RL 17a: Deep RL

(certainly interesting, but not examinable)

Michael Herrmann

University of Edinburgh, School of Informatics

(bonus lecture)

RL

- "The use of punishments and rewards can at best be a part of the teaching process" (A. Turing)

  Russell and Norvig: AI, Ch.21
- "A microcosm for the entire AI problem"

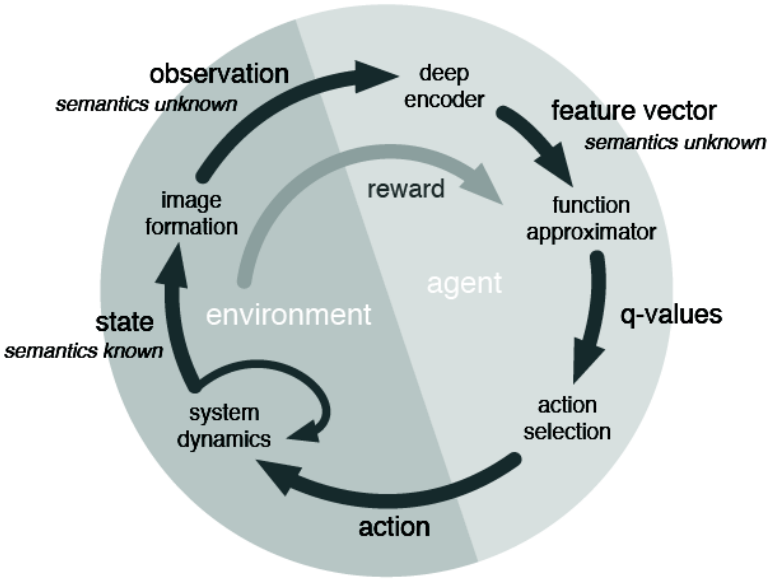
  Russell and Norvig: AI, Ch.20
- We seek a single agent which can solve any human-level task
  The essence of an intelligent agent
  Reinforcement learning + deep learning = AI

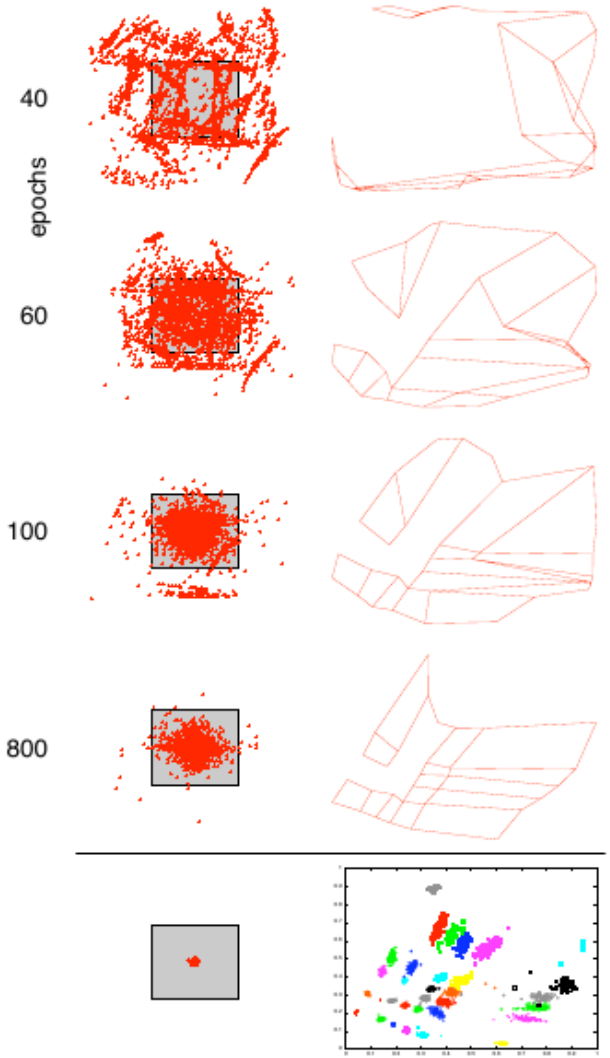
  David Silver, Google DeepMind

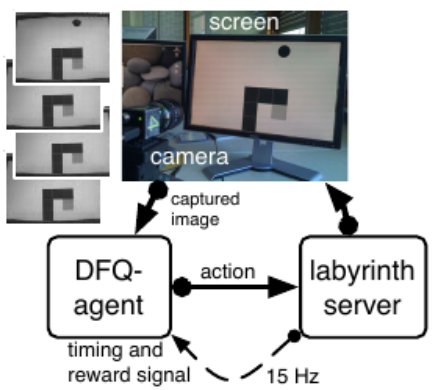
Deep RL

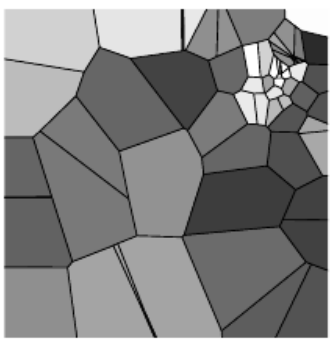
- State classification by deep learning
- Representation of sample trajectories
- Deep network representation of the value function
- Deep network representation of the policy
- Deep network model of state transitions

Deep Auto-Encoder Neural Networks in RL







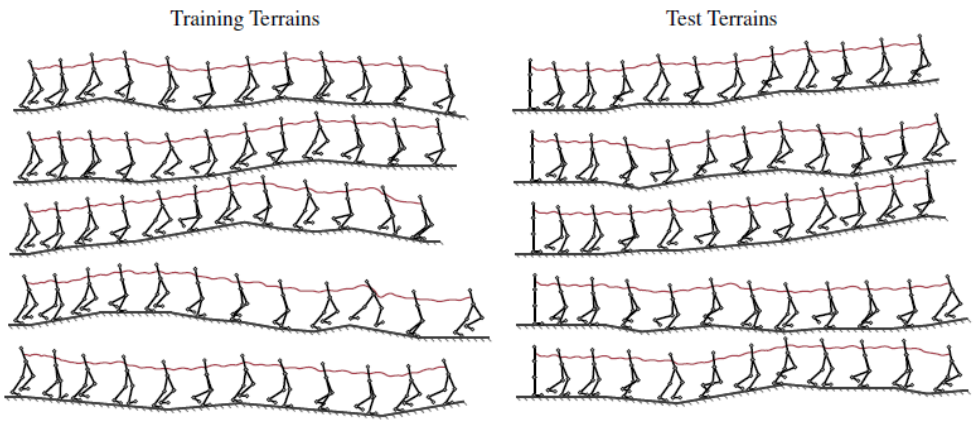


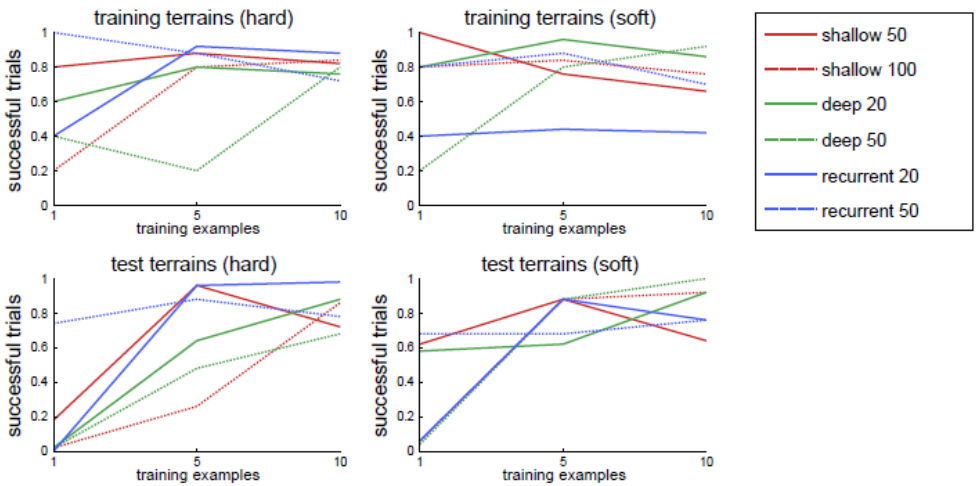
value function in feature space

Sascha Lange and Martin Riedmiller, IJCNN 2010

bonus lecutre in 2016 Michael Herrmann RL 17a

Deep and Recurrent Architectures for Optimal Control





- 9 degrees of freedom
- 10 training terrains
- 10 test terrains
- No x-y input to controller
- Policy gradient
- Soft or hard rectified units
- Equal number of parameters in all models

Sergey Levine, 2013

bonus lecutre in 2016 Michael Herrmann RL 17a

Deep $\mathcal{Q}$-learning is still $\mathcal{Q}$-learning

- Function approximation of the$\mathcal{Q}$-function

$$\mathcal{Q}(s;a;w) \approx \mathcal{Q}(s;a)$$

- $\delta = r + \gamma \max_{a'} \mathcal{Q}(s', a'; w) - \mathcal{Q}(s, a; w)$
- Update by stochastic gradient descent

$$\begin{aligned} \Delta w &= -\eta \frac{\partial \delta^2}{\partial w} \\ &= \eta \left(r + \gamma \max_{a'} \mathcal{Q}(s', a'; w) - \mathcal{Q}(s, a; w) \right) \frac{\partial \mathcal{Q}(s, a; w)}{\partial w} \end{aligned}$$

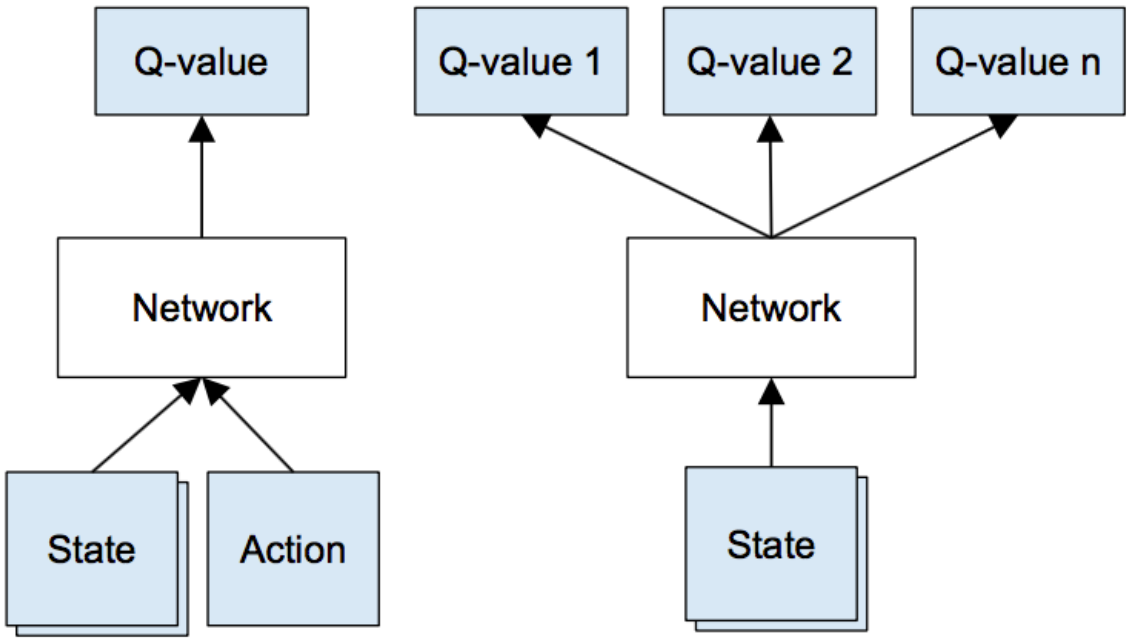
DQL: Lessons from RL with function approximation

Countermeasures against divergence of $\mathcal{Q}(s; a; w)$

- Diversify data to reduce episodic correlation
- Use lots of data, i.e. many quadruples (s, a, r, s')
- Use one networks for $\mathcal{Q}(s, a; w_0)$ and another one for $\mathcal{Q}(s', a'; w_1)$
- Use information about the reward distribution
- Normalisation to get robust gradients

Adapted from: Deep Reinforcement Learning by David Silver, Google DeepMind

Demystifying Deep Reinforcement Learning (T. Matiisen)



Naive formulation of DQN vs. DQN in DeepMind

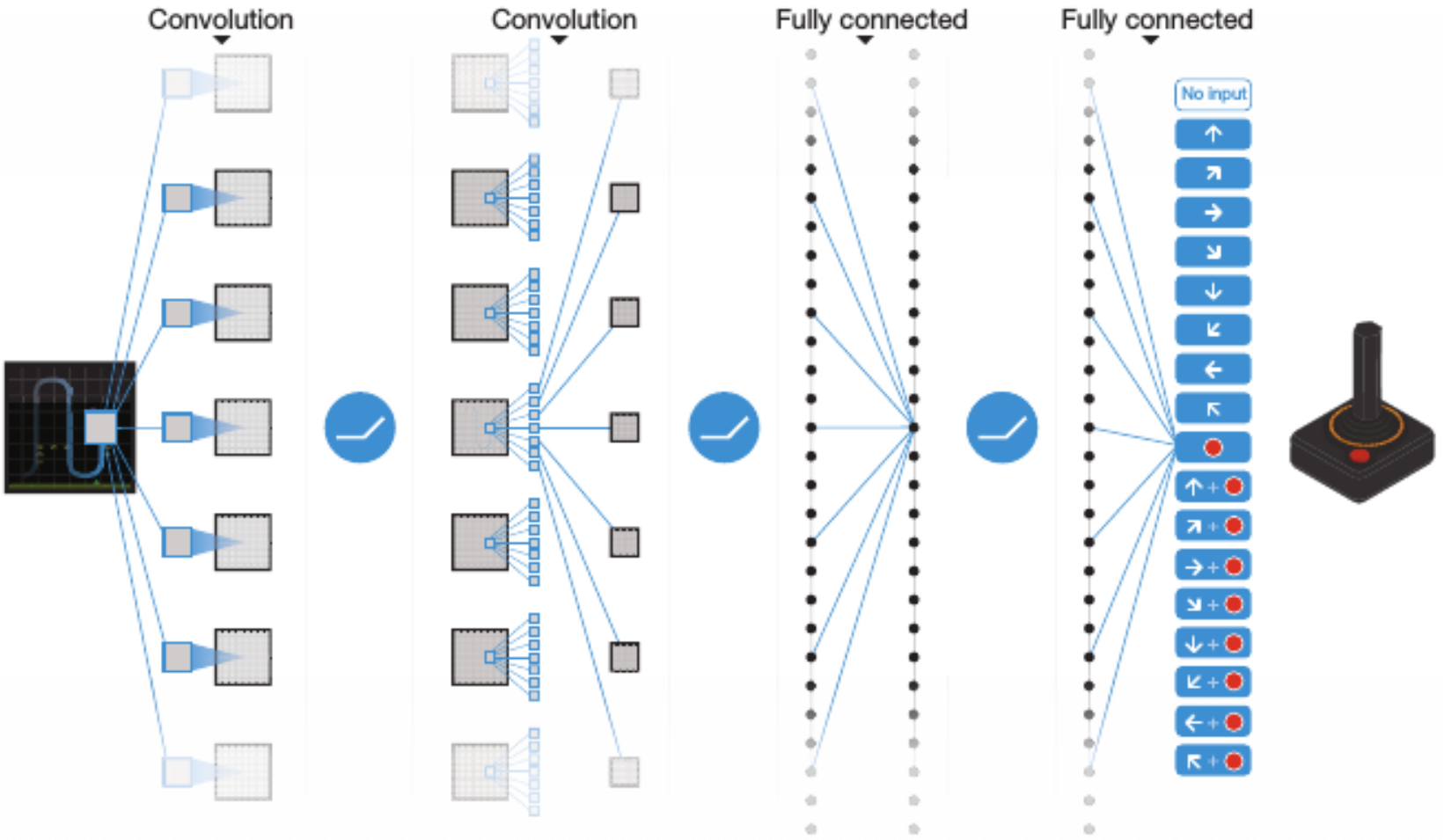
Deep $\mathcal{Q}$ Network (Volodymyr Mnih ea. 2015)

- Represent artificial agent by deep neural network: D$\mathcal{Q}$N
- Learn successful policies directly from high-dimensional sensory inputs
- End-to-end reinforcement learning
- Tested classic Atari 2600 games
- Reward: Game score
- Achieve a level comparable to a professional human games tester across a set of 49 games (about half of the games better than human)
- Unchanged metaparameters for all games

Algorithm for Deep Reinforcement Learning (T. Matiisen)

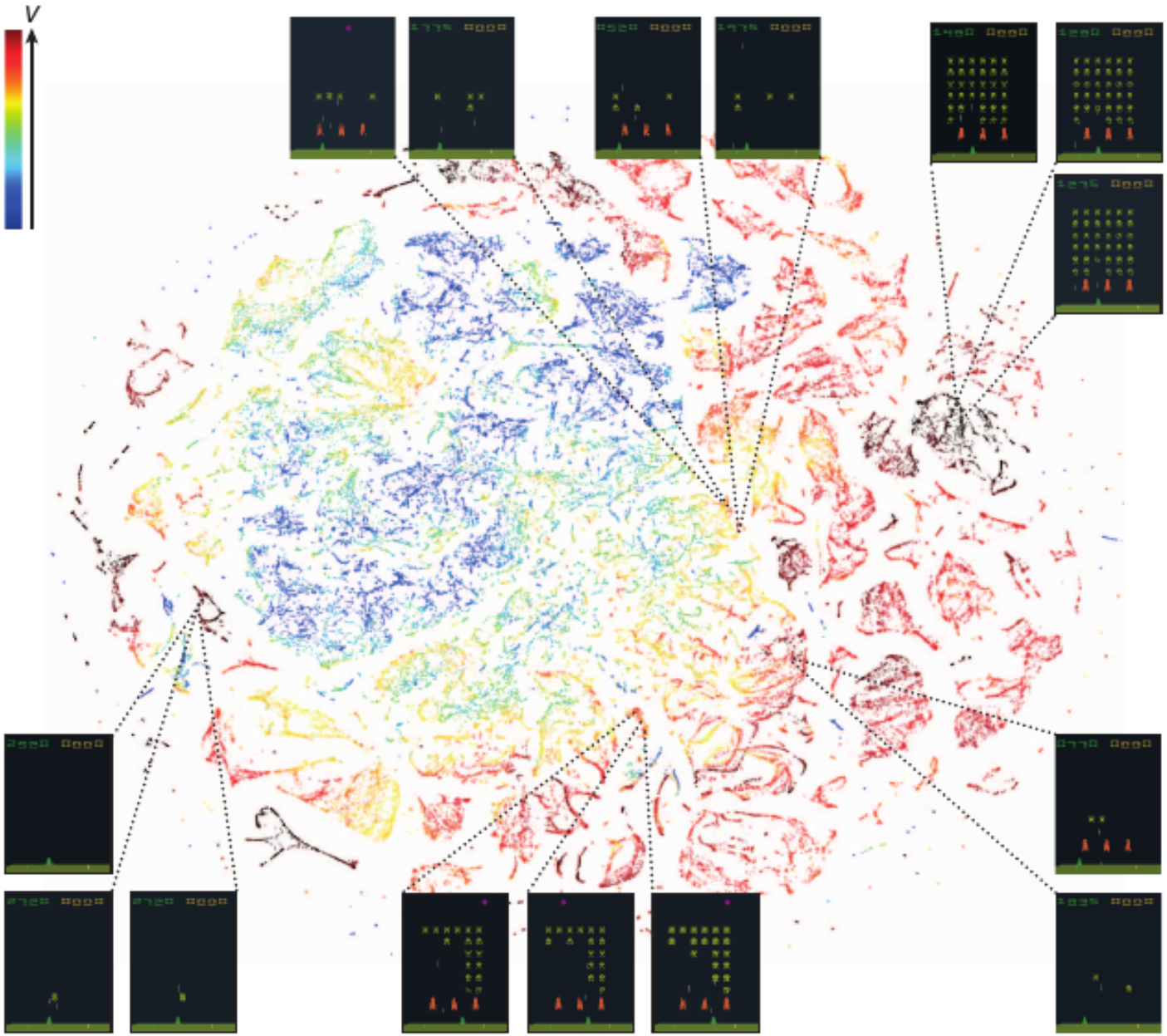
```
initialize replay memory D
initialize action-value function Q with random weights
observe initial state s
repeat
      select an action a
            with probability ε select a random action
            otherwise select a = argmax_a' Q(s,a')
      carry out action a
      observe reward r and new state s'
      store experience <s, a, r, s'> in replay memory D

      sample random transitions <ss, aa, rr, ss'> from replay memory D
      calculate target for each minibatch transition
            if ss' is terminal state then tt = rr
            otherwise tt = rr + γmax_a' Q(ss', aa')
      train the Q network using (tt - Q(ss, aa))^2 as loss

      s = s'
until terminated
```

Convolutional network for DQN (Volodymyr Mnih ea. 2015)



Space invaders: Visualising last hidden layer (Mnih ea 2015)

t-distributed stochastic neighbour embedding



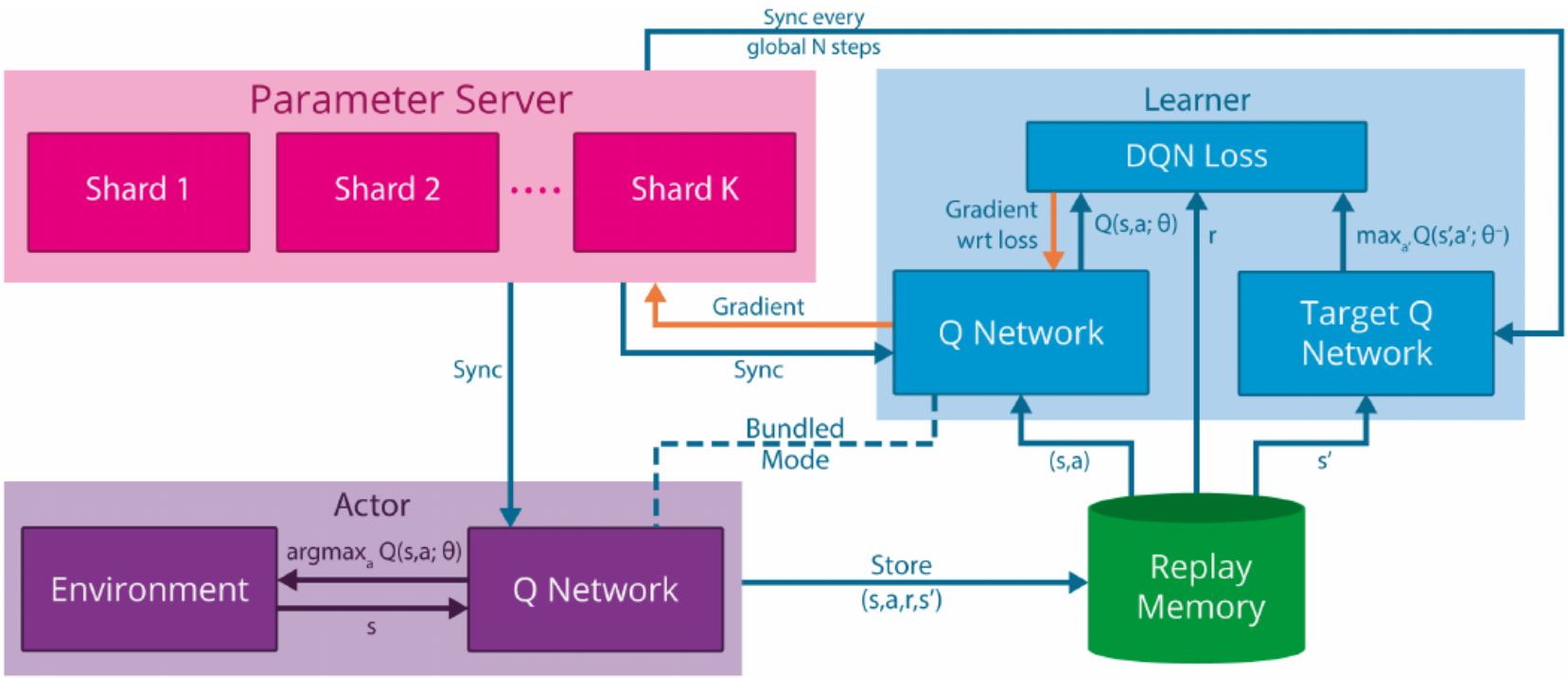
bonus lecutre in 2016 Michael Herrmann RL 17a

Hyperparameter values (Volodymyr Mnih ea. 2015)

Hyperparameter	Value	Description
minibatch size	32	Number of training cases over which each stochastic gradient descent (SGD) update is computed.
replay memory size	1000000	SGD updates are sampled from this number of most recent frames.
agent history length	4	The number of most recent frames experienced by the agent that are given as input to the Q network.
target network update frequency	10000	The frequency (measured in the number of parameter updates) with which the target network is updated (this corresponds to the parameter *C* from Algorithm 1).
discount factor	0.99	Discount factor gamma used in the Q-learning update.
action repeat	4	Repeat each action selected by the agent this many times. Using a value of 4 results in the agent seeing only every 4th input frame.
update frequency	4	The number of actions selected by the agent between successive SGD updates. Using a value of 4 results in the agent selecting 4 actions between each pair of successive updates.
learning rate	0.00025	The learning rate used by RMSProp.
gradient momentum	0.95	Gradient momentum used by RMSProp.
squared gradient momentum	0.95	Squared gradient (denominator) momentum used by RMSProp.
min squared gradient	0.01	Constant added to the squared gradient in the denominator of the RMSProp update.
initial exploration	1	Initial value of ε in ε-greedy exploration.
final exploration	0.1	Final value of ε in ε-greedy exploration.
final exploration frame	1000000	The number of frames over which the initial value of ε is linearly annealed to its final value.
replay start size	50000	A uniform random policy is run for this number of frames before learning starts and the resulting experience is used to populate the replay memory.
no-op max	30	Maximum number of "do nothing" actions to be performed by the agent at the start of an episode.

Deep RL: Parallelisation

Gorila (GOogle ReInforcement Learning Architecture)



(Arun Nair et al., 2015)

Many Learners send gradient information to a parameter server: Shards update disjoint subsets of the parameters of the deep network.

Further work

- Policy Gradient for Continuous Actions
- Model-Based RL: Deep transition model of the environment
- Deep AC
- POMDPs
- IRL
- Biological modelling
- Robot control
- Text based games (incl. language understadning)
- Dropout for robust representation

Model-based Deep RL: Challenges (David Silver)

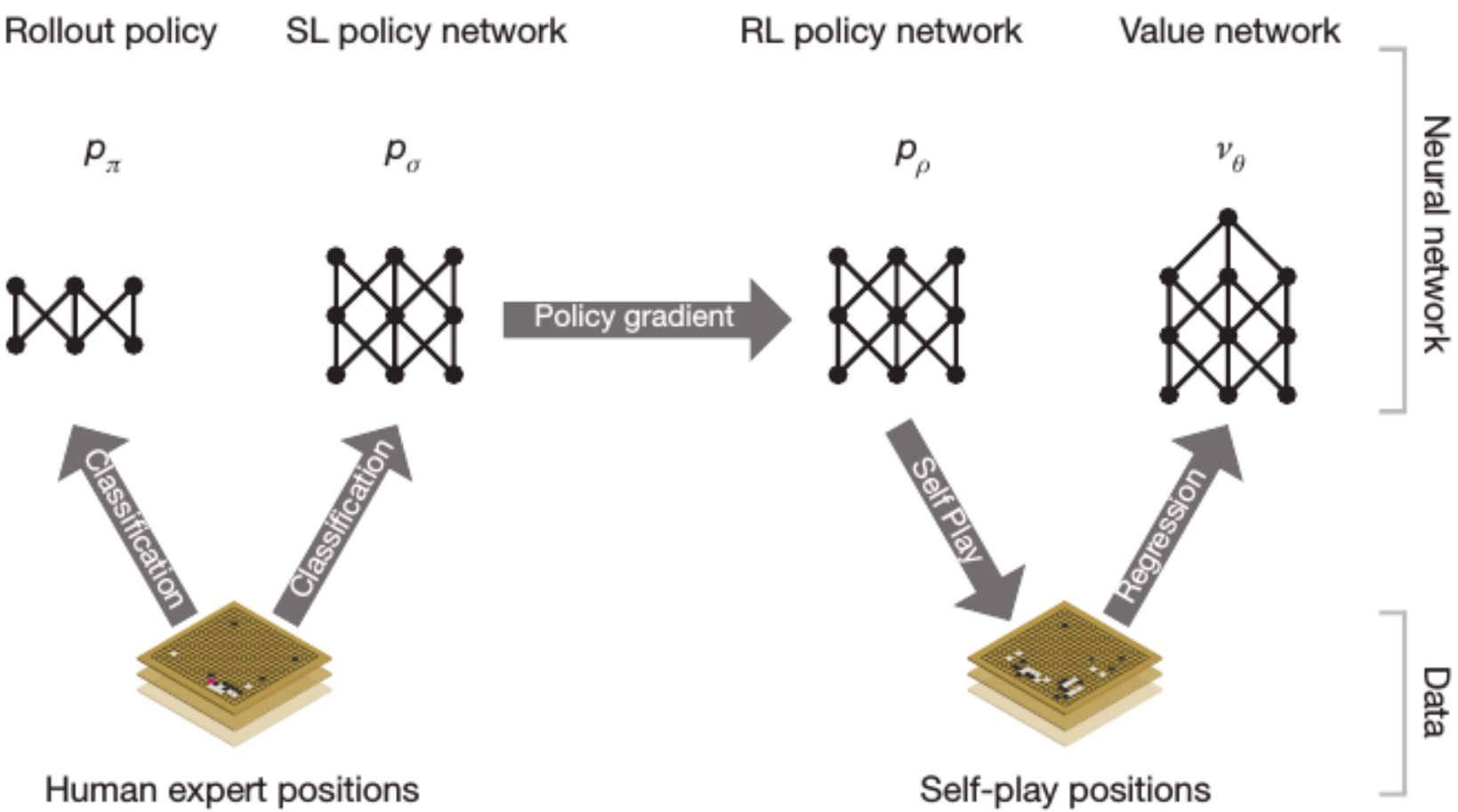
- Errors in the transition model compound over the trajectory
- By the end of a long trajectory, rewards can be totally wrong
- Model-based RL has failed (so far) in Atari
- Deep networks of value/policy can "plan" implicitly
- Each layer of network performs arbitrary computational step
  n-layer network can "lookahead" n steps
- Are transition models required at all?

AlphaGo

- Chess: ≈ 30 possible moves, ≈ 80 moves per game
- Go: ≈ 250 possible moves, ≈ 150 moves per game
- Approach
  - Reduce depth by truncating search tree at state s and replacing subtree below s by approximate value function $v(s)$
  - Reduce width of search tree by policy $p(a, s)$
  - Train using supervised learning to predict human player moves
  - Train policy network to optimise outcome by self-play
  - Train a value network to predict the winner

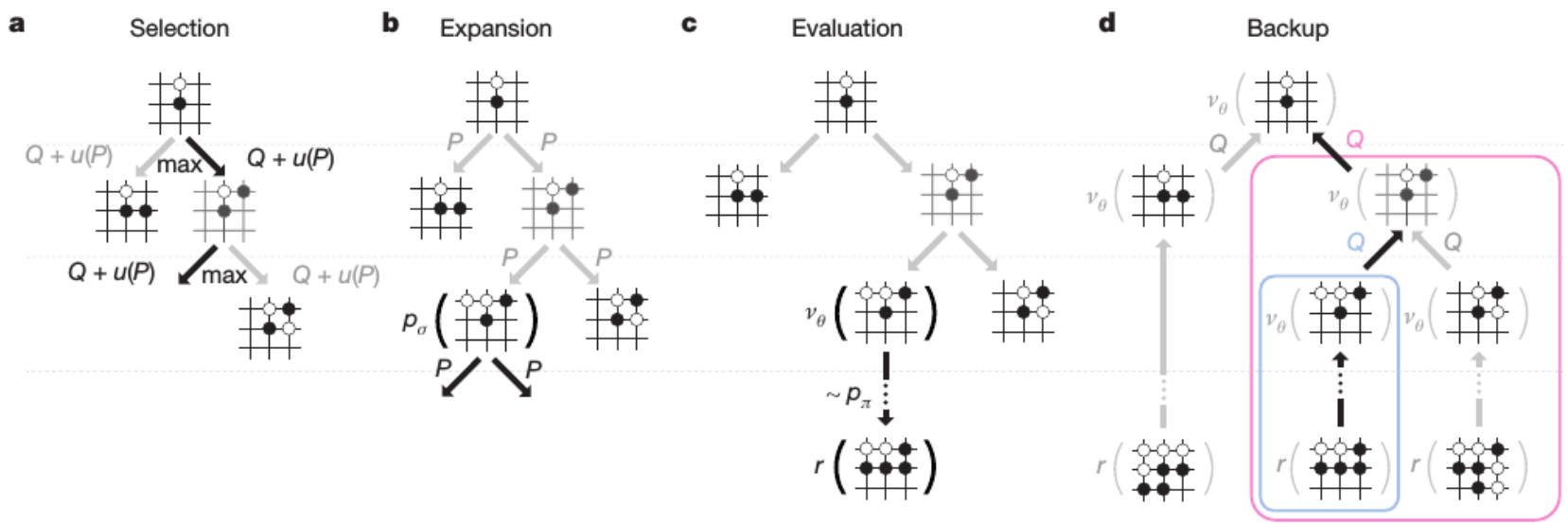
Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.

AlphaGo



Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.

Monte Carlo tree search in AlphaGo



Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.

Remarks

- Non-zero rewards only at win/loss states
- During learning a fast policy (based on human data) is used that plays to the end , the learned policy becomes later more general
- Combination of SL and RL (policy + value) and MCTS
- Neural networks of AlphaGo trained directly from gameplay purely through general-purpose methods (without a handcrafted evaluation function as in DeepBlue)
- AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0, and a human world champion 4-1 or 3-2 (t.b.d. today!), a feat previously thought to be at least a decade away.
- Achieving one of AI's "grand challenges"

Silver, David, et al. "Mastering the game of Go with deep neural networks and tree search." Nature 529.7587 (2016): 484-489.