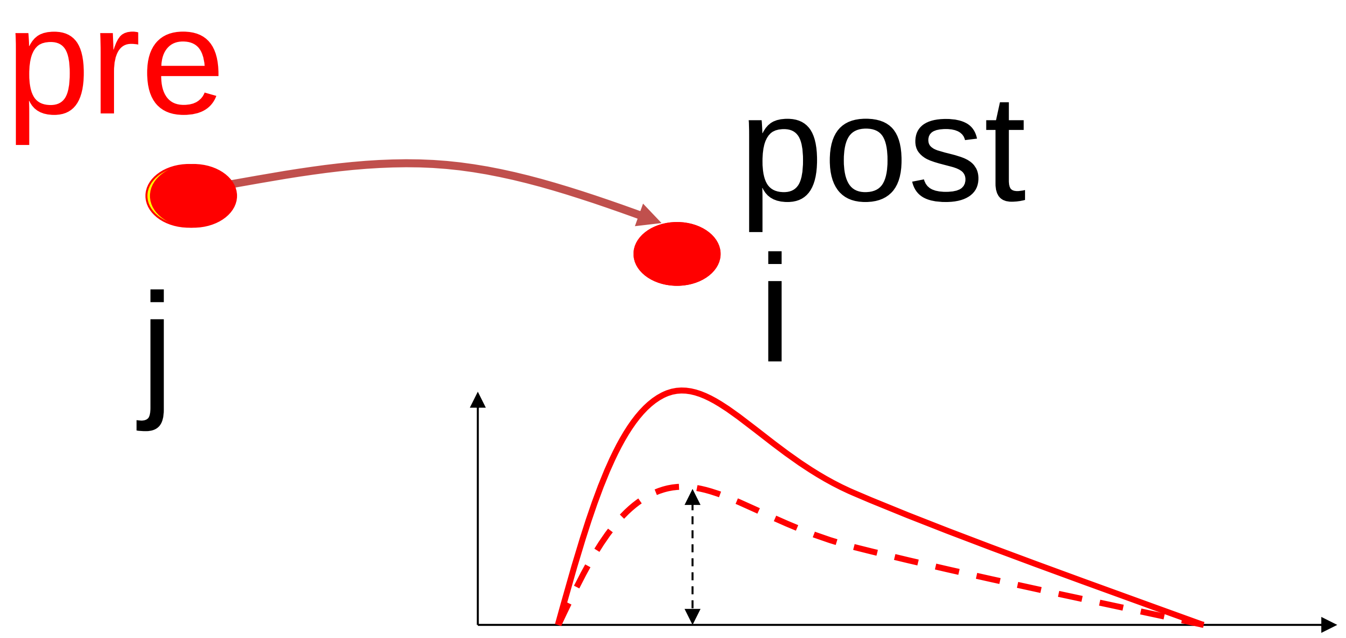
For the moment we say that reinforcement learning depends on three factors: the Hebbian pre- and postsynaptic factor plus a success signal related to reward.
We will get more precise later.

Classification of synaptic changes

unsupervised vs reinforcement

LTP/LTD/Hebb
Theoretical concept

- passive changes
- exploit statistical correlations

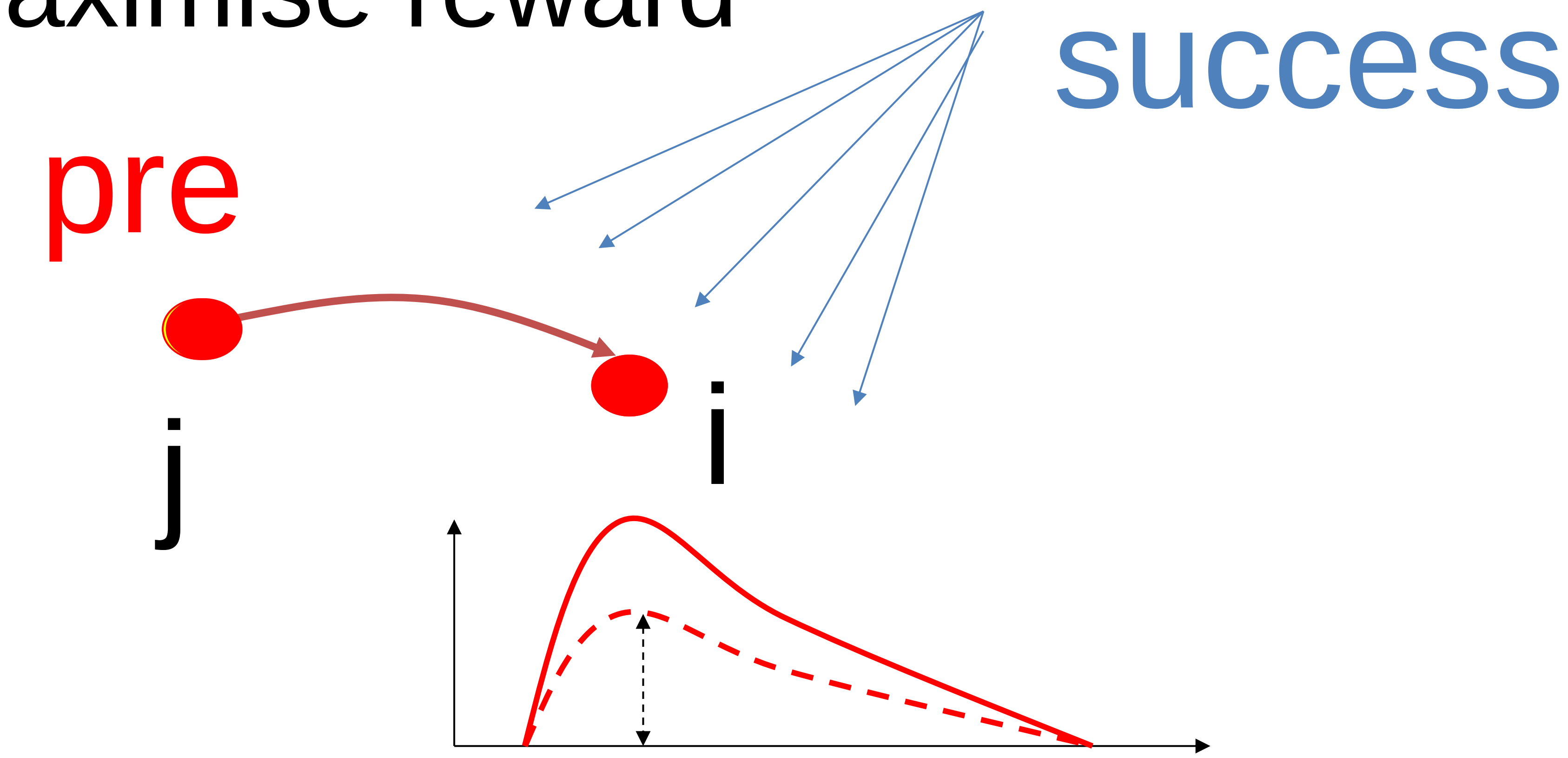


Functionality

-useful for development
(develop good filters)

Reinforcement Learning
Theoretical concept

- conditioned changes
- maximise reward



Functionality

- useful for learning
a new behavior

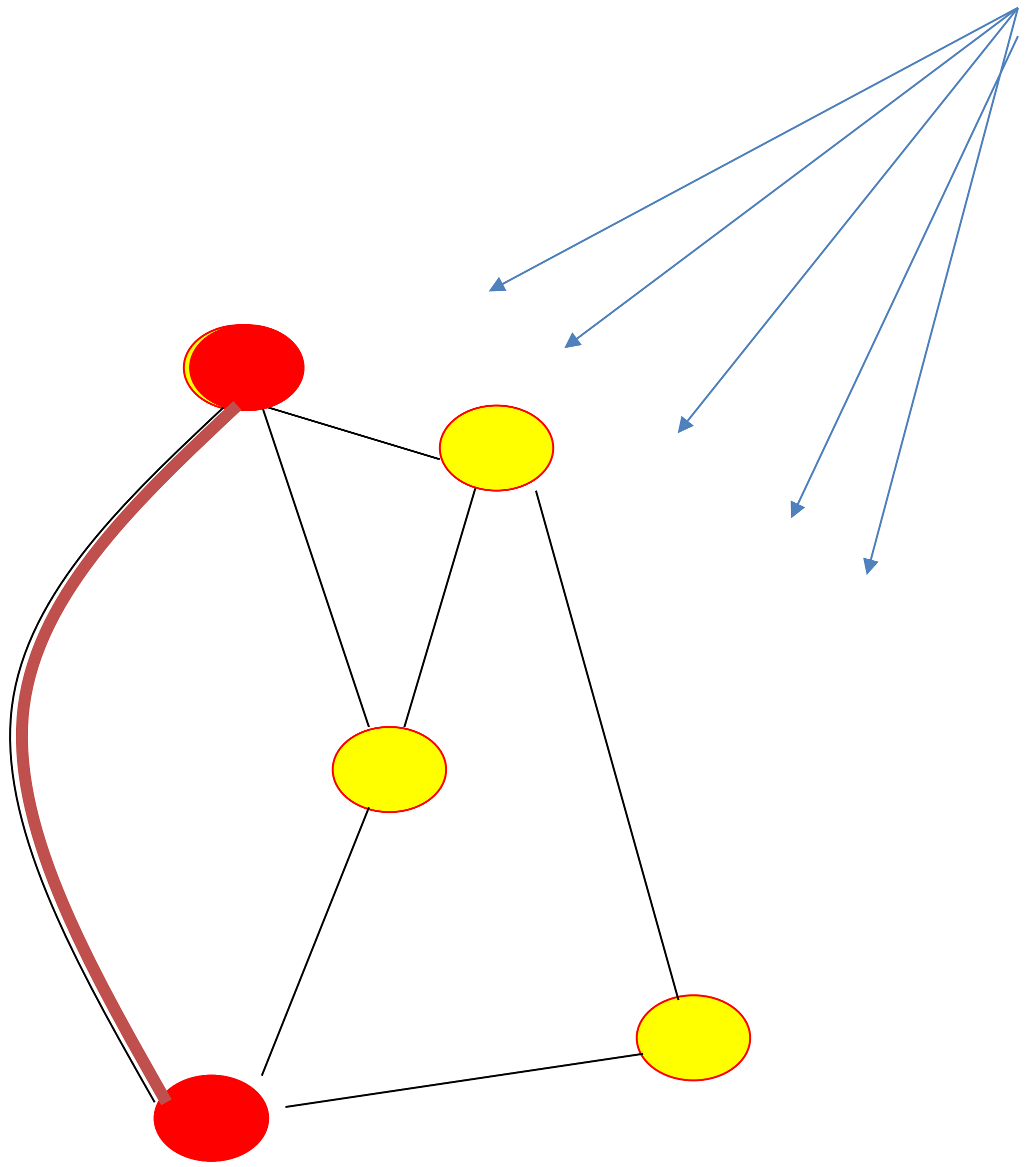
Previous slide.

This does not mean the standard Hebbian learning is wrong: in fact it is very useful for the development of generic synaptic connections, e.g., to make neurons develop good filtering properties that pick up relevant statistical signals in the stream of input. Unsupervised Hebbian learning can for example implement Principal Component Analysis or Independent Component Analysis.

The three-factor rules are relevant for learning novel behaviors via feedback through reward.

Three-factor rule: the role of neuromodulators

= Hebb-rule gated by a neuromodulator



Neuromodulators: Interestingness, surprise; attention; novelty

$$\Delta w_{ij} \propto F(pre, post, MOD; w_{ij})$$

local global

Previous slide.

To summarized: The three-factor rules have a Hebbian component: pre- and postsynaptic activity together, but in addition the third factor which is related to neuromodulators.

There are several neuromodulators in the brain.

Neuromodulator projections

- 4 or 5 neuromodulators
- near-global action

Dopamine/reward/TD:
Schultz et al., 1997,
Schultz, 2002

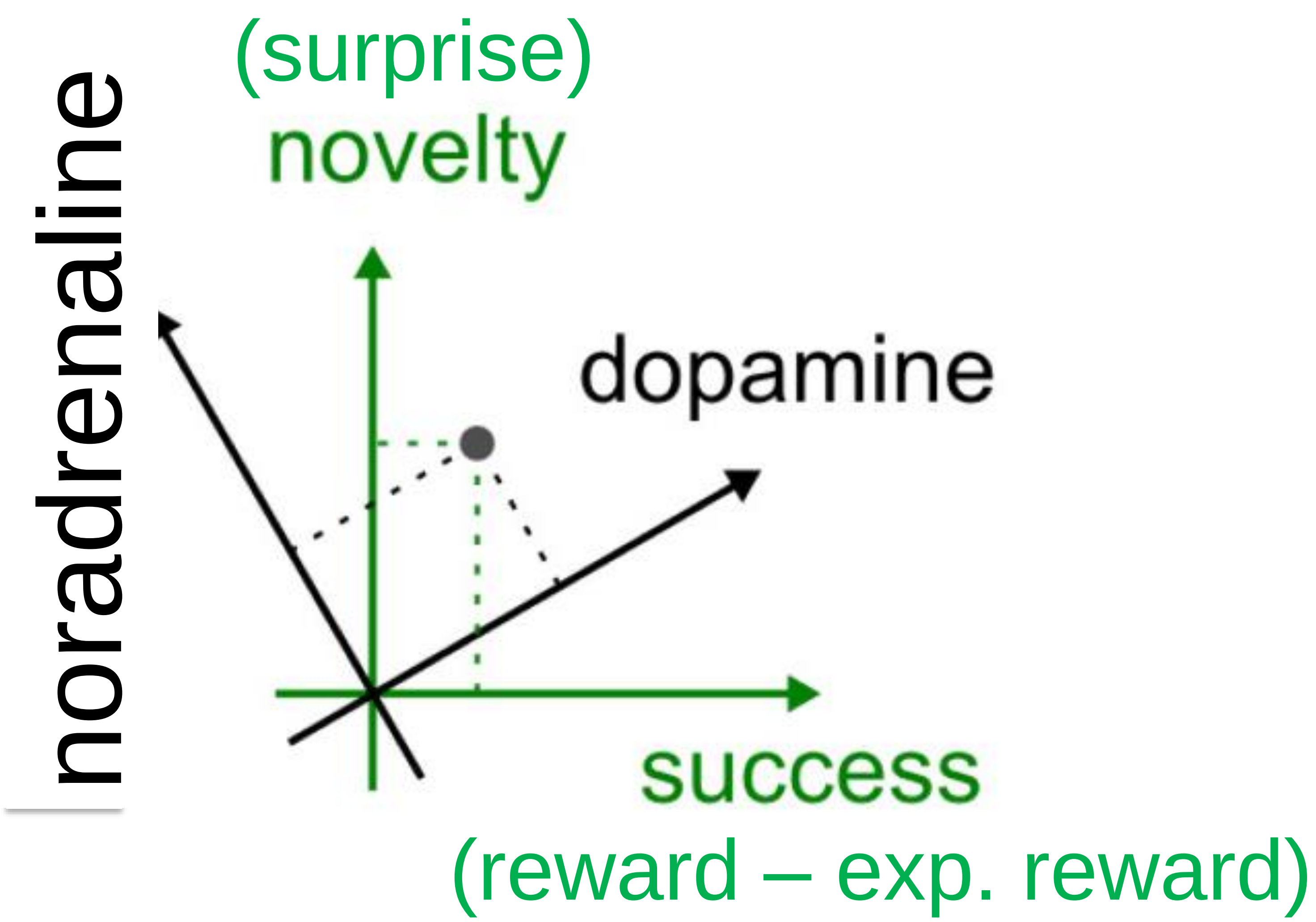
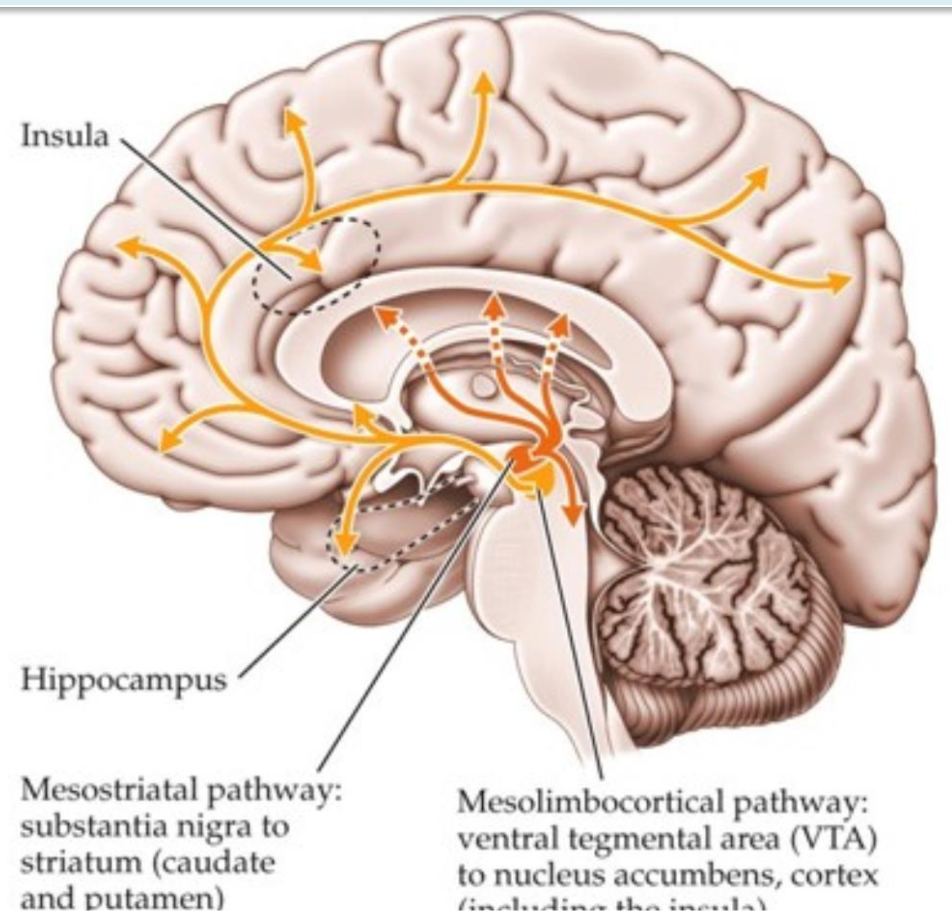


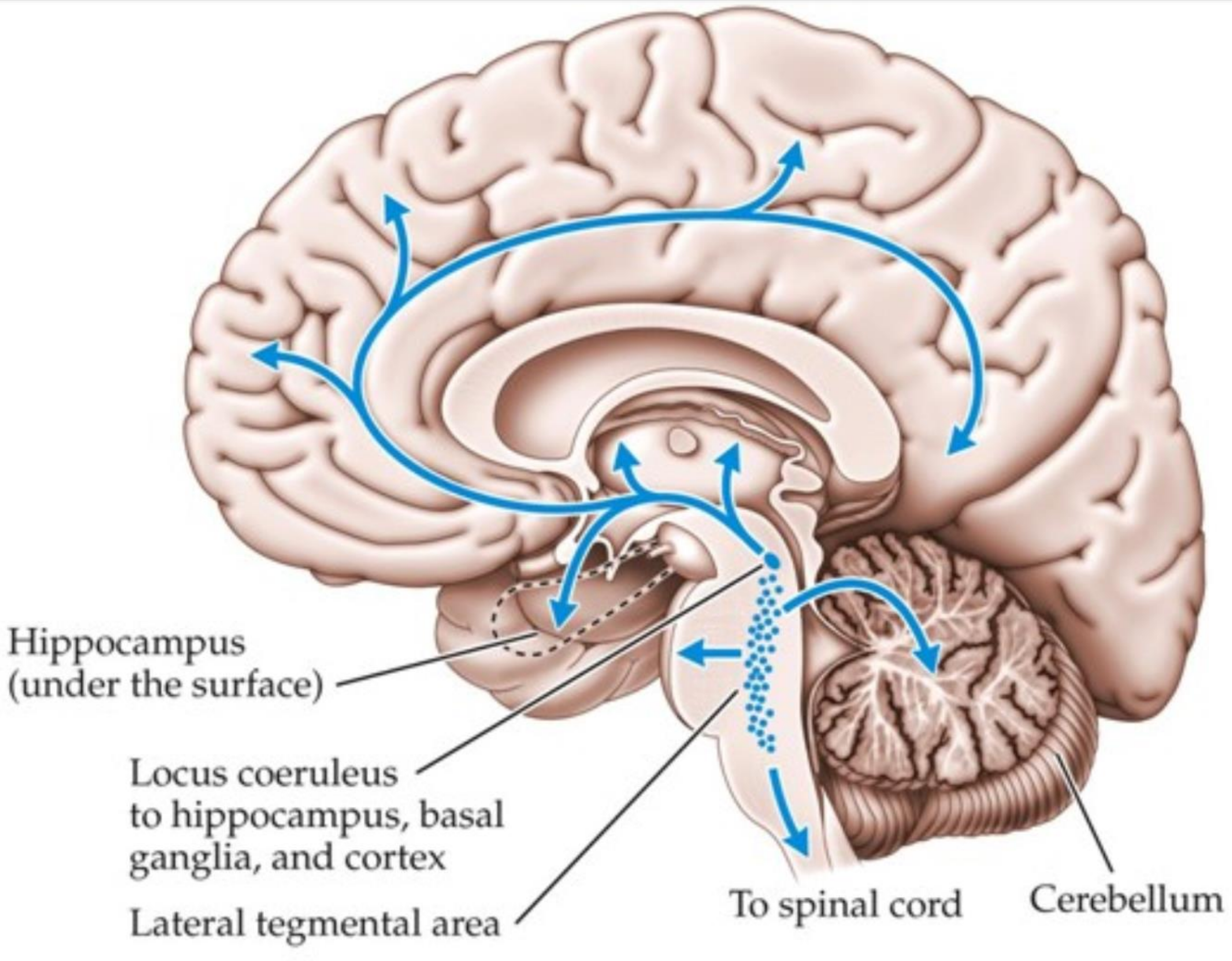
Image:
Fremaux and Gerstner, Frontiers (2016)

Image: Biological Psychology, Sinauer

Dopamine (DA)


Insula
Hippocampus
Mesostriatal pathway: substantia nigra to striatum (caudate and putamen)
Mesolimbocortical pathway: ventral tegmental area (VTA) to nucleus accumbens, cortex (including the insula)

Noradrenaline (NE)


Hippocampus (under the surface)
Locus coeruleus to hippocampus, basal ganglia, and cortex
Lateral tegmental area
To spinal cord
Cerebellum

BIOLOGICAL PSYCHOLOGY 7e, Figure 4.5

Previous slide.

The most famous neuromodulator is dopamine (DA) which is related to reward, as we will see.

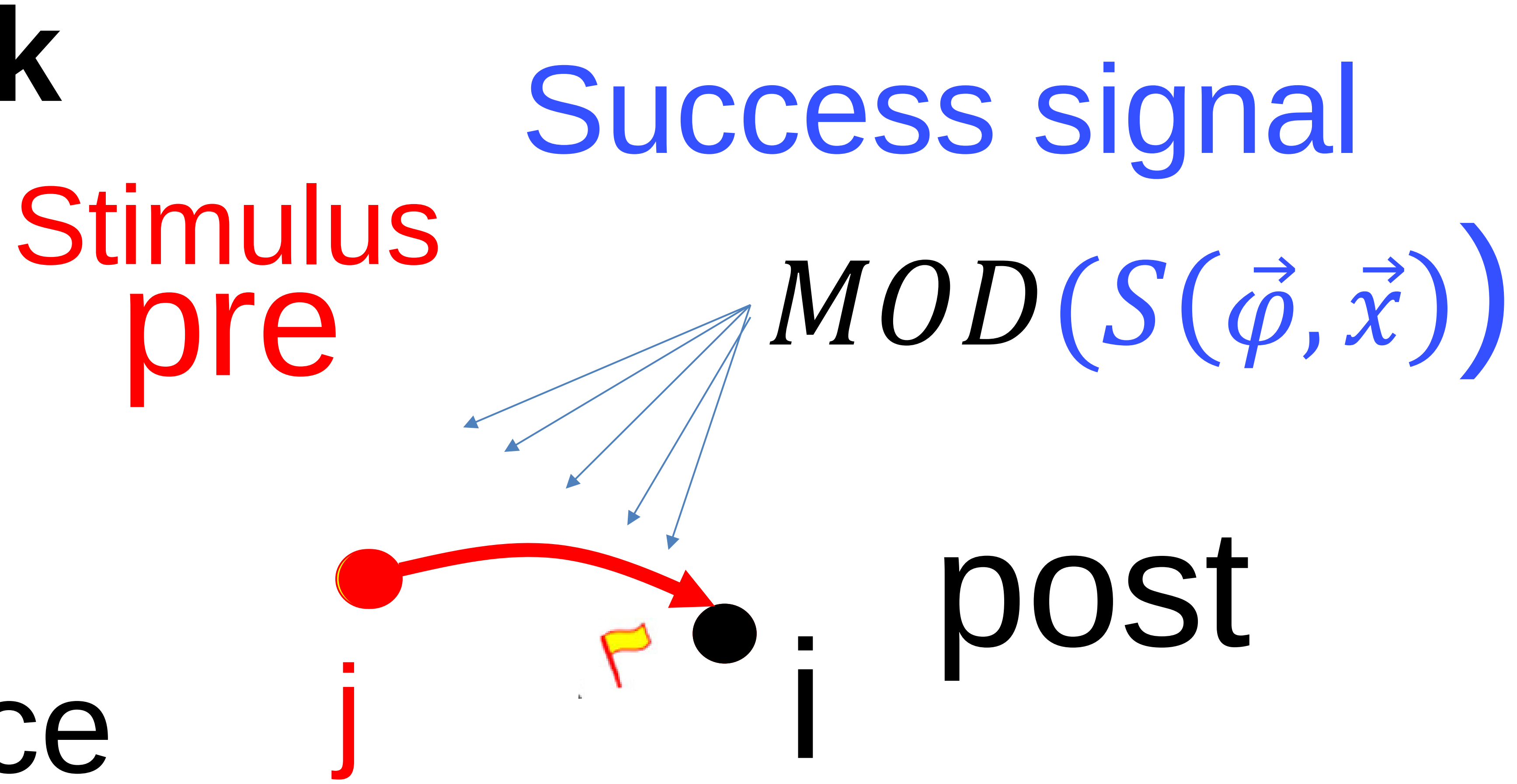
But there are other neuromodulators such as noradrenaline (also called norepinephrine, NE) which is related to surprise.

Left: the mapping between neuromodulators and functions is not one-to-one. Indeed, dopamine also has a ‘surprise’ component.

Right: most neuromodulators send axons to large areas of the brain, in particular to several cortical areas. The axons branch out in thousands of branches.
Thus the information transmitted by a neuromodulator arrives nearly everywhere.
In this sense, it is a ‘global’ signal, available in nearly all brain areas.

Formalism of Three-factor rules with eligibility trace

Three-factor rule defines a framework



x_j = activity of presynaptic neuron

φ_i = activity of postsynaptic neuron

Step 1: co-activation sets eligibility trace

$$\Delta z_{ij} = \eta \ f(\varphi_i)\, g(x_j)$$

Step 2: eligibility trace decays over time

$$z_{ij} \leftarrow \lambda \ z_{ij}$$

Step 3: eligibility trace translated into weight change

$$\Delta w_{ij} = \eta\, MOD\big(S(\vec{\varphi},\vec{x})\big) \cdot z_{ij}$$

Previous slide. Why this is a good algo will become clear in a few weeks!
Three-factor rules are implementable with eligibility traces.

1. The joint activation of pre- and postsynaptic neuron sets a ‘flag’. This step is similar to the Hebb-rule, but the change of the synapse is not yet implemented. The exact condition for setting the eligibility trace COULD be the one from the actor-critic/policy gradient framework, but could also be some other combination of pre-and postsynaptic factors.

2. The eligibility trace decays over time

3. However, if a neuromodulatory signal M arrives before the eligibility trace has decayed to zero, an actual change of the weight is implemented.
The change is proportional to

- the momentary value of the eligibility trace
- the value of the success signal

The success signal can be broadcasted by a neuromodulator signaling

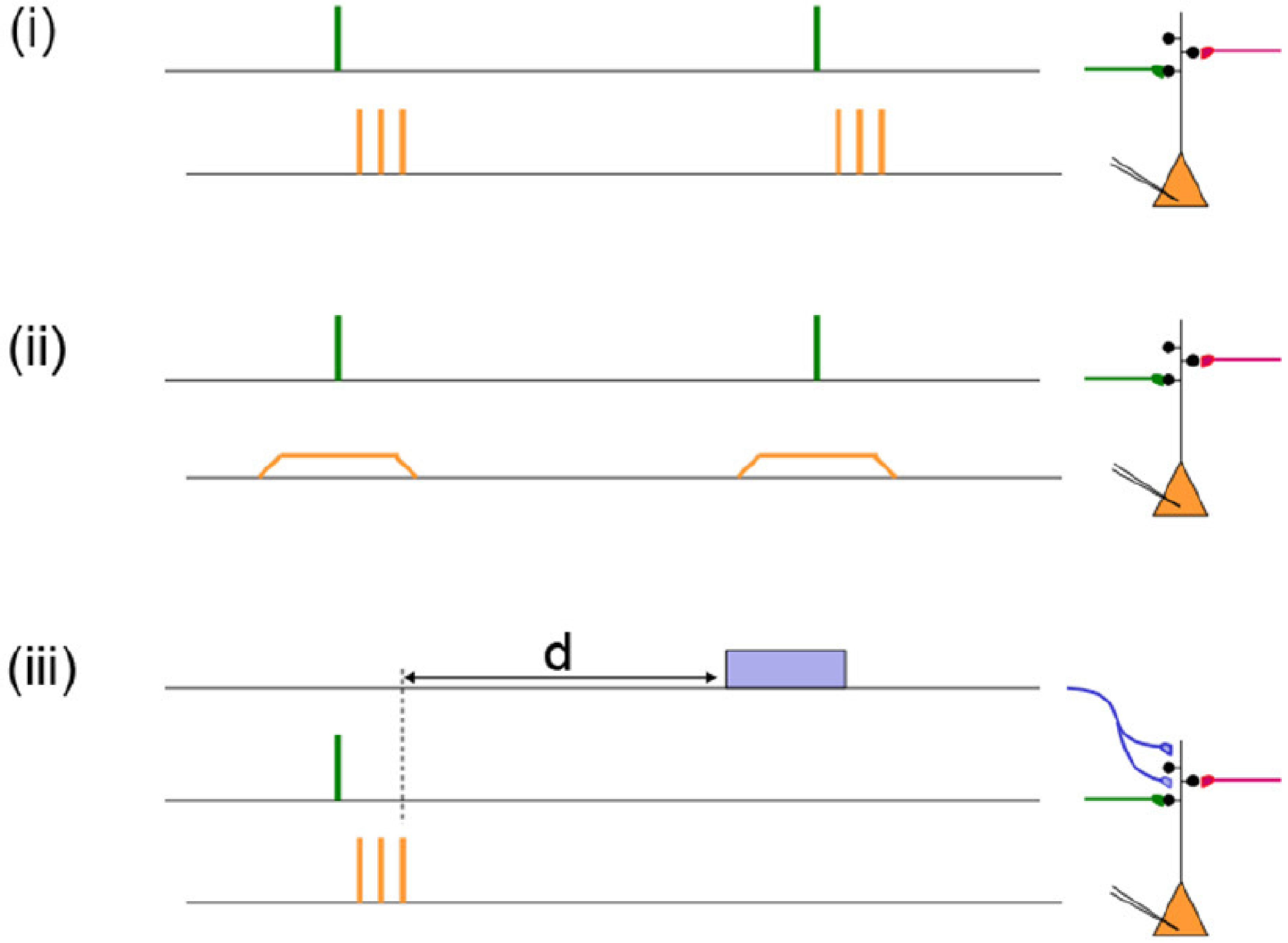
- Reward (minus reward-baseline) OR
- TD-error

Hebbian rules versus Three-factor rules

Hebbian coactivation:
pre-post-post-post

Hebbian coactivation:
but no post-spikes

Scenario of three-factor rule: Hebb+modulator



Neuromodulator can come with a delay of 1s

Image: Gerstner et al. (2018, review paper in Frontiers)

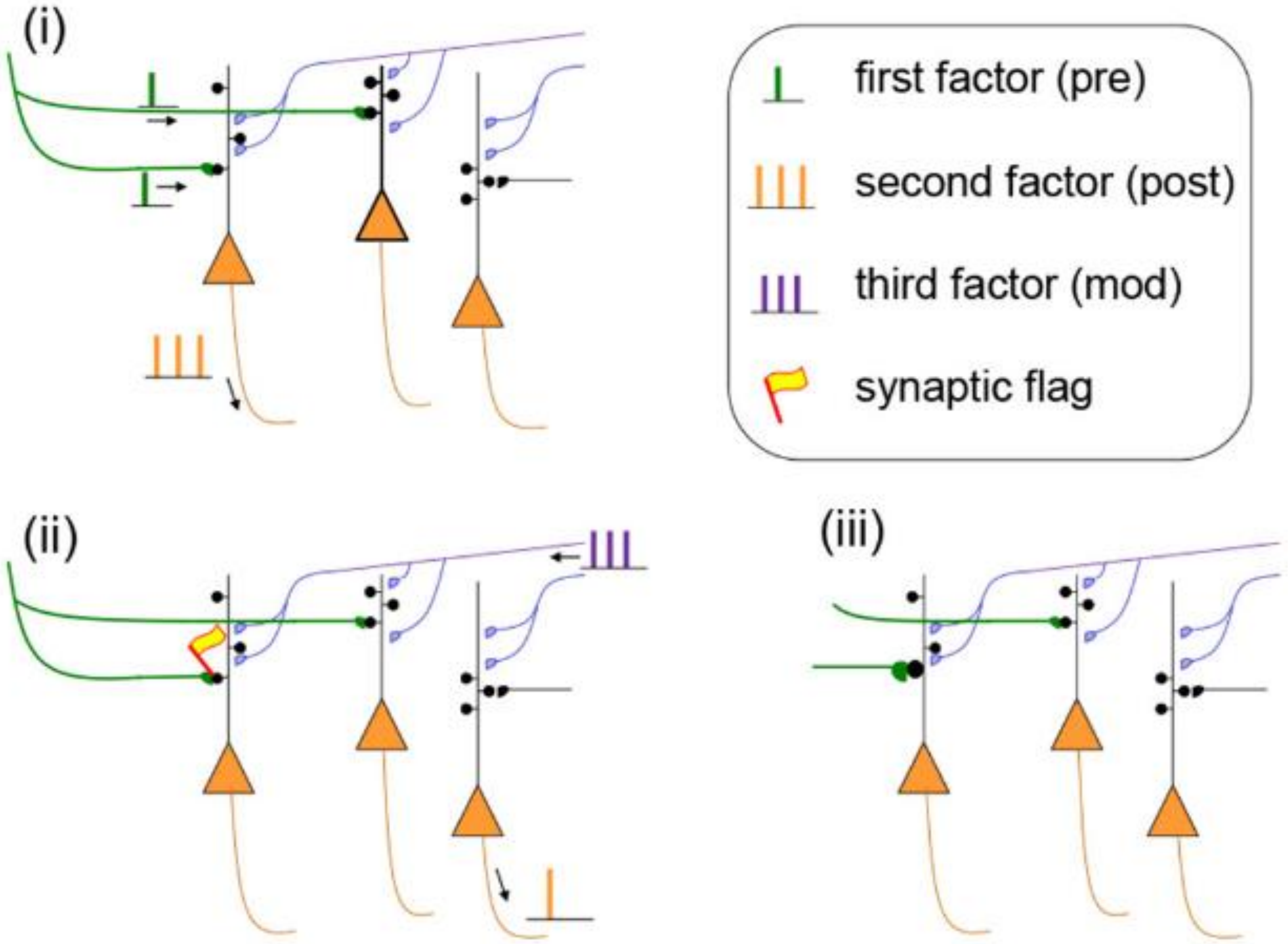
Previous slide.

The joint activation of pre- and postsynaptic neuron sets a ‘flag’. This step is similar to the Hebb-rule, but the change of the synapse is not yet implemented.
Note that joint activation can imply spikes of pre- (green) and postsynaptic (orange) neuron (top);
Or spikes of a presynaptic neuron combined with a weak voltage increase in the postsynaptic neuron (middle).

Bottom: three-factor rule only if a neuromodulatory signal M arrives before the eligibility trace has decayed to zero, an actual change of the weight is implemented. The neuromodulater arrives through the branches

The ideas of three-factor rules can be traced back over several decades.
Early papers were Crow 1968, Barto, 1983/1985, Schultz 1997,
First experimental papers Schultz 1997

Three-factor rules: synaptic flags and delayed reward (mod)



synaptic flag plays role of eligibility trace

Fig: Gerstner et al. 2018, Frontiers

Previous slide.

Specificity of three-factor learning rules.

(i) Presynaptic input spikes (green) arrive at two different neurons, but only one of these also shows postsynaptic activity (orange spikes).

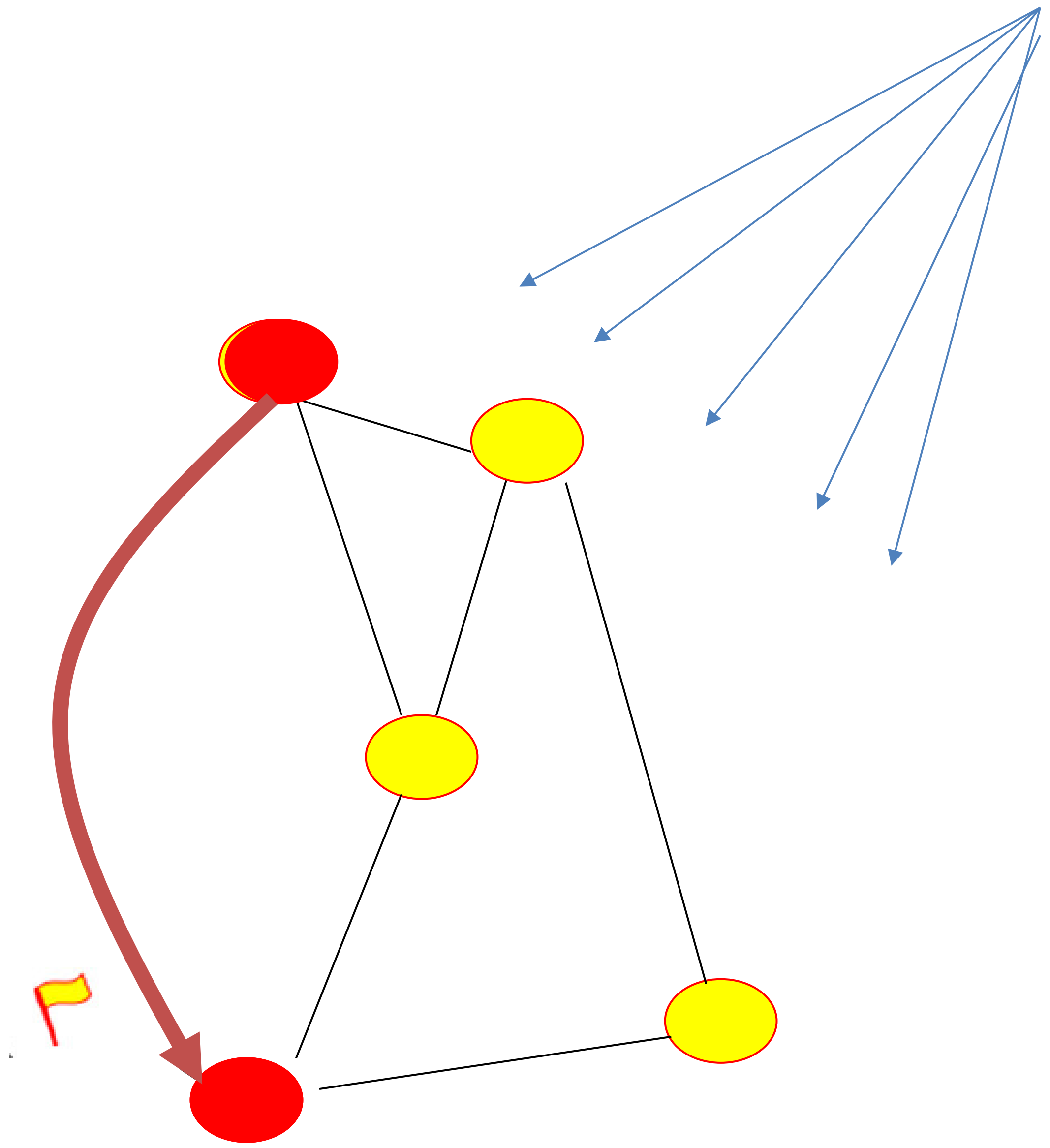
(ii) A synaptic flag is set only at the synapse with a Hebbian co-activation of pre- and postsynaptic factors; the synapse become then eligible to interact with the third factor (blue). Spontaneous spikes of other neurons do not interfere.

(iii) The interaction of the synaptic flag (eligibility trace) with the third factor leads to a strengthening of the synapse (green).

Fig caption: Gerstner et al. 2018

3. Recent experiments for Three-factor rules

Neuromodulators for reward; interestingness; surprise; attention; novelty



Step 1: co-activation sets eligibility trace

Step 2: eligibility trace decays over time

Step 3: (delayed) neuromodulator:
eligibility trace translated into weight change

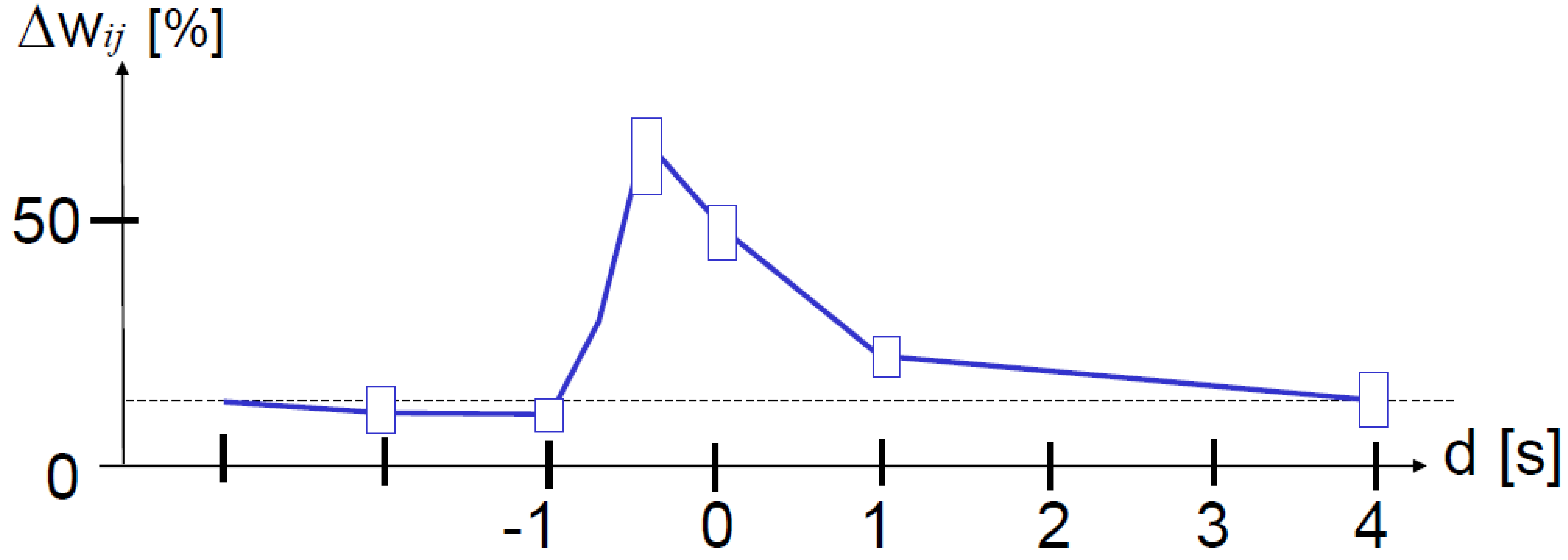
Previous slide.

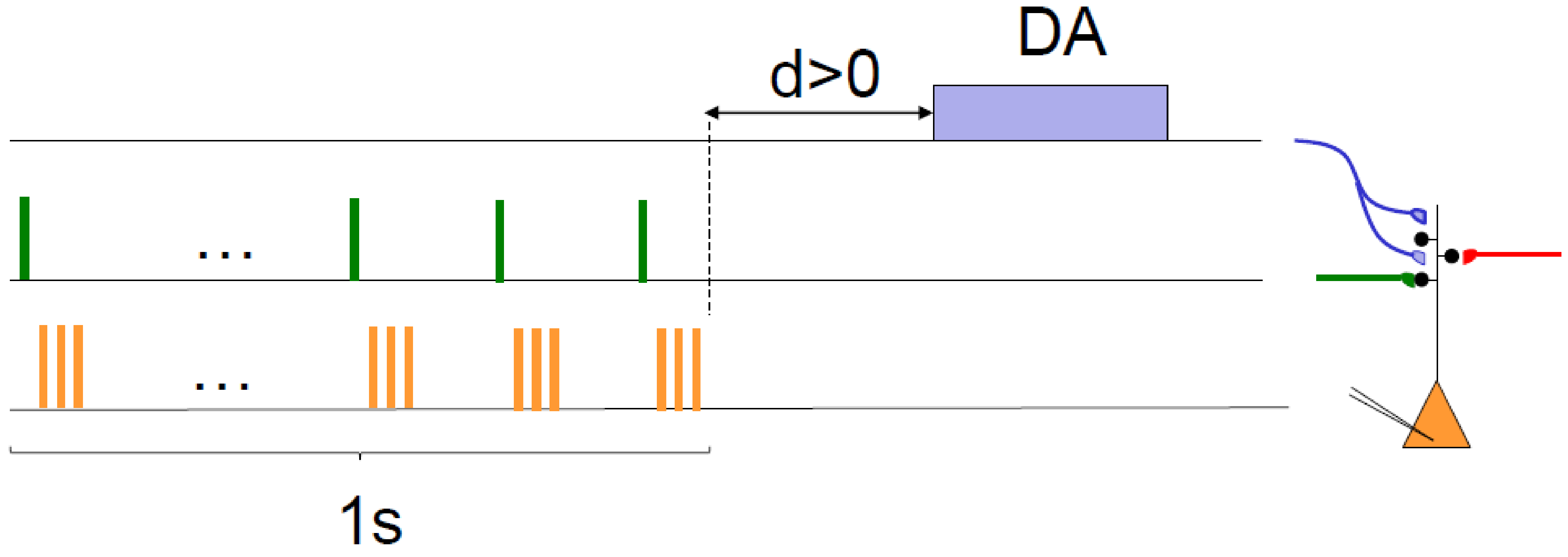
three-factor learning rules are a theoretical concept.

But are there any experiments? Only quite recently, a few experimental results were published that directly address this question.

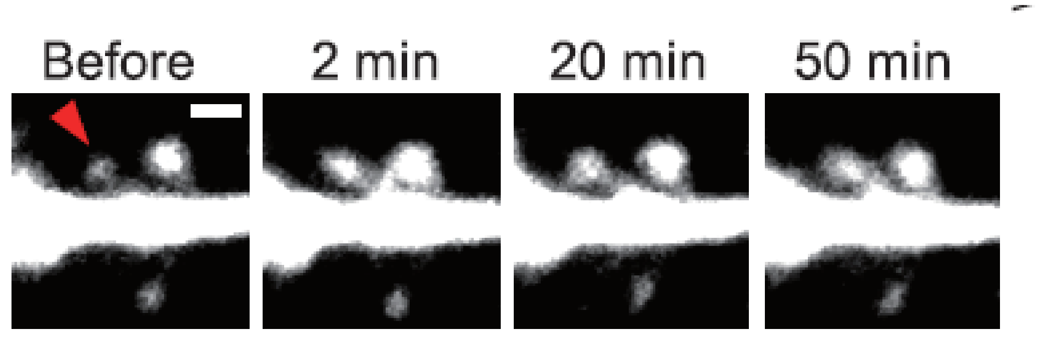
Three-factor rules in striatum: eligibility trace and delayed DA

Yagishita et al. 2014, SCIENCE
Kasai lab



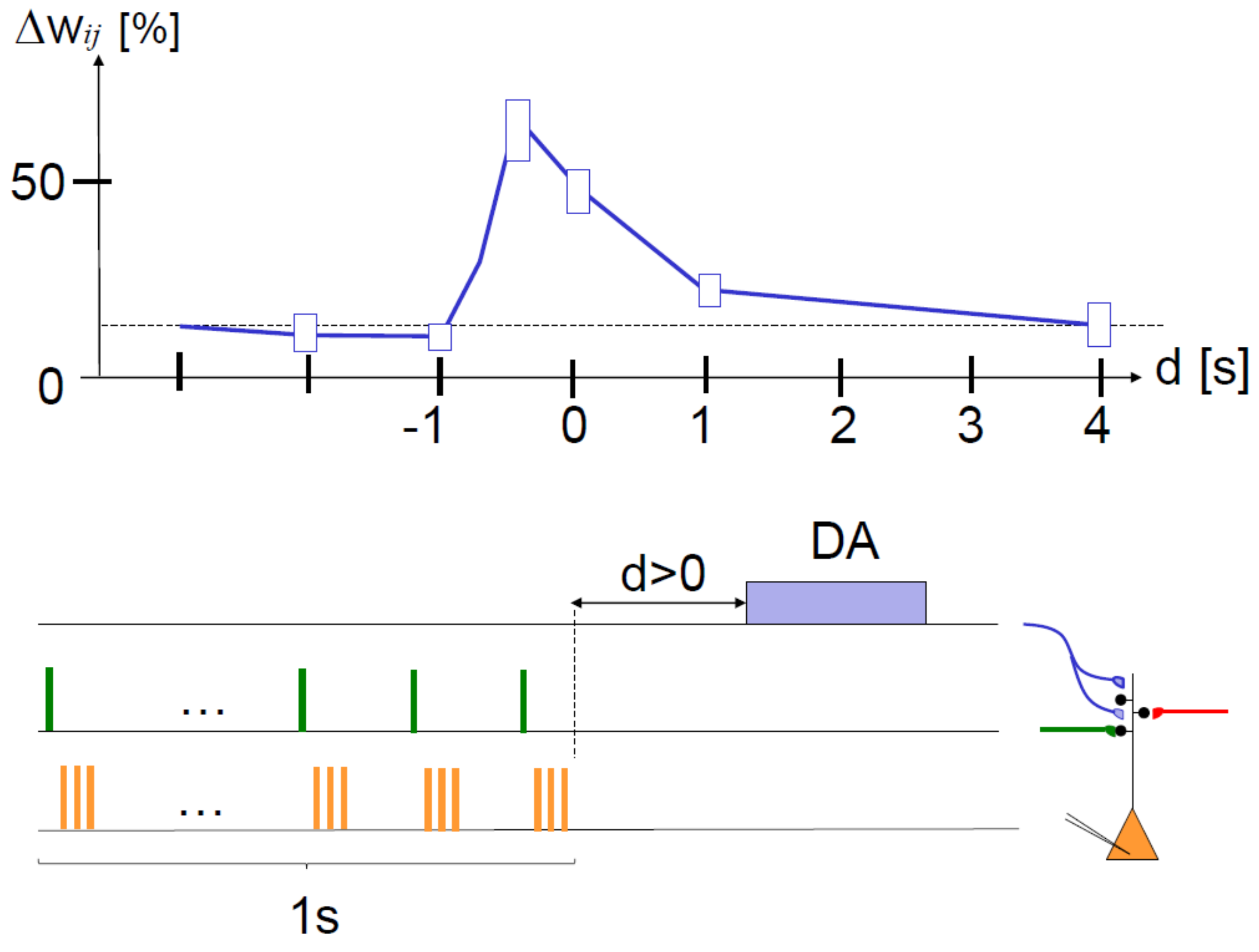


Striatum involved
in action selection
(later today)



-Dopamine (DA) can come with a delay of 1s
-Long-Term stability over at least 50 min.

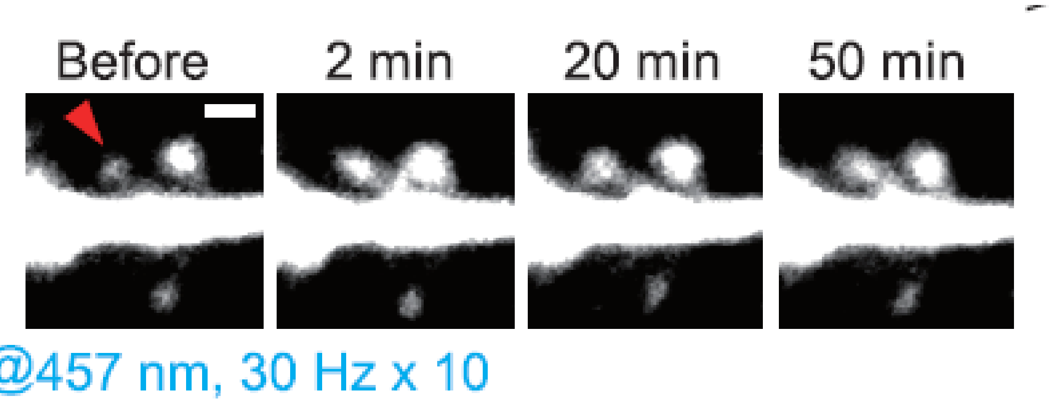
3. Three-factor rules in striatum: eligibility trace and delayed Da



Yagishita et al. 2014

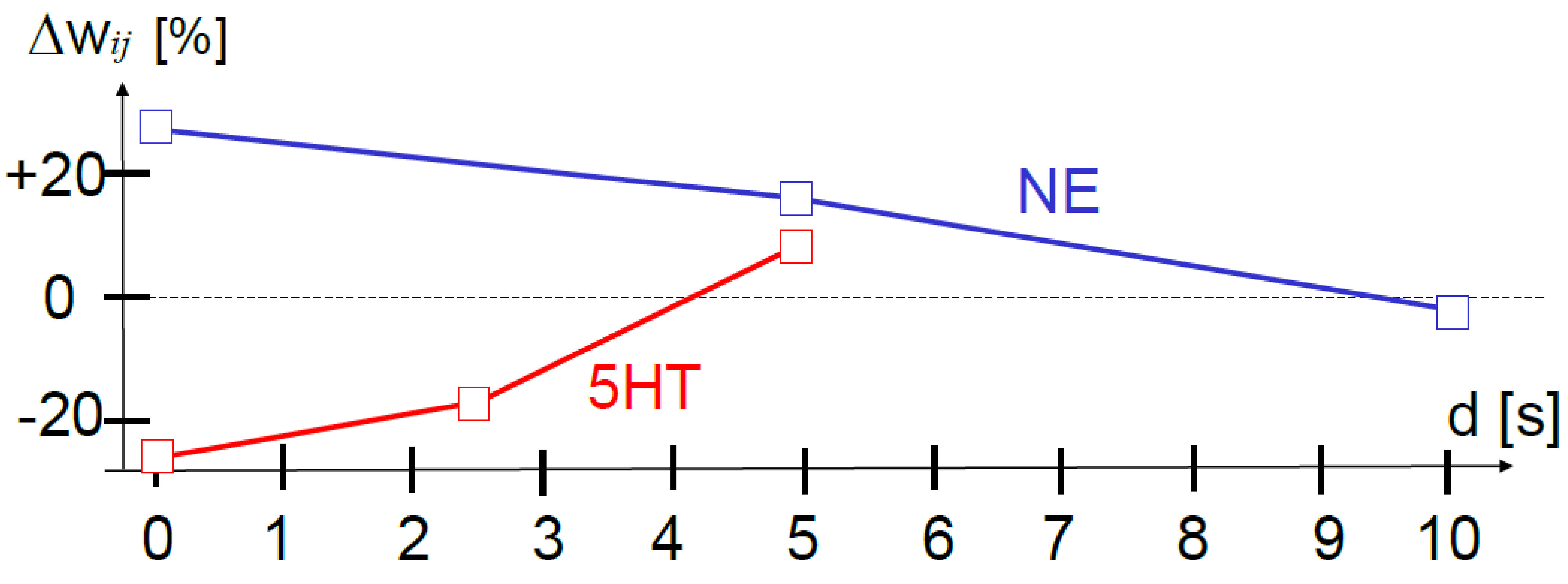
In striatum medial spiny cells, stimulation of presynaptic glutamatergic fibers (green) followed by three postsynaptic action potentials (STDP with pre-post-post-post at +10ms) repeated 10 times at 10Hz yields LTP if dopamine (DA) fibers are stimulated during the presentation (d < 0) or shortly afterward (d = 0s or d = 1s) but not if dopamine is given with a delay d = 4s; redrawn after Fig. 1 of (Yagishita et al., 2014), with delay d defined as time since end of STDP protocol.

Lower left: the image from the beginning of this lecture comes from this experiment of Yagishita. This image demonstrates the Long-Term Stability over at least 50 min

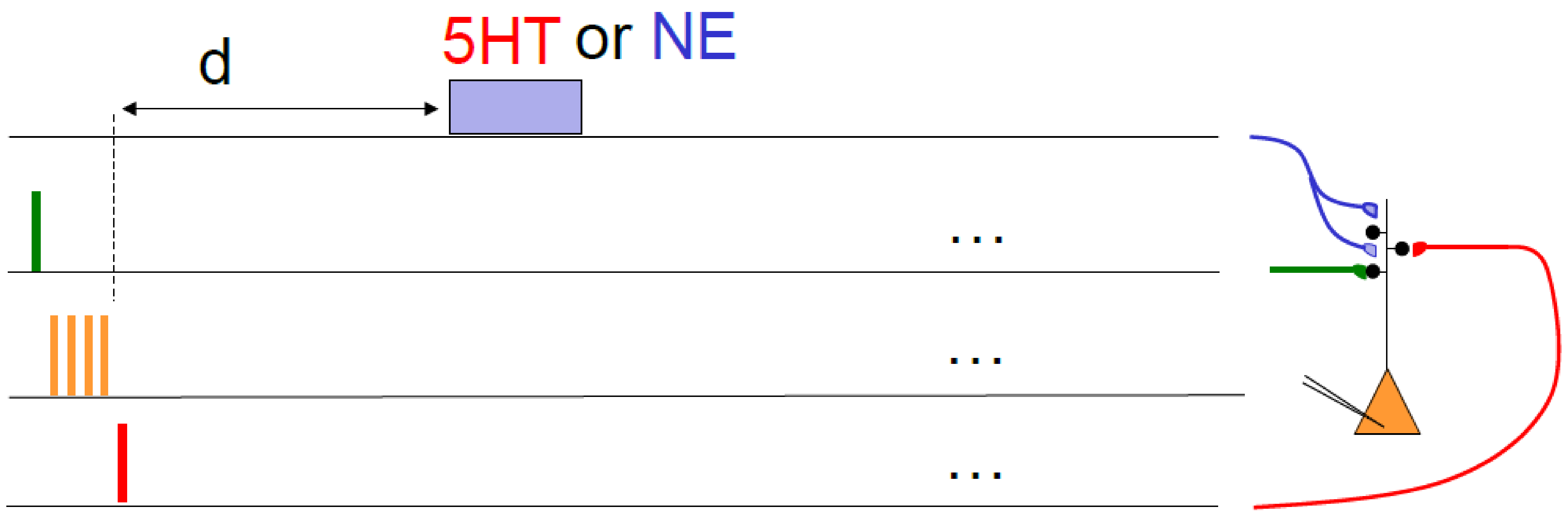


Three-factor rules in cortex: eligibility trace and delayed NE

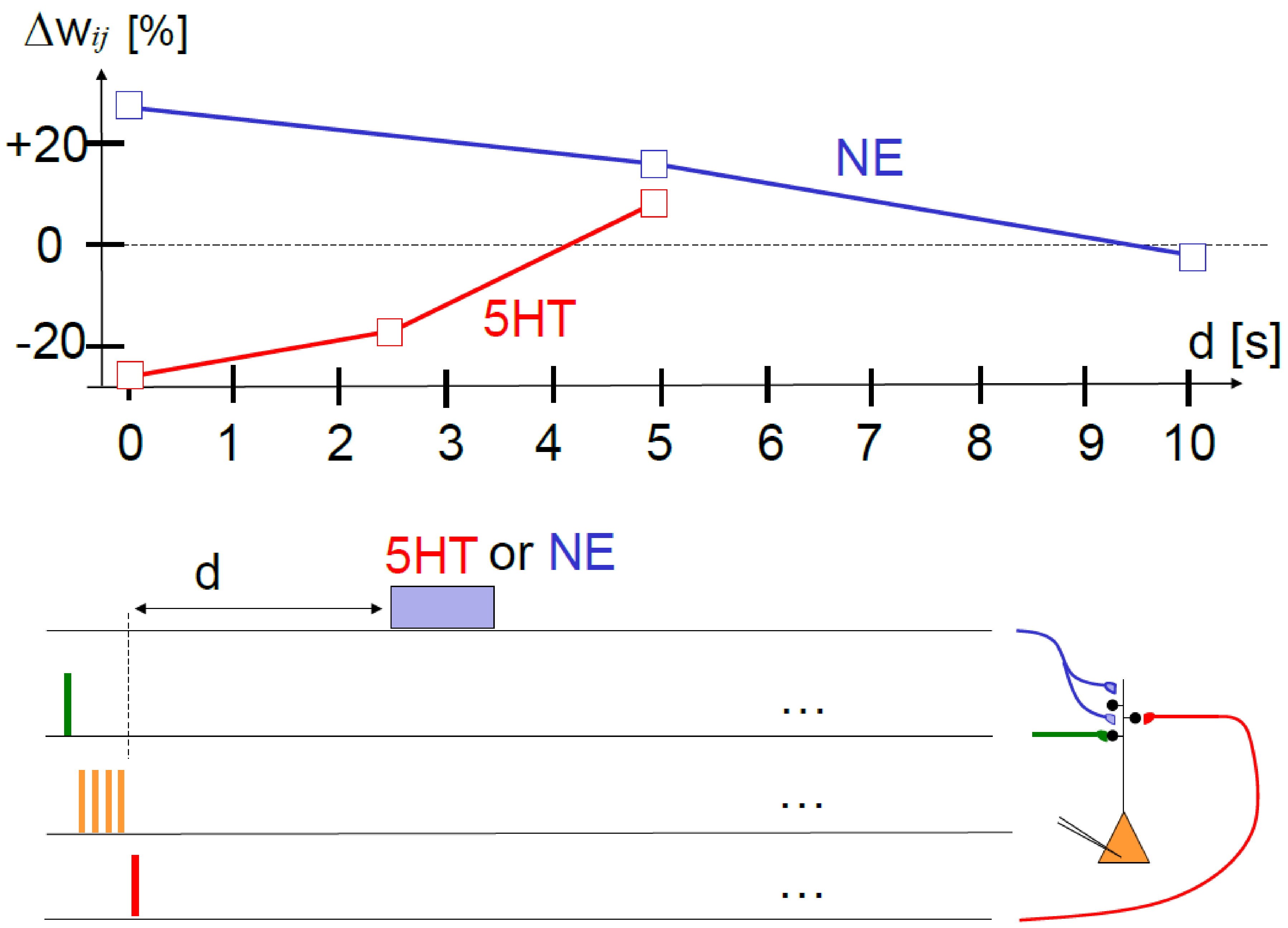
He et al., 2015, NEURON
Kirkwood lab.



NE = norepinephrine
5HT=serotonin



second example



In cortical pyramidal cells, stimulation of two independent presynaptic pathways (green and red) from layer 4 to layer 2/3 by a single pulse is paired with a burst of four postsynaptic spikes (orange).

If the pre-before-post stimulation was combined with a pulse of norepinephrine (NE) receptor agonist isoproterenol
with a delay of 0 or 5s, the protocol gave LTP (blue trace).

If the post-before-pre stimulation
was combined with a pulse of serotonin (5-HT) of a delay of 0 or 2.5s, the protocol gave LTD (red trace).

(He et al., 2015).

Three-factor rules: summary

Three factors are needed for synaptic changes:

- Presynaptic factor = spikes of presynaptic neuron
  or the effect of spike arrival at the synapse

- Postsynaptic factor = spikes of postsynaptic neuron
  or increased voltage or a function of both

- Third factor = Neuromodulator such as dopamine

Previous slide.

three-factor learning rules are a theoretical concept.

But recent experiments show that the brain really can implement three-factor rules. Importantly, the third factor (neuromodulator) can come with a delay of one or two seconds after the Hebbian induction protocol that sets the eligibility trace. Minimal delays work better than longer delays.

Quiz. Synaptic Plasticity and Learning Rules

Learning rules in the brain

[ ] Hebbian learning depends on presynaptic activity
AND on state of postsynaptic neuron

[ ] Reinforcement learning depends on neuromodulators
such as dopamine indicating reward (or ‘success’) [ ]
Three-factor rule: presynaptic signal, postsynaptic
signal, and neuromodulator signal (e.g., DA) MUST
arrive at the same time.

Literature.

Yagishita, S., Hayashi-Takagi, A., Ellis-Davies, G. C., Urakubo, H., Ishii, S., and Kasai, H. (2014). A critical time window for dopamine actions on the structural plasticity of dendritic spines. Science 345, 1616–1620. doi: 10.1126/science.1255514

He, K., Huertas, M., Hong, S. Z., Tie, X., Hell, J.W., Shouval, H., and Kirkwood, A. (2015). Distinct eligibility traces for LTP and LTD in cortical synapses. Neuron 88, 528–538. doi: 10.1016/j.neuron.2015.09.037

Gerstner W, Lehmann M, Liakoni V, Corneil D and Brea J (2018) Eligibility Traces and Plasticity on Behavioral Time Scales: Experimental Support of NeoHebbian Three-Factor Learning Rules. Front. Neural Circuits 12:53. doi: 10.3389/fncir.2018.00053

Frémaux N and Gerstner W (2016) Neuromodulated Spike-Timing-Dependent Plasticity, and Theory of Three-Factor Learning Rules. Front. Neural Circuits 9:85. doi: 10.3389/fncir.2015.00085

Reinforcement Learning Lecture 1
Reinforcement Learning and SARSA

Wulfram Gerstner
EPFL, Lausanne, Switzerland

Part 1: Examples of Reward-based Learning

Objectives for Lecture RL1 (Part 1-3)

- Reinforcement Learning (RL) is learning by rewards
- Agents and actions, states and rewards
- Convergence in expectation, online and batch.

Reading:
Sutton and Barto, Reinforcement Learning (MIT Press, 2nd edition 2018)
Chapters: 1.1-1.4; 2.1-2.6; 3.1-3.5; 6.4

Reading for this week:

Sutton and Barto, Reinforcement Learning (MIT Press, 2nd edition 2018, also online)

Chapters: 1.1-1.4; 2.1-2.6; 3.1-3.5; 6.4

Video for most of this lecture:

RL Lecture 1 on https://lcnwww.epfl.ch/gerstner/VideoLecturesRL-Gerstner.html Parts 1-3.

Background reading:

Silver et al. 2017,

Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

REPETITION: Artificial Neural Networks for action learning

No labeled data?

Replaced by:

‘Value of action’

- ‘goodie’ for dog
- ‘success’
- ‘compliment’

BUT:

Reward is rare:

‘sparse feedback’ after a long action sequence

Previous slide. (already shown before the break)

How does a human learn to play table tennis: How does a child learn to play the piano? How does a dog learn to perform tricks?

In all these cases there is no supervisor. No master guides the hand of the players during the learning phase. Rather the player ‘discovers’ good movements by rather coarse feedback. For example, the ball in table tennis does not land on the table as it should. That is bad (negative feedback). The ball has a great spin so that the opponent does not get. This is good (positive feedback).

Similarly, it is hard to tell a dog what to do. But if you reinforce the dog’s behavior by giving a ‘goodie’ at the moment when it spontaneously performs a nice action, then it can learn quite amazing things.

In all these cases it is the ‘reward’ that guides the learning. Rewards can be the goodie for the dog, or just the feeling ‘now I did well’ for humans.

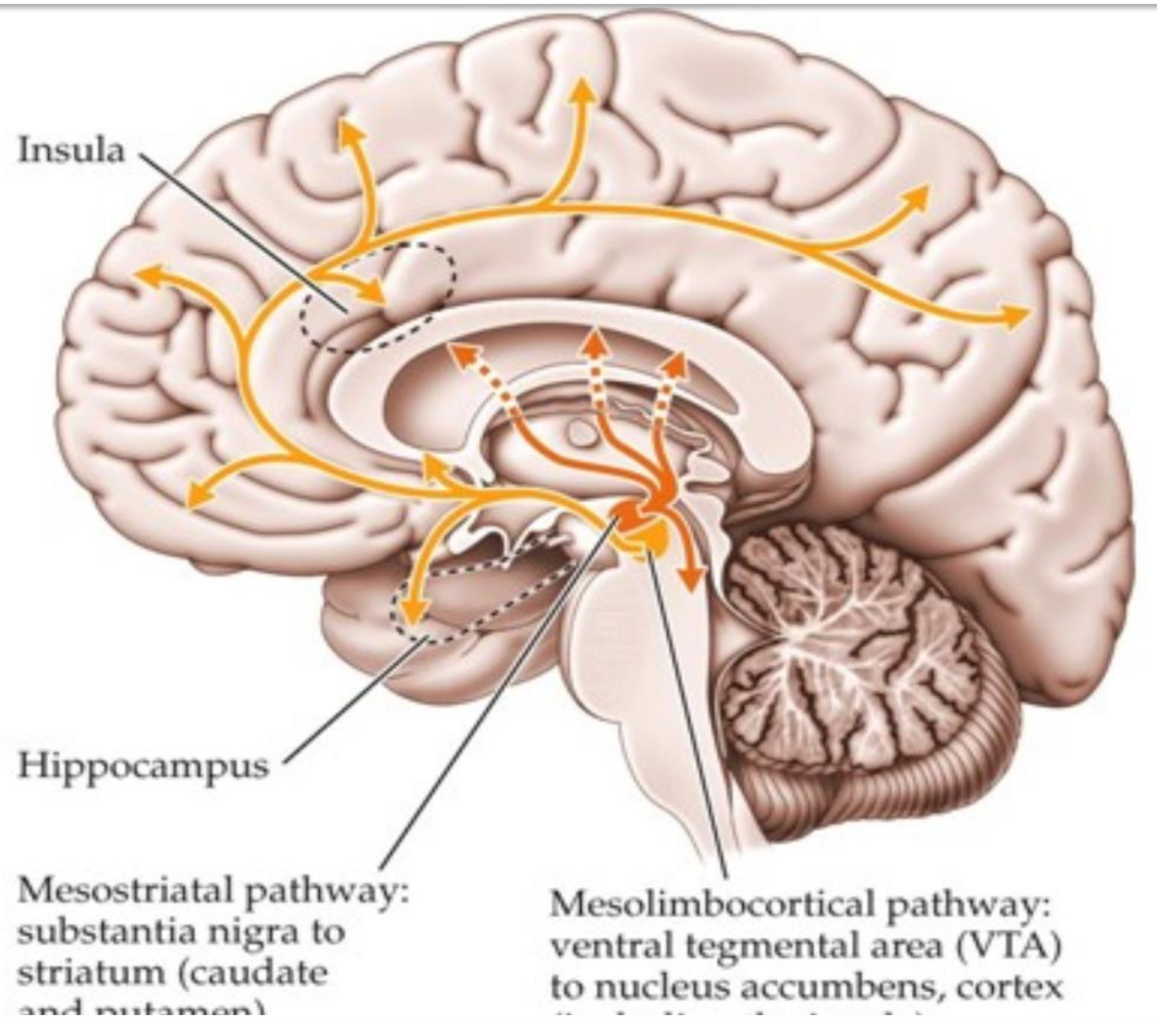
Reward information is available in the brain

Neuromodulator **dopamine**:
Signals “reward minus expected reward”

Schultz et al., 1997,
Waelti et al., 2001
Schultz, 2002

‘success signal’

Dopamine


Insula
Hippocampus
Mesostriatal pathway: substantia nigra to striatum (caudate and putamen)
Mesolimbocortical pathway: ventral tegmental area (VTA) to nucleus accumbens, cortex

Previous slide.

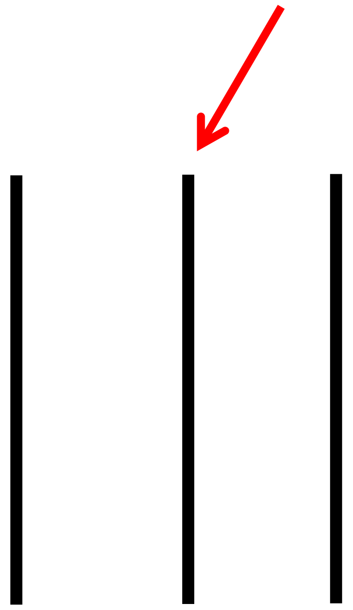
Inside the brain, reward information is transmitted by the neuromodulator dopamine. Neurons that use dopamine as their chemical transmission signal are situated in nuclei below the cortex and have cables (axons) that reach out to vast areas of the brain.

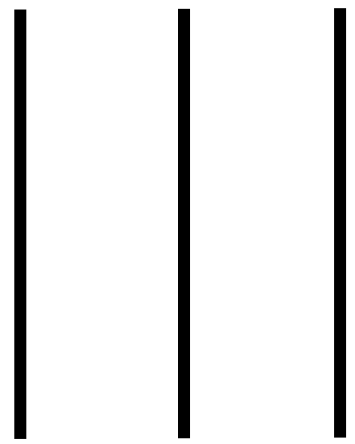
As we will see later, neurons that communicate with the neuromodulator dopamine transmit a generic success signal that is not just reward, but something like ‘reward minus expected reward’.

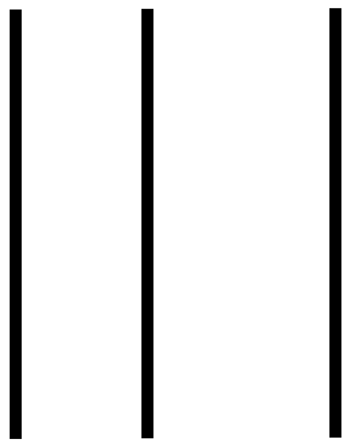
To conclude, reward information is available throughout the brain.

Examples of reinforcment learning

Middle bar: shifted left or shifted right?

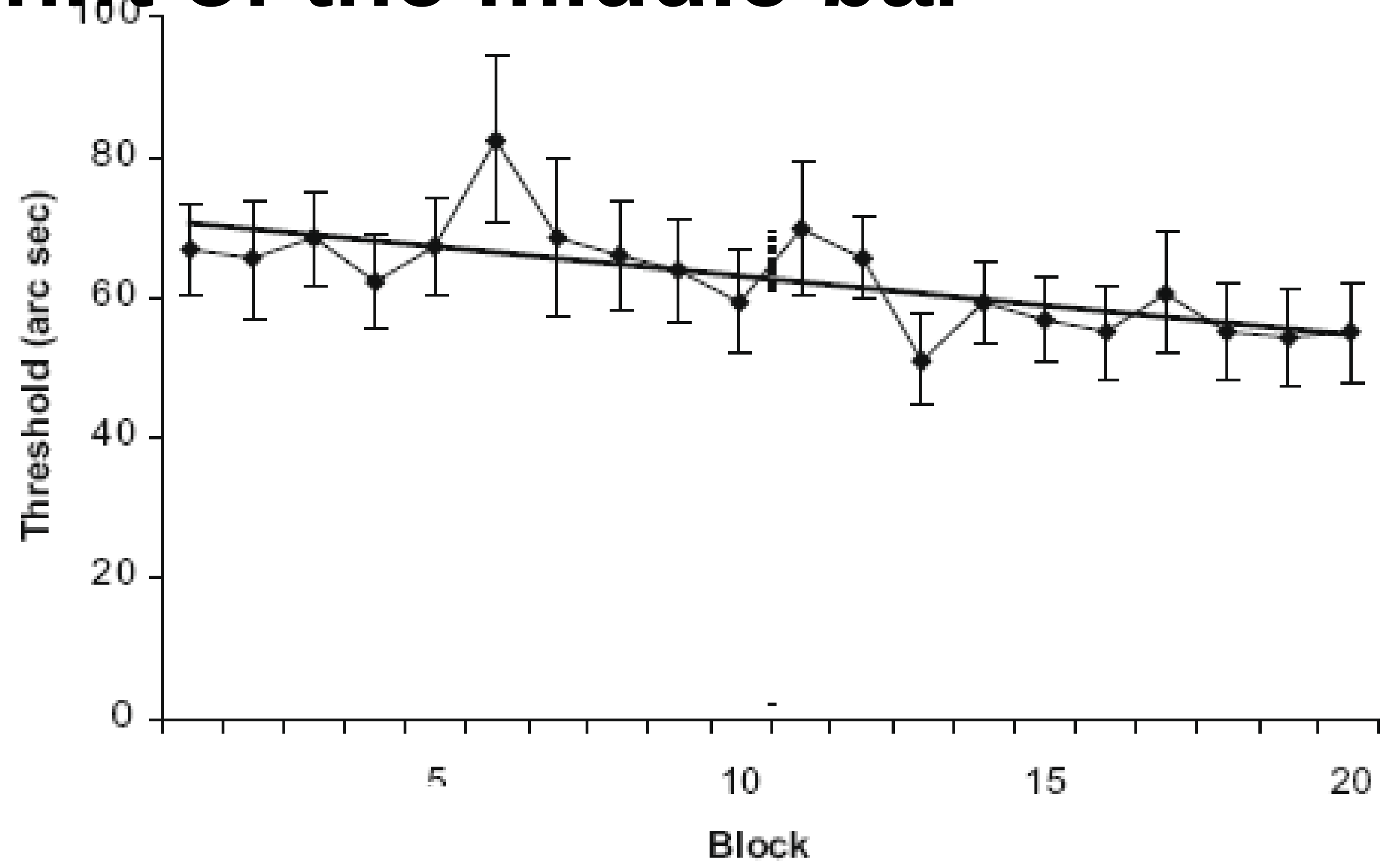






Observers get better at seeing the shift of the middle bar

Min. shift



Feedback:
tone for wrong response

Tartaglia,Aberg,Herzog 2009

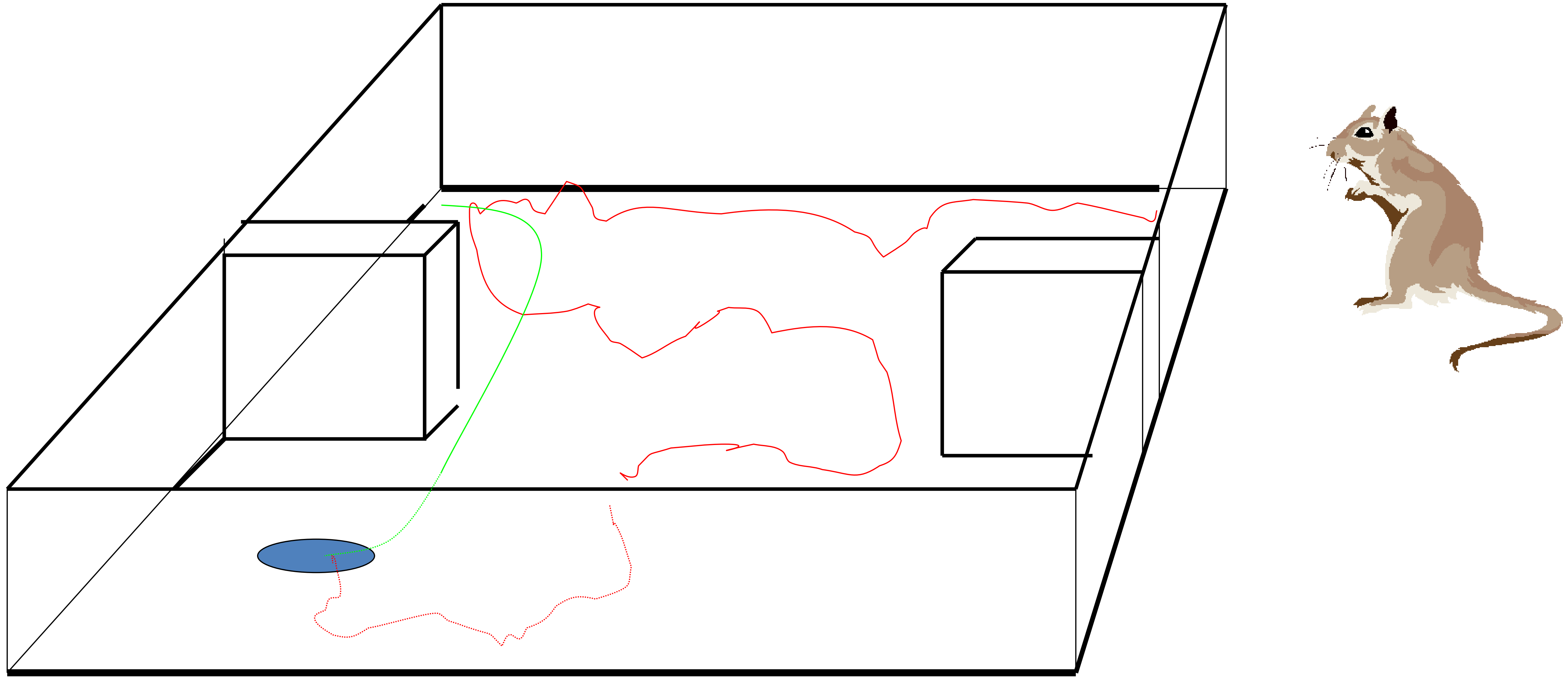
Previous slide (This example is not shown in class)

Let us look at a few additional examples, beyond table tennis.

Humans can get, by practice and feedback, better at recognizing a visual pattern with three bars. The task is to distinguish cases where the middle bar is shifted to the left from those where it is shifted to the right.

Bottom right:
The minimal shift that is just recognizable decreases over time (1 block = 1 practice session) indicating learning.
The feedback signal is just right or wrong.

Examples of reinforcement learning: animal conditioning

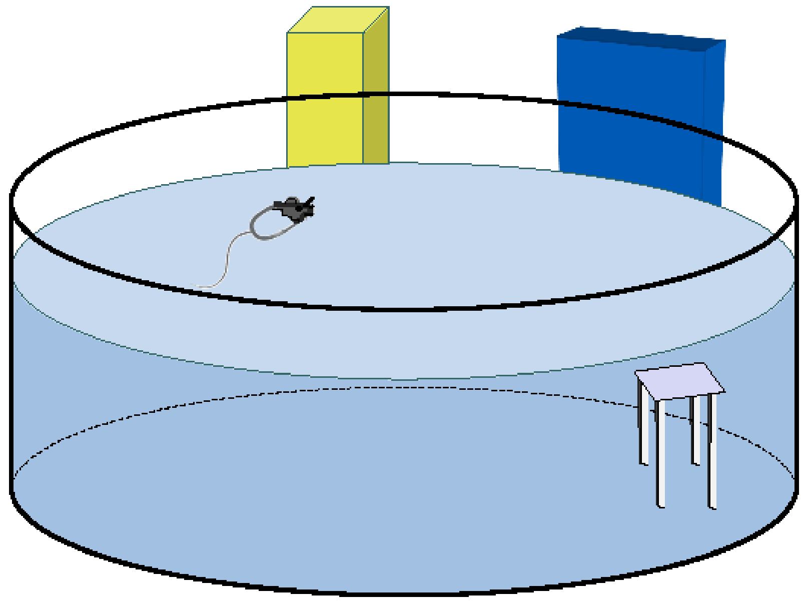


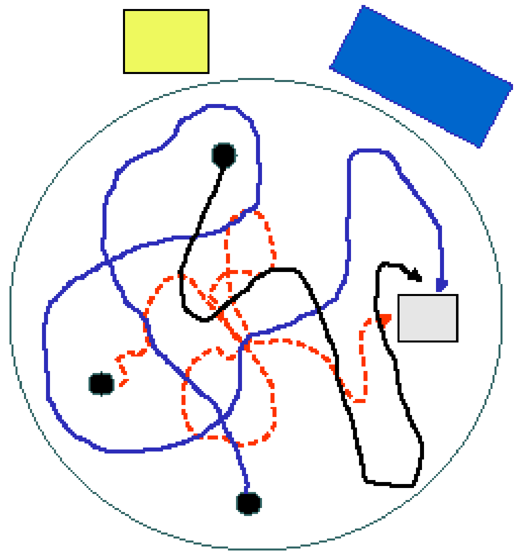
Previous slide. (already shown before the break)

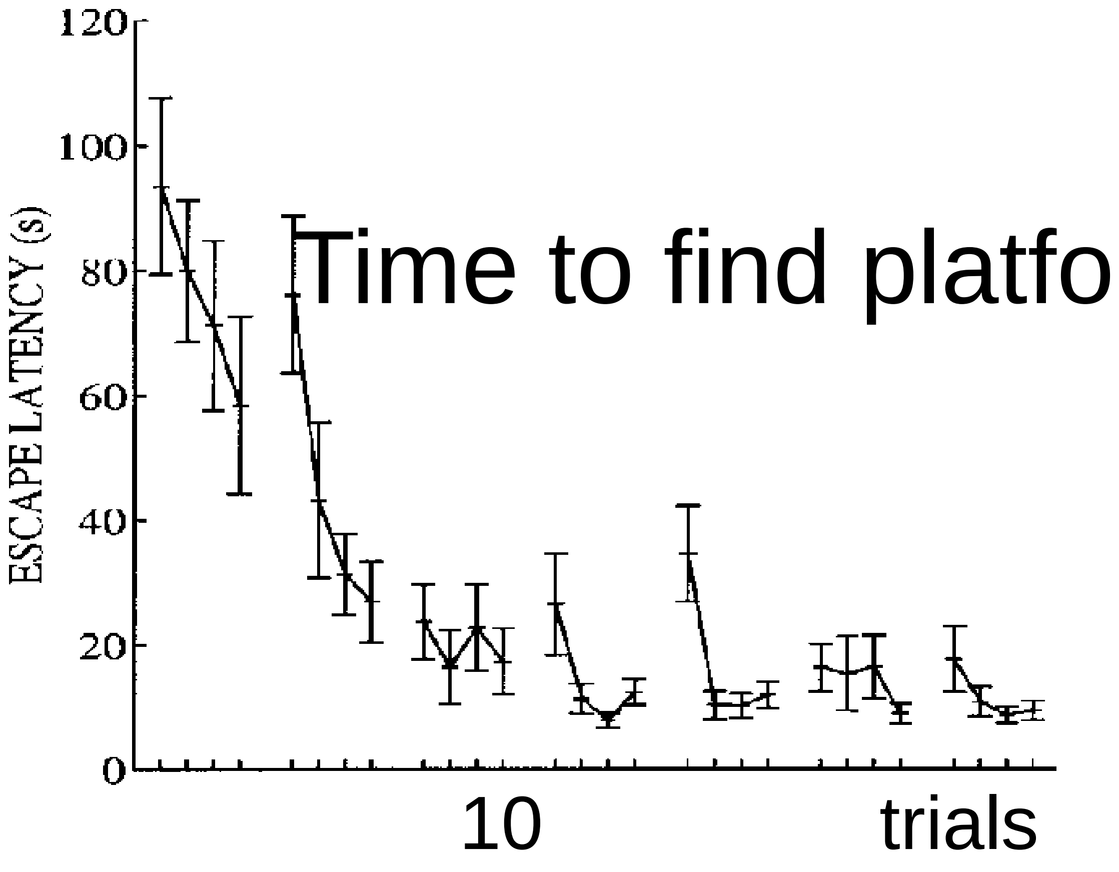
If you put a rat into an environment it will wander around. Suppose that, at some place, it discovers a food source hidden below the sand of the surface.
After a couple of trials it will go straight to the location of the food source which implies that it has learned the appropriate sequence of actions in the environment to find the food source.

Examples of reinforcement learning: animal conditioning

Morris Water Maze







Time to find platform

Rats learn to find
the hidden platform

(Because they like to
get out of the cold water)

Foster, Morris, Dayan 2000

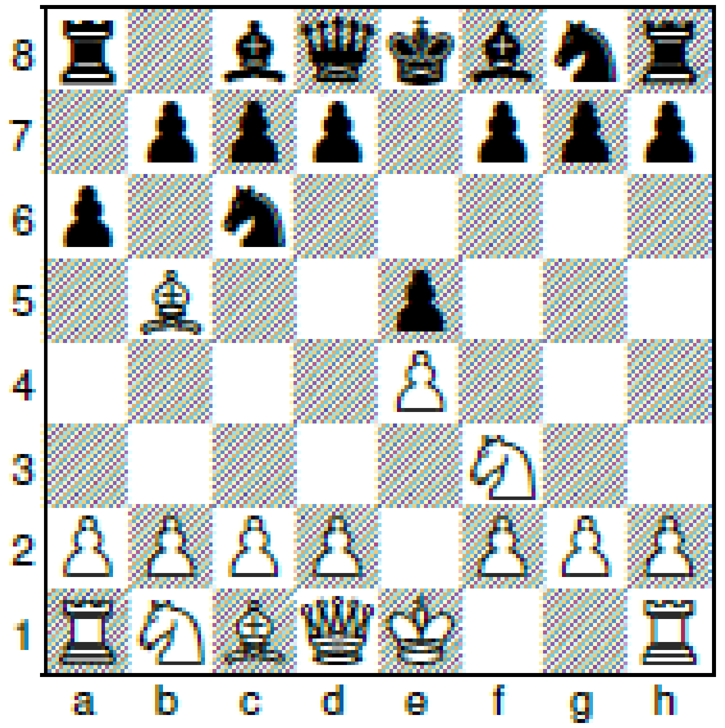
Previous slide. (This example is not shown in class)

Actual experiments for location learning are often performed in a Morris water maze. In the maze, there are 4 starting points and one target location which is a platform hidden (in milky water) just below the water surface. The rat does not like to swim in cold water and therefore tries to find the platform.

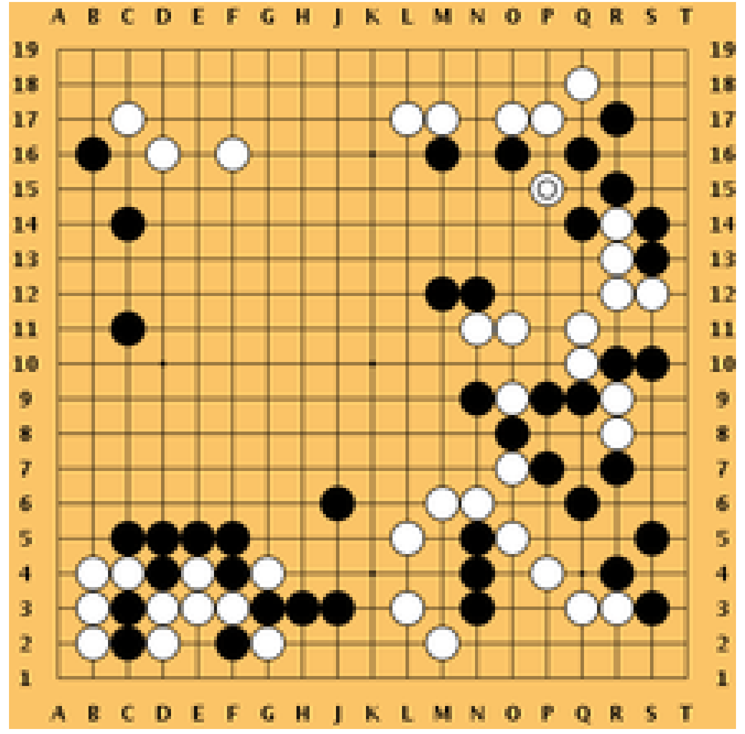
After a few trials it swims straight to the platform.
Bottom right: the time to reach the platform decreases over trials, indicating learning.

REPETITION: Deep reinforcement learning

Chess



Go



Artificial neural network (*AlphaZero*) discovers different strategies by playing against itself.

In Go, it beats Lee Sedol

Previous slide.

In chess a neural network trained by reinforcement learning discovers winning strategies by playing against itself. Similarly, a neural network playing Go against itself learns to play at a level so as to beat one of the world champions.

The aim of the class is to arrive at Deep Reinforcement Learning (Deep RL): Today we start with (standard) RL, in a few weeks we turn to deep networks, and in May we will turn to Deep RL.

Deep reinforcement learning

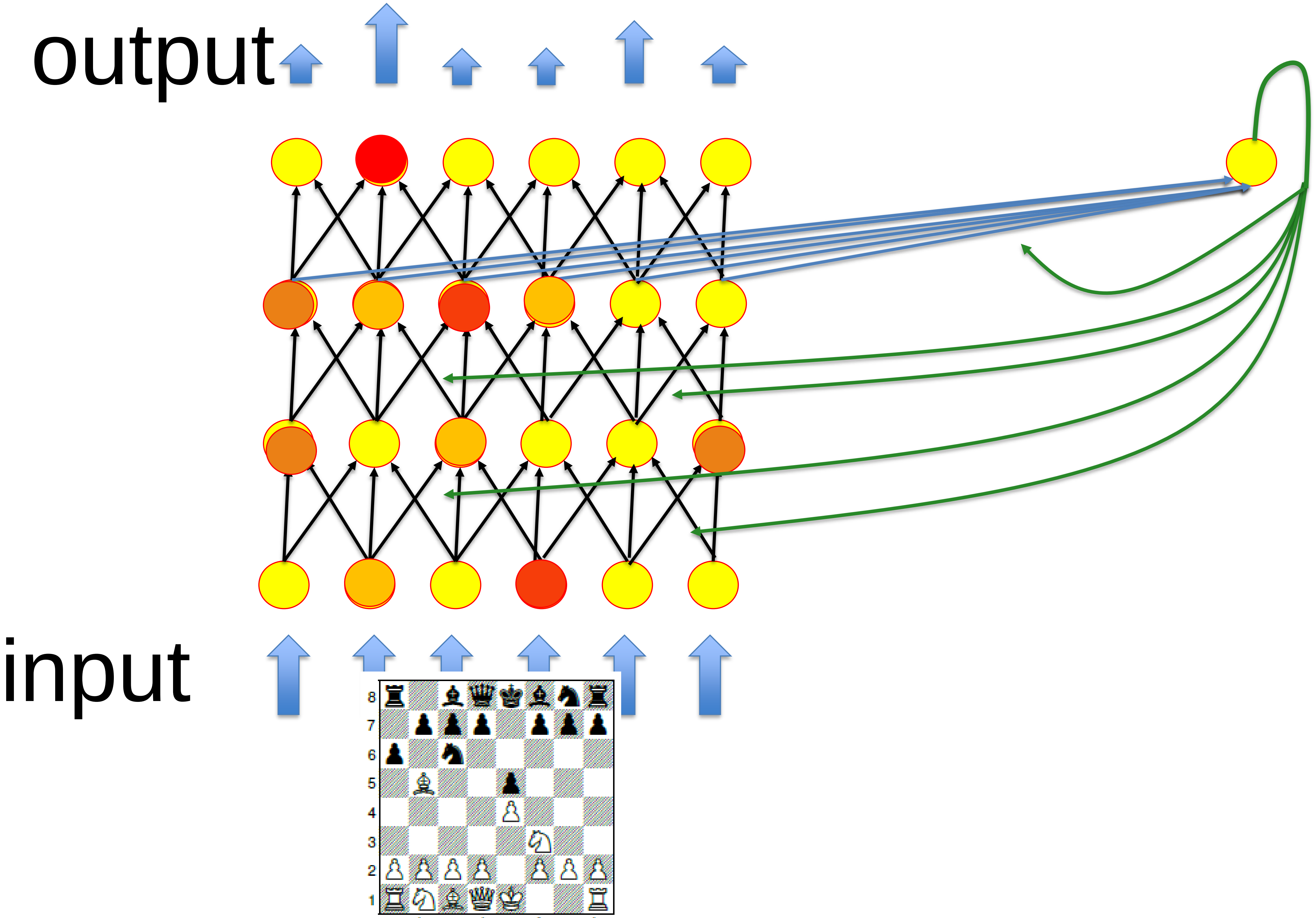
Network for choosing action

action:

Advance king

2nd output for **value** of state:

probability to win



Learning by success signal

- change connections

aim:

- choose next action to win

aim for value unit:

- predict value of current position

Previous slide. (already shown before the break)

At the end of this semester, you will be able to understand the algorithms and network structure used to achieve these astonishing performances. Important are two types of outputs.

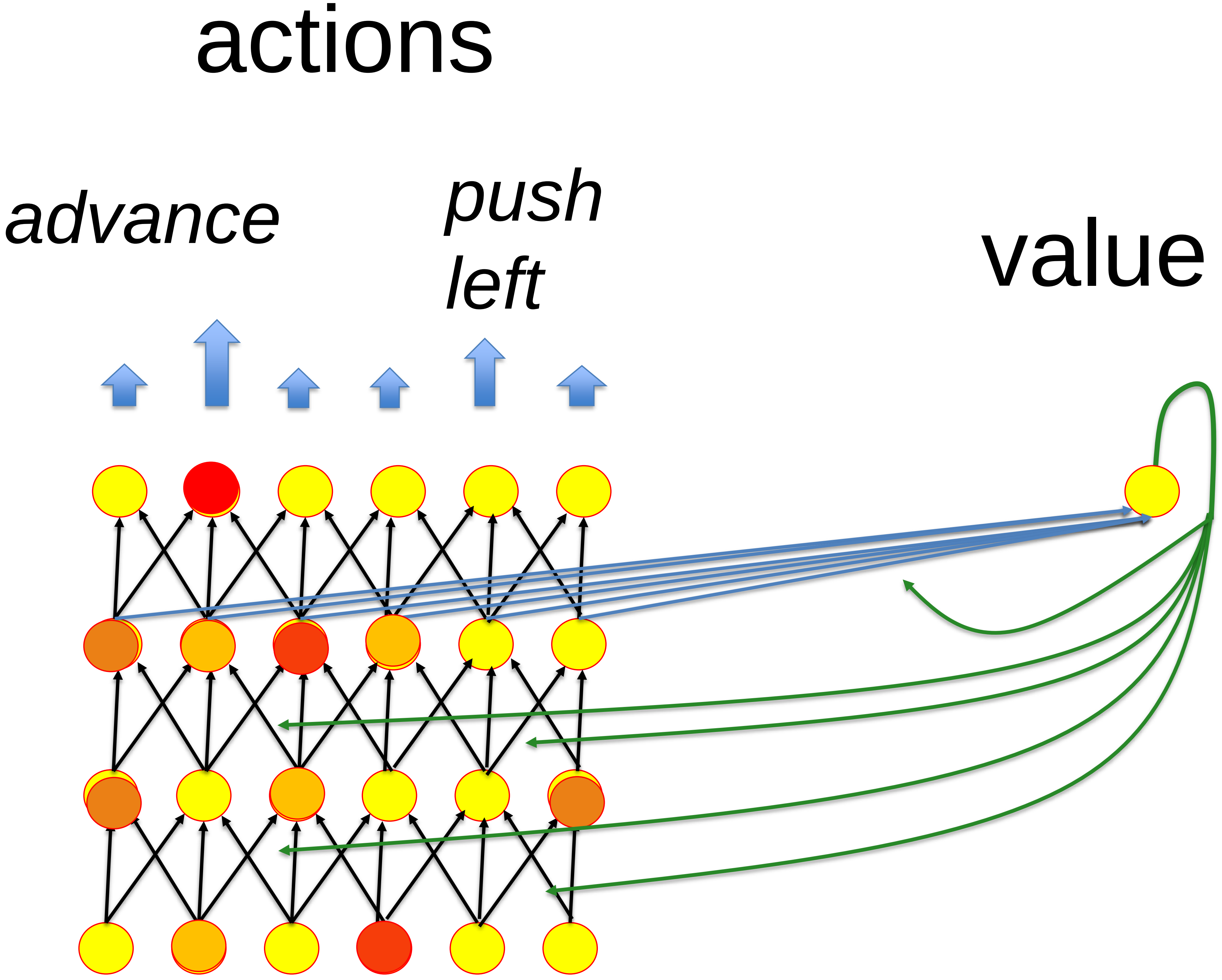
Left: different output neurons represent different actions.
Right: an additional output neuron represents the value of the present state; we can loosely define the value as the probability to win, or the ‘average reward’ that you can get starting from this state.

The input is a representation of the present state of the game.

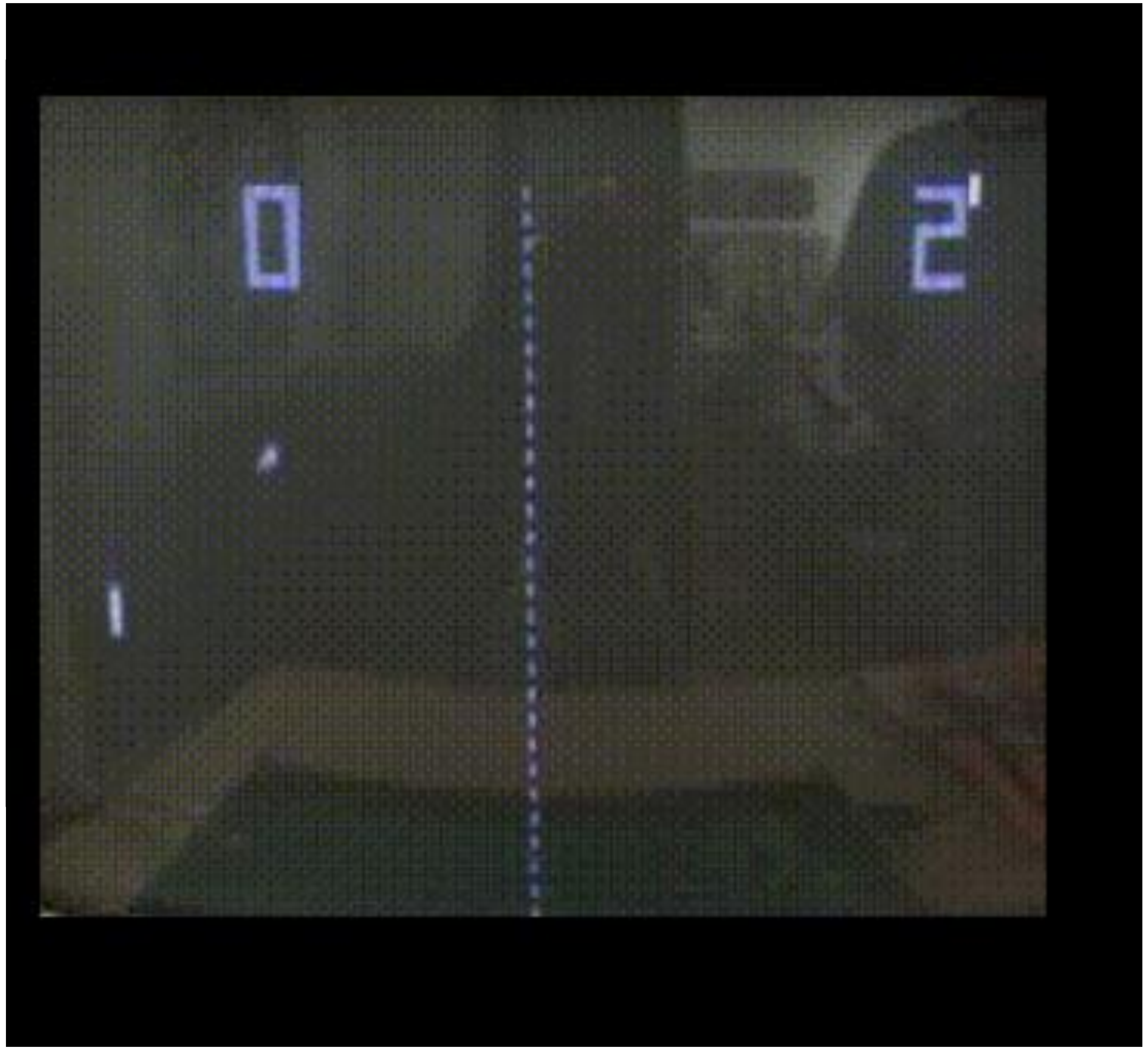
Details will become clear toward the end of the semester; at the moment the aim is just to give you a flavor of the high-level concepts.

Deep Reinforcement Learning:

Control a dynamic system (example of past minproject)



Example: Play Pong (Atari game)



Previous slide.

In one of the miniprojects training will be based on reward: successful behavior of the simulated agent will give positive rewards.

Quiz: Rewards in Reinforcement Learning

[ ] Reinforcement learning is based on rewards
[ ] Reinforcement learning aims at optimal action choices
[ ] In chess, the player gets an external reward after every move
[ ] In table tennis, the player gets a reward when he makes a point
[ ] A dog can learn to do tricks if you give it rewards at appropriate moments

Previous slide. Your notes (already shown before the break)

.

Reinforcement Learning Lecture 1
Reinforcement Learning and SARSA

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Part 2: Elements of Reinforcement Learning

- Examples of Reward-based Learning
- **Elements of Reinforcement Learning**

Previous slide.

We now start with the formalization of reinforcement learning

Elements of Reinforcement Learning:

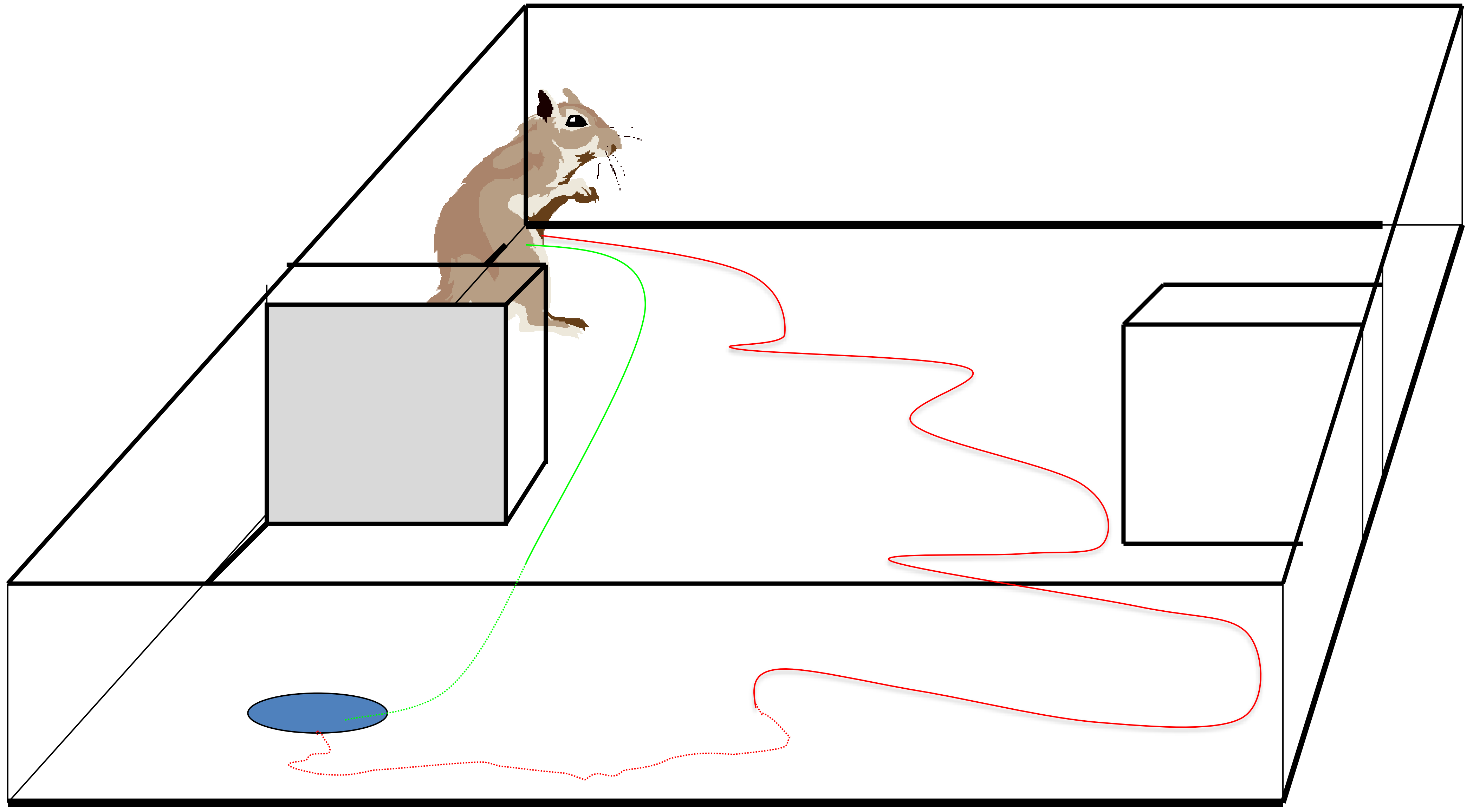
-states
-actions
-rewards

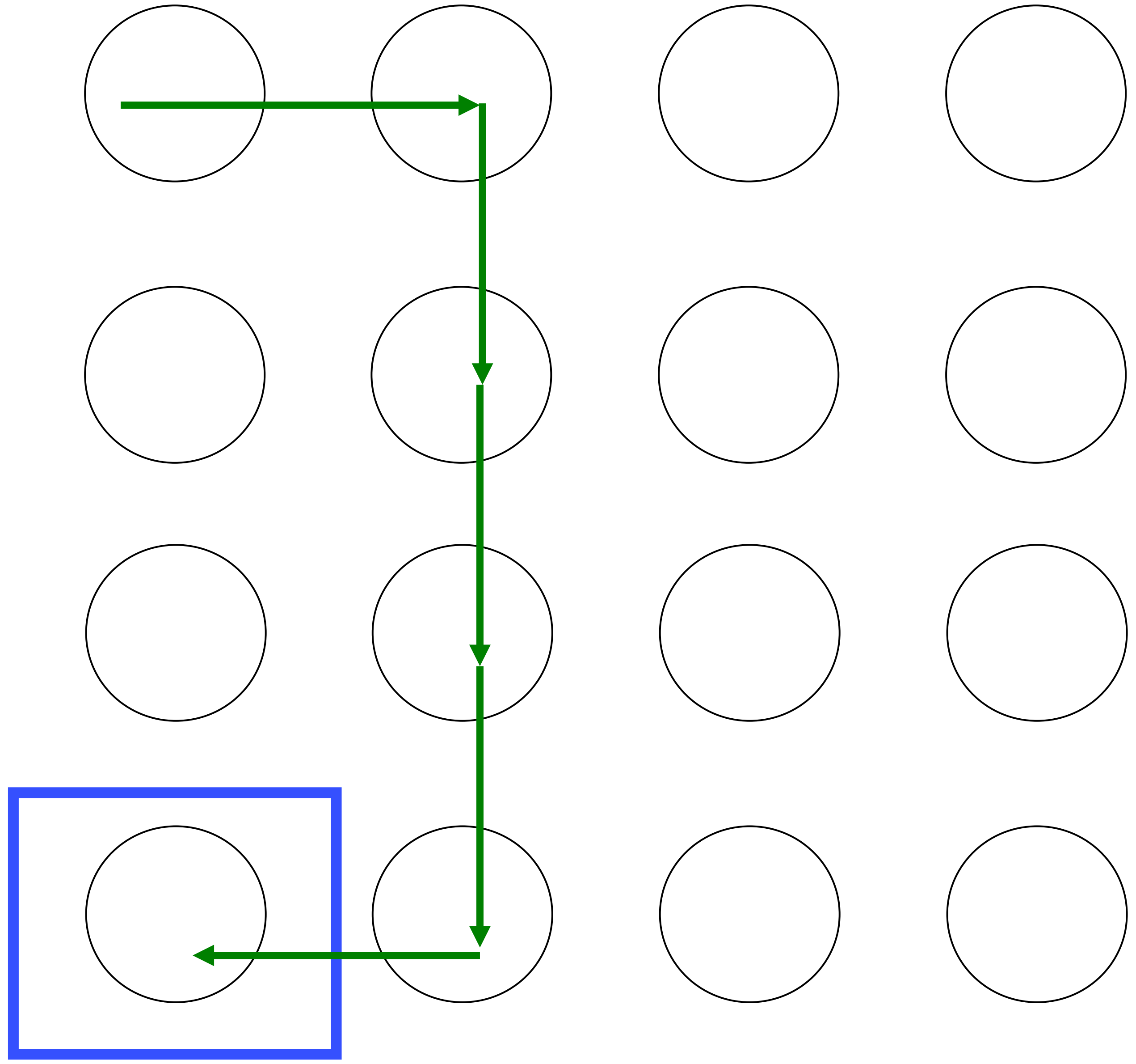
Previous slide.

Reinforcement learning needs states, actions, and rewards.

Elements of Reinforcement Learning:

- discrete states
- discrete actions
- sparse rewards





Previous slide (already shown before the break)

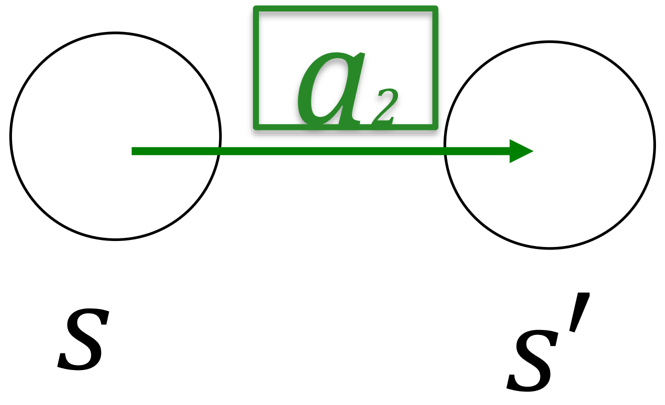
.

Note that, for standard formulations of Reinforcement Learning Theories this (normally) implies discretizing space and actions.

We will study continuous-space formulations only next week.

Elements of Reinforcement Learning:

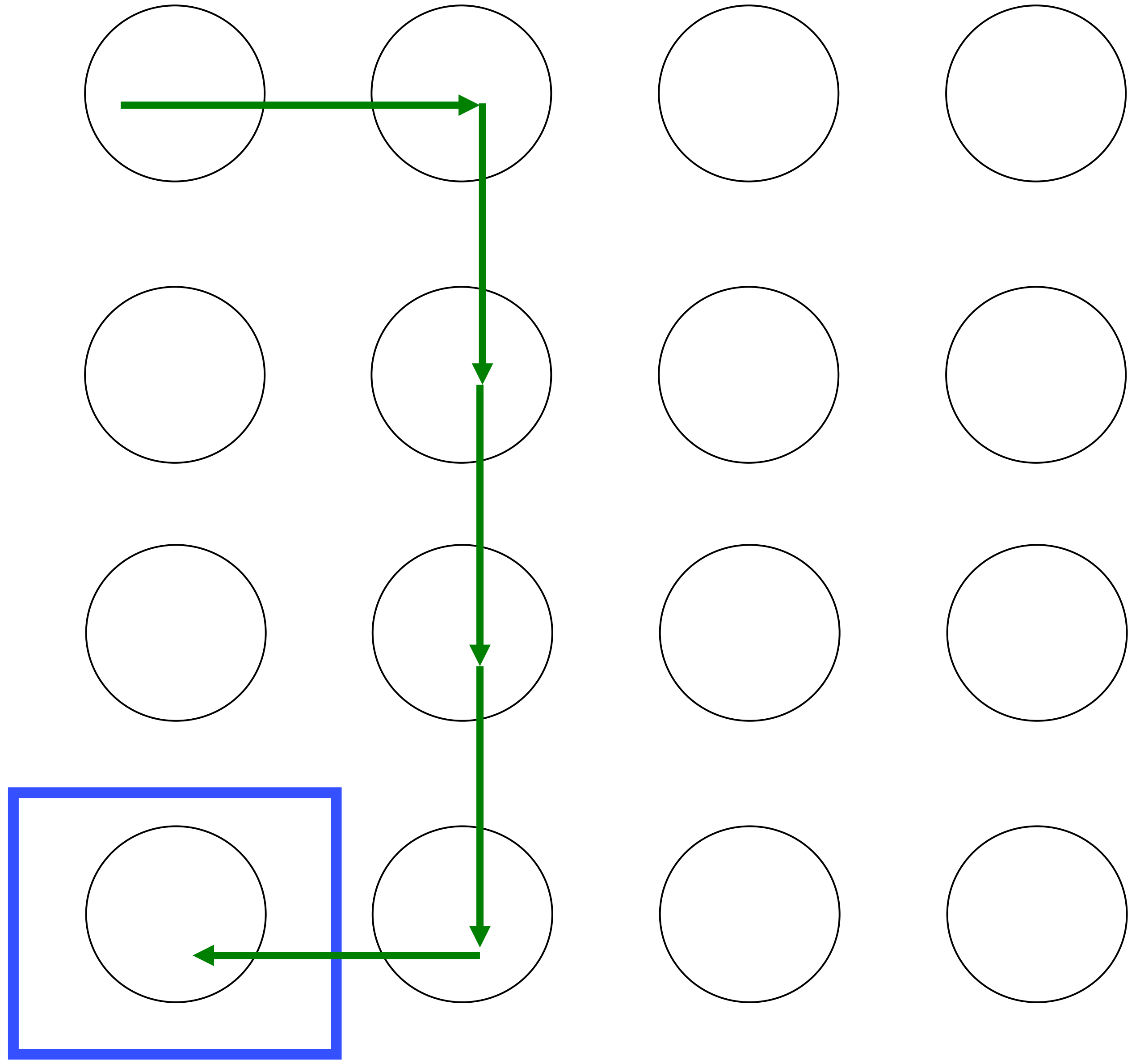
- discrete states:
  - old state s
  - new state s'
- current state: s_t
- discrete actions: $a_1, \; a_2 \; \dots \; a_A$
- current action: a_t
- current reward: r_t



- Mean rewards for transitions:

$$R^a_{s \to s'}$$

often most transitions have zero reward



Previous slide.

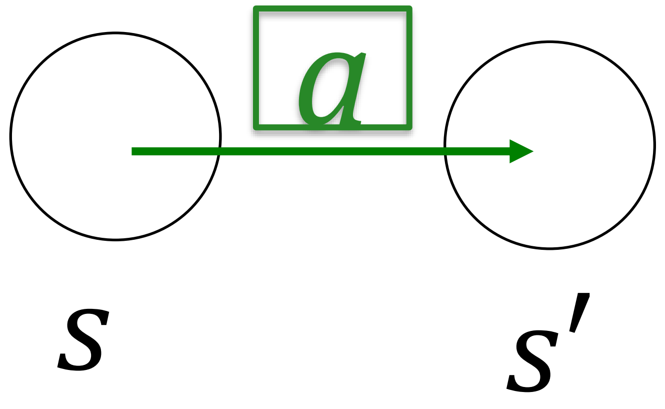
The elementary step is:
The agent starts in state s.
It takes action a
It arrives in a new state s’
Potentially receiving reward r (during the transition or upon arrival at s’).

Since rewards are stochastic we have to distinguish the mean reward at the transition (capital R with indices identifying the transition) from the actual reward (lower-case r with index t) that is received at time t on a transition.

Note that in many practical situations most transitions or states have zero rewards, except a single ‘goal’ state at the end.

REPETITION: States in Reinforcement Learning:

- discrete states:
    - starting state s
    - arrival state s'
- current state: s_t



state = current configuration/well-defined situation

= generalized ‘location’ of actor in environment

Previous slide.

What are these discrete states?
Loosely speaking a state is the current configuration that **uniquely** describes the momentary situation. We can think of the generalized ‘location’ of the actor in the environment

To get acquainted with this, let us look at an example.

Reinforcement Learning: Example Acrobot

3 actions: a_1 = no torque,

a_2 = torque +1 at elbow,

a_3 = torque -1 at elbow

States?

→ discretize!

reward if tip above line

Suppose 5 states per dimension,
How many states in total?
[ ] 5
[ ] 25
[ ] 125
[ ] 625

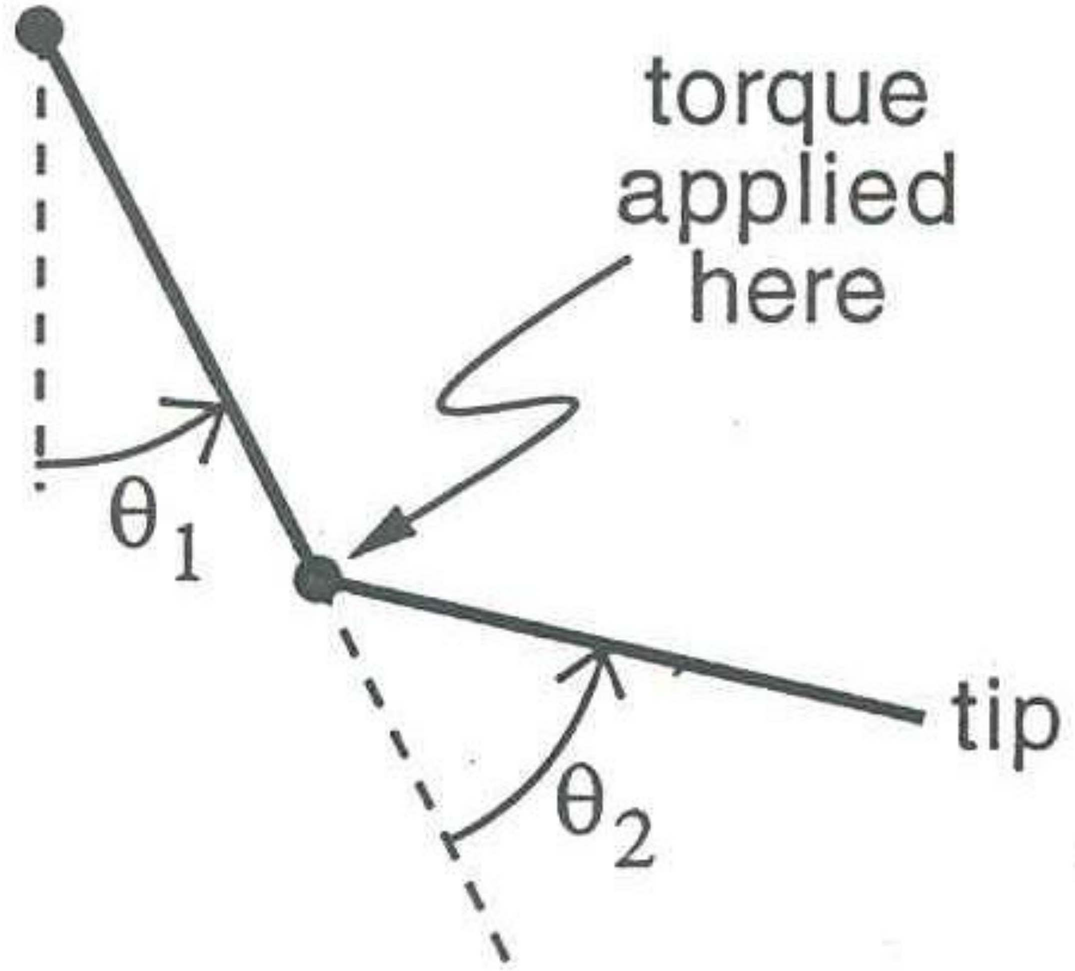


Figure 11.4 The acrobot.

*From Book:
Sutton and Barto*

Previous slide.
The aim of the acrobat is to move the tip above the blue line. To achieve this torque can be applied at the ‘elbow’ link. The second link is the ‘shoulder’.

There are three possible actions.
But what are the states? How many states do we have?

Reinforcement Learning: Example Acrobot

1st episode: long sequence of random actions

400th episode: short sequence of ‘smart’ actions

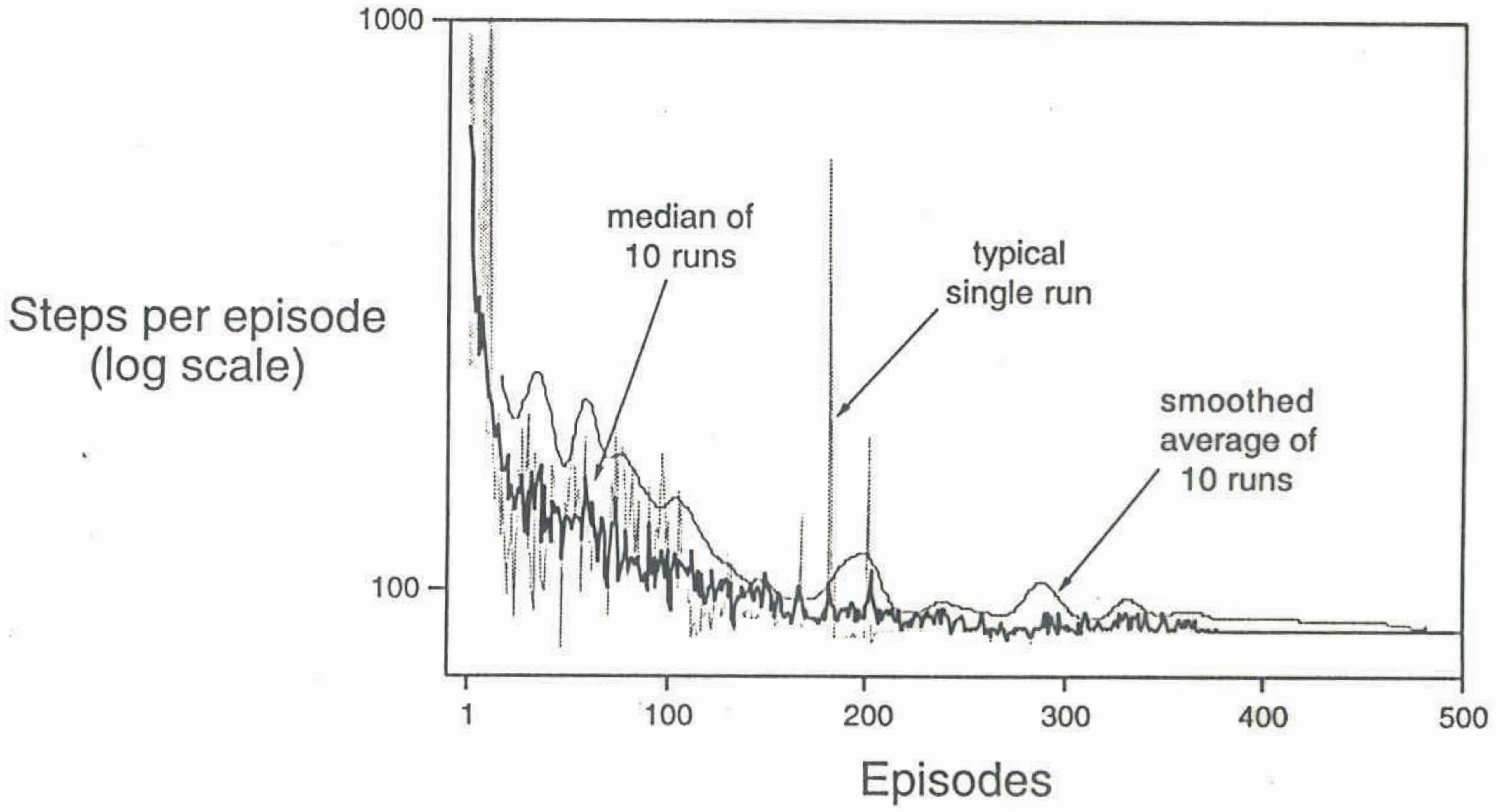


Figure 11.6 Learning curves for Sarsa(λ) on the acrobot task.

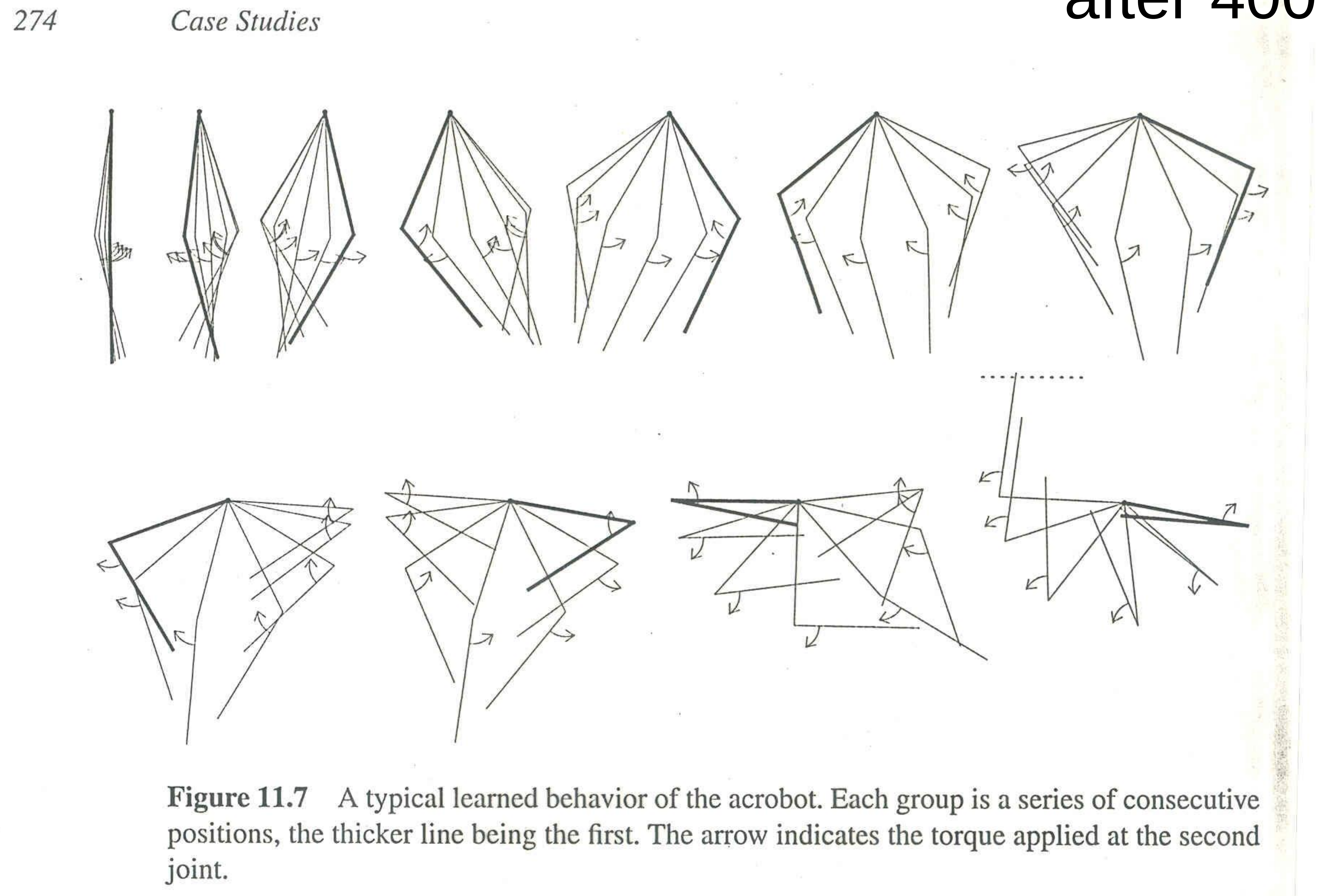
From Book:
Sutton and Barto

Previous slide.

An episode finishes if the target is reached. Over time episodes get shorter and shorter indicating that the acrobat has discovered (via reinforcement learning) a smart sequence of actions so as to reach the target (i.e., move the tip above the reference line)

Reinforcement Learning: Example Acrobot

after 400 episodes



274 Case Studies

Figure 11.7 A typical learned behavior of the acrobot. Each group is a series of consecutive positions, the thicker line being the first. The arrow indicates the torque applied at the second joint.

From Book:
Sutton and Barto

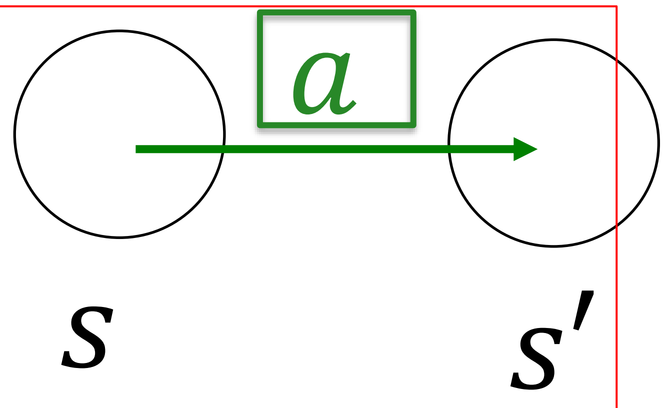
Previous slide.
One example of an action sequence, after learning, is shown.

Summary: Elements of Reinforcement Learning

There can be MANY states

Often need to discretize first

(→ later we try to model in continuum)



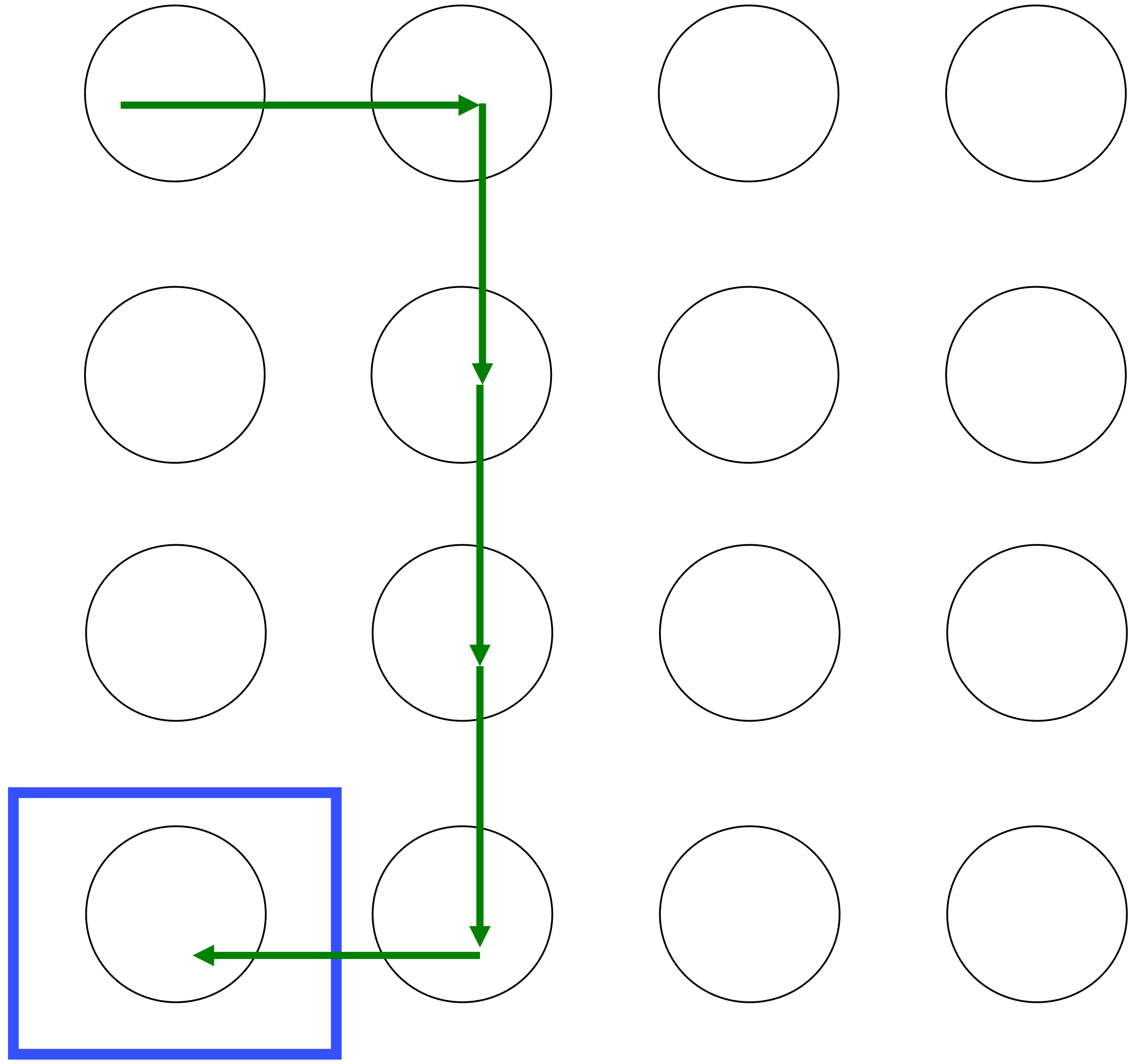
- discrete actions: a

- Mean reward for transition:

$$R^{a}_{s\to s'} = E\left(r \middle| s, a, s'\right)$$

- current actual reward: r_t

often most transitions have zero reward

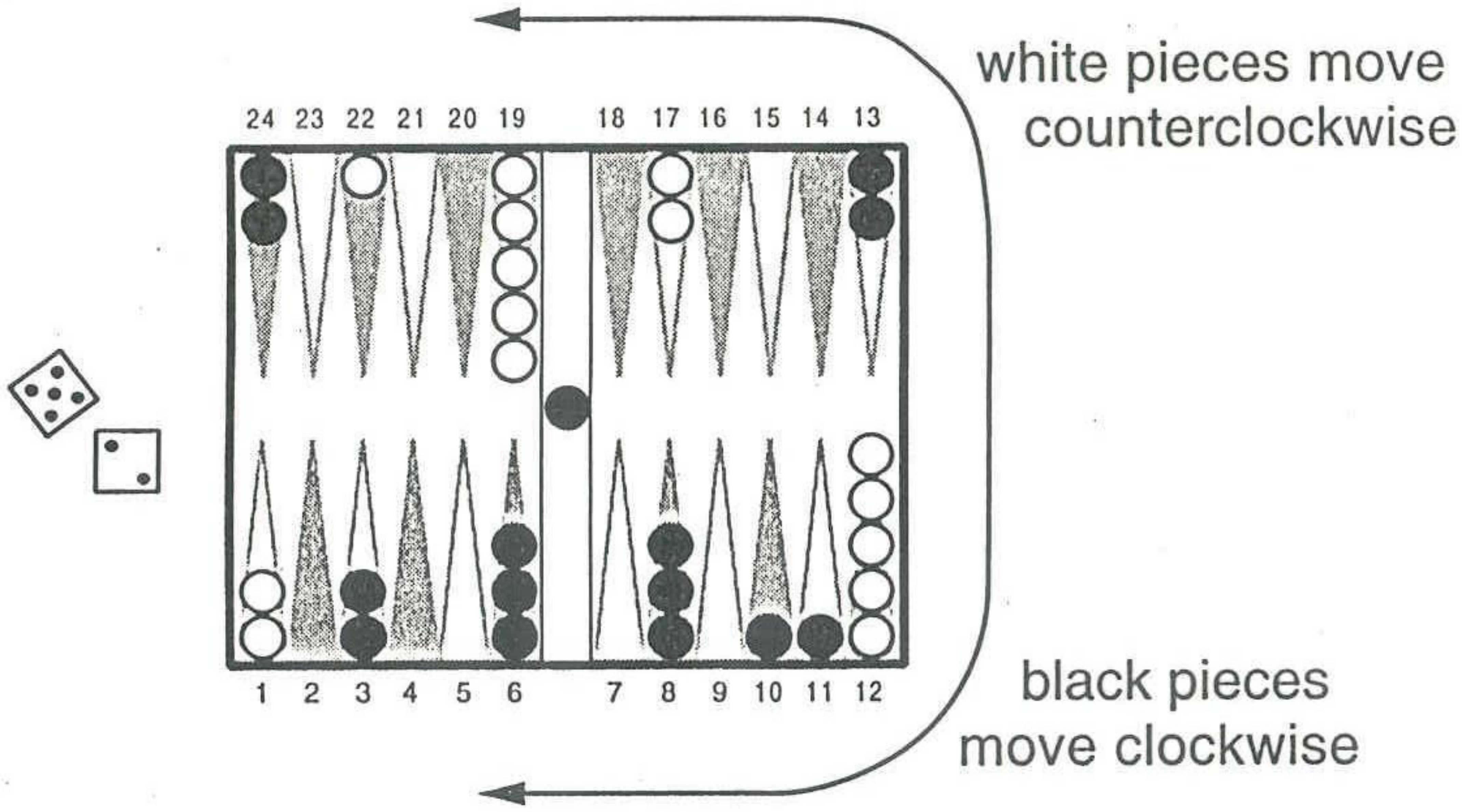


Previous slide.
Conclusion: In all practical situations, there is an enormous number of states.

In many situations we can think of the actions as discrete. For the moment we also think of the states as discrete (but next week we will go to continuous state space)

Quiz: Reinforcement Learning for backgammon

Case Studies



white pieces move counterclockwise
24 23 22 21 20 19 18 17 16 15 14 13
1 2 3 4 5 6 7 8 9 10 11 12
black pieces move clockwise

Figure 11.1 A backgammon position.

From Book:
Sutton and Barto

Game position = discrete states!

Suppose 2 pieces per player,
How many states in total?
[ ] 100<n<500
[ ] 500<n<5000
[ ] 5 000<n<50 000
[ ] n>50 000

Previous slide.

Backgammon game. There are 24 fields on the board. Players have several pieces. Pieces are protected if there are two of the same color on the same field.

To make it simply, we now consider that both players have two pieces each left. How many different states are there in total?

Reinforcement Learning and the Brain:

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Coarse Brain Anatomy and Reinforcement Learning

Previous slide.

Before we can make a link to Reinforcement Learning we need to know a bit more about the brain.

For most of the RL part, I also have videos on this page:
https://lcnwww.epfl.ch/gerstner/VideoLecturesRL-Gerstner.html

Video for this section:
https://www.youtube.com/watch?v=16d6XEO7sHY
Which is part 2 of lecture:
Reinforcement Learning and the Brain: **3-factor rules and brain-style computing**
on
https://lcnwww.epfl.ch/gerstner/VideoLecturesRL-Gerstner.html

Coarse Brain Anatomy and Reinforcement Learning

Reinforcement learning needs:

- states / sensory representation
  → where are states encoded in the brain?
- action selection
  → where is action selection encoded in the brain?
- reward signals
  → how is reward encoded in the brain?
  → is a ‘TD-error’ signal implemented in the brain?

Previous slide.

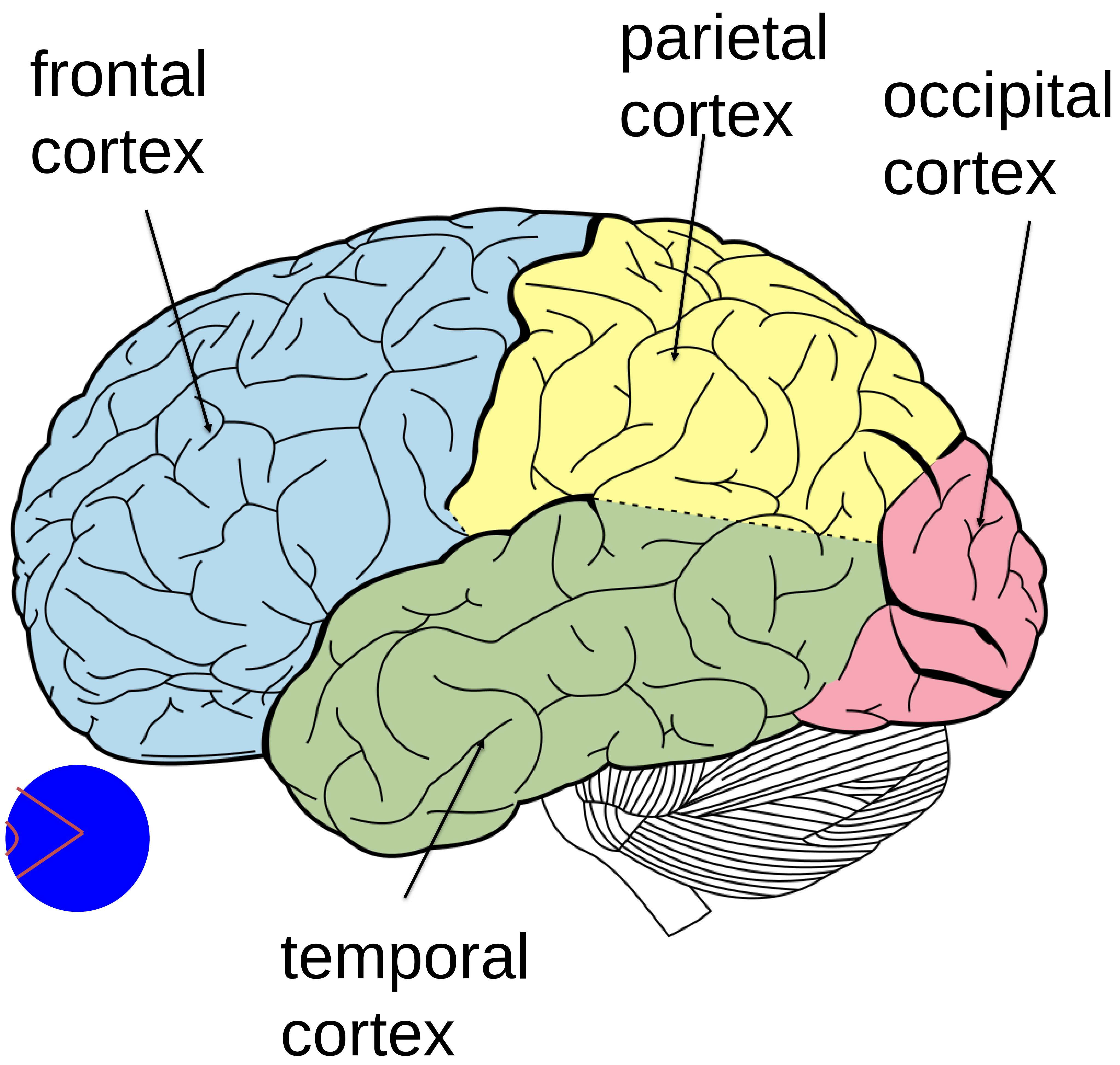
In reinforcement learning, the essential variables that define the update step of the learning rule are the states (defined by sensory representation), a policy for action selection, the actions themselves, and the rewards given by the environment.

If we want to link reinforcement learning to the brain, we will have to search for corresponding substrates and functions in the brain.
Therefore we now take a rather coarse and simplified look at the anatomy of the brain.

The Wikipedia articles give more information for those who are interested.

Coarse Brain Anatomy: Cortex

Sensory representation in visual/somatosensory/auditory cortex



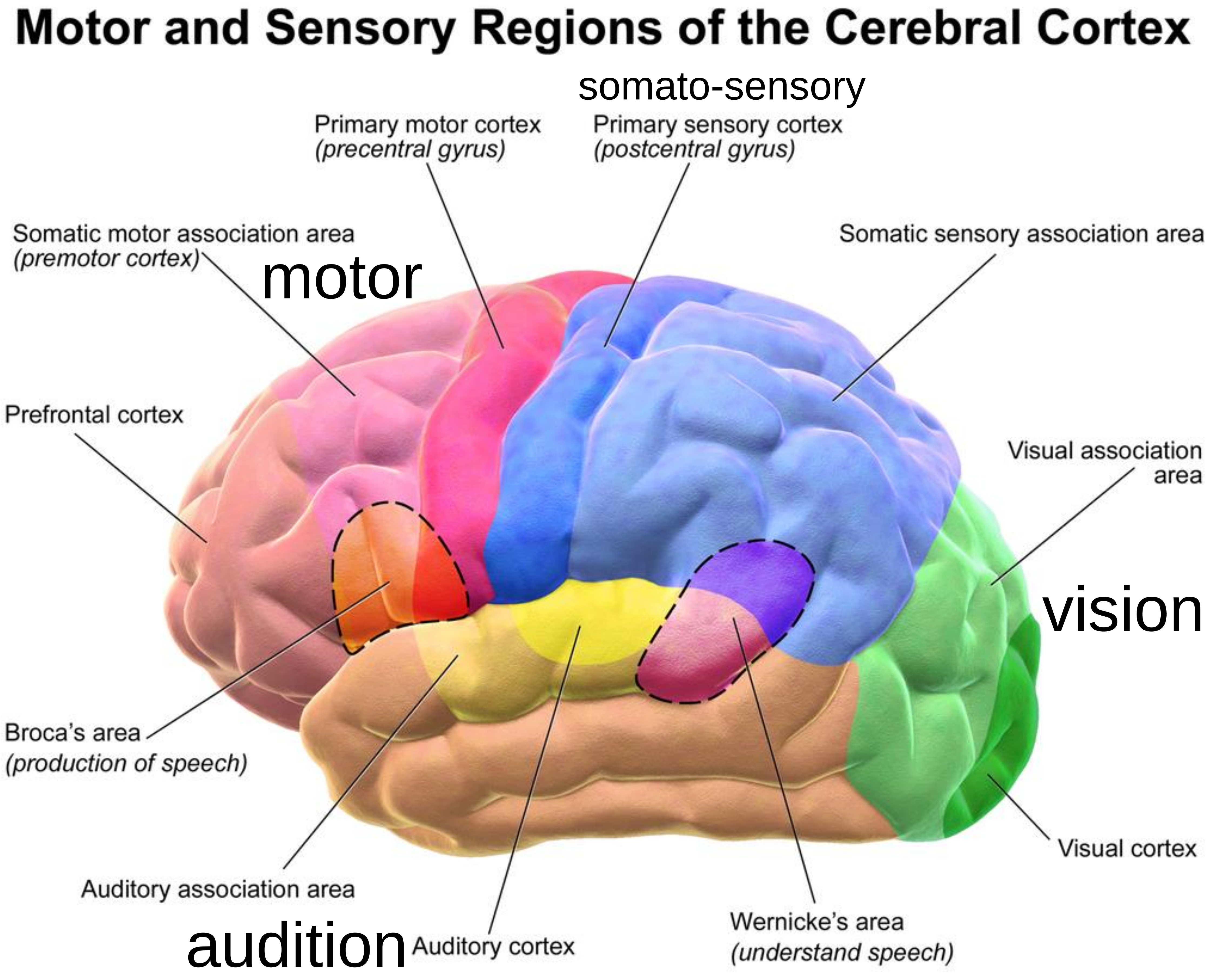


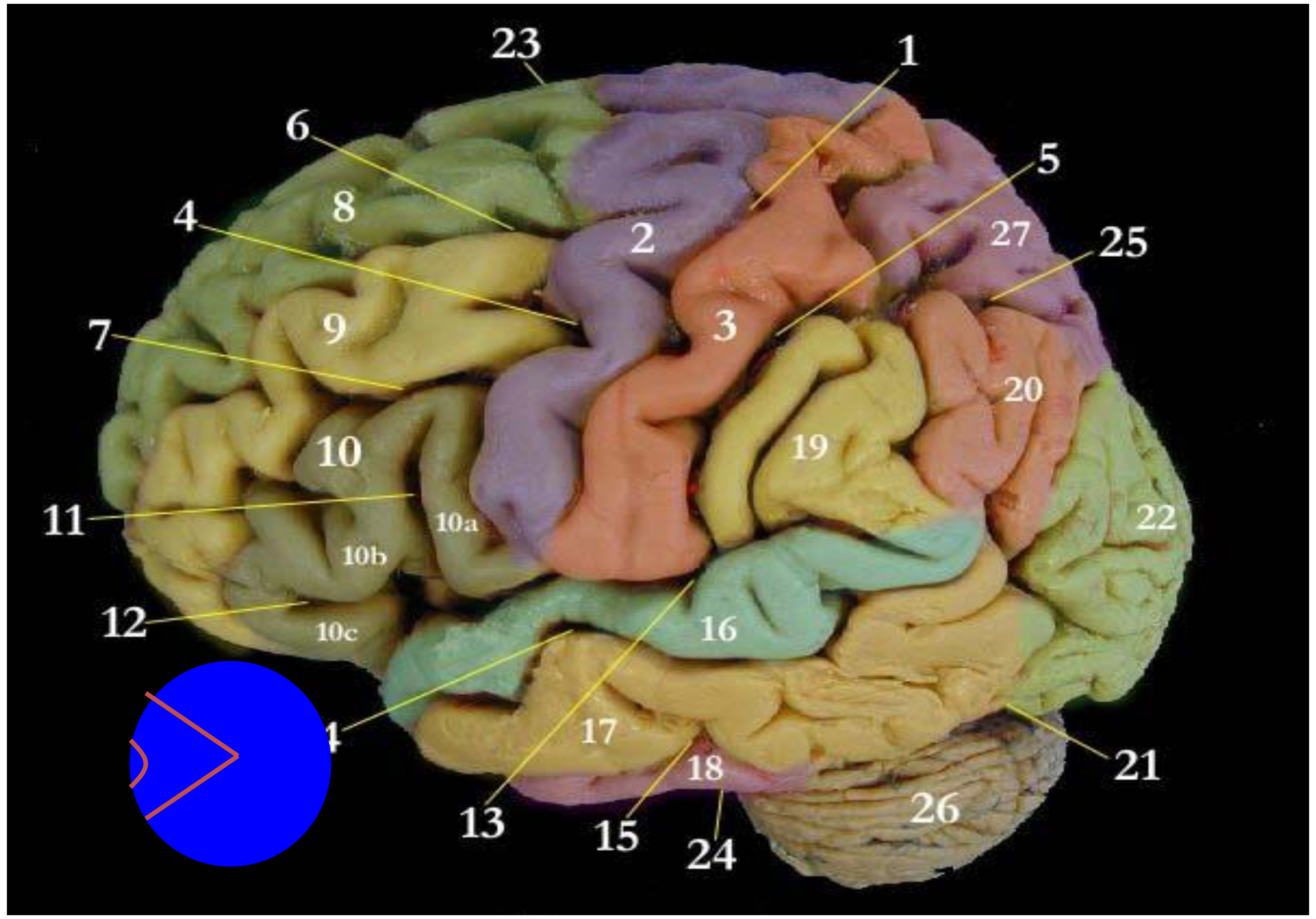
fig: Wikipedia

Previous slide.
Left: Anatomy. The Cortex is the part of the brain directly below the skull. It is a folded sheet of densely packed neurons. The biggest folds separate the four main parts of cortex (frontal, parietal, occipital, and temporal cortex)

Right: Functional assignments. Different parts of the brain are involved in different tasks. For example there several areas involved in processing visual stimuli (called primary and secondary visual cortex). Other areas are involved in audition (auditory cortex) or the presentation of the body surface (somatosensory cortex). Yet other areas are prepared in the preparation of motor commands for e.g., arm movement (primary motor cortex)

Coarse Brain Anatomy

- many different cortical areas
- but also several brain nuclei sitting below the cortex
- Some of these nuclei send dopamine signals
- Dopamine sent from: VTA and substantia nigra
- **Dopamine is related to reward, surprise, and pleasure**



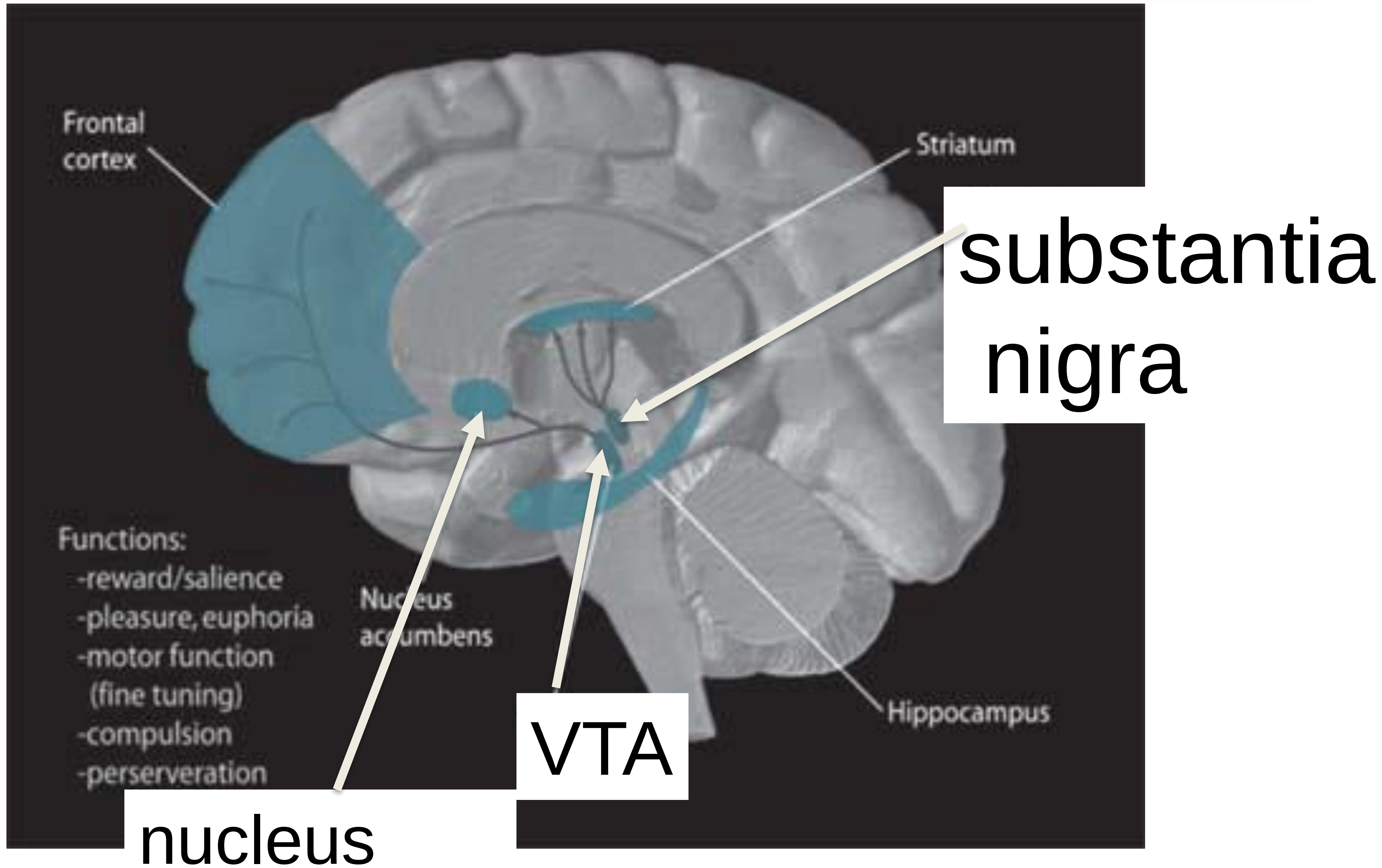


fig: Wikipedia commons

Previous slide.
Left: Anatomy. View on the folds of the cortex, and main cortical areas in different color.

Right: Below the cortex sit different nuclei. Some of these nuclei use dopamine as their signaling molecule. Important nuclei for dopamine are the Ventral Tegmental Area (VTA) and the Substantia Nigra pars compacte (SNc). These dopamine neurons send their signals to large areas of the cortex as well as to the striatum (and nucleus accumbens).
Since dopamine is involved in reward, these dopamine neurons will play a role in this lecture that links reinforcement learning and the brain.
Frontal Cortex is also involved in many aspects related to Reinforcement Learning.

In the next slides we will focus on striatum and hippocampus.

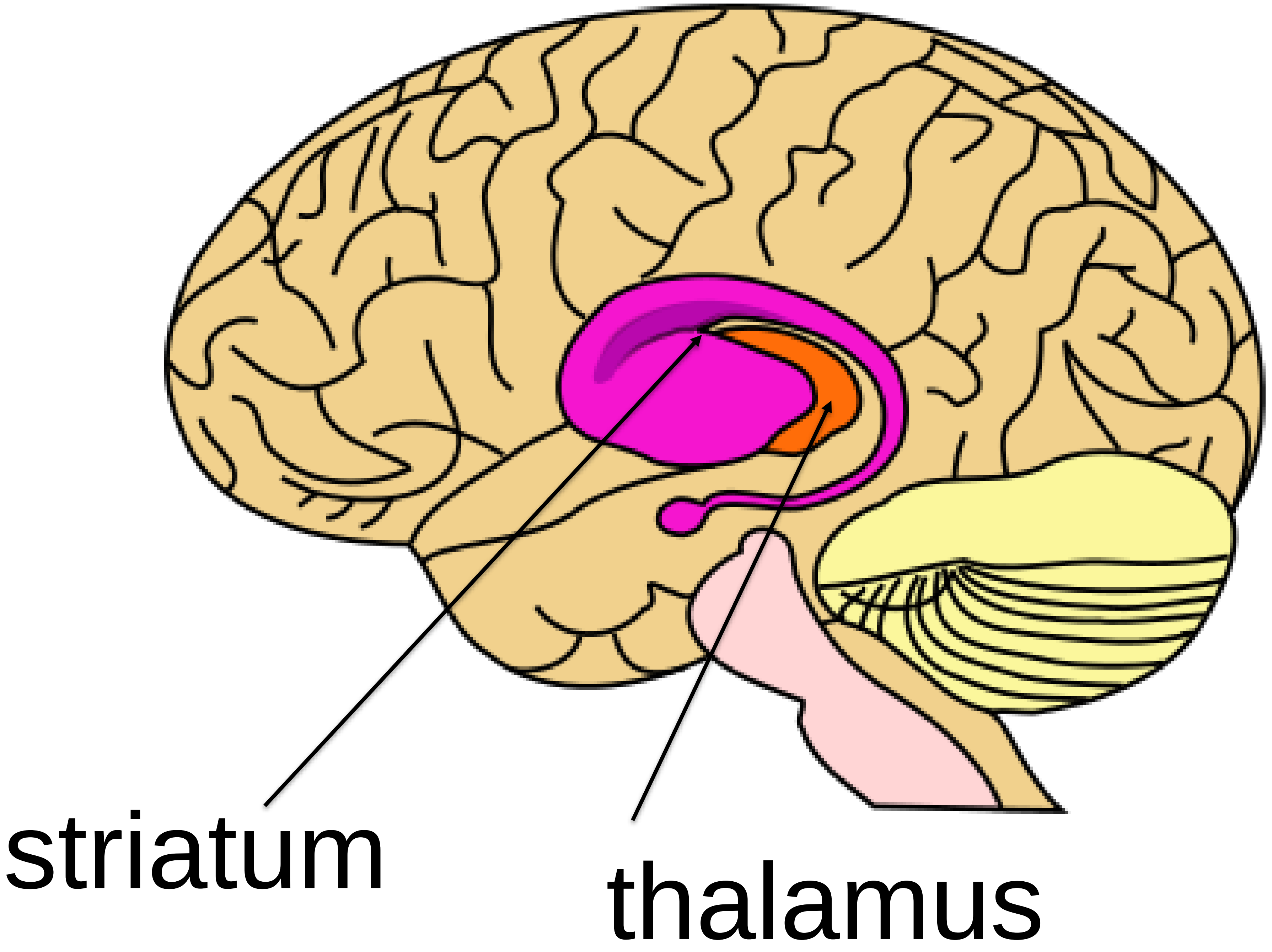
Coarse Brain Anatomy: Striatum

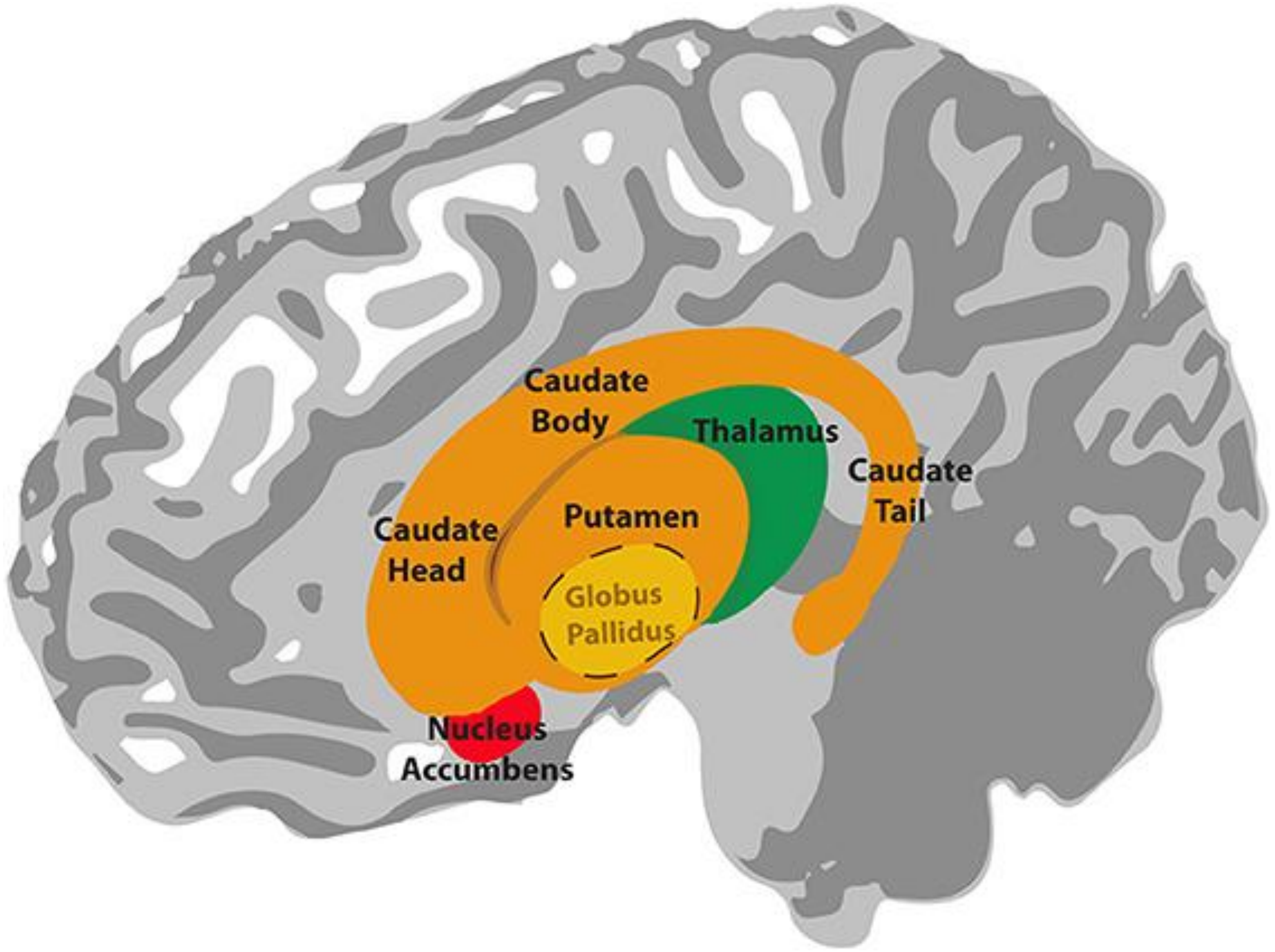
- Striatum sits below cortex
- Part of the ‘basal ganglia’
- **Dorsal striatum involved in action selection, decisions**

Striatum consists of

- Caudate (dorsal striatum)
- Putamen (dorsal striatum)

https://en.wikipedia.org/wiki/Striatum





Nucleus Accumbens is part of **ventral striatum**

fig: Wikipedia

Previous slide.

Left: Sketch of the Anatomical location of striatum and thalamus.

Right: the striatum lies also below the cortex. Since the striatum is involved in action selection it will play an important role in this lecture.

From Wikipedia:

The **striatum** is a nucleus (a cluster of neurons) in the subcortical basal ganglia of the forebrain. The striatum is a critical component of the motor and reward systems; receives glutamatergic and dopaminergic inputs from different sources; and serves as the primary input to the rest of the basal ganglia.

Functionally, the striatum coordinates multiple aspects of cognition, including both motor and action planning, decision-making, motivation, reinforcement, and reward perception.The striatum is made up of the caudate nucleus and the lentiform nucleus. The lentiform nucleus is made up of the larger putamen, and the smaller globus pallidus.

In primates, the striatum is divided into a **ventral striatum**, and a **dorsal striatum**, subdivisions that are based upon function and connections. The ventral striatum consists of the nucleus accumbens and the olfactory tubercle. The dorsal striatum consists of the caudate nucleus and the putamen. A white matter, nerve tract (the internal capsule) in the dorsal striatum separates the caudate nucleus and the putamen.[4] Anatomically, the term *striatum* describes its striped (striated) appearance of grey-and-white matter

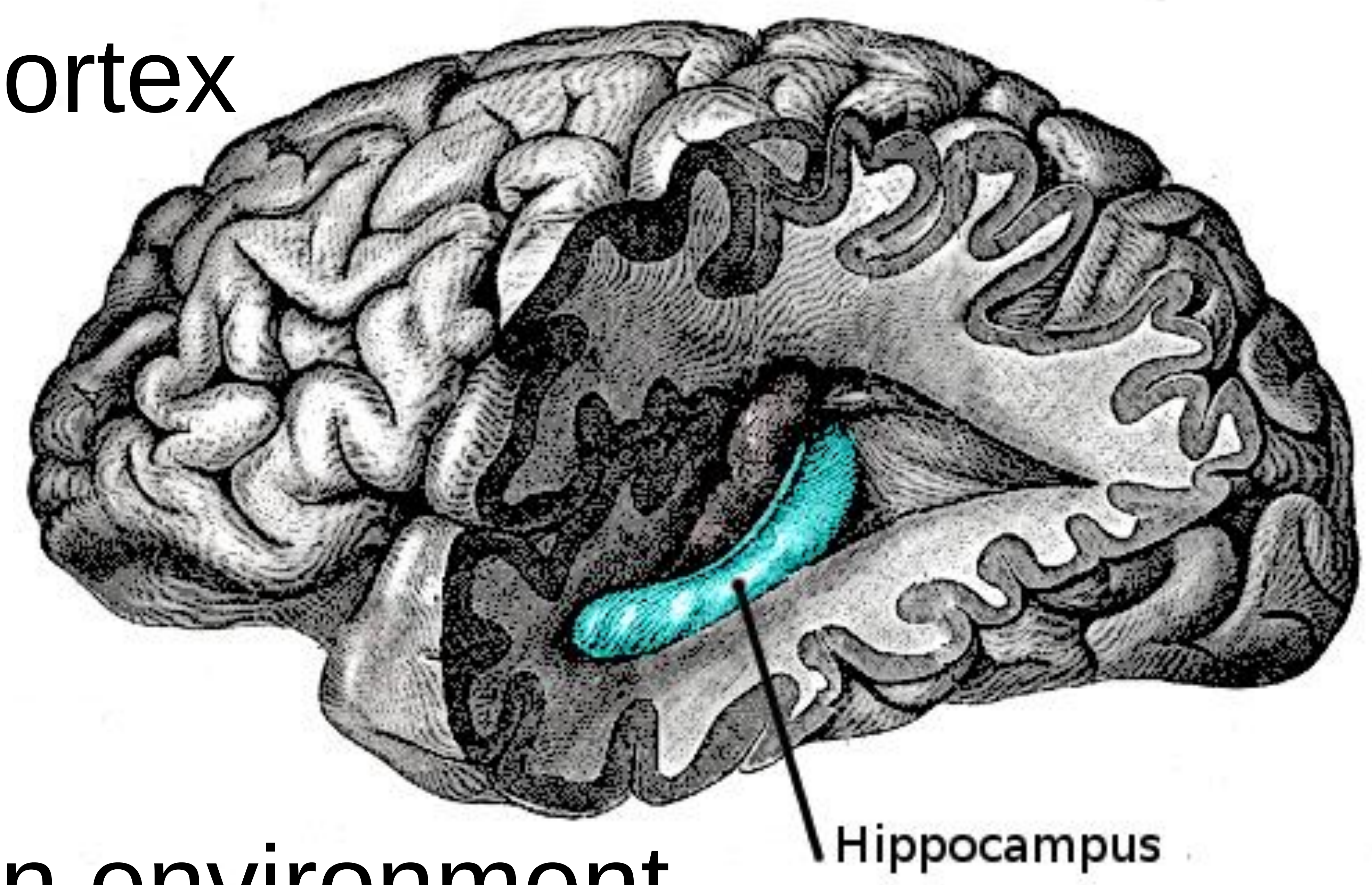
Coarse Brain Anatomy: hippocampus

Henry Gray (1918) *Anatomy of the Human Body*

fig: Wikipedia

Hippocampus

- sits below/part of temporal cortex
- involved in memory
- involved in **spatial** memory



Spatial memory:

knowing where you are,

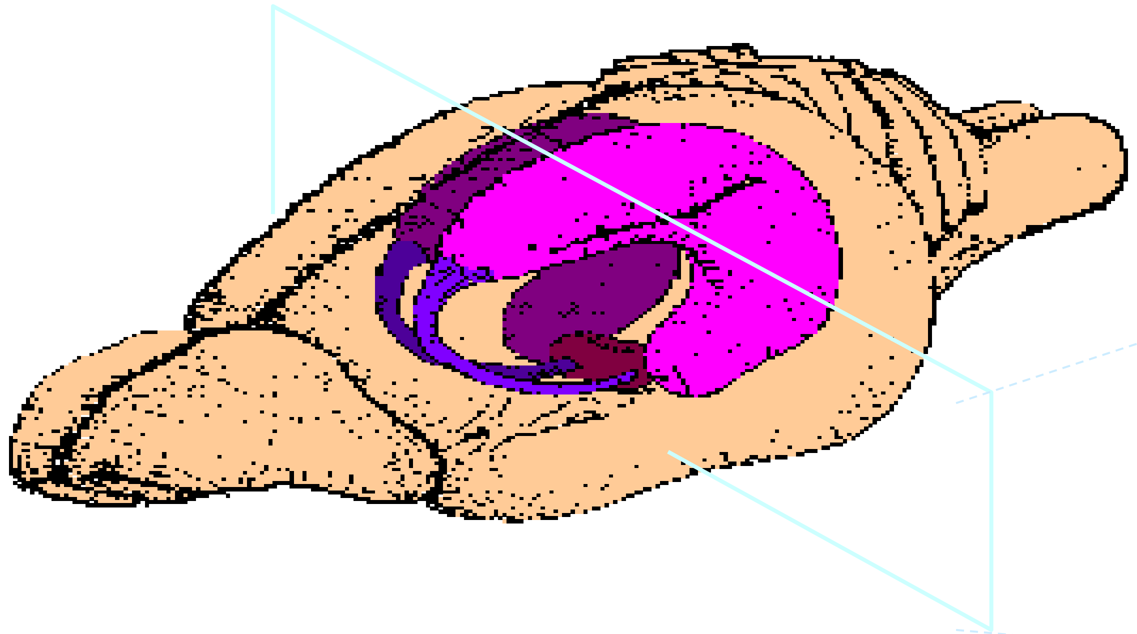
knowing how to navigate in an environment

Hippocampus involved in spatial memory
→ ‘state representation’

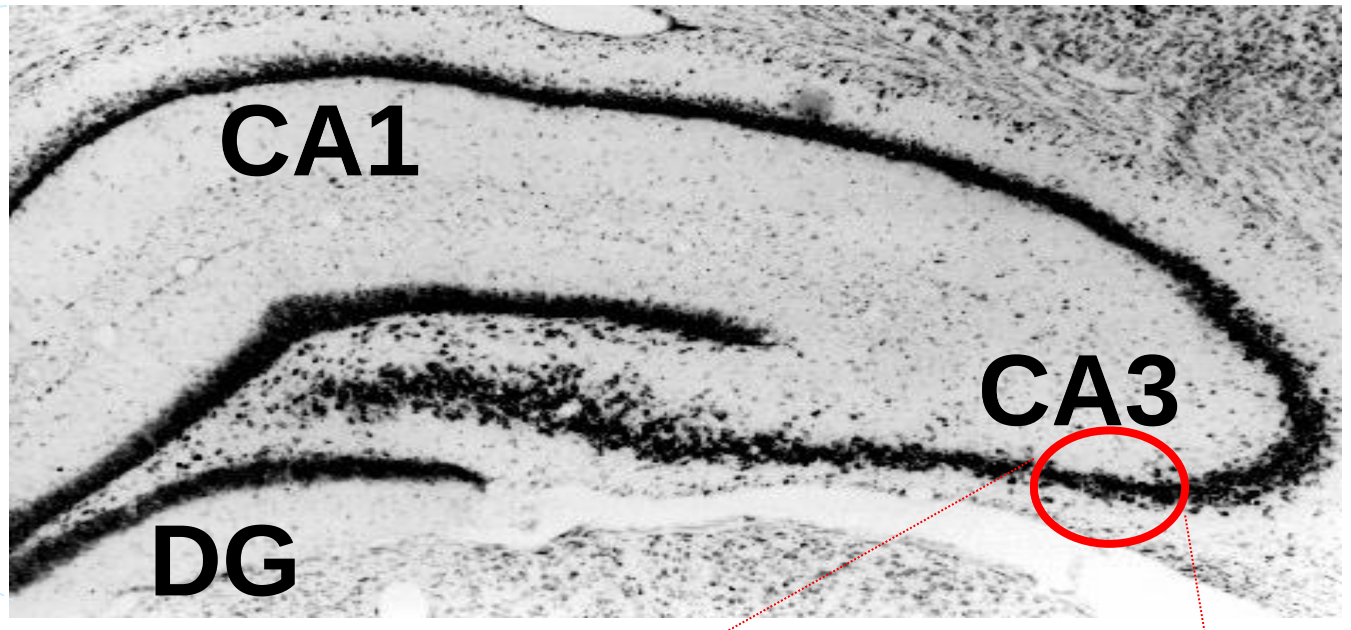
Previous slide.
From Wikipedia:

The **hippocampus** (named after its resemblance to the seahorse, from the Greek ἱππόκαμπος, "seahorse" from ἵππος *hippos*, "horse" and κάμπος *kampos*, "sea monster") is a major component of the brains of humans and other vertebrates. Humans and other mammals have two hippocampuses, one in each side of the brain. The hippocampus belongs to the limbic system and plays important roles in the consolidation of information from short-term memory to long-term memory, and in spatial memory that enables navigation. The hippocampus is located under the cerebral cortex (allocortical)[1][2][3] and in primates in the medial temporal lobe.

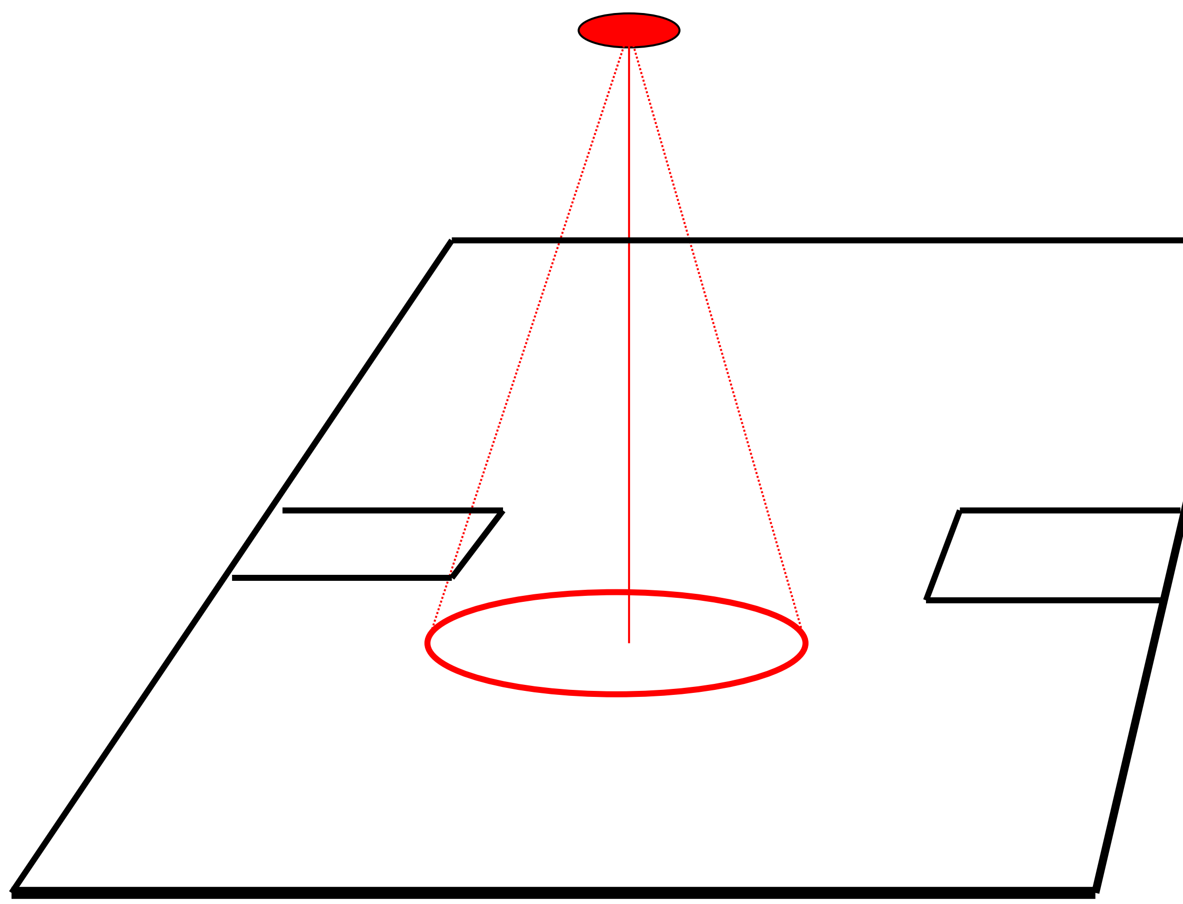
Hippocampal place cells

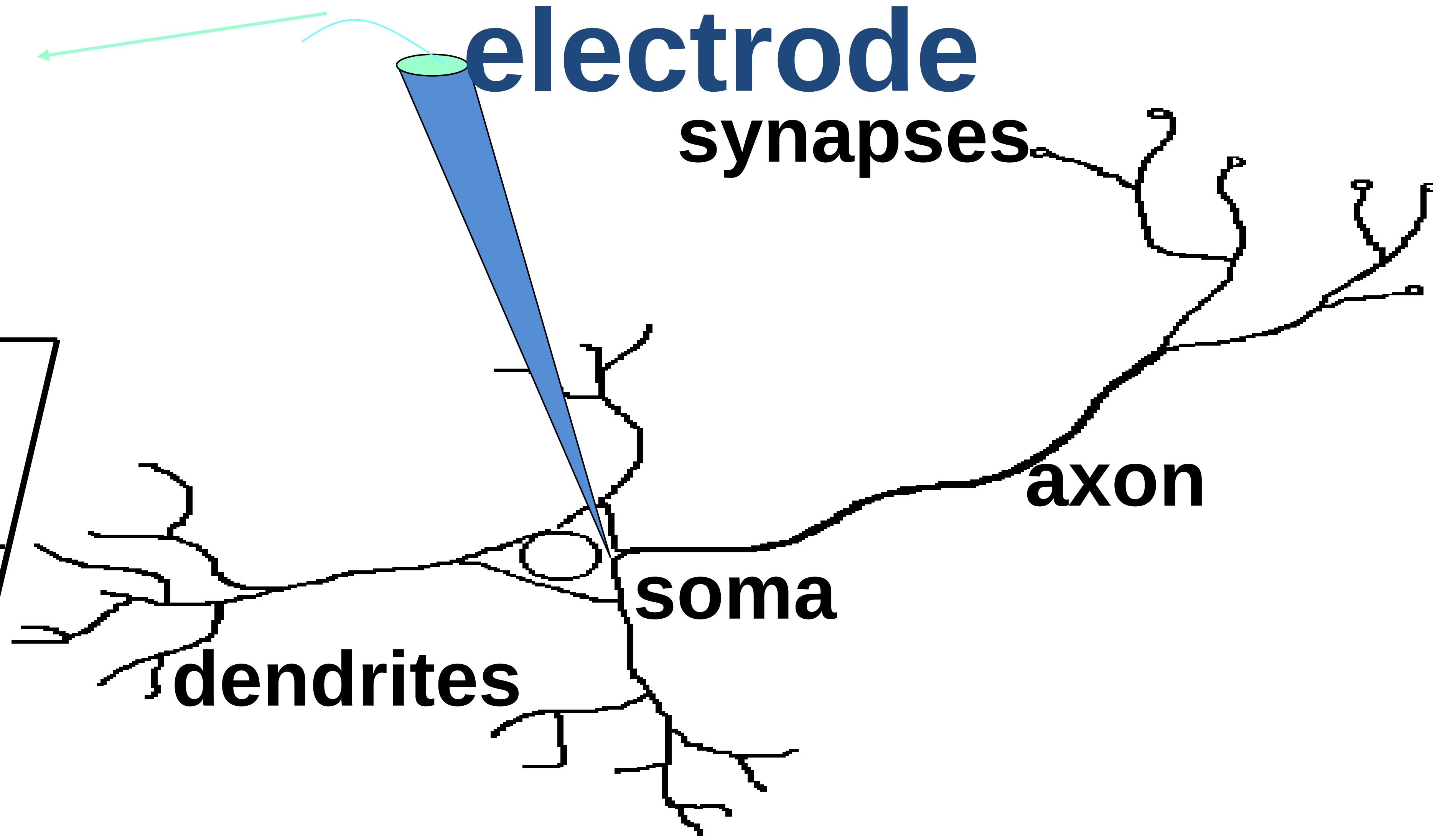


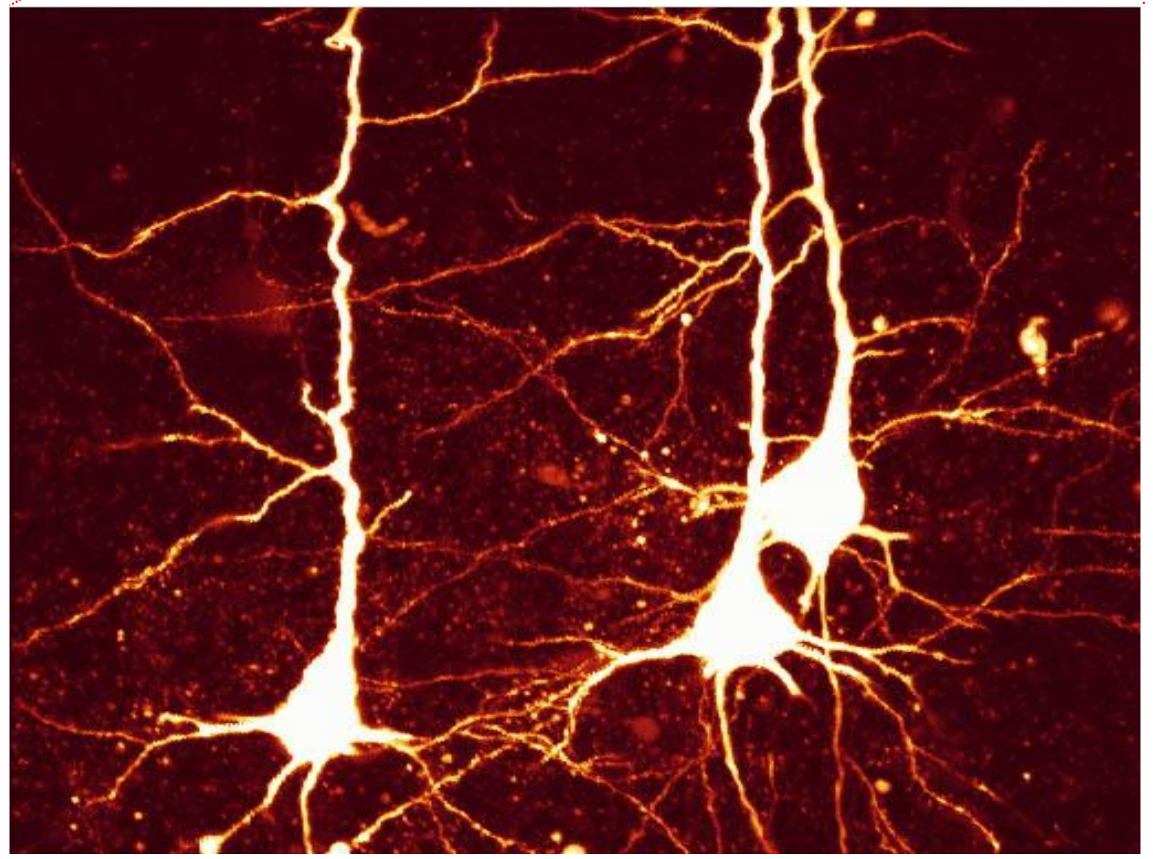
rat brain



Place fields







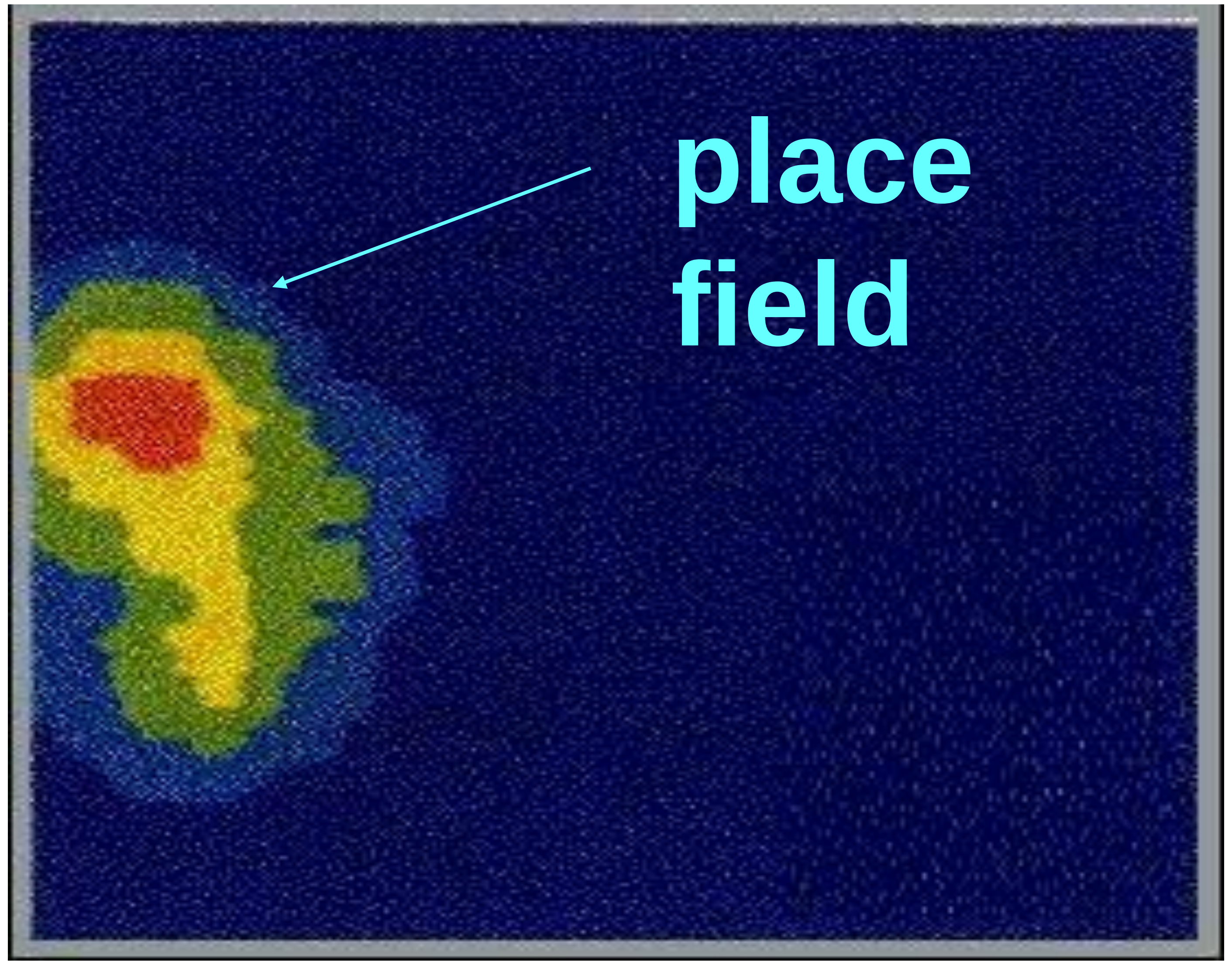
pyramidal cells

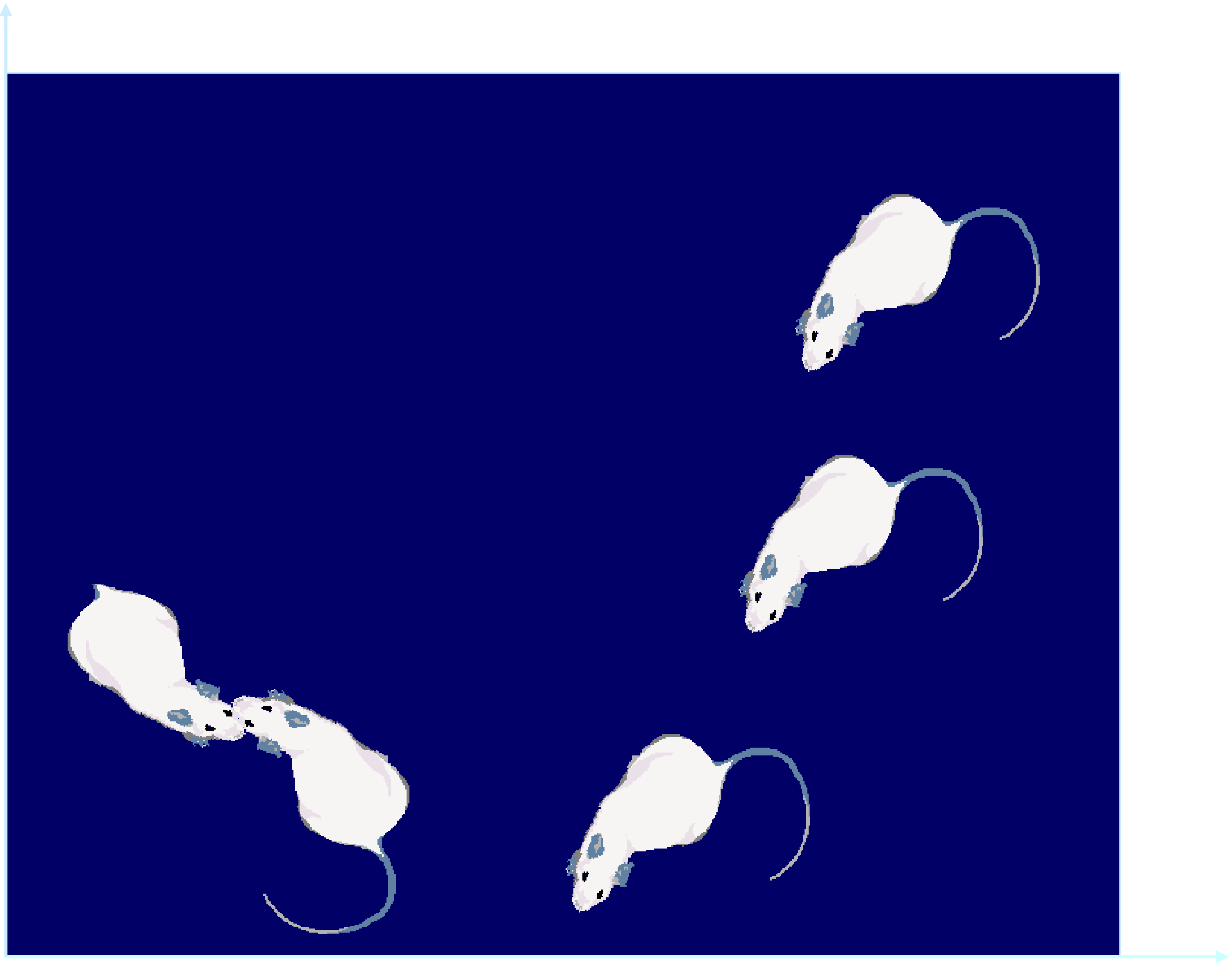
Previous slide.

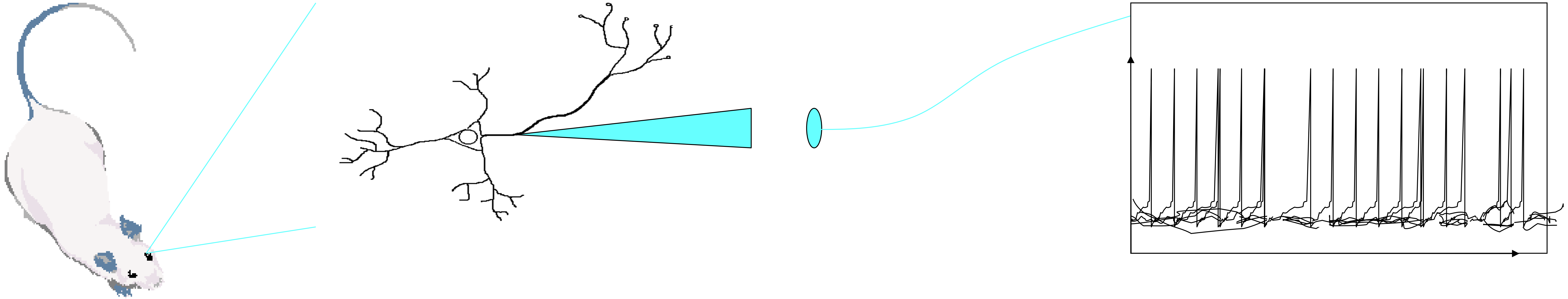
The upper left figure shows the rat brain with the hippocampus highlighted. Rats are animals that walk around a lot in their environment and have a large hippocampus. The next two images are a slice of hippocampus with three regions marked: Dentate Girus (DT), area CA3, and area CA1.
In CA 3 and CA1 it is common to find place cells, i.e., cells that respond only in a small region of the environment.

Hippocampal place cells

Main property: encoding the animal’s location







Previous slide.
in a large 2D environment, the place cells show activity in a localized region (place field). Importantly, this activity is mostly independent of the direction of the head or the walking direction. In that sense, the place cell is really a location in the environment, as opposed a certain configuration of visual cues on the retina.

It is known that the place field depends on (highly processed) visual input as well as self-motion information (path integration).

Coarse Brain Anatomy and Reinforcement Learning

Reinforcement learning needs:

- representation of states / sensory input / ‘where’
  → hippocampus? / sensory cortex?

- action selection → striatum?, motor cortex?

- reward signals → dopamine?

→ Candidate brain areas and brain signals!

Previous slide.

In reinforcement learning, the essential variables are the states (defined by sensory representation), a policy for action selection, the actions themselves, and the rewards given by the environment.

If we want to link reinforcement learning to the brain, we will have to search for corresponding substrates and functions in the brain.

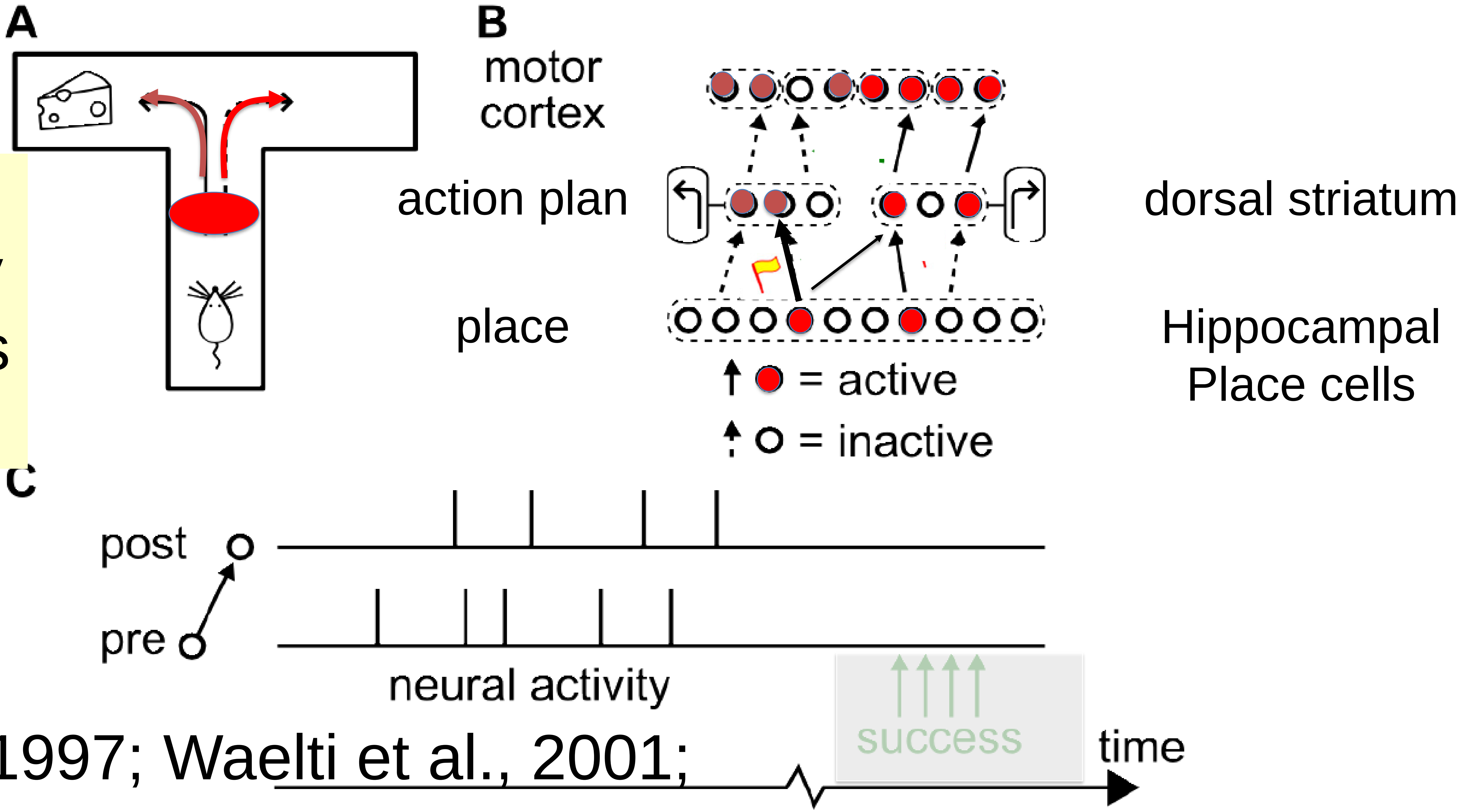
The potential relations show candidate brain region for a mapping to state, actions, and reward. The above rough ideas need to be defined during the rest of this lecture.

Action Learning reconsidered

Image: Fremaux and Gerstner, Front. Neur. Circ., 2015

Eligibility trace:
Synapse keeps memory of pre-post coincidences over a few seconds

Dopamine: Reward/success



Schultz et al. 1997; Waelti et al., 2001;

→ **Reinforcement learning: success = reward – (expected reward)**

TD-learning, SARSA, Policy gradient   (book: Sutton and Barto, 2018)

Previous slide (repetition).
Hebbian learning as it stands is not sufficient to describe learning in a setting were rewards play a role. If joint activity of pre- and post causes stronger synapses, the rat is likely to repeat the same unrewarded action a second time. A three-factor rule adds the influence of a neuromodulator (e.g., dopamine): reward-modulate plasticity.

Hypothetical functional role of neuromodulated synaptic plasticity.
(A) Schematic reward-based learning experiment. An animal learns to perform a desired sequence of actions (e.g.,move straight,then turn left) in a T-maze through trial-and-error with rewards (cheese).
(B) The current position (“place”) of the animal in the environment is represented by an assembly of active cells in the hippocampus, called place cells. These cells connect to neurons (e.g.,in the dorsal striatum) which code for high-level actions at the decision point, e.g., “turn left” or “turn right.” These neurons in turn project to motorcortex neurons, responsible for the detailed implementation of actions. Connections between neurons that are active together are marked (flag/eligibility trace).
(C) Neuromodulator timing. While spikes occur on the time scale of milliseconds, the success signal (green arrows/shaded) may come a few seconds later.

1. Quiz: Coarse Functional Brain anatomy

[ ] the brain = the cortex (synonyms)
[ ] the cortex consists of several areas
[ ] some areas are more involved in controlling motor output, others in the representation of the body surface
[ ] below the cortex there are groups (clusters) of neurons
[ ] Hippocampus sends out dopamine signals
[ ] VTA sends out dopamine signals

[ ] dopamine is linked to reward, pleasure, surprise
[ ] striatum is involved in action selection
[ ] hippocampus is involved in the representation of ‘WHERE’

Previous slide. Your comments

Reinforcement Learning Lecture 1
Reinforcement Learning and SARSA

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Part 3: One-step horizon (bandit problems)

- Examples of Reward-based Learning
- Elements of Reinforcement Learning
- **One-step horizon (bandit problems)**

Previous slide.

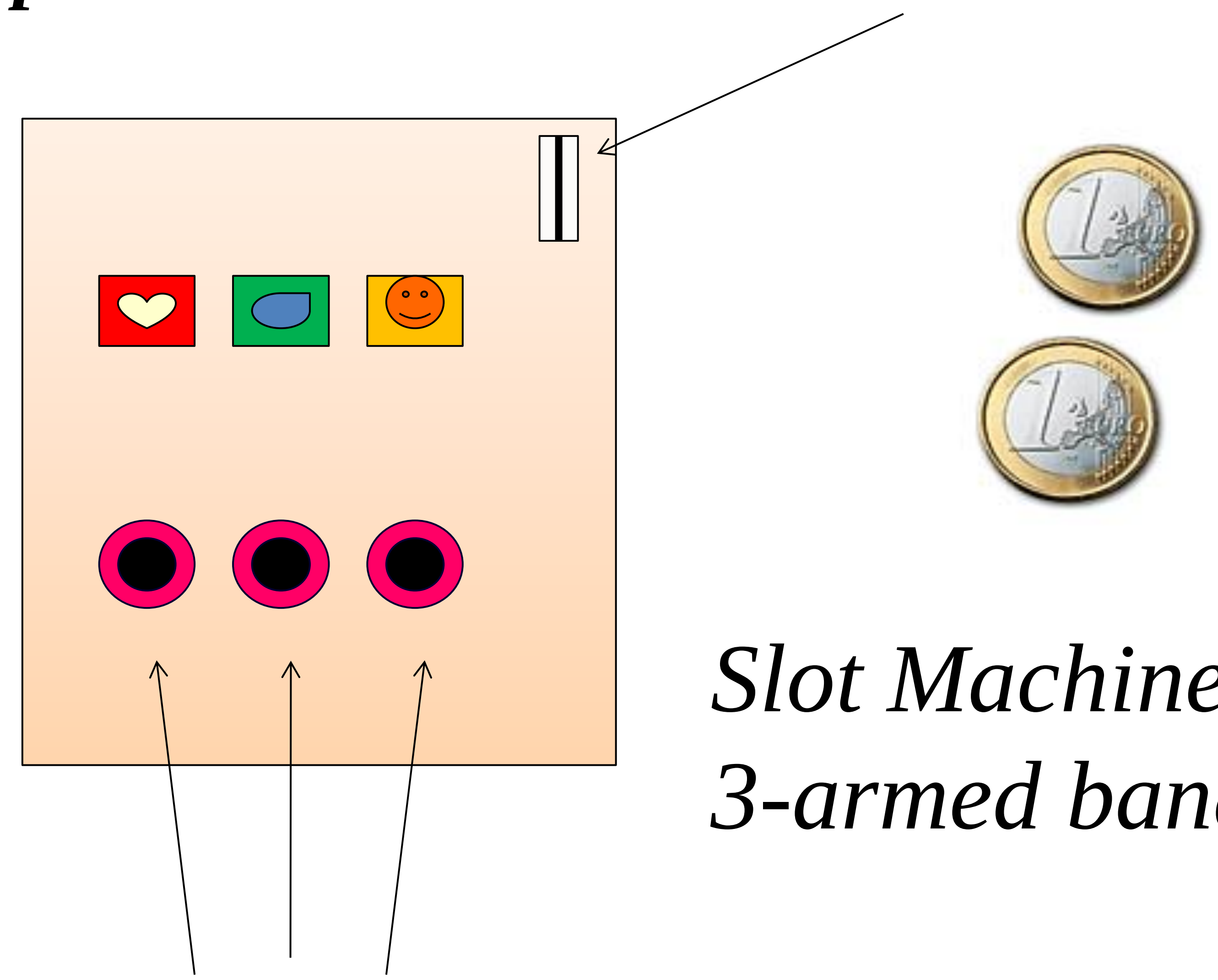
We start with the simplest discrete example: the agent takes an action. Immediately afterwards, the game is over and reward is given. I call this a one-step scenario or 1-step horizon.

Video for most of this lecture:

RL Lecture 1 on https://lcnwww.epfl.ch/gerstner/VideoLecturesRL-Gerstner.html
Parts 1-3.

One-step horizon games (bandit)

action=button press



coins

buttons

Slot Machine
3-armed bandid

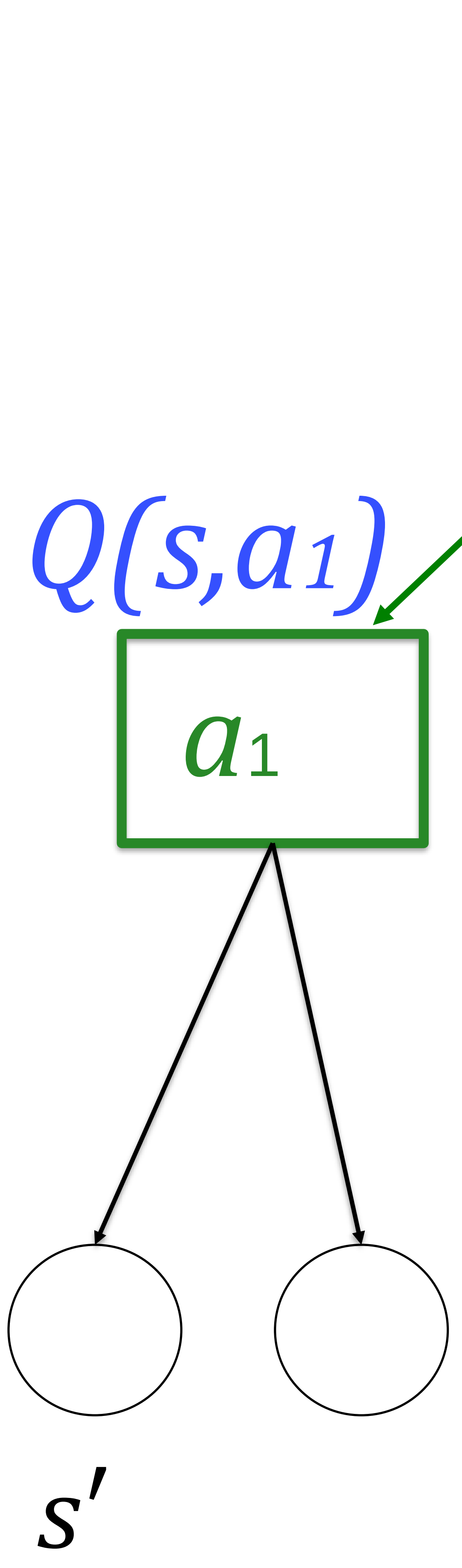
Previous slide.

The standard example is a multi-armed bandit, or slot machine: you have to choose between a few actions, and once you have pressed the button you can just wait and see whether you get reward or not.

One-step horizon games

Blackboard1:
Q-values

Q-value: $Q(s,a)$
Expected reward for action a starting from s



Previous slide.

One of the most central notion in reinforcement learning is the Q-value.
Q(s,a) has two indices: you start in state s and take action a.
The Q-value Q(s,a) is (an estimate of) the mean expected reward that you will get if you take action a starting from state s.

One-step horizon games

Blackboard1:
Q-values

Your notes.

One-step horizon games: Q-value

Q-value $Q(s,a)$

Expected reward for action a starting from s

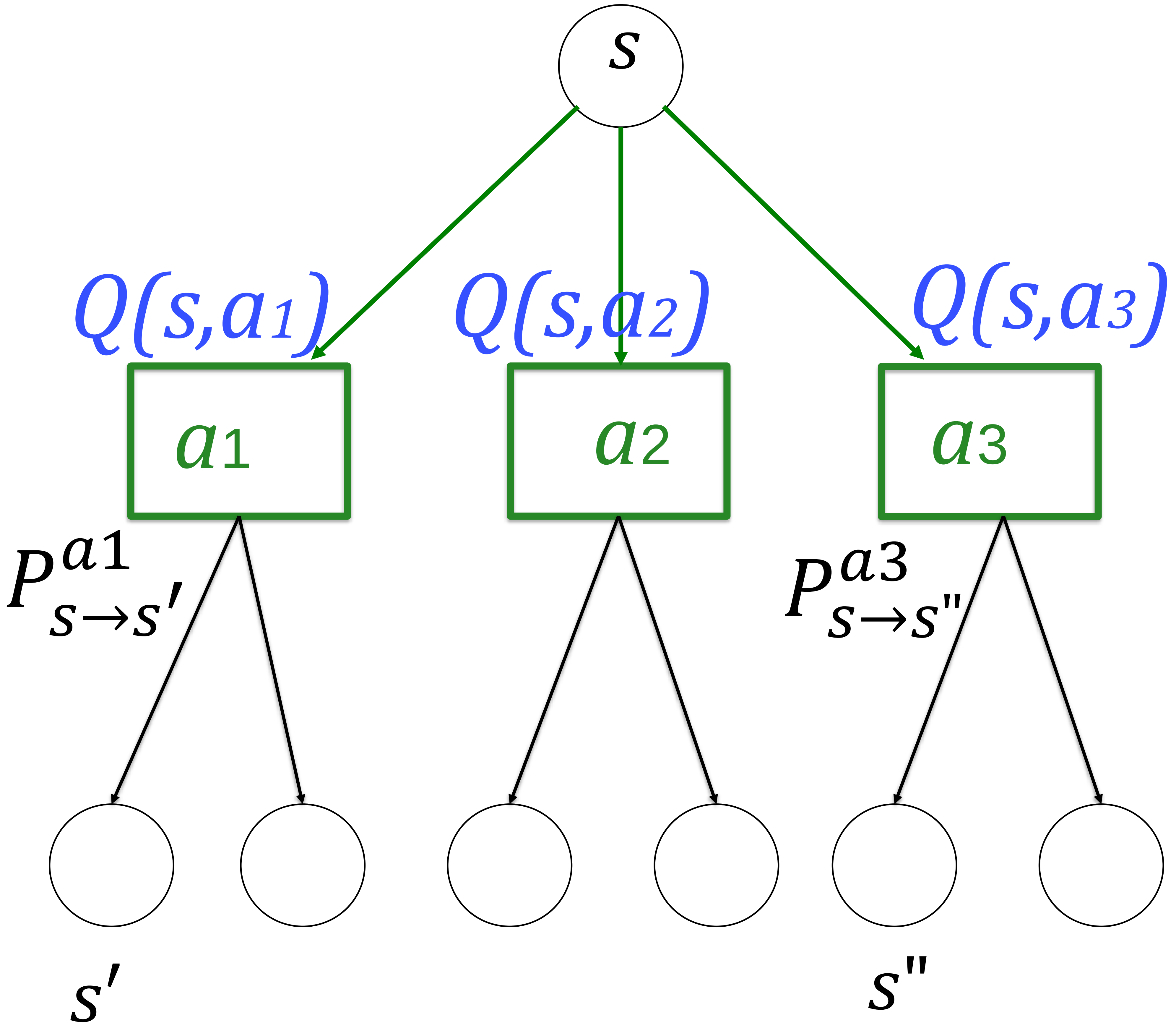
$$Q(s,a) = \sum_{s'} P^a_{s\to s'} R^a_{s\to s'}$$

Reminder:

$$R^a_{s\to s'} = E(r|s',a,s\ )$$

Similarly:

$$Q(s,a) = E(r|s,a\ )$$



Now we know the Q-values: which action should you choose?

Previous slide.

$P_{s \to s'}^{a1}$ is the probability that you end up in a specific state s’ if you take action a1 in state s.
We refer to this sometimes as the ‘branching ratio’ below the ‘actions’.

Q(s,a) is attached to the branches linking the state s with the actions.

actions are indicated by green boxes; states are indicated by black circles.

The mean reward $R_{s \to s'}^{a}$ is defined as the expected reward given that you start in state s with action a and end up in state s’ (see Blackboard 1).

Given the branching ratio and the mean rewards, it is easy to calculate the Q-values (Blackboard 1).

Optimal policy (greedy)

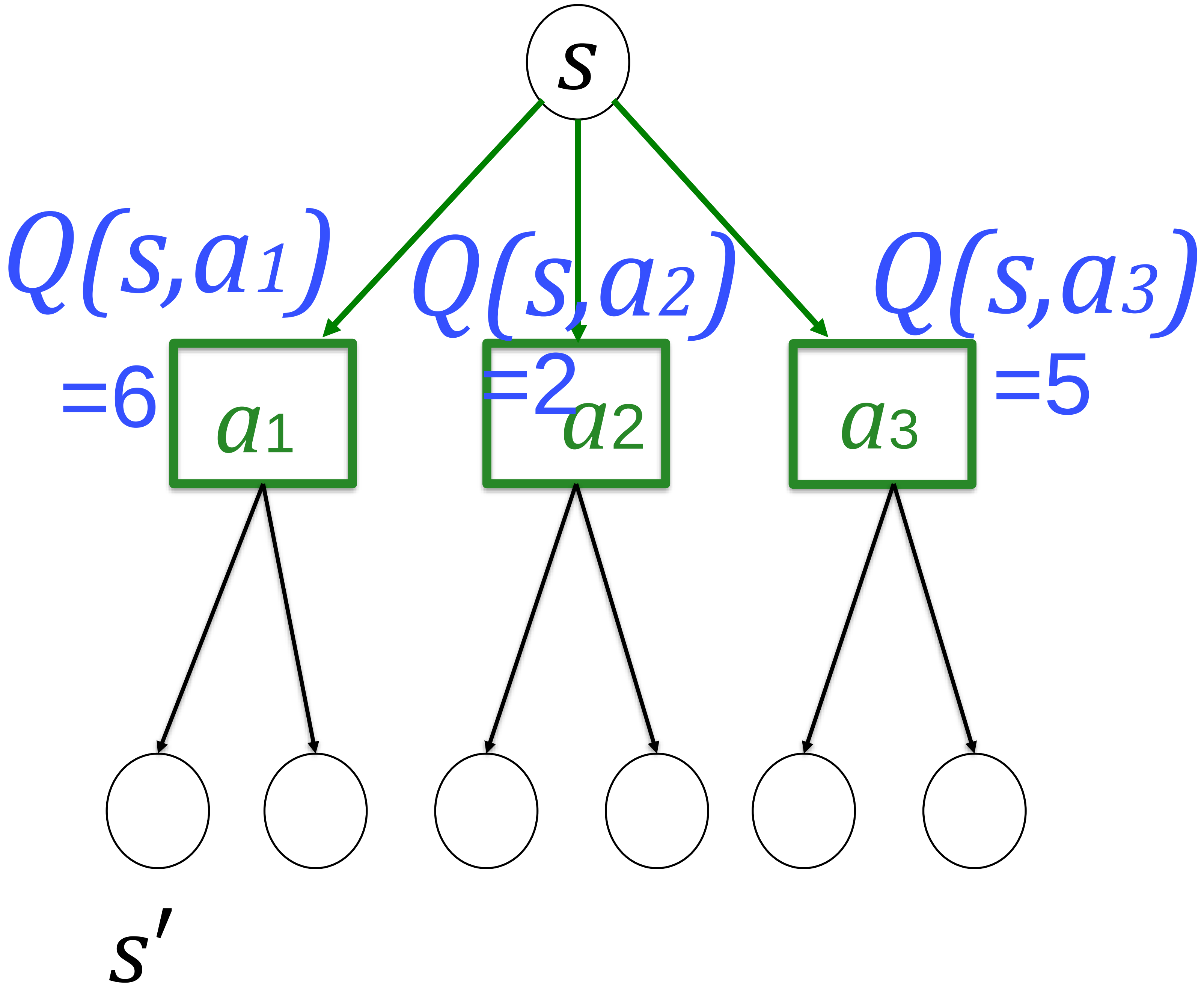
Suppose all Q-values are known:

take action a with*

$$Q(s,a^*) \geq Q(s,a_j)$$

↑

other actions



optimal action:

$$a^* = argmax_a\,[Q(s,a)]$$

Optimal policy is also called ‘greedy policy’

Previous slide.

And once you have the Q-values it is easy to choose the optimal action:

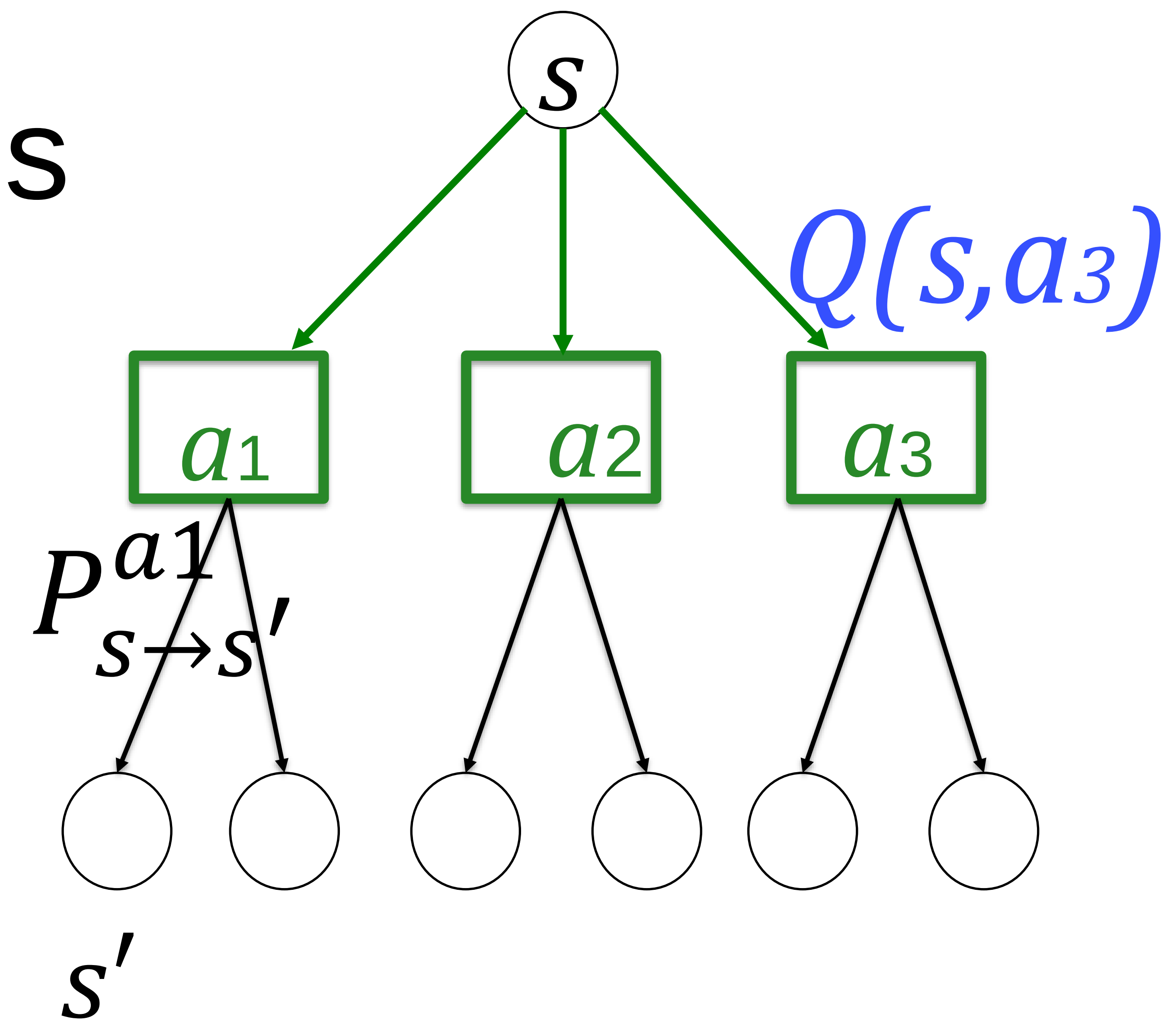
Just take the one with maximal Q-value.

One-step horizon games

Q-value = expected reward for state-action pair

If Q-value is known, choice of action is simple
→ take action with highest Q-value

BUT: we normally do not know the Q-values
→ estimate by trial and error



Previous slide.
The only remaining problem is that we do not know the Q-values, because the casino gives you neither the branching ratio nor the reward scheme.

Hence the only way to find out is by trial and error (that is, by playing many times – the casino will love this!).

Teaching monitoring – monitoring of understanding

[ ] today, up to here, at least 60% of material was new to me.

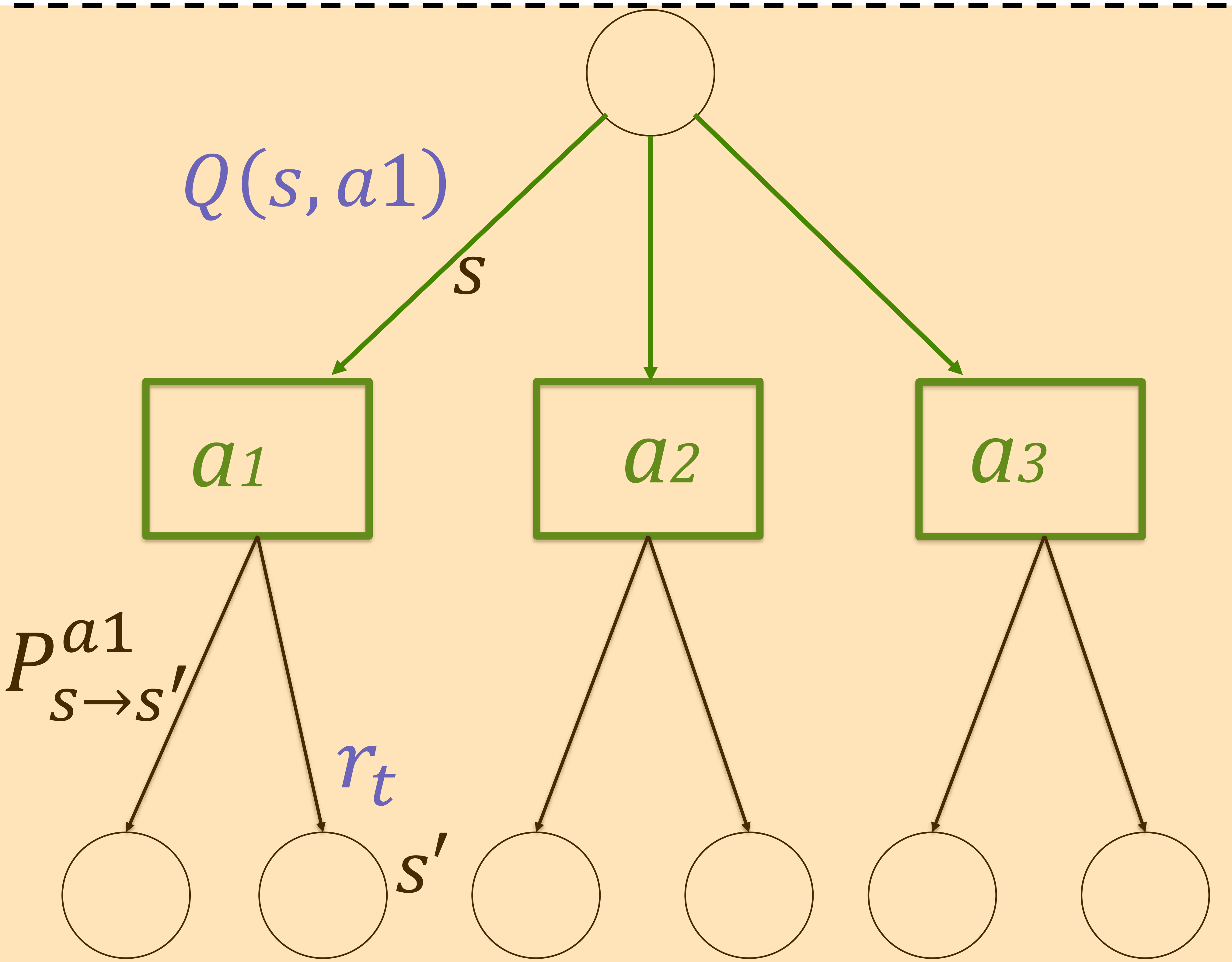
[ ] up to here, I have the feeling that I have been able to follow (at least) 80% of the lecture.

Previous slide.
Teaching monitoring – feedback for the teacher.

Exercise 1 (Exercise session)

Expected reward

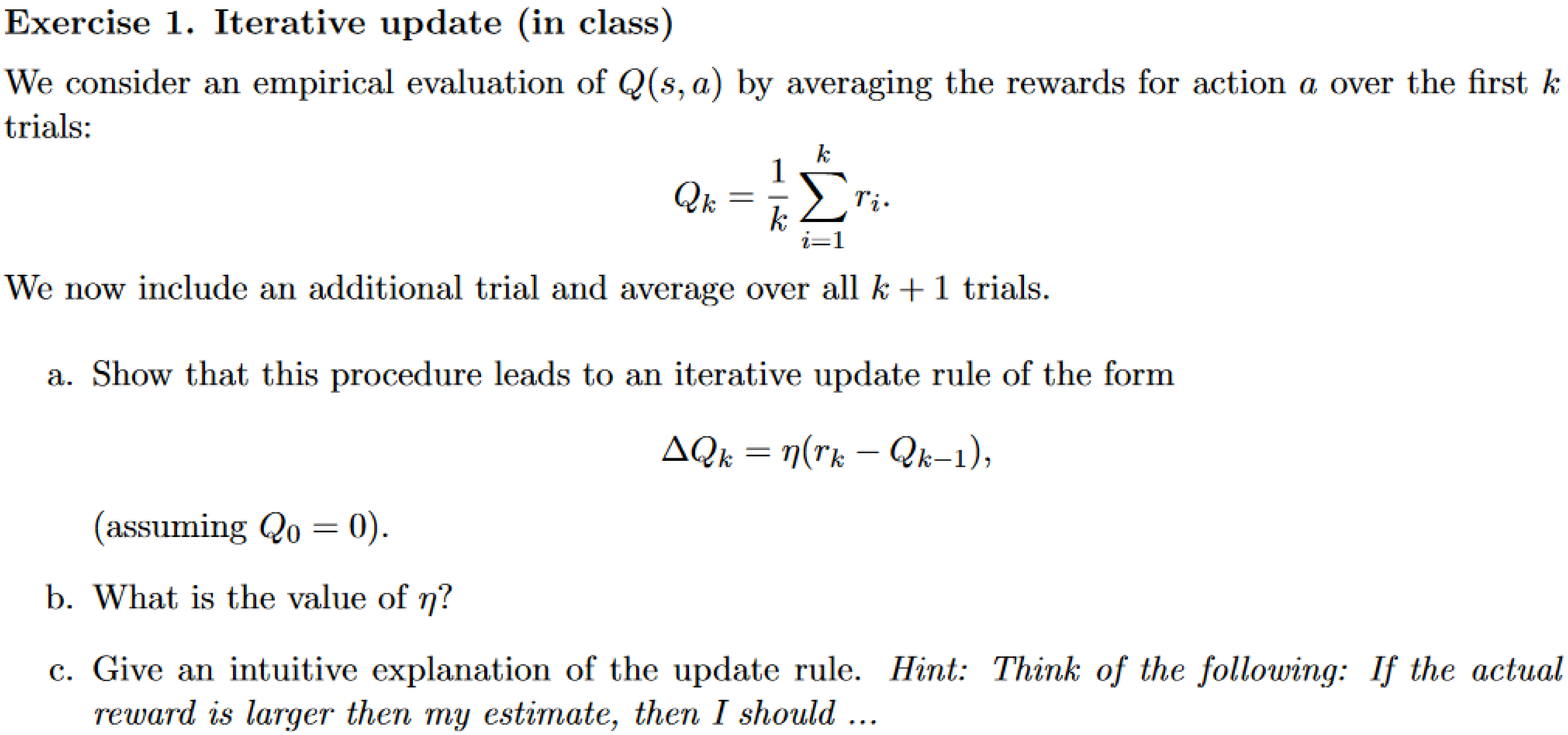
$$Q(s,a) = \sum_{s'} P^{a}_{s\to s'} \, R^{a}_{s\to s'}$$



Show that empirical averaging over k trials gives an update rule

$$\Delta Q(s,a) = \eta\,[r_t - Q(s,a)]$$

Exercise 1 (in class)



Exercise 1. Iterative update (in class)

We consider an empirical evaluation of $Q(s,a)$ by averaging the rewards for action a over the first k trials:

$$Q_k = \frac{1}{k}\sum_{i=1}^{k} r_i.$$

We now include an additional trial and average over all $k+1$ trials.

a. Show that this procedure leads to an iterative update rule of the form

$$\Delta Q_k = \eta(r_k - Q_{k-1}),$$

(assuming $Q_0 = 0$).

b. What is the value of η?

c. Give an intuitive explanation of the update rule. *Hint: Think of the following: If the actual reward is larger then my estimate, then I should …*

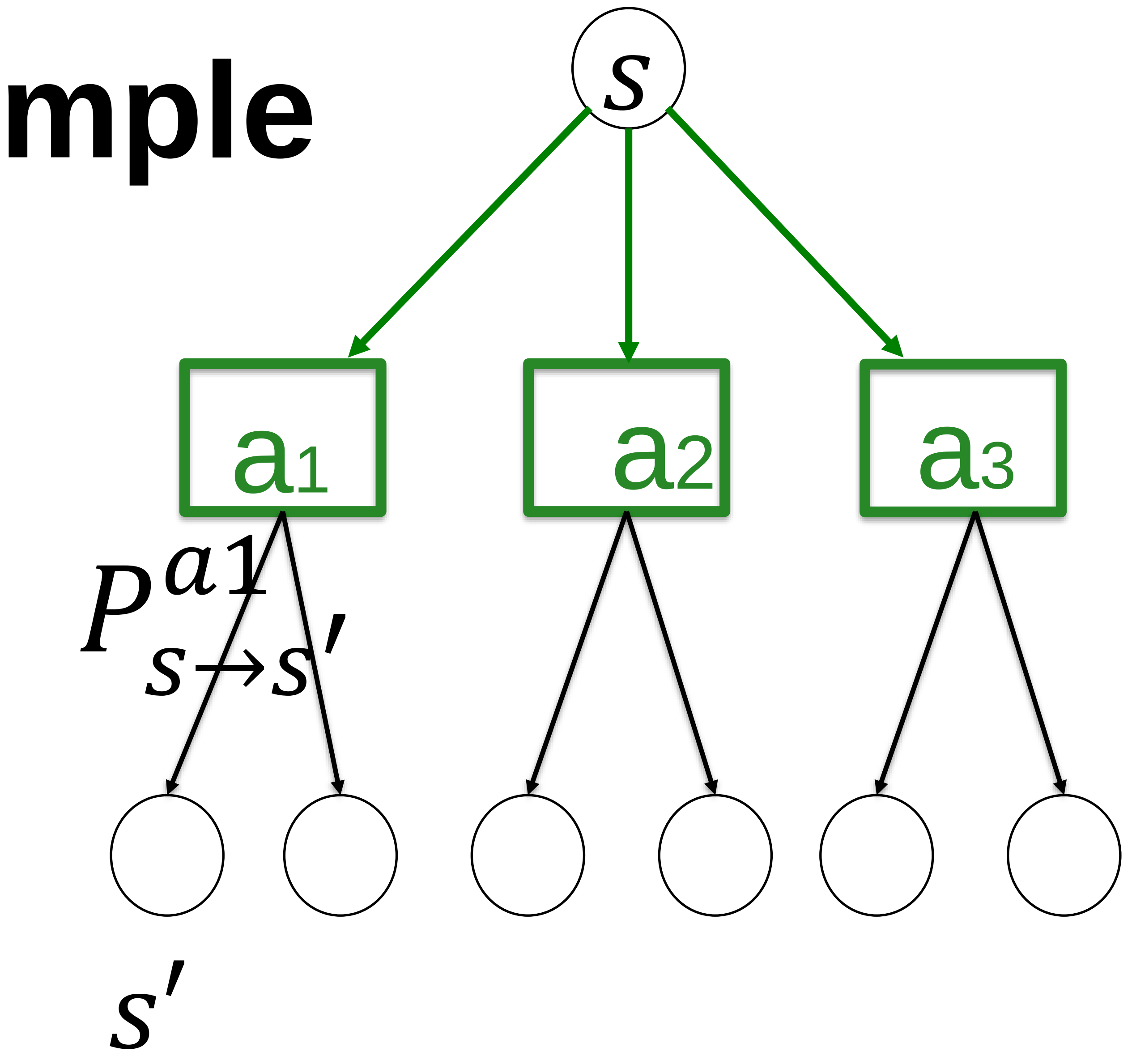
Next lecture at 13h15

One-step horizon: Proposition

Q-value = expected reward for state-action pair

If Q-value is known, choice of action is simple

→ take action with highest Q-value



If Q-value not known:

→ estimate $\hat{Q}$ by trial and error

→ update with rule

$$\Delta\hat{Q}(s,a) = \eta\,[r_t - \hat{Q}(s,a)] \qquad \textbf{(1)}$$

→Let learning rate η decrease over time

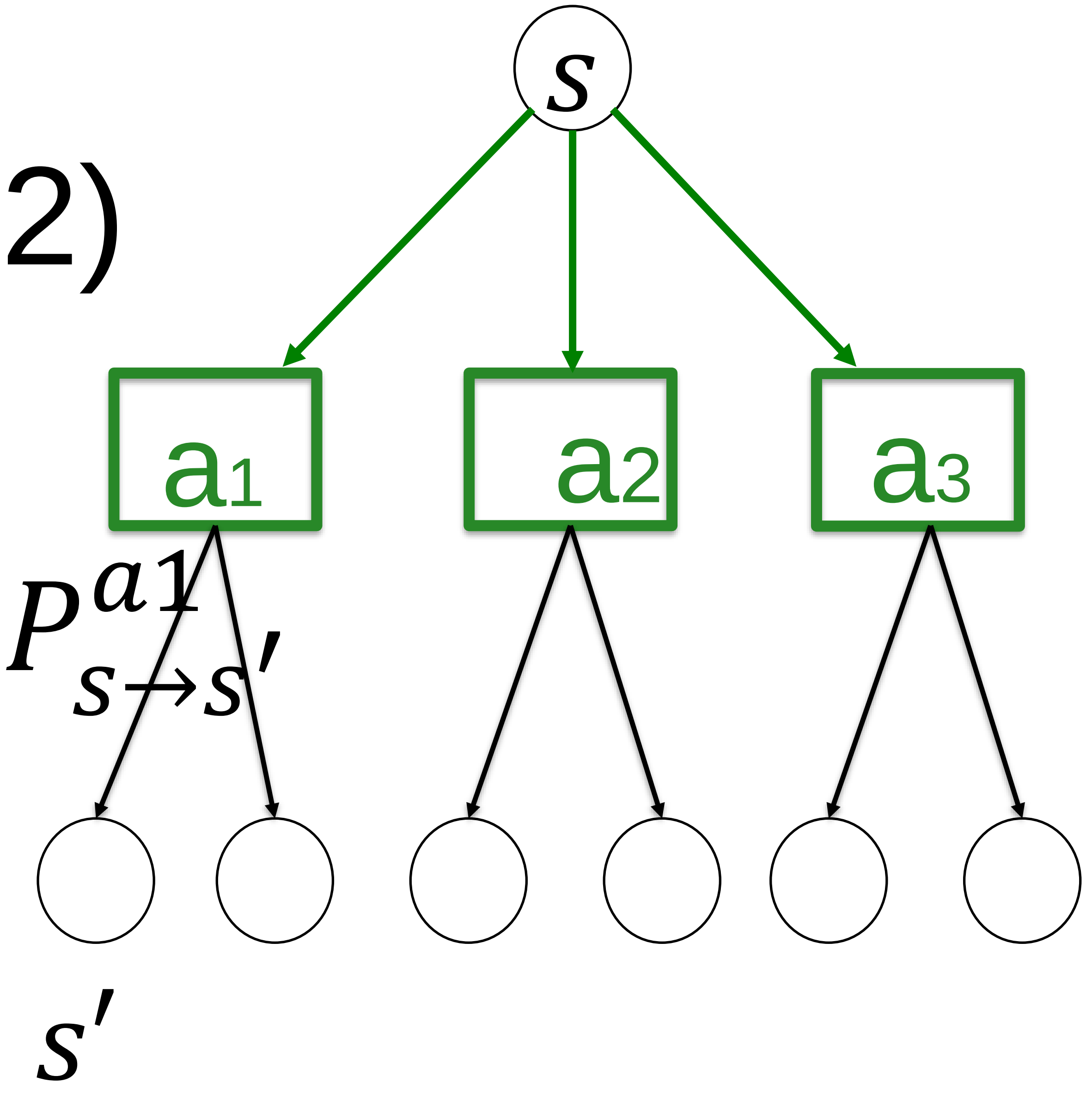
Convergence in Expectation

After taking action a in state s, we update with

$$\Delta\hat{Q}(s,a) = \eta\ [r_t - \hat{Q}(s,a)] \qquad (1)$$

(i) If (1) has **converged in expectation given (s,a),** then $\hat{Q}(s,a)$ has a value,

$$\hat{Q}(s,a) = E\ [\hat{Q}(s,a)|s,a] = Q(s,a) = \sum_{s'} P^a_{s\to s'}\, R^a_{s\to s'} \qquad (2)$$



(ii) If the learning rate η decreases, fluctuations around the **empirical mean** $\langle\boldsymbol{\hat{Q}(s,a)}\rangle_{\boldsymbol{t|s,a}}$ decrease. If $\langle\boldsymbol{\hat{Q}(s,a)}\rangle_{\boldsymbol{t|s,a}}$ converges for fixed η, then **the empirical mean approaches $\boldsymbol{Q(s,a)}$** .

Previous slide.

When evaluating the **expectation value given (s,a),** the learning rate drops out since we set the left-hand-side to zero. The exact value of η is not relevant, as discussed in the theorem. Part (i) of the theorem states that the expectation value of $\hat{Q}(s,a)$ is the correct Q-value. For a quick proof of $E\,[\hat{Q}(s,a)|s,a] = Q(s,a)$ see the video. On the blackboard a stronger statement was shown:

$$\hat{Q}(s,a) = Q(s,a).$$

Convergence in expectation is equivalent to imagining that you start millions of trials with the same value $\hat{Q}(s,a)$ without any intermediate update. So in that sense it is like an infinite 'batch' of examples. The stochastic variables are the **next** state s' and the received reward r_t. The value of $\hat{Q}(s,a)$ is not stochastic but 'frozen'. Therefore (trivially) $E\,[\hat{Q}(s,a)|s,a] = \hat{Q}(s,a)$.

In practice, we do not have expectations but online updates with fluctuations. It is important that η is small at the end of learning so as to limit the amount of fluctuations. Part (ii) states that **online mean** for small learning rate also goes to the correct Q-value.

Indeed, since the equations are linear (for the bandit problem = 1-step horizon), the calculation of part (i) apply analogously to the long-term empirical temporal average (denoted by angular brackets). The average is across all those time steps where action a was chosen in state s, denoted as $\langle \hat{\boldsymbol{Q}}(\boldsymbol{s},\boldsymbol{a}) \rangle_{\boldsymbol{t}|\boldsymbol{s},\boldsymbol{a}}$. We assume convergence, hence our hypothesis reads

$$\langle \Delta\hat{\boldsymbol{Q}}(\boldsymbol{s},\boldsymbol{a}) \rangle_{\boldsymbol{t}|\boldsymbol{s},\boldsymbol{a}} = \boldsymbol{\eta}\, \langle r_t - \hat{\boldsymbol{Q}}(\boldsymbol{s},\boldsymbol{a}) \rangle_{\boldsymbol{t}|\boldsymbol{s},\boldsymbol{a}} = 0\ .$$

The specific result $\langle \hat{\boldsymbol{Q}}(\boldsymbol{s},\boldsymbol{a}) \rangle_{\boldsymbol{t}|\boldsymbol{s},\boldsymbol{a}} = \boldsymbol{Q}(\boldsymbol{s},\boldsymbol{a})$ is based on linearity and is not true for the multi-step horizon that we discuss later.

Proof: Convergence in Expectation

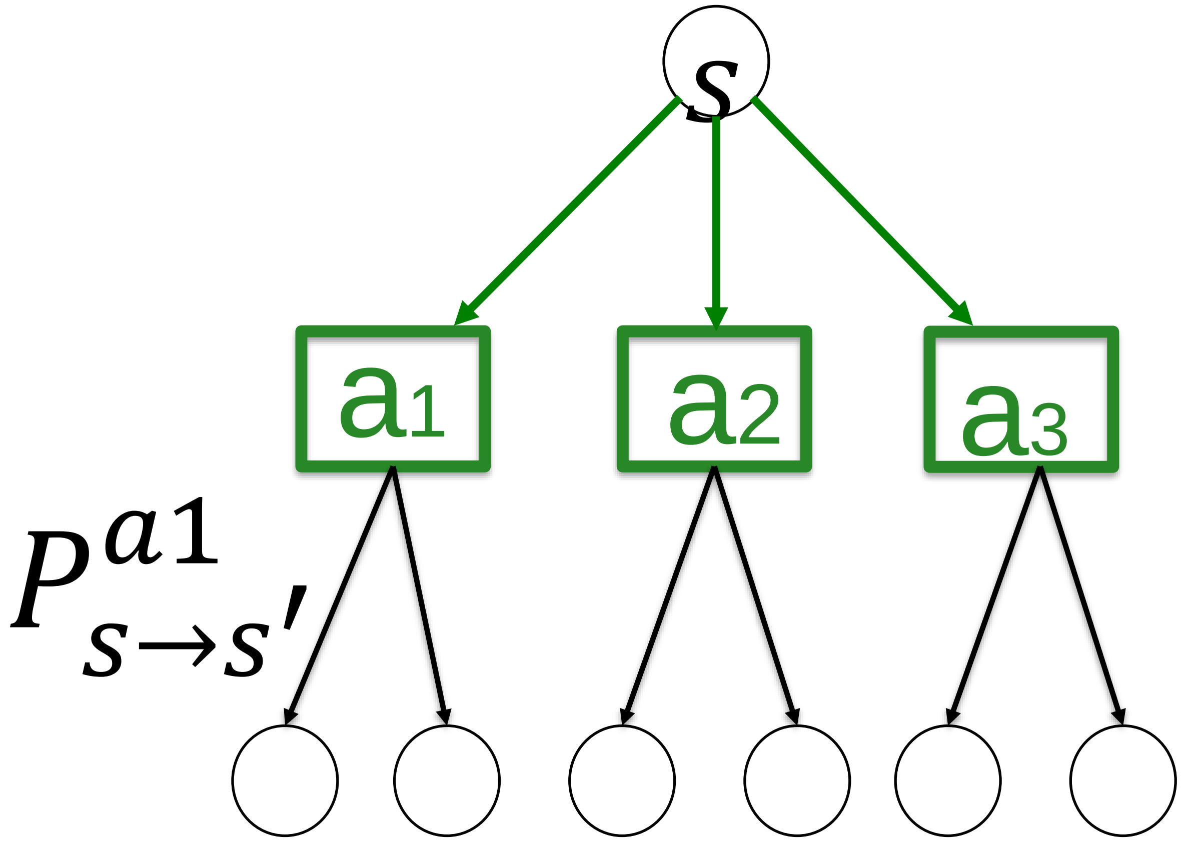
Blackboard2:
Proof of (i)

After taking action a in state s, we update with

$$\Delta\hat{Q}(s,a) = \eta\,[r_t - \hat{Q}(s,a)] \qquad (1)$$

(i) If (1) has converged in expectation, then $\hat{Q}(s,a)$ has an expectation value,

$$E\left[\widehat{Q}(s,a)\right] = \widehat{Q}(s,a) = \sum_{s'} P^{a}_{s\to s'}\, R^{a}_{s\to s'} = Q(s,a) \qquad (2)$$



Note: the expectation is over all possible ‘futures’. For the bandit problem the future is defined by the possible next states and possible rewards.

Your notes.

Blackboard2

Part (i) of Theorem
converged in expectation → $E(\Delta\hat{Q}(s,a)|s,a)=0$

expectation of all possible futures with correct statistical weight

we always start in (s,a) while the system is frozen

Perspective similar to a batch mode:
update only **after** (infinitely) many trials that all start in (s,a) with the same value $\hat{Q}(s,a)$
=
update the expectation over all possibilities that may occur in the next time step.

Previous slide:
$\hat{Q}(s,a)$ denotes the current estimate of the Q-value. Claim: If Q no longer changes (in expectation) then it must be the correct Q-value.

There are different views on how to interpret the ‘expectation;:
- Formally from a mathematical point of view: average over all possible outcomes of the next time step given (s,a).
- In a simulation this would correspond to the following sampling procedure:

You freeze the value of $\hat{Q}(s,a)$ and run MANY times (N to infinity) a test with the state-action pair (s,a) as a starting condition. Then you evaluate the resulting ‘batch update’ averaged across all these examples. If the batch update with millions of examples is zero, that implies that you have converged to the correct value.
In the copies of the blackboard notes, there are two versions of the proof:

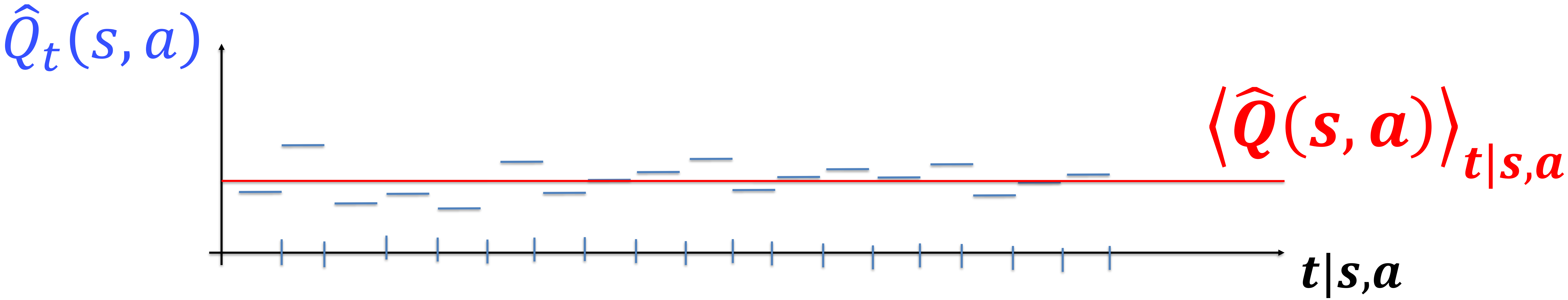
First, on page 2, top half of page a SIMPLE proof. $E\left[\widehat{Q}(s,a)|s,a\right] = Q(s,a) = \sum_{s'} P^a_{s\to s'} R^a_{s\to s'}$
Second, on page 4 (final page), the stronger proof with more in-between steps showing $\widehat{Q}(s,a) = E\left[\widehat{Q}(s,a)|s,a\right] = Q(s,a) = \sum_{s'} P^a_{s\to s'} R^a_{s\to s'}$

Blackboard3

Part (ii) of Theorem:

We work with the online update $\Delta\hat{Q}(s,a)$. With finite learning rate, the value of $\hat{Q}_t(s,a)$ fluctuates around a mean

$\boldsymbol{\langle\hat{Q}(s,a)\rangle_{t|s,a}}$



Under the hypothesis of the theorem (i.e., the mean converges), then the mean is equal to the ‘correct’ Q-value.

Notes.

Proof of part (ii) of the theorem is in the Blackboard notes on page 3 – think about it. The proof works because of linearity.

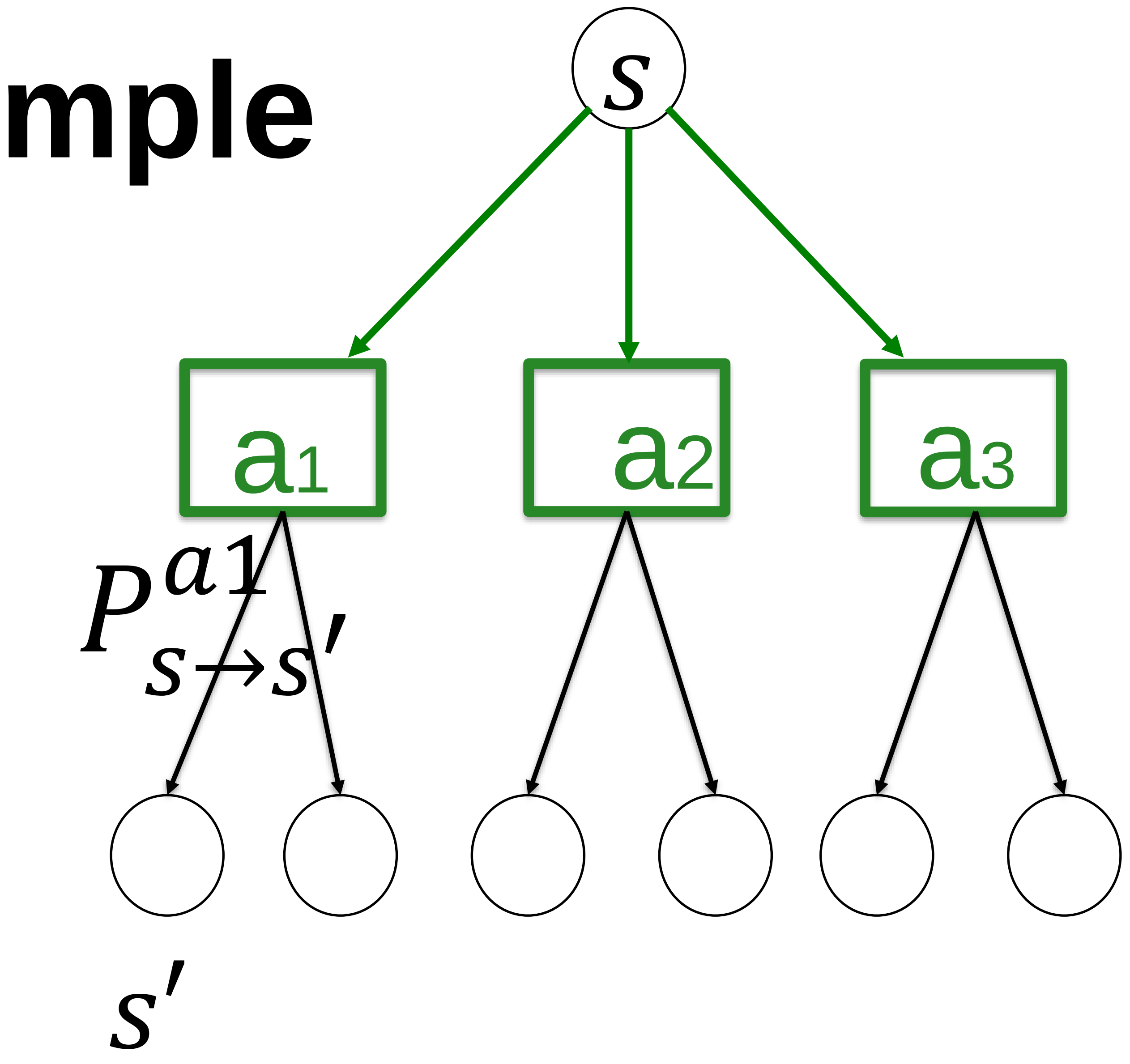
More information regarding the philosophy of different averaging procedures also in Exercise 3 this week and beginning of the lecture of next week.

One-step horizon: summary

Q-value = expected reward for state-action pair

If Q-value is known, choice of action is simple

→ take action with highest Q-value



If Q-value not known:

→ estimate $\hat{Q}$ by trial and error

→ update with rule

$$\Delta\hat{Q}(s,a) = \eta\,[r_t - \hat{Q}(s,a)] \qquad \textbf{(1)}$$

→Let learning rate η decrease over time

Iterative algorithm (1) converges in expectation

Previous slide.
Let us distinguish the ESTIMATE $\hat{Q}(s,a)$ from the real Q-value $Q(s,a)$

The update rule can be interpreted as follows:
if the actual reward is larger than (my estimate of) the expected reward, then I should increase (a little bit) my expectations.

The learning rate η :

In exercise 1, we found a rather specific scheme for how to reduce the learning rate over time. But many other schemes also work in practice. For example you keep η constant for a block of time, and then you decrease it for the next block.

Note: in later lectures I will often use the symbol α instead of η

Both symbols indicate what is called the ‘learning rate’ in Deep Learning.

Teaching monitoring – monitoring of understanding

[ ] today, at least 60% of material was new to me.

[ ] I have the feeling that I have been able to follow (at least) 80% of the lecture.

Previous slide.
Teaching monitoring – feedback for the teacher.