Dueling Network

Shusen Wang

Advantage Function

Return

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Value Functions

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[U_t | S_t = s_t, A_t = a_t\right].$

Value Functions

Definition: Discounted return.

- $U_t = R_t + \gamma \cdot R_{t+1} + \gamma^2 \cdot R_{t+2} + \gamma^3 \cdot R_{t+3} + \cdots$

Definition: Action-value function.

- $Q_\pi(s_t, a_t) = \mathbb{E}\left[U_t | S_t = s_t, A_t = a_t\right].$

Definition: State-value function.

- $V_\pi(s_t) = \mathbb{E}_A\left[Q_\pi(s_t, A)\right]$

Optimal Value Functions

Definition: Optimal action-value function.

- $Q^{\star}(s, a) = \max_{\pi} Q_{\pi}(s, a).$

Optimal Value Functions

Definition: Optimal action-value function.

- $Q^\star(s, a) = \max_\pi Q_\pi(s, a)$.

Definition: Optimal state-value function.

- $V^\star(s) = \max_\pi V_\pi(s)$.

Optimal Value Functions

Definition: Optimal action-value function.

- $Q^{\star}(s, a) = \max_{\pi} Q_{\pi}(s, a)$.

Definition: Optimal state-value function.

- $V^{\star}(s) = \max_{\pi} V_{\pi}(s)$.

Definition: Optimal advantage function.

- $A^{\star}(s, a) = Q^{\star}(s, a) - V^{\star}(s)$.

Properties of Advantage Function

Theorem 1: $V^{\star}(s) = \max_{a} Q^{\star}(s, a)$.

Properties of Advantage Function

Theorem 1: $V^\star(s) = \max_{a} Q^\star(s, a)$.

- Recall the definition of the optimal advantage function:

$$A^\star(s, a) = Q^\star(s, a) - V^\star(s).$$

Properties of Advantage Function

Theorem 1: $V^{\star}(s) = \max_{a} Q^{\star}(s, a).$

- Recall the definition of the optimal advantage function:

$$A^{\star}(s, a) = Q^{\star}(s, a) - V^{\star}(s).$$

- It follows that

$$\max_{a} A^{\star}(s, a) = \underbrace{\max_{a} Q^{\star}(s, a) - V^{\star}(s).}_{= 0}$$

Properties of Advantage Function

Theorem 1: $V^\star(s) = \max_a Q^\star(s, a)$.

- Recall the definition of the optimal advantage function:

$$A^\star(s, a) = Q^\star(s, a) - V^\star(s).$$

- It follows that

$$\max_a A^\star(s, a) = 0$$

Properties of Advantage Function

Definition of advantage: $A^{\star}(s, a) = Q^{\star}(s, a) - V^{\star}(s)$.

$$Q^{\star}(s, a) = V^{\star}(s) + A^{\star}(s, a)$$

Properties of Advantage Function

Definition of advantage: $A^{\star}(s, a) = Q^{\star}(s, a) - V^{\star}(s)$.

$$Q^{\star}(s, a) = V^{\star}(s) + A^{\star}(s, a) - \underbrace{\max_{a} A^{\star}(s, a)}_{= 0}.$$

Properties of Advantage Function

Definition of advantage: $A^\star(s,a) = Q^\star(s,a) - V^\star(s)$.

Theorem 2: $Q^\star(s,a) = V^\star(s) + A^\star(s,a) - \underbrace{\max_{a} A^\star(s,a)}_{=\,0}$.

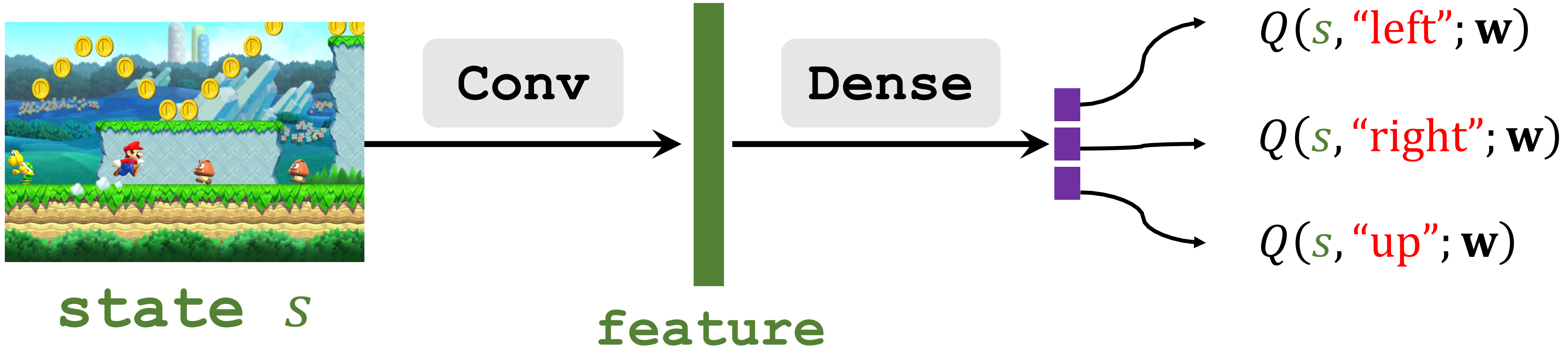
Dueling Network

Reference:

1. Wang et al. Dueling network architectures for deep reinforcement learning. In *ICML*, 2016.

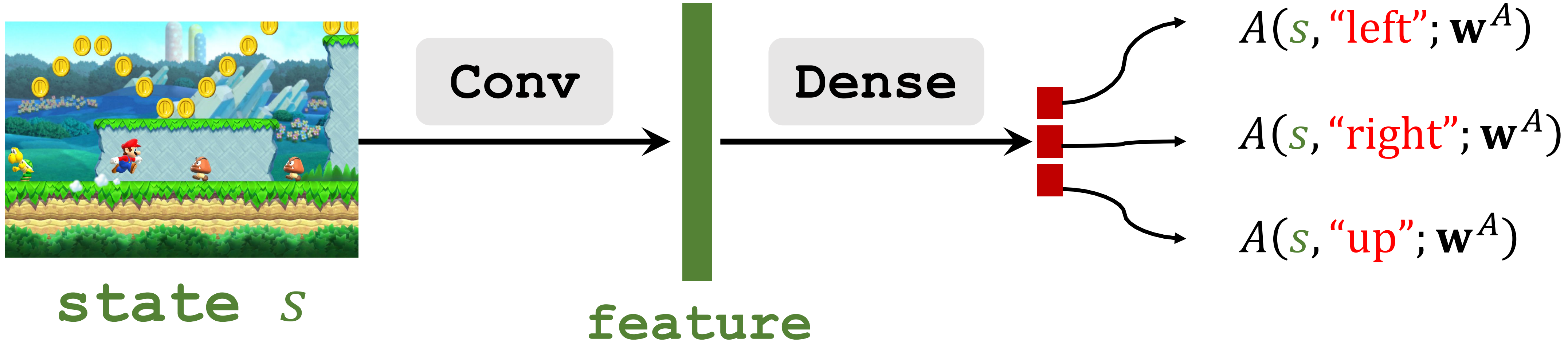
Revisiting DQN

- Approximate $Q^{\star}(s, a)$ by a neural network, $Q(s, a; \mathbf{w})$.



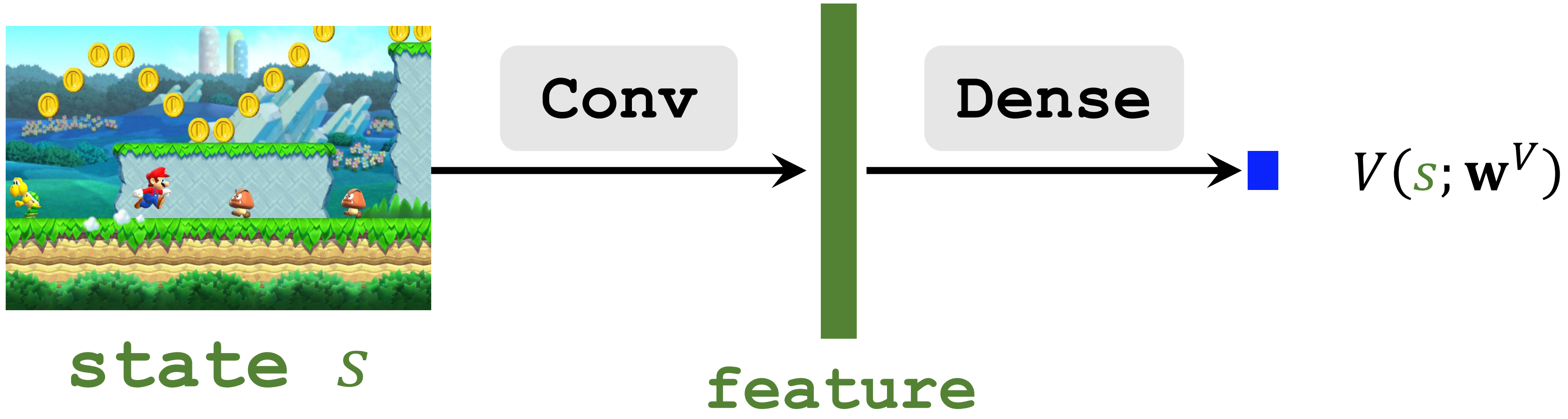
Approximating Advantage Function

- Approximate $A^{\star}(s, a)$ by a neural network, $A(s, a; \mathbf{w}^{A})$.



Approximating State-Value Function

- Approximate $V^{\star}(s)$ by a neural network, $V(s;\mathbf{w}^V)$.



Dueling Network: Formulation

Theorem 2: $Q^{\star}(s, a) = V^{\star}(s) + A^{\star}(s, a) - \max_{a} A^{\star}(s, a).$

Dueling Network: Formulation

Theorem 2: $Q^{\star}(s, a) = V^{\star}(s) + A^{\star}(s, a) - \max_{a} A^{\star}(s, a).$

- Approximate $V^{\star}(s)$ by a neural network, $V(s; \mathbf{w}^{V})$.

Dueling Network: Formulation

Theorem 2: $Q^\star(s,a) = V^\star(s) + A^\star(s,a) - \max_a A^\star(s,a)$.

- Approximate $V^\star(s)$ by a neural network, $V(s;\mathbf{w}^V)$.
- Approximate $A^\star(s,a)$ by a neural network, $A(s,a;\mathbf{w}^A)$.

Dueling Network: Formulation

Theorem 2: $Q^\star(s, a) = V^\star(s) + A^\star(s, a) - \max_a A^\star(s, a).$

- Approximate $V^\star(s)$ by a neural network, $V(s; \mathbf{w}^V)$.
- Approximate $A^\star(s, a)$ by a neural network, $A(s, a; \mathbf{w}^A)$.
- Thus, approximate $Q^\star(s, a)$ by the dueling network:

$$Q(s, a; \mathbf{w}^A, \mathbf{w}^V) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \max_a A(s, a; \mathbf{w}^A).$$

Dueling Network: Formulation

Theorem 2: $Q^{\star}(s,a) = V^{\star}(s) + A^{\star}(s,a) - \max_{a} A^{\star}(s,a).$

- Approximate $V^{\star}(s)$ by a neural network, $V(s;\mathbf{w}^{V})$.
- Approximate $A^{\star}(s,a)$ by a neural network, $A(s,a;\mathbf{w}^{A})$.
- Thus, approximate $Q^{\star}(s,a)$ by the dueling network:

$$Q(s,a;\mathbf{w}^{A},\mathbf{w}^{V}) = V(s;\mathbf{w}^{V}) + A(s,a;\mathbf{w}^{A}) - \max_{a} A(s,a;\mathbf{w}^{A}).$$

Dueling Network: Formulation

Theorem 2: $Q^{\star}(s,a) = V^{\star}(s) + A^{\star}(s,a) - \max_{a} A^{\star}(s,a).$

- Approximate $V^{\star}(s)$ by a neural network, $V(s;\mathbf{w}^{V})$.
- Approximate $A^{\star}(s,a)$ by a neural network, $A(s,a;\mathbf{w}^{A})$.
- Thus, approximate $Q^{\star}(s,a)$ by the dueling network:

$$Q(s,a;\boxed{\mathbf{w}^{A},\mathbf{w}^{V}}) = V(s;\mathbf{w}^{V}) + A(s,a;\mathbf{w}^{A}) - \max_{a} A(s,a;\mathbf{w}^{A}).$$

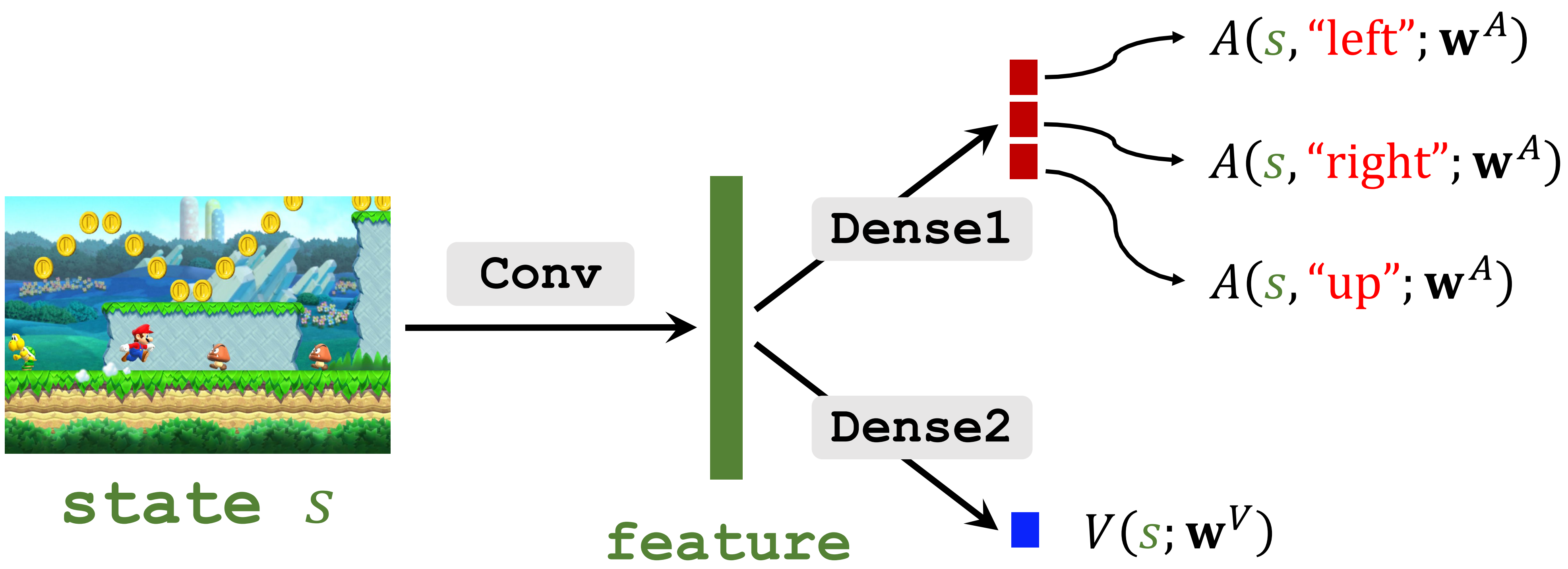
$$\mathbf{w} = (\mathbf{w}^{A},\mathbf{w}^{V})$$

Dueling Network

$$Q(s, a; \mathbf{w}) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \max_{a} A(s, a; \mathbf{w}^A).$$

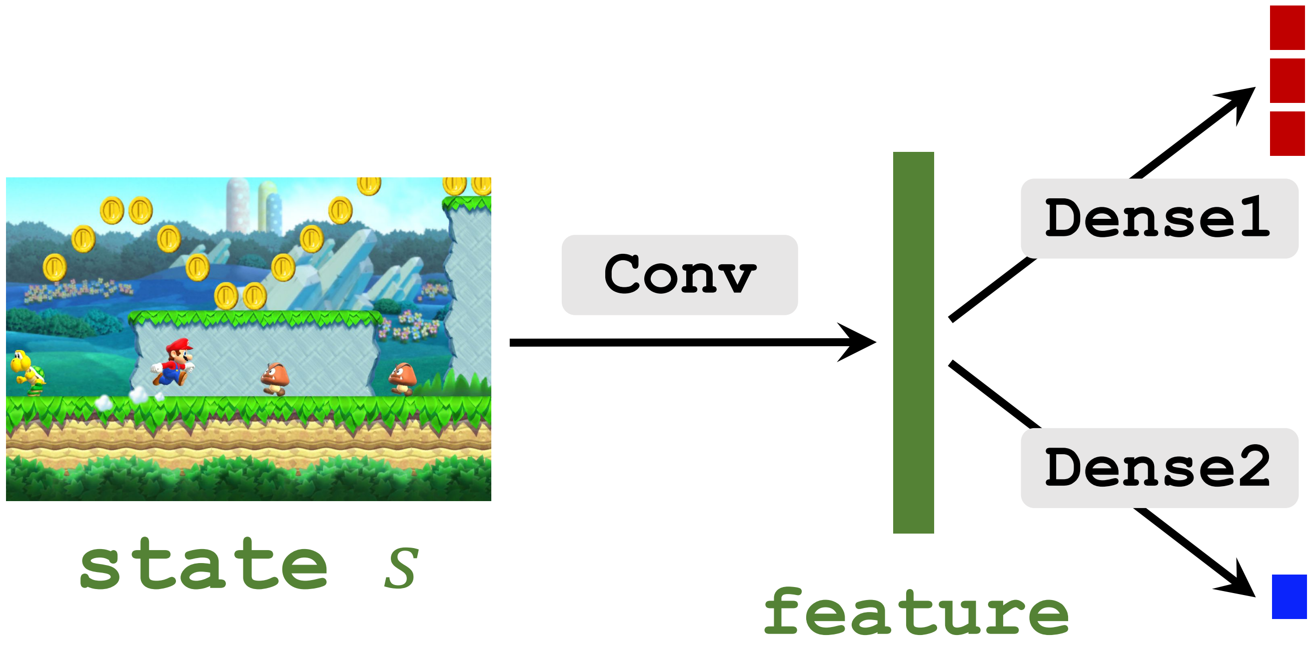
Dueling Network

$$Q(s, a; \mathbf{w}) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \max_a A(s, a; \mathbf{w}^A).$$



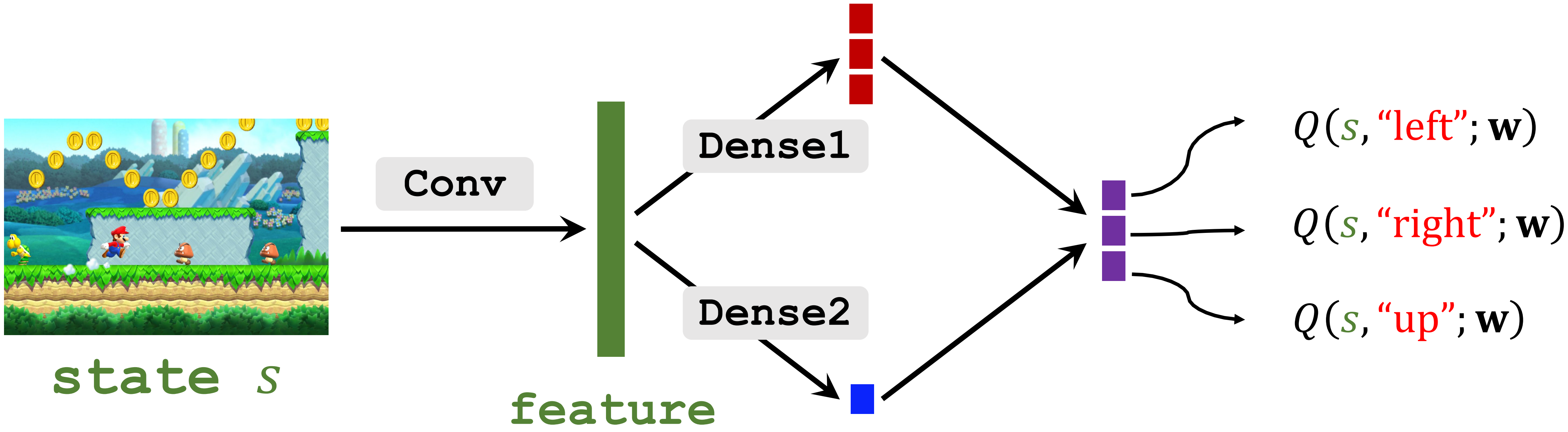
Dueling Network

$$Q(s, a; \mathbf{w}) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \max_a A(s, a; \mathbf{w}^A).$$



Dueling Network

$$Q(s, a; \mathbf{w}) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \max_a A(s, a; \mathbf{w}^A).$$



Training

- Dueling network, $Q(s, a; \mathbf{w})$, is an approximation to $Q^\star(s, a)$.
- Learn the parameter, $\mathbf{w} = (\mathbf{w}^A, \mathbf{w}^V)$, in the same way as the other DQNs.
- Tricks can be used in the same way.
    - Prioritized experience replay.
    - Double DQN.
    - Multi-step TD target.

Overcome Non-identifiability

Problem of Non-identifiability

- **Equation 1:** $Q^\star(s,a) = V^\star(s) + A^\star(s,a).$
- **Equation 2:** $Q^\star(s,a) = V^\star(s) + A^\star(s,a) - \boxed{\max_a A^\star(s,a)}$

Question: Why is the zero term necessary?

Problem of Non-identifiability

- **Equation 1:** $Q^\star(s,a) = V^\star(s) + A^\star(s,a).$
- Equation 1 has the problem of *non-identifiability*.

Problem of Non-identifiability

- **Equation 1:** $Q^{\star}(s,a) = V^{\star}(s) + A^{\star}(s,a)$.
- Equation 1 has the problem of *non-identifiability*.
  - Let $V' = V^{\star} + 10$ and $A' = A^{\star} - 10$.

Problem of Non-identifiability

- **Equation 1:** $Q^\star(s,a) = V^\star(s) + A^\star(s,a).$
- Equation 1 has the problem of *non-identifiability*.
  - Let $V' = V^\star + 10$ and $A' = A^\star - 10.$
  - Then $Q^\star(s,a) = V^\star(s) + A^\star(s,a) = V'(s) + A'(s,a).$

Problem of Non-identifiability

- **Equation 1:** $Q^\star(s,a) = V^\star(s) + A^\star(s,a)$.
- Equation 1 has the problem of *non-identifiability*.
  - Let $V' = V^\star + 10$ and $A' = A^\star - 10$.
  - Then $Q^\star(s,a) = V^\star(s) + A^\star(s,a) = V'(s) + A'(s,a)$.
- Why is non-identifiability a problem?

Problem of Non-identifiability

- **Equation 2:** $Q^{\star}(s,a) = V^{\star}(s) + A^{\star}(s,a) - \max_{a} A^{\star}(s,a).$
- Equation 2 does not have the problem.

Dueling Network

$$Q(s, a; \mathbf{w}) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \max_{a} A(s, a; \mathbf{w}^A).$$

Alternative:

$$Q(s, a; \mathbf{w}) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \operatorname*{mean}_{a} A(s, a; \mathbf{w}^A).$$

Summary

- **Dueling network:**

$$Q(s, a; \mathbf{w}) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \underset{a}{\text{mean}}\, A(s, a; \mathbf{w}^A).$$

Summary

- **Dueling network:**

$$Q(s, a; \mathbf{w}) = V(s; \mathbf{w}^V) + A(s, a; \mathbf{w}^A) - \underset{a}{\mathrm{mean}}\, A(s, a; \mathbf{w}^A).$$

- Dueling network controls the agent in the same way as DQN.
- Train dueling network by TD in the same way as DQN.
- (Do not train V and A separately.)

Thank you!