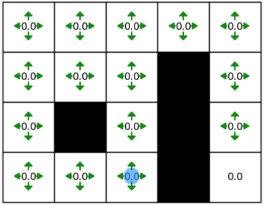
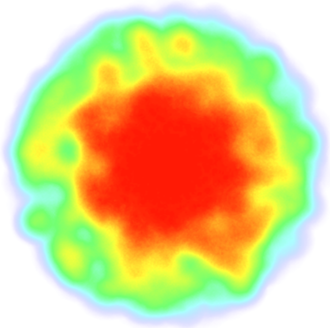
# DQN

Olivier Sigaud

Sorbonne Université
http://people.isir.upmc.fr/sigaud

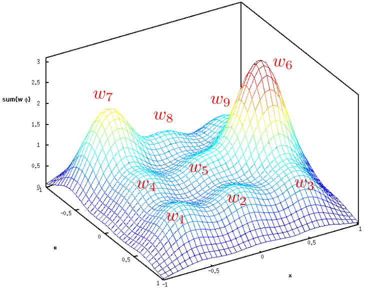
## Parametrized representations



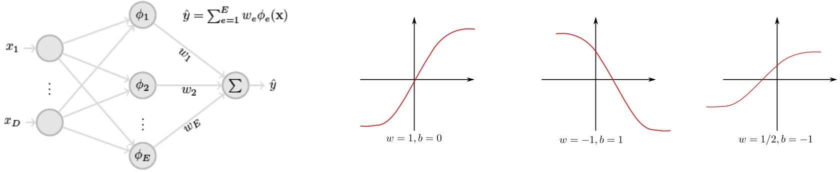
- If the state or action spaces become continuous, we need to represent a function over a continuous domain
- We cannot enumerate all values

## Linear representations




- To represent a continuous function, use features and a vector of parameters
- Learning tunes the weights
- Linear architecture: linear combination of features

## The case of (feedforward) neural networks




- Last layer: linear combination of features (as in linear architectures)
- Sigmoids instead of Gaussians: better split of space in high dimensions
- Weight of input layer(s): tuning basis functions
- Weight of output layer: regression
- The backprop algo tunes both output and hidden weights
- Discovers adequate features by itself in a large space

## General motivations for Deep RL

- Approximation with deep networks provided enough computational power can be very accurate
- All processes rely on efficient backpropagation in deep networks
- Available in CPU/GPU libraries: theano, caffe, ..., TensorFlow, pytorch

Kingma, D. P. & Ba, J. (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*

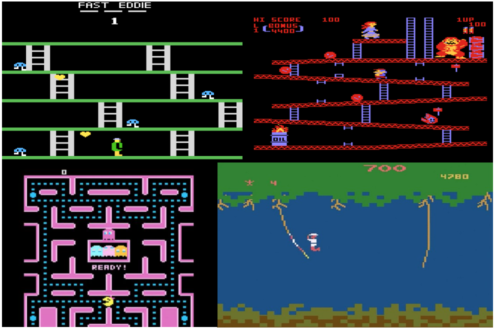
## Early success

- The first world champion was using RL with neural networks
- But it was shown that RL with function approximation can diverge

Tesauro, G. (1995) Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68

Baird, L. C. (1994) Reinforcement learning in continuous time: Advantage updating. *Proceedings of the International Conference on Neural Networks*, Orlando, FL

## DQN: the breakthrough

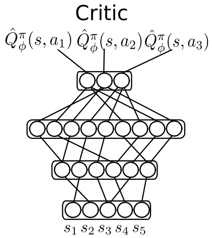



- DQN: Atari domain, Nature paper, small discrete actions set
- Learned very different representations with the same tuning

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015) Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

## The Q-network in DQN

| state / action | $a_0$ | $a_1$ | $a_2$ | $a_3$ |
|---|---|---|---|---|
| $\mathbf{s}_0$ | 0.66 | 0.88* | 0.81 | 0.73 |
| $\mathbf{s}_1$ | 0.73 | 0.63 | 0.9* | 0.43 |
| $\mathbf{s}_2$ | 0.73 | 0.9 | 0.95* | 0.73 |
| $\mathbf{s}_3$ | 0.81 | 0.9 | 1.0* | 0.81 |
| $\mathbf{s}_4$ | 0.81 | 1.0* | 0.81 | 0.9 |
| $\mathbf{s}_5$ | 0.9 | 1.0* | 0.0 | 0.9 |



DQN

- Parametrized representation of the critic $Q_{\phi}(s_t, a_t)$
- The Q-network is the equivalent of the Q-Table (with an infinity of state rows)
- Select action by finding the max (as in Q-LEARNING)
- Limitation: requires one output neuron per action

## Learning the neural Q-function

- Supervised learning: minimize a loss-function, often the squared error w.r.t. the output:

$$L(s,a) = (y^*(s,a) - Q_{\phi}(s,a))^2 \tag{1}$$

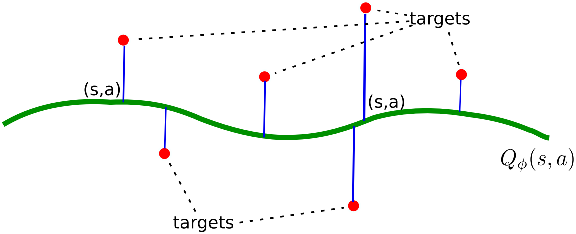
with backprop on weights $\phi$

- For each sample $i$, the Q-network should minimize the RPE:

$$\delta_i = r_{i+1} + \gamma \max_a Q_{\phi}(s_{i+1}, a) - Q_{\phi}(s_i, a_i)$$

- Thus, given a minibatch of $N$ samples $\{s_i, a_i, r_{i+1}, s_{i+1}\}$, compute $y_i = r_{i+1} + \gamma \max_a Q_{\phi}(s_{i+1}, a)$

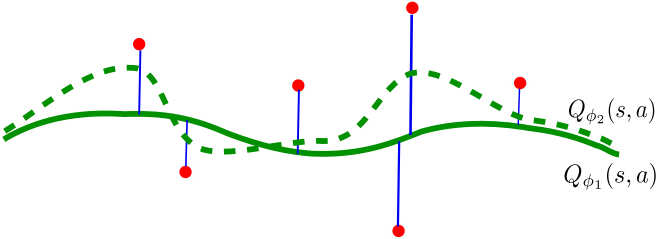
## Learning the neural Q-function




- In the tabular case, each Q-value is updated separately
- In the continuous function approximation setting, interdependencies
- Thus update $\phi$ by minimizing the (squared error) loss function

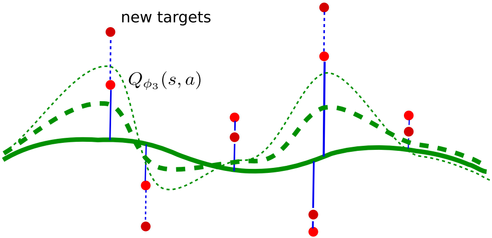
$$L = 1/N \sum_i (y_i - Q_{\phi}(s_i, a_i))^2 \quad (2)$$

## Learning the neural Q-function



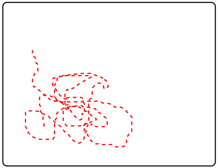

- The neural network weights are updated so as to decrease all errors on average
- Using many mini-batches, one gets global minimization over the whole Q-function

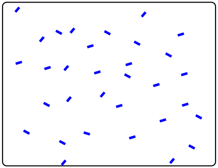
## Trick 1: Stable Target Q-function

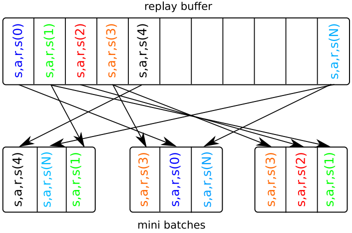



- The target $y_i = r_{i+1} + \gamma \max_a Q_{\boldsymbol{\phi}}(s_{i+1}, a))$ is itself a function of $Q$
- Thus this is not truly supervised learning, and this is unstable
- Key idea: "periods of supervised learning"
- Compute the loss function from a separate *target network* $Q'_{\boldsymbol{\phi}'}(...)$
- So rather compute $y_i = r_{i+1} + \gamma \max_a Q'_{\boldsymbol{\phi}'}(s_{i+1}, a)$
- $\phi'$ is updated to $\phi$ only each $K$ iterations

## Trick2: Replay buffer shuffling




- Agent samples are not independent and identically distributed (i.i.d.)
- Shuffling a replay buffer (RB) makes them more i.i.d.
- It improves a lot the sample efficiency
- Recent data in the RB come from policies close to the current one

Lin, L.-J. (1992) Self-Improving Reactive Agents based on Reinforcement Learning, Planning and Teaching. *Machine Learning*, 8(3/4), 293–321

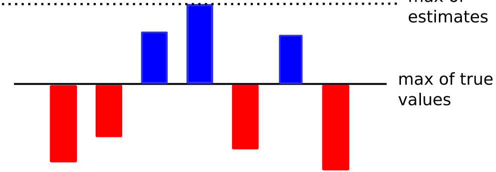
de Bruin, T., Kober, J., Tuyls, K., & Babuška, R. (2015) The importance of experience replay database composition in deep reinforcement learning. In *Deep RL workshop at NIPS 2015*

Zhang, S. & Sutton, R. S. (2017) A deeper look at experience replay. *arXiv preprint arXiv:1712.01275*

## Maximization in RL

- Two maximization steps:
  - In action selection:
    $$\pi(s) \sim \operatorname*{argmax}_{a \in A} Q(s,a)$$
    might be stochastic or contain some exploration
  - In Q-LEARNING, in the value update rule
    $$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max_{a \in A} Q(s_{t+1},a) - Q(s_t,a_t)]$$

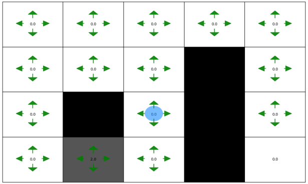
## Maximization bias




- In action selection, a maximum over estimated $Q(s,a)$ is performed
- "In these algorithms, a maximum over estimated values is used implicitly as an estimate of the maximum value, which can lead to a significant positive bias."
- Example: imagine all true $Q(s,a)$ values are null

Sutton, R. S. & Barto, A. G. (2018) *Reinforcement Learning: An Introduction (Second edition)*. MIT Press

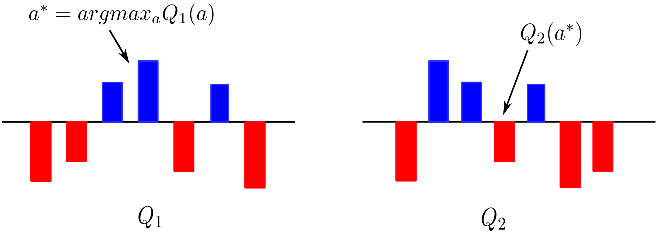
## Over-estimation bias propagation



- Some initial bias cannot be prevented due to Q-Table initialization
- In Q-LEARNING, due to the max operator, the maximization bias propagates
- No propagation of under-estimation
- The same holds for DDPG without a max operator!

Fujimoto, S., van Hoof, H., & Meger, D. (2018) Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*

## Double Q-LEARNING




- Solution: using two Q-Tables, one for value estimation and one for action selection
- $a^* = \operatorname{argmax}_a Q_1(a)$
- $Q_2(a^*) = Q_2(\operatorname{argmax}_a Q_1(a))$ unbiased estimate of $Q(a^*)$
- $a'^* = \operatorname{argmax}_a Q_2(a)$
- $Q_1(a'^*) = Q_1(\operatorname{argmax}_a Q_2(a))$ unbiased estimate of $Q(a'^*)$
- Randomly select one of each at all steps

Van Hasselt, H. (2010) Double q-learning. *Advances in Neural Information Processing Systems*, pages 2613–2621

## Double Q-LEARNING: results

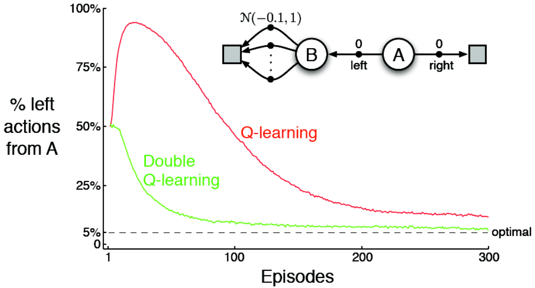



**Figure 6.5:** Comparison of Q-learning and Double Q-learning on a simple episodic MDP (shown inset). Q-learning initially learns to take the left action much more often than the right action, and always takes it significantly more often than the 5% minimum probability enforced by $\varepsilon$-greedy action selection with $\varepsilon = 0.1$. In contrast, Double Q-learning is essentially unaffected by maximization bias. These data are averaged over 10,000 runs. The initial action-value estimates were zero. Any ties in $\varepsilon$-greedy action selection were broken randomly.

## Double-DQN

- The max operator in Q-LEARNING results in the maximization bias
- Double Q-LEARNING: use $Q_1$ and $Q_2$ functions
- Double-DQN: make profit of the target network: propagate on target Q-network, select max on Q-network,
- Minor change with respect to DQN (one line of code)
- Converges twice faster

Van Hasselt, H., Guez, A., & Silver, D. (2015) Deep reinforcement learning with double q-learning. *CoRR, abs/1509.06461*

## Prioritized Experience Replay

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $S_0$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ | ... ... |
| $\delta = 0.8$ | $\delta = 0.7$ | $\delta = 0.5$ | $\delta = 0.2$ | $\delta = 0.1$ | ... | ... | $\delta = 0.01$ | |

- Samples with a greater TD error improve the critic faster
- Give them a higher probability of being selected
- Favors the replay of new $(s, a)$ pairs (largest TD error), as in $R - max$
- Several minor hacks, and interesting discussion
- Converges twice faster

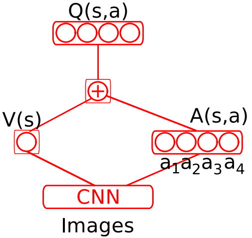
Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015) Prioritized Experience Replay. *arXiv preprint arXiv:1511.05952*

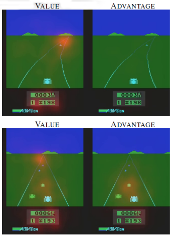
## Advantage function

- $A_\phi(s_i, a_i) = Q_\phi(s_i, a_i) - max_a Q_\phi(s_i, a)$
- Corresponds to a regret for not performing the best action
- $V(s_i) = max_a Q_\phi(s_i, a)$
- Why use it?
  - Put forward by Baird to stabilize function approximation
  - In the likelihood ratio view, corresponds to the optimal baseline (minimizing variance)
  - In compatible actor-critic architecture, corresponds to the natural gradient
  - Link to minimizing the KL divergence between subsequent policies

Deisenroth, M. P., Neumann, G., Peters, J., et al. (2013) A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2), 1–142

## Dueling networks

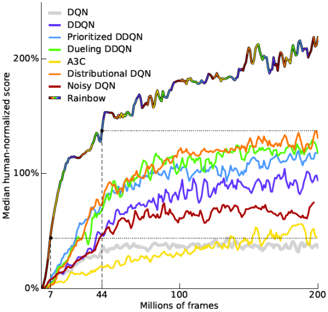



- Instead of $A_{\phi^A}(s_i, a_i) = Q_{\phi^Q}(s_i, a_i) - max_a Q_{\phi^Q}(s_i, a) = Q_{\phi^Q}(s_i, a_i) - V_{\phi^V}(s_i)$
- Rather use $Q_{\phi^Q}(s_i, a_i) = A_{\phi^A}(s_i, a_i) + V_{\phi^V}(s_i)$
- Note that $A_{\phi}(s_i, a_i^*) = 0$
- Center around average A to stabilize: $Q = V + A - average(A)$
- Better captures some relevant aspects of a control task
- Similar idea in NAF with continuous actions

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., & de Freitas, N. (2015) Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*

## Rainbow



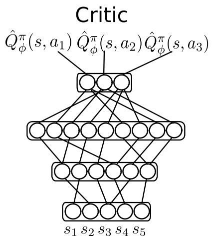

- A3C, distributional DQN and Noisy DQN presented later
- Combining all local improvements

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2017) Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*

# DDPG

## The Q-network in DQN

| state / action | $a_0$ | $a_1$ | $a_2$ | $a_3$ |
|---|---|---|---|---|
| $\mathbf{s}_0$ | 0.66 | 0.88* | 0.81 | 0.73 |
| $\mathbf{s}_1$ | 0.73 | 0.63 | 0.9* | 0.43 |
| $\mathbf{s}_2$ | 0.73 | 0.9 | 0.95* | 0.73 |
| $\mathbf{s}_3$ | 0.81 | 0.9 | 1.0* | 0.81 |
| $\mathbf{s}_4$ | 0.81 | 1.0* | 0.81 | 0.9 |
| $\mathbf{s}_5$ | 0.9 | 1.0* | 0.0 | 0.9 |




DQN

- Parametrized representation of the critic $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)$
- Q-network equivalent to the Q-Table (with an infinity of state rows)
- For each observed $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$:

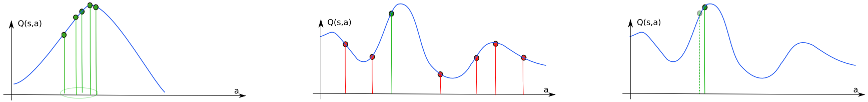
$$Q(\mathbf{s}_t, \mathbf{a}_t) \leftarrow Q(\mathbf{s}_t, \mathbf{a}_t) + \alpha[r_t + \gamma \max_{\mathbf{a} \in A} Q(\mathbf{s}_{t+1}, \mathbf{a}) - Q(\mathbf{s}_t, \mathbf{a}_t)]$$

- Select action by finding $\max_{\mathbf{a} \in A} Q(\mathbf{s}, \mathbf{a})$ (as in Q-LEARNING)
- Limitation: requires one output neuron per action

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015) Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

## Moving to continuous actions

- Two things become too hard:
  - Selecting actions by finding $\max_{\mathbf{a} \in A} Q(\mathbf{s}, \mathbf{a})$
  - Computing $\max_{\mathbf{a} \in A} Q(\mathbf{s}_{t+1}, \mathbf{a})$ in the update rule

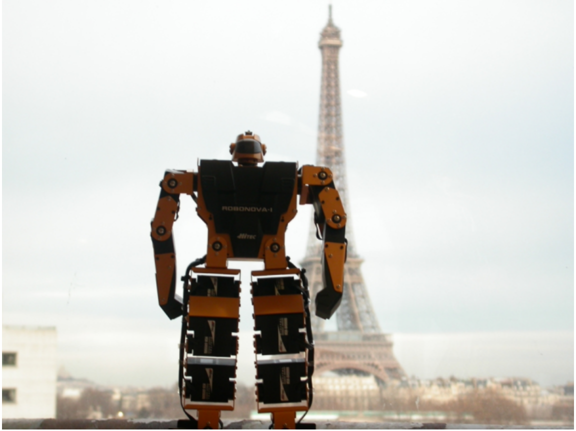



- Three classes of solutions
  1. Use an easily optimized model (e.g. convex) (NAF, Wang et al. 2016)
  2. Sample a limited set of actions (QT-Opt, Kalashnikov et al., 2018)
  3. DDPG: train a side estimator of the best action (also true of SAC)

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Rémi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016

## Any question?




Send mail to: `Olivier.Sigaud@isir.upmc.fr`

Baird, L. C. (1994).
Reinforcement learning in continuous time: Advantage updating.
In *Proceedings of the International Conference on Neural Networks*, Orlando, FL.

de Bruin, T., Kober, J., Tuyls, K., and Babuška, R. (2015).
The importance of experience replay database composition in deep reinforcement learning.
In *Deep RL workshop at NIPS 2015*.

Deisenroth, M. P., Neumann, G., Peters, J., et al. (2013).
A survey on policy search for robotics.
*Foundations and Trends® in Robotics*, 2(1–2):1–142.

Fujimoto, S., van Hoof, H., and Meger, D. (2018).
Addressing function approximation error in actor-critic methods.
In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1582–1591. PMLR.

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M. G., and Silver, D. (2018).
Rainbow: Combining improvements in deep reinforcement learning.
In McIlraith, S. A. and Weinberger, K. Q., editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3215–3222. AAAI Press.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al. (2018).
QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation.
*arXiv preprint arXiv:1806.10293*.

Kingma, D. P. and Ba, J. (2015).
Adam: A method for stochastic optimization.
In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015).
Human-level control through deep reinforcement learning.
*Nature*, 518(7540):529–533.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016).
Prioritized experience replay.
In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Sutton, R. S. and Barto, A. G. (2018).
*Reinforcement Learning: An Introduction (Second edition)*.
MIT Press.

Tesauro, G. (1995).
Temporal difference learning and td-gammon.
*Communications of the ACM*, 38(3):58–68.

van Hasselt, H. (2010).
Double q-learning.
In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, pages 2613–2621. Curran Associates, Inc.

van Hasselt, H., Guez, A., and Silver, D. (2016).
Deep reinforcement learning with double q-learning.
In Schuurmans, D. and Wellman, M. P., editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2094–2100. AAAI Press.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. (2017).
Sample efficient actor-critic with experience replay.
In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Wang, Z., Schaul, T., Hessel, M., van Hasselt, H., Lanctot, M., and de Freitas, N. (2016).
Dueling network architectures for deep reinforcement learning.
In Balcan, M. and Weinberger, K. Q., editors, *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1995–2003. JMLR.org.

Zhang, S. and Sutton, R. S. (2017).
A deeper look at experience replay.
*arXiv preprint arXiv:1712.01275*.