Martin Mundt, Dr. Iuliia Pliushch, Prof. Dr. Visvanathan Ramesh

Pattern Analysis & Machine Intelligence Praktikum: MLPR-SS21

Week 11: Introduction into DQN

DQN-RL: Supervised learning setup

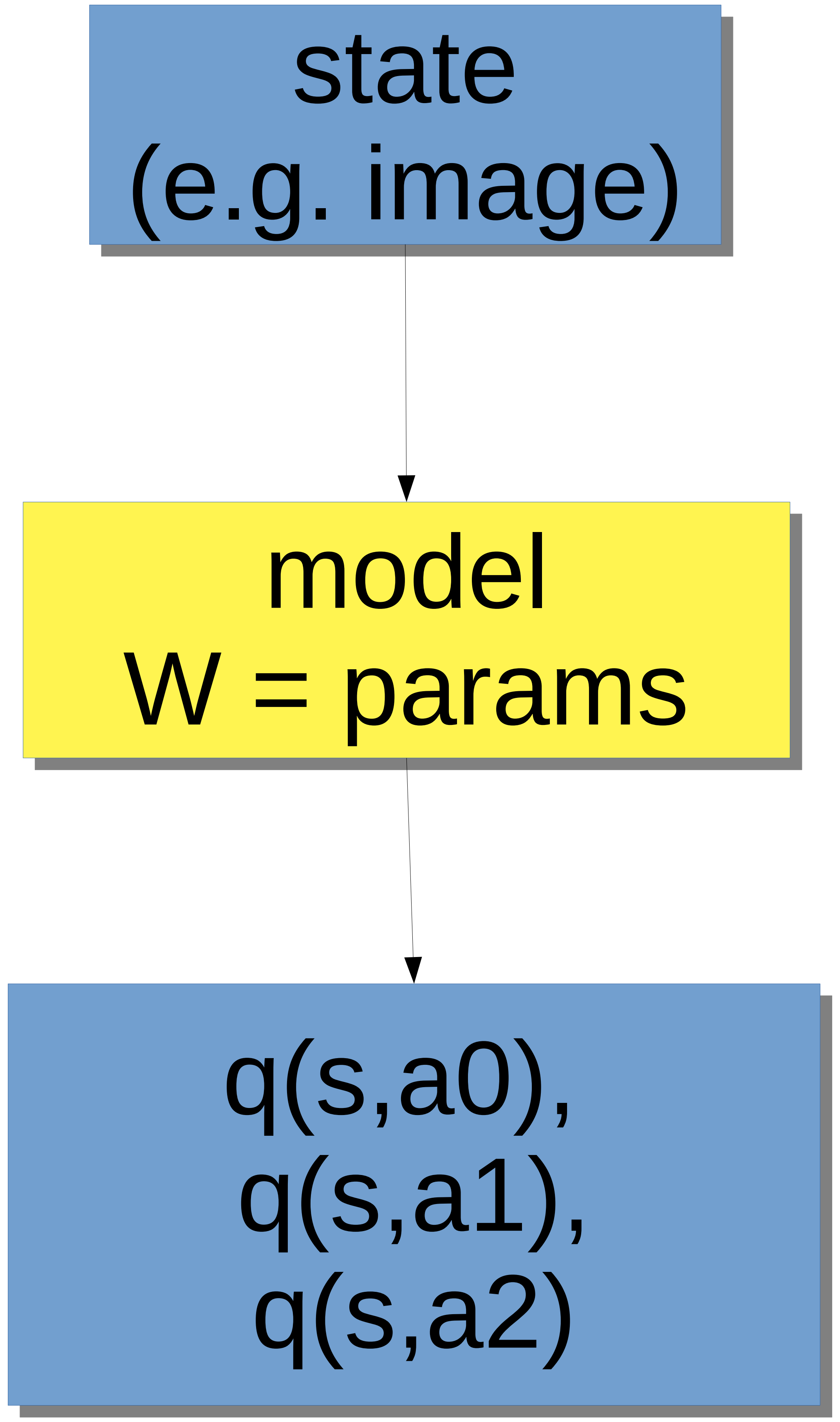
- Motivation: reduce number of **parameters**:

$$\hat{v}(s,w)=v_{\pi}(s)$$

$$\hat{q}(s,a,w)=q_{\pi}(s,a)$$

- The input-output relation to learn: $s,a \rightarrow q_{\pi}(s,a)$

- Important:
  Any **single parameter** affects values of **all states**



https://www.coursera.org/learn/practical-rl/home/welcome

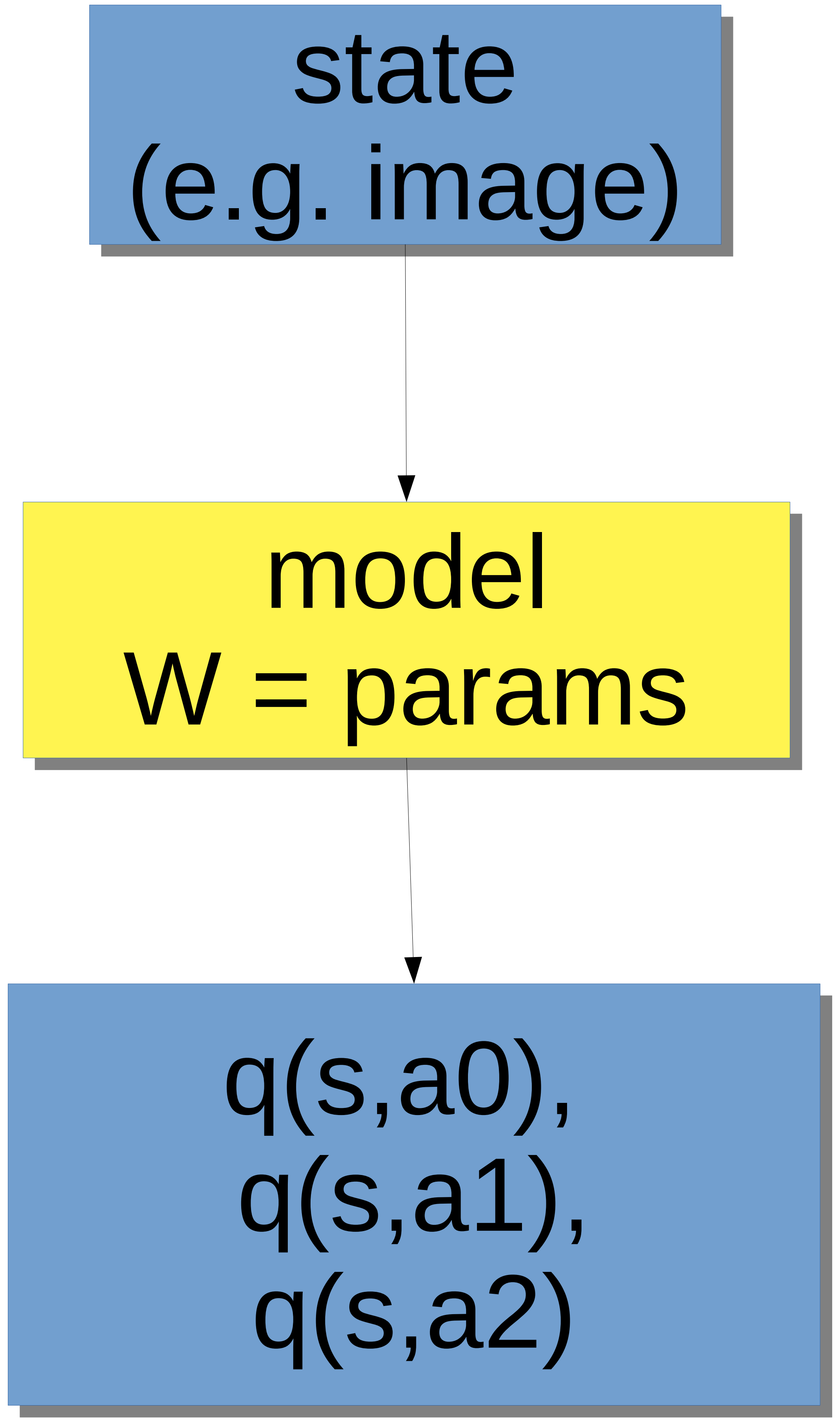
DQN-RL: Supervised learning setup

- Motivation: reduce number of **parameters**:

$$\hat{v}(s,w)=v_{\pi}(s)$$

$$\hat{q}(s,a,w)=q_{\pi}(s,a)$$

- The input-output relation to learn: $s,a \rightarrow q_{\pi}(s,a)$



Actions :	↑	→	↓	←
Start				
Nothing / Blank				
Power				
Mines				
END				

https://www.freecodecamp.org/news/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc/

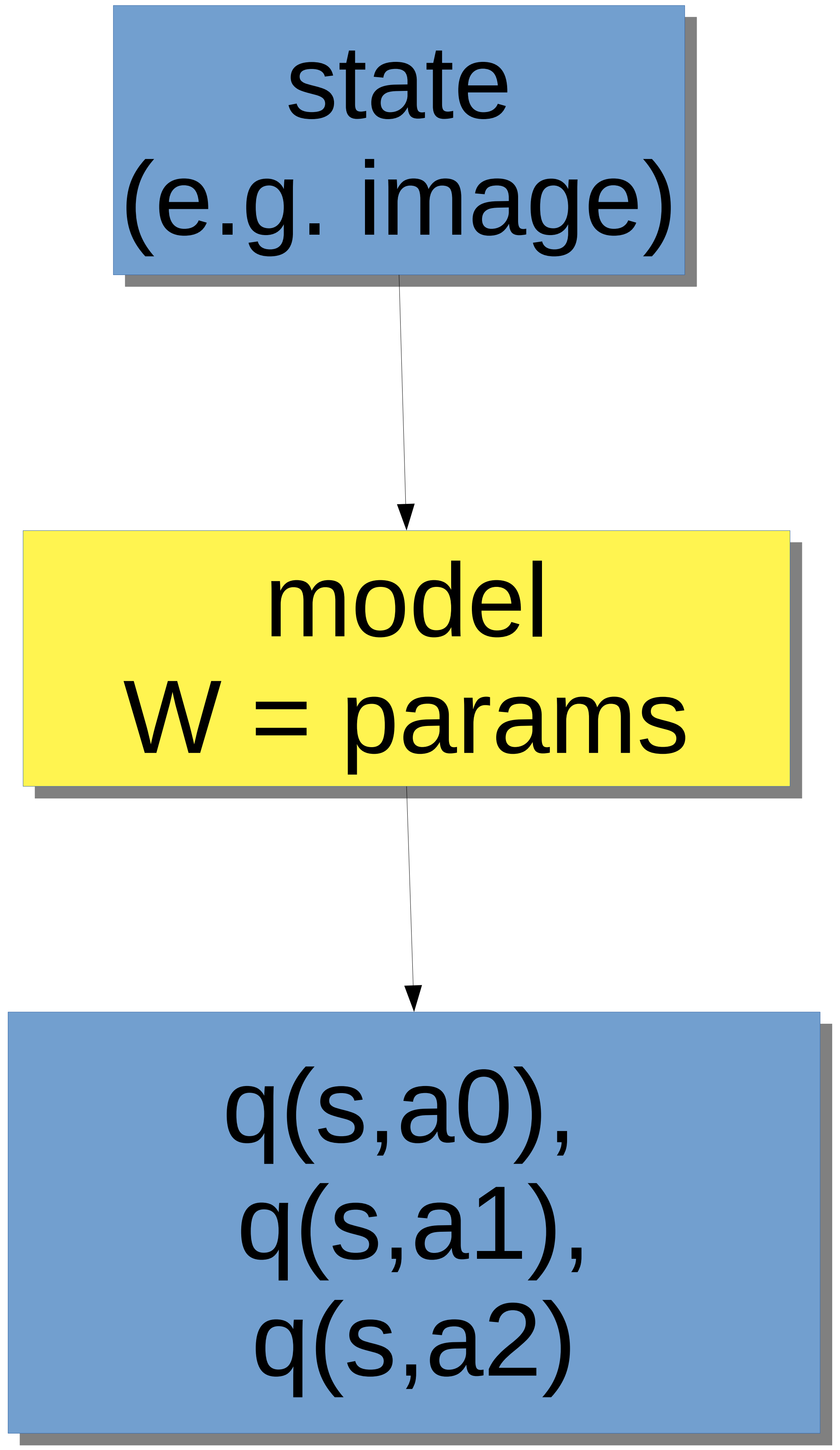
DQN-RL: Temporal Difference (TD)

- Reminder: cumulative expected reward $G_t = \sum_{i=0}^{\infty} \gamma^i * r_{t+i}$

- Ideal goal: $s, a \rightarrow q_\pi(s, a) = E_\pi[G_t | S_t = s, A_t = a]$

- TD: sample-based approximation
  1. approximate the **value function** with parameters *w*
  2. approximate **expectation** with a *sample-based estimate*
  3. approximate the **value of the next state**

$$s, a \rightarrow r(s, a) + \gamma * G_{t+1} \stackrel{\text{def}}{=} r(s, a) + \gamma * \hat{v}_\pi(s_{t+1}, w)$$



https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: Loss and semi-gradient update

- Defined goal to learn: $g(s,a) = r(s,a) + \gamma * max_a \hat{q}_\pi(s_{t+1}, a, w)$

- The loss is the same as for a regression problem:

$$L(w) = \frac{1}{2} \sum_{s,a} \rho_{s,a} \underbrace{[g(s,a) - \hat{q}_\pi(s,a,w)]^2}_{L_{s,a}(w)}$$

- Mean squared error between **targets (goals)** and **our estimates**.

- $\rho_{s,a}$ - measure of “importance” of a (s,a)-pair
  (how often they were encountered)

https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: Loss and semi-gradient update

- Defined goal to learn: $g(s,a)=r(s,a)+\gamma * max_a \hat{q}_\pi(s_{t+1},a,w)$

- Loss function: $L(w)=\frac{1}{2}\sum_{s,a}\rho_{s,a}\underbrace{[g(s,a)-\hat{q}_\pi(s,a,w)]^2}_{L_{s,a}(w)}$

- SGD: $w \leftarrow w-\alpha * \nabla_w L_{s,a}(w)$

https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: Loss and semi-gradient update

- Defined goal to learn: $g(s,a)=r(s,a)+\gamma * max_a \hat{q}_\pi(s_{t+1},a,w)$
- Loss function: $L(w)=\frac{1}{2}\sum_{s,a}\rho_{s,a} * [g(s,a)-\hat{q}_\pi(s,a,w)]^2$
- SGD: $w \leftarrow w-\alpha * \nabla_w L_{s,a}(w)$
- Consider goals to be fixed: $\nabla_w g(s,a)=0$
- Apply **semi-gradient** update:

$$w \leftarrow w+\alpha * [g(s,a)-\hat{q}_\pi(s,a,w)] * \nabla_w \hat{q}_\pi(s,a,w)$$

https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: Tabular vs. approx. Q-Learning

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

- Tabular Q-Learning:

$$q(s,a) \leftarrow \alpha * \underbrace{\tilde{q}(s,a)}_{r(s,a)+\gamma * max_a q_\pi(s_{t+1},a)} + (1-\alpha) * q(s,a)$$

$$= q(s,a) + \alpha * [r + \gamma * max_a q(s_{t+1},a) - q(s,a)]$$

- Approximate Q-Learning:

$$w \leftarrow w + \alpha * [\underbrace{g(s,a)}_{r(s,a)+\gamma * max_a \hat{q}_\pi(s_{t+1},a,w)} - \hat{q}_\pi(s,a,w)] * \nabla_w \hat{q}_\pi(s,a,w)$$

https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: approx Q-Learning

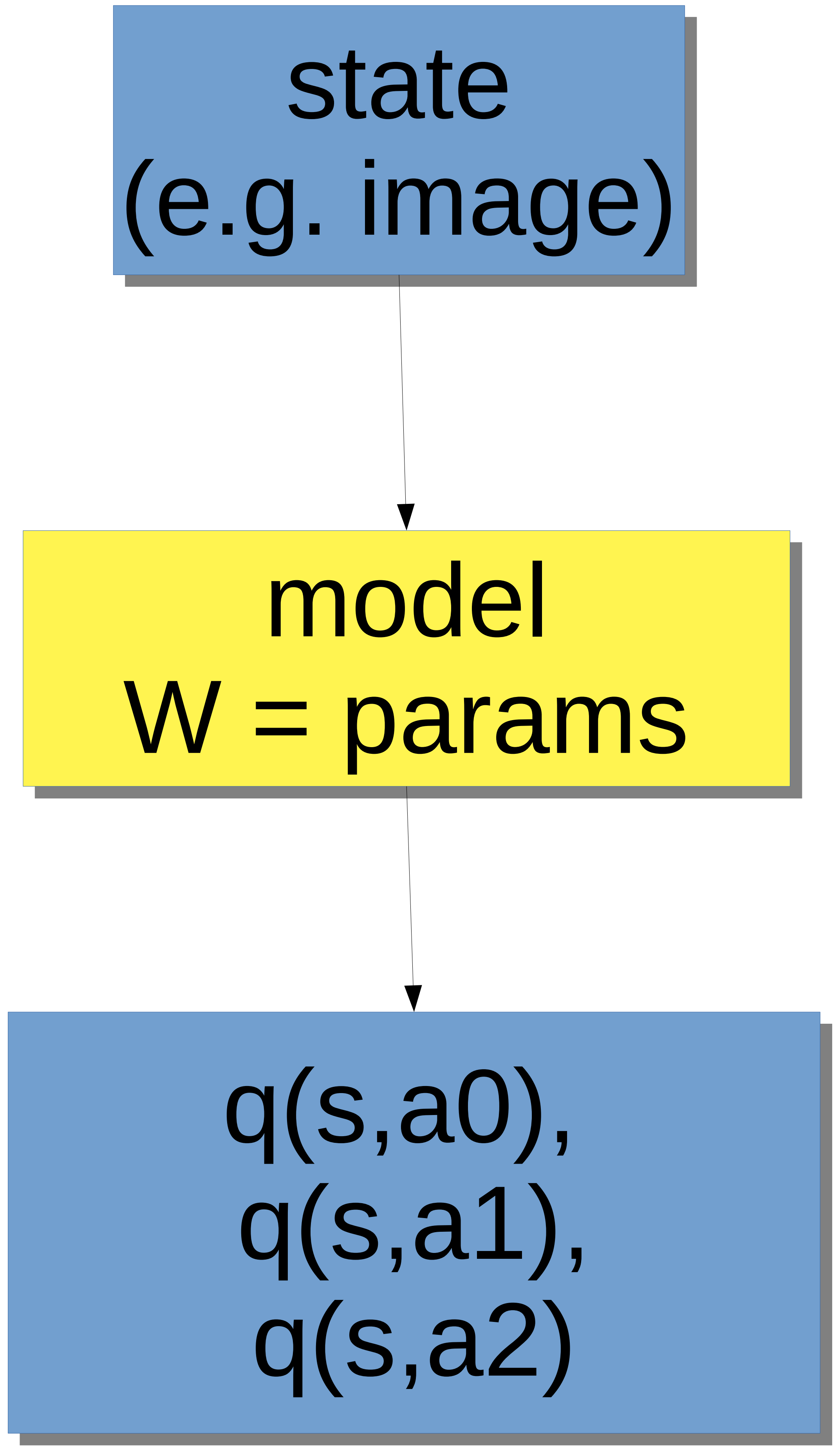
- Ideal goal: $s,a \rightarrow q_\pi(s,a) = E_\pi[G_t | S_t = s, A_t = a]$

- TD: sample-based approximation

$$s,a \rightarrow r(s,a) + \gamma * G_{t+1} \stackrel{\text{def}}{=} r(s,a) + \gamma * \hat{v}_\pi(s_{t+1}, w)$$

- Semi-gradient update:

$$w \leftarrow w + \alpha * [\underbrace{g(s,a)}_{r(s,a) + \gamma * max_a \hat{q}_\pi(s_{t+1}, a, w)} - \hat{q}_\pi(s,a,w)] * \nabla_w \hat{q}_\pi(s,a,w)$$



https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: Instability issues

1) **Sequential correlated data** may hurt *convergence* and *performance*

2) **Instability of the data distribution** because of *policy oscillation*

3) **Unstable gradients** because q-values vary a lot

https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: Instability issues

1) **Sequential correlated data** may hurt *convergence* and *performance*

Experience replay

2) **Instability of the data distribution** because of *policy oscillation*

Target networks

3) **Unstable gradients** because q-values vary a lot

Gradient clipping

https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: Instability issues

1) **Sequential correlated data** may hurt *convergence* and *performance*

Experience replay: store (s,a,r,ŝ)-tuples in a pool and sample at random

2) **Instability of the data distribution** because of *policy oscillation*

Target networks

3) **Unstable gradients** because q-values vary a lot

Gradient clipping

https://www.coursera.org/learn/practical-rl/home/welcome

DQN-RL: Instability issues

1) **Sequential correlated data** may hurt *convergence* and *performance*

Experience replay: store (s,a,r,ŝ)-tuples in a pool and sample at random

2) **Instability of the data distribution** because of *policy oscillation*

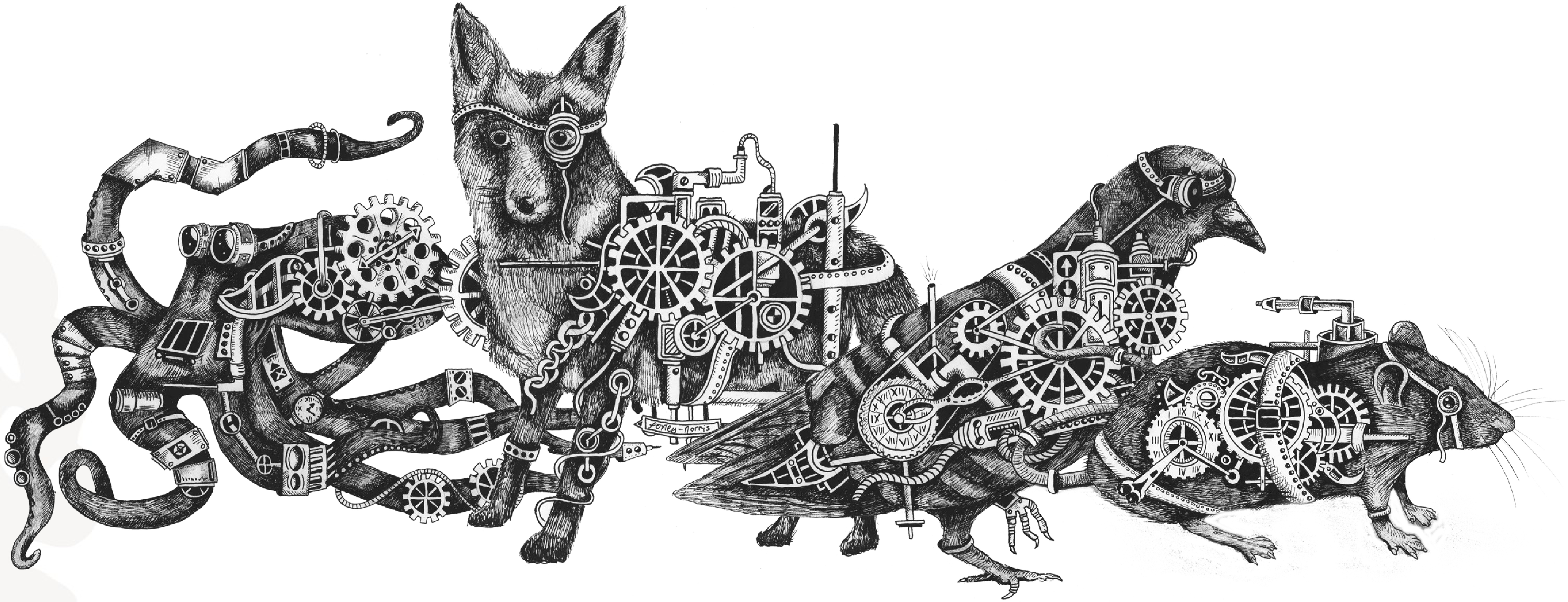
Target networks $w \leftarrow w + \alpha * [\underbrace{g(s,a)}_{r(s,a)+\gamma * max_a \hat{q}_\pi(s_{t+1},a,w)} - \hat{q}_\pi(s,a,w)] * \nabla_w \hat{q}_\pi(s,a,w)$

3) **Unstable gradients** because q-values vary a lot

Gradient clipping

https://www.coursera.org/learn/practical-rl/home/welcome

Application: Animal AI Olympics



https://github.com/beyretb/AnimalAI-Olympics

Application: RL as a basis of AI?

- **intelligence / associated abilities** maximize (different kinds of) reward

Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward Is Enough. Artificial Intelligence, 299, 103535. https://doi.org/10.1016/j.artint.2021.103535