

RL: Deep

Dueling Networks

Marius Lindauer



Leibniz
Universität
Hannover



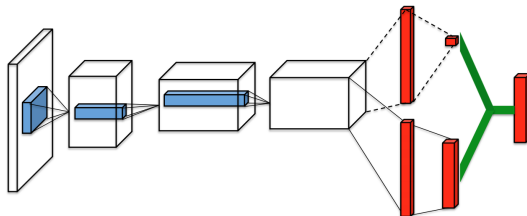
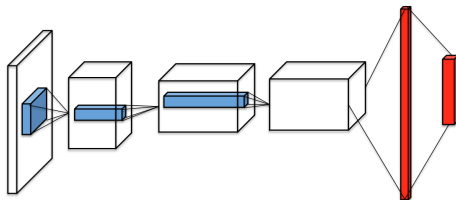
Winter Term 2021

Value & Advantage Function

- ▶ Intuition: Features need to accurately represent value may be different than those needed to specify difference in actions
- ▶ E.g.
 - ▶ Game score may help accurately predict $V(s)$
 - ▶ But not necessarily in indicating relative action values $Q(s, a_1)$ vs $Q(s, a_2)$
- ▶ Advantage function [Baird 1993] $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$

Dueling DQN [Wang et al. 2016]

Standard DQN



Dueling DQN

Dueling DQN [Wang et al. 2016]

- Advantage function [Baird

1993] $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$ Consider a network that outputs $V(s; \vec{w}_1, \vec{w}_2)$ as well as advantage $A(s, a; \vec{w}_1, \vec{w}_3)$ where \vec{w}_i are the weights of the different parts of the network

- To construct Q could use

$$Q(s, a; \vec{w}_1, \vec{w}_2, \vec{w}_3) = V(s; \vec{w}_1, \vec{w}_2) + A(s, a; \vec{w}_1, \vec{w}_3)$$

Dueling DQN [Wang et al. 2016]

- Advantage function [Baird

1993] $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$ Consider a network that outputs $V(s; \vec{w}_1, \vec{w}_2)$ as well as advantage $A(s, a; \vec{w}_1, \vec{w}_3)$ where \vec{w}_i are the weights of the different parts of the network

- To construct Q could use

$$Q(s, a; \vec{w}_1, \vec{w}_2, \vec{w}_3) = V(s; \vec{w}_1, \vec{w}_2) + A(s, a; \vec{w}_1, \vec{w}_3)$$

- Challenge: There doesn't have to be a unique advantage function

Uniqueness

- ▶ Consider a network that outputs $V(s; \vec{w}_1, \vec{w}_2)$ as well as advantage $A(s, a; \vec{w}_1, \vec{w}_3)$
- ▶ To construct Q could use

$$Q(s, a; \vec{w}_1, \vec{w}_2, \vec{w}_3) = V(s; \vec{w}_1, \vec{w}_2) + A(s, a; \vec{w}_1, \vec{w}_3)$$

- ▶ Option 1: Force $Q(s, a) = V(s)$ for the best action suggested by the advantage:

$$\hat{Q}(s, a; \vec{w}) = \hat{V}(s; \vec{w}) + \left(\hat{A}(s, a; \vec{w}) - \max_{a' \in \mathcal{A}} \hat{A}(s, a'; \vec{w}) \right)$$

- ▶ This helps to force the V network to approximate V
- ▶ Option 2: Use mean as baseline (more stable)

$$\hat{Q}(s, a; \vec{w}) = \hat{V}(s; \vec{w}) + \left(\hat{A}(s, a; \vec{w}) - \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} \hat{A}(s, a'; \vec{w}) \right)$$

- ▶ More stable often because averaging over all advantages instead of the advantage of the current max action.