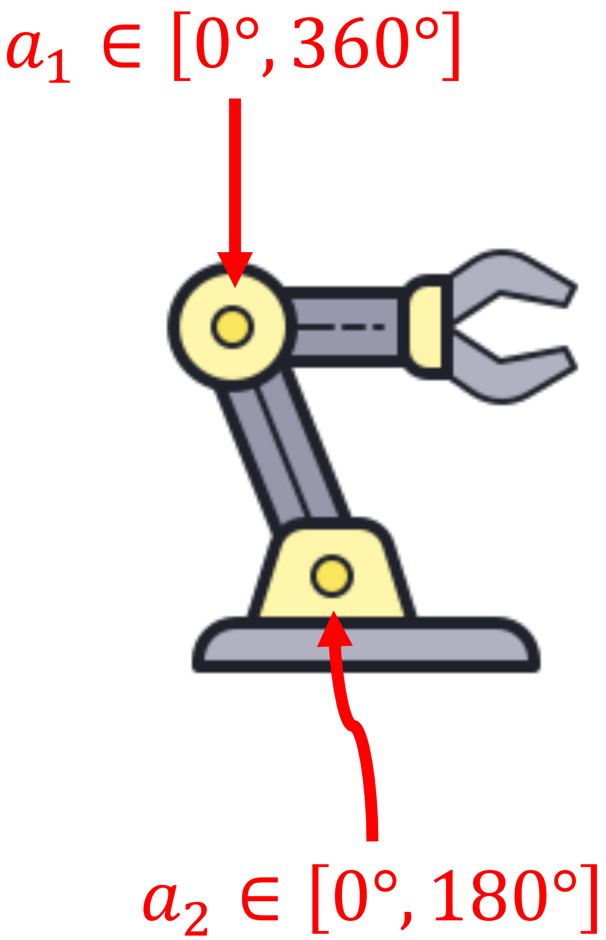
Stochastic Policy for Continuous Control

Shusen Wang

http://wangshusen.github.io/

Continuous Action Space



- The action space $\mathcal{A}$ is a subset of $\mathbb{R}^2$.
- The action space $\mathcal{A}$ is continuous:

$$\mathcal{A} = [0°, 360°] \times [0°, 180°].$$

- Actions are 2-dim vectors.

Policy Network

Univariate Normal Distribution

- Assume the degree of freedom is one, i.e., $\mathcal{A} \subset \mathbb{R}$.
- Let μ (mean) and σ (std) be functions of s.
- Let policy function be the PDF of normal distribution:

$$\pi(a|s) = \frac{1}{\sqrt{6.28}\,\sigma} \cdot \exp\left(-\frac{(a-\mu)^2}{2\sigma^2}\right).$$

Multivariate Normal Distribution

- Let the degree of freedom be d, i.e., action $\mathbf{a}$ is d-dim.
- Let $\boldsymbol{\mu}, \boldsymbol{\sigma}$: $\mathcal{S} \mapsto \mathbb{R}^d$ be functions of s.
- Let μ_i and σ_i be the i-th elements of $\boldsymbol{\mu}(s)$ and $\boldsymbol{\sigma}(s)$, respectively.
- Let policy function be the PDF of multivariate normal:

$$\pi(\mathbf{a}|s) = \prod_{i=1}^{d} \frac{1}{\sqrt{6.28}\,\sigma_i} \cdot \exp\left(-\frac{(a_i-\mu_i)^2}{2\sigma_i^2}\right).$$

Multivariate Normal Distribution

- Let the degree of freedom be d, i.e., action $\mathbf{a}$ is d-dim.
- Let $\boldsymbol{\mu}, \boldsymbol{\sigma}$: $\mathcal{S} \mapsto \mathbb{R}^d$ be functions of s.
- Let μ_i and σ_i be the i-th elements of $\boldsymbol{\mu}(s)$ and $\boldsymbol{\sigma}(s)$, respectively.
- Let policy function be the PDF of multivariate normal:

$$\pi(\mathbf{a}|s) = \prod_{i=1}^{d} \frac{1}{\sqrt{6.28}\,\sigma_i} \cdot \exp\left(-\frac{(a_i-\mu_i)^2}{2\sigma_i^2}\right).$$

Problem: $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ (which are functions of s) are unknown.

Function Approximation

- Approximate the mean, $\boldsymbol{\mu}(s)$, by the neural network, $\boldsymbol{\mu}(s; \boldsymbol{\theta}^{\mu})$.
- ~~Approximate the std, $\boldsymbol{\sigma}(s)$, by the neural network, $\boldsymbol{\sigma}(s; \boldsymbol{\theta}^{\sigma})$.~~

Function Approximation

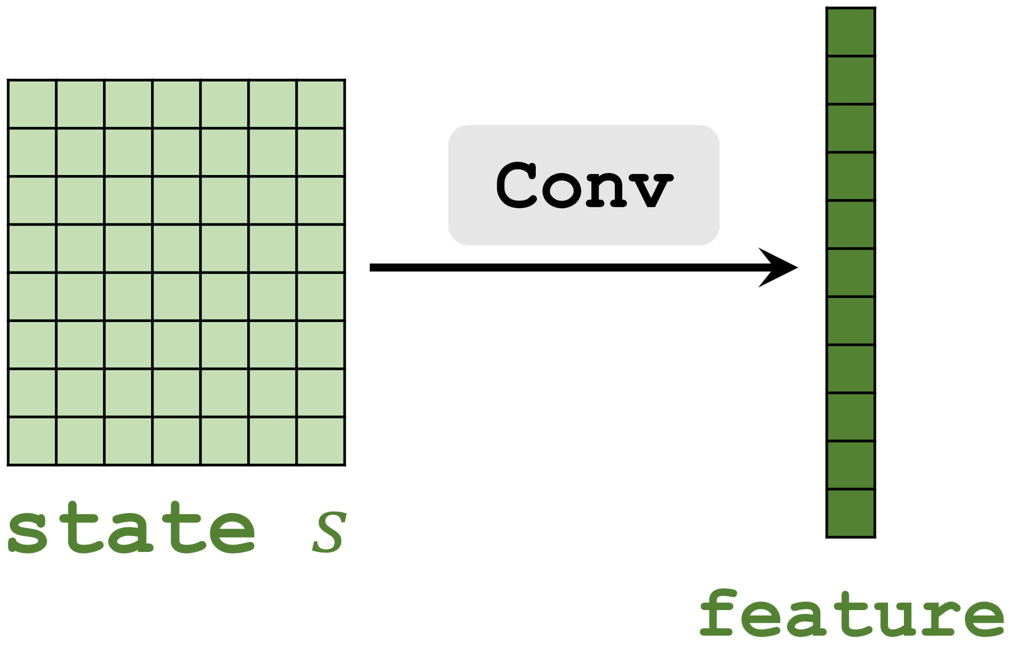
- Approximate the mean, $\boldsymbol{\mu}(s)$, by the neural network, $\boldsymbol{\mu}(s; \boldsymbol{\theta}^{\mu})$.
- ~~Approximate the std, $\boldsymbol{\sigma}(s)$, by the neural network, $\boldsymbol{\sigma}(s; \boldsymbol{\theta}^{\sigma})$.~~
- A better practice is to approximate the log variance:

$$\rho_i = \ln \sigma_i^2, \quad \text{for } i = 1, \cdots, d.$$

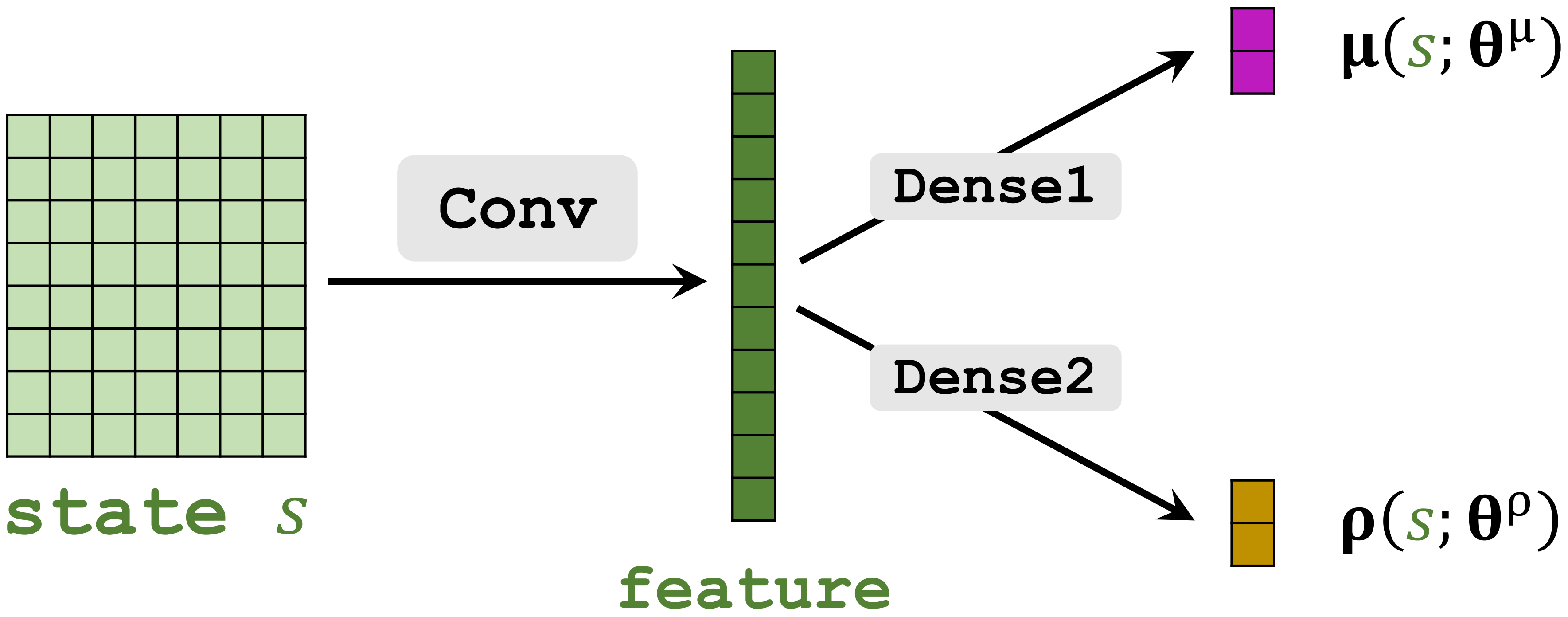
- Approximate $\boldsymbol{\rho}$, by the neural network, $\boldsymbol{\rho}(s; \boldsymbol{\theta}^{\rho})$.

Function Approximation

Function Approximation



Function Approximation



Continuous Control

- Observe state s.
- Compute mean and log variance using the neural network:

$$\widehat{\boldsymbol{\mu}} = \boldsymbol{\mu}(s; \boldsymbol{\theta}^{\mu}) \quad \text{and} \quad \widehat{\boldsymbol{\rho}} = \boldsymbol{\rho}(s; \boldsymbol{\theta}^{\rho}).$$

Continuous Control

- Observe state s.
- Compute mean and log variance using the neural network:

$$\widehat{\boldsymbol{\mu}} = \boldsymbol{\mu}(s; \boldsymbol{\theta}^{\mu}) \quad \text{and} \quad \widehat{\boldsymbol{\rho}} = \boldsymbol{\rho}(s; \boldsymbol{\theta}^{\rho}).$$

- Compute $\hat{\sigma}_i^2 = \exp(\hat{\rho}_i)$, for all $i = 1, \cdots, d$.
- Randomly sample action $\mathbf{a}$ by

$$a_i \sim N\left(\hat{\mu}_i, \hat{\sigma}_i^2\right), \quad \text{for all } i = 1, \cdots, d.$$

Training Policy Network

1. Auxiliary network (for computing policy gradient).
2. Policy gradient.
3. Algorithms: actor-critic and REINFORCE.

Training (1/4): Auxiliary Network

Policy Network

- The policy network is:

$$\boxed{\pi(\mathbf{a}|s;\boldsymbol{\theta}^{\mu},\boldsymbol{\theta}^{\rho})} = \boxed{\prod_{i=1}^{d} \frac{1}{\sqrt{6.28}\,\sigma_i} \cdot \exp\left(-\frac{(a_i-\mu_i)^2}{2\sigma_i^2}\right).}$$

Policy Network

- The policy network is:

$$\pi(\mathbf{a}|s;\boldsymbol{\theta}^{\mu},\boldsymbol{\theta}^{\rho}) = \prod_{i=1}^{d} \frac{1}{\sqrt{6.28}\,\sigma_i} \cdot \exp\left(-\frac{(a_i-\mu_i)^2}{2\sigma_i^2}\right).$$

- The natural log of the policy network is:

$$\boxed{\ln\pi(\mathbf{a}|s;\boldsymbol{\theta}^{\mu},\boldsymbol{\theta}^{\rho})} = \sum_{i=1}^{d}\left[-\ln\sigma_i - \frac{(a_i-\mu_i)^2}{2\sigma_i^2}\right] + \text{const}$$

Policy Network

- The policy network is:

$$\pi(\mathbf{a}|s;\boldsymbol{\theta}^{\mu},\boldsymbol{\theta}^{\rho}) = \prod_{i=1}^{d} \frac{1}{\sqrt{6.28}\,\sigma_i} \cdot \exp\left(-\frac{(a_i-\mu_i)^2}{2\sigma_i^2}\right).$$

- The natural log of the policy network is:

$$\ln \pi(\mathbf{a}|s;\boldsymbol{\theta}^{\mu},\boldsymbol{\theta}^{\rho}) = \sum_{i=1}^{d} \underline{\left[-\ln \sigma_i - \frac{(a_i-\mu_i)^2}{2\sigma_i^2}\right]} + \text{const}$$

$$= \sum_{i=1}^{d} \underline{\left[-\frac{\rho_i}{2} - \frac{(a_i-\mu_i)^2}{2\cdot\exp(\rho_i)}\right]} + \text{const.}$$

Auxiliary Network

Identity: $\ln \pi(\mathbf{a}|s;\boldsymbol{\theta}) = \text{const} + \sum_{i=1}^{d}\left[-\frac{\rho_i}{2} - \frac{(a_i-\mu_i)^2}{2\cdot\exp(\rho_i)}\right].$

Here, $\boldsymbol{\theta} = (\boldsymbol{\theta}^{\mu}, \boldsymbol{\theta}^{\rho})$.

Auxiliary Network

Identity: $\ln \pi(\mathbf{a}|s;\boldsymbol{\theta}) = \text{const} + \sum_{i=1}^{d}\left[-\frac{\rho_i}{2} - \frac{(a_i-\mu_i)^2}{2\cdot\exp(\rho_i)}\right].$

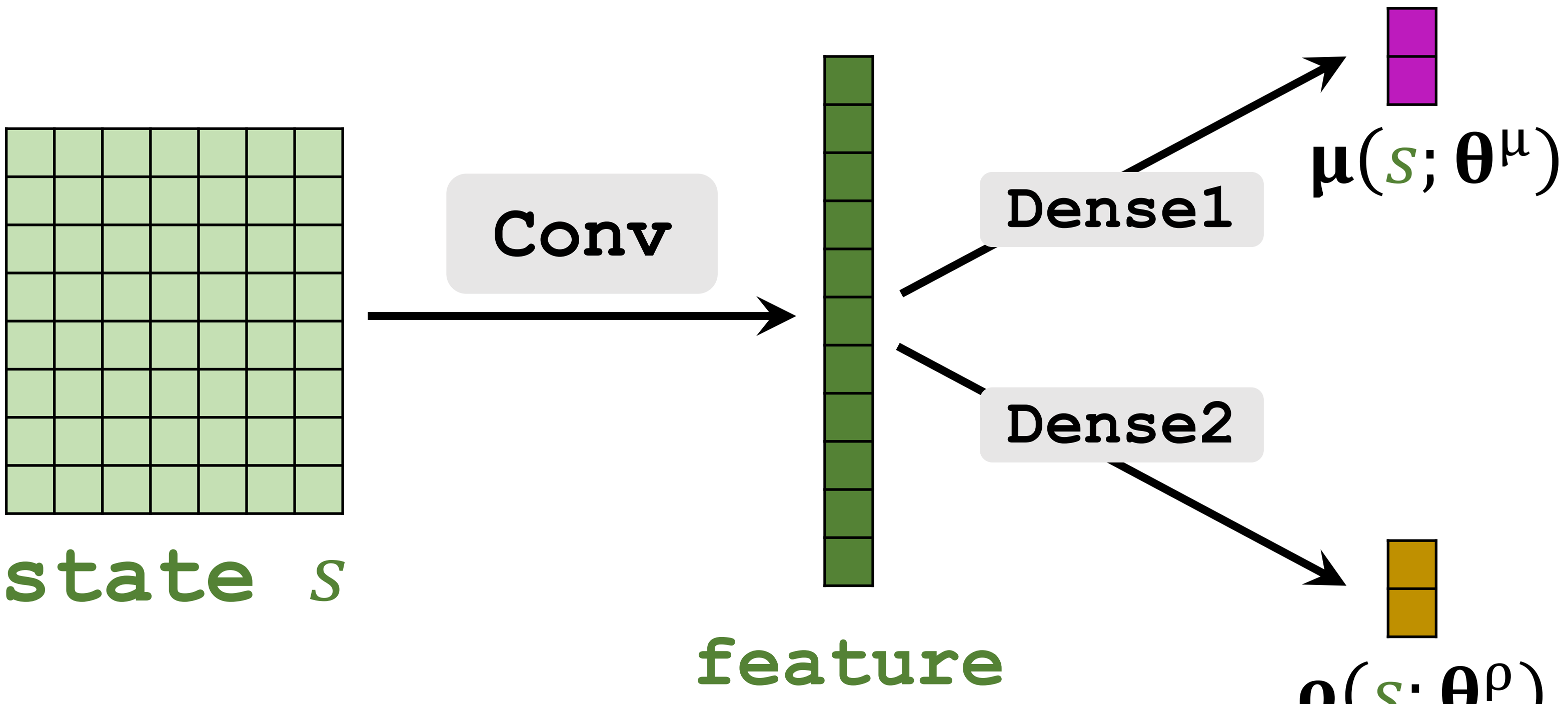
$= f(s, \mathbf{a}; \boldsymbol{\theta})$ (Auxiliary Network)

Auxiliary Network

Auxiliary Network: $f(s, \mathbf{a}; \boldsymbol{\theta}) = \sum_{i=1}^{d} \left[-\frac{\rho_i}{2} - \frac{(a_i - \mu_i)^2}{2 \cdot \exp(\rho_i)} \right].$

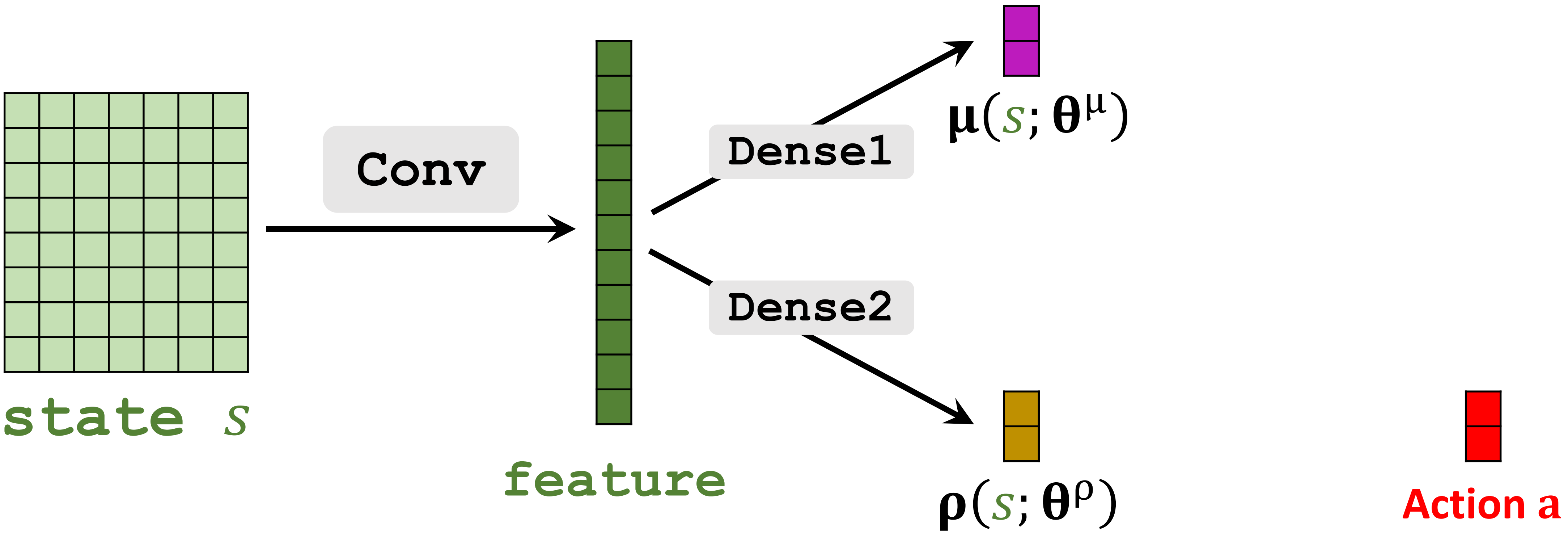
Auxiliary Network

Auxiliary Network: $f(s, \mathbf{a}; \boldsymbol{\theta}) = \sum_{i=1}^{d} \left[-\frac{\rho_i}{2} - \frac{(a_i - \mu_i)^2}{2 \cdot \exp(\rho_i)} \right].$



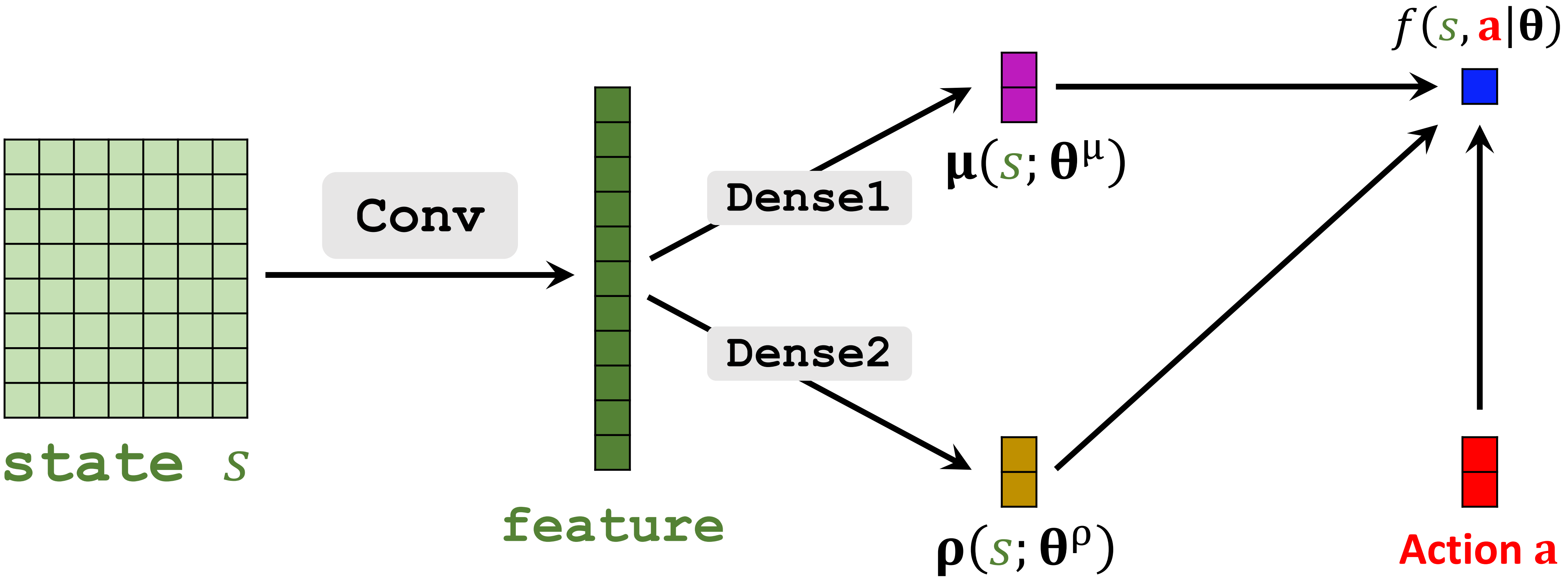
Auxiliary Network

Auxiliary Network: $f(s, \mathbf{a}; \boldsymbol{\theta}) = \sum_{i=1}^{d} \left[-\frac{\rho_i}{2} - \frac{(a_i - \mu_i)^2}{2 \cdot \exp(\rho_i)} \right].$



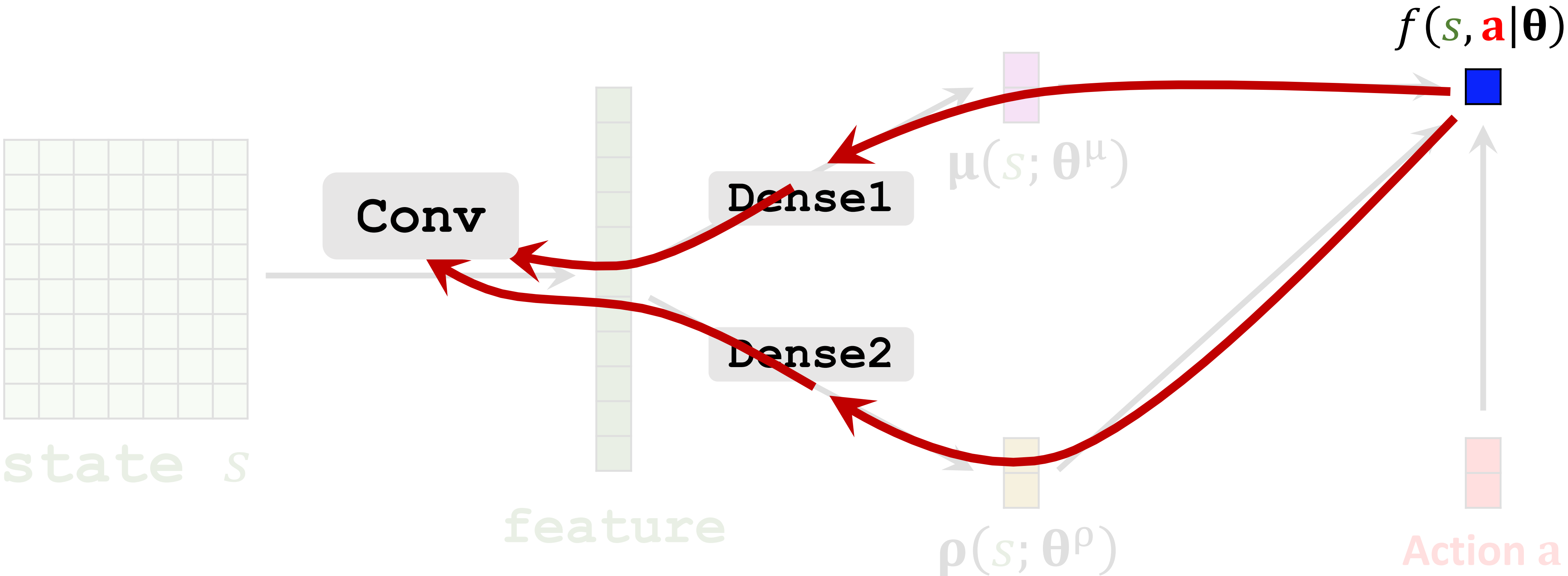
Auxiliary Network

Auxiliary Network: $f(s, \mathbf{a}; \boldsymbol{\theta}) = \sum_{i=1}^{d} \left[-\frac{\rho_i}{2} - \frac{(a_i - \mu_i)^2}{2 \cdot \exp(\rho_i)} \right].$



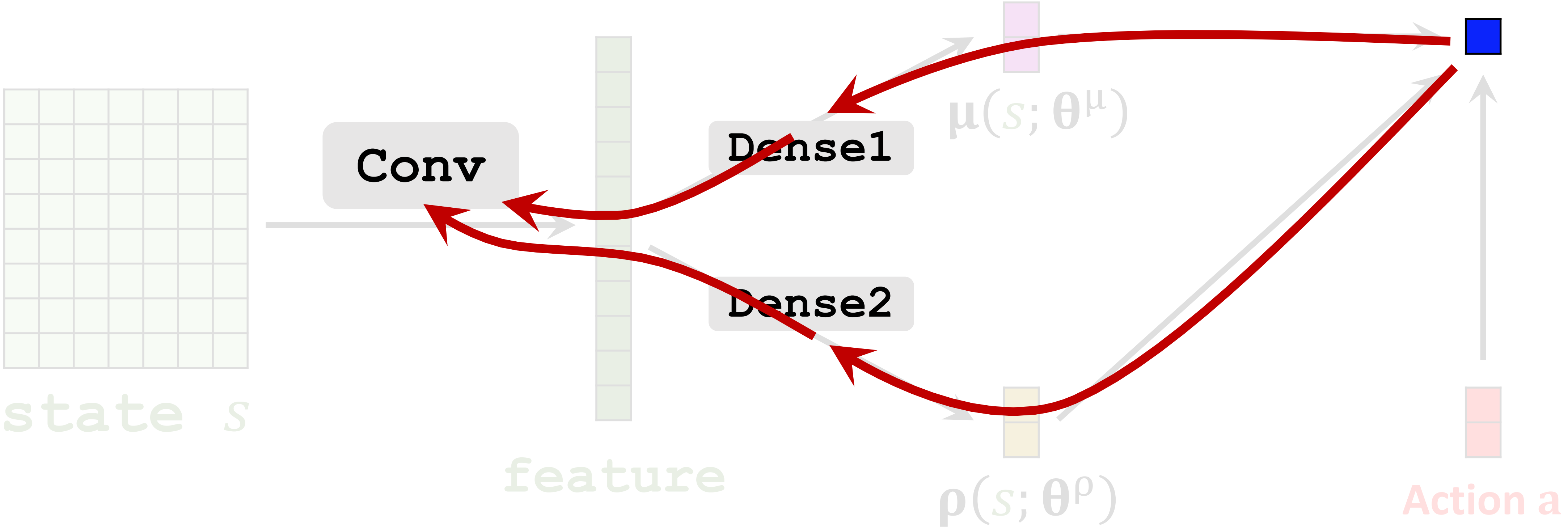
Auxiliary Network

Auxiliary Network: $f(s, \mathbf{a}; \boldsymbol{\theta}) = \sum_{i=1}^{d} \left[-\frac{\rho_i}{2} - \frac{(a_i - \mu_i)^2}{2 \cdot \exp(\rho_i)} \right].$



Auxiliary Network

The gradient, $\frac{\partial f}{\partial \boldsymbol{\theta}}$, can be automatically computed.



Recap

We have built three neural networks:

$$\underline{\boldsymbol{\mu}(s;\boldsymbol{\theta}^{\mu}),\quad \boldsymbol{\rho}(s;\boldsymbol{\theta}^{\rho})},\quad \text{and} \quad \underline{f(s,\mathbf{a};\boldsymbol{\theta})}.$$

$$\boldsymbol{\theta} = (\boldsymbol{\theta}^{\mu}, \boldsymbol{\theta}^{\rho})$$

Recap

We have built three neural networks:

$$\boldsymbol{\mu}(s;\boldsymbol{\theta}^{\mu}),\quad \boldsymbol{\rho}(s;\boldsymbol{\theta}^{\rho}),\quad \text{and}\quad f(s,\mathbf{a};\boldsymbol{\theta}).$$

- $\boldsymbol{\mu}(s;\boldsymbol{\theta}^{\mu})$ computes the mean.
- $\boldsymbol{\rho}(s;\boldsymbol{\theta}^{\rho})$ computes the log variance.

for controlling the agent

Recap

We have built three neural networks:

$$\boldsymbol{\mu}(s; \boldsymbol{\theta}^{\mu}), \quad \boldsymbol{\rho}(s; \boldsymbol{\theta}^{\rho}), \quad \text{and} \quad f(s, \mathbf{a}; \boldsymbol{\theta}).$$

- Auxiliary network, $f(s, \mathbf{a}; \boldsymbol{\theta})$, helps with the training.
- We will use $\frac{\partial f}{\partial \boldsymbol{\theta}}$ for computing policy gradient.

Training (2/4): Policy Gradient

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.

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

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.

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

Definition: State-value function.

- $V_\pi(s) = \mathbb{E}_A\left[Q_\pi(s, A)\right]$.
- The expectation is taken w.r.t. $A \sim \pi(\cdot\ |s; \boldsymbol{\theta})$.

Policy Gradient

Policy gradient: $$\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \boxed{\mathbb{E}_A} \left[\boxed{\frac{\partial \ln \pi(A \mid s; \boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} \cdot \boxed{Q_\pi(s, A)} \right].$$

Policy Gradient

Policy gradient: $\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_A\left[\frac{\partial\, \boxed{\ln \pi(A \mid s;\boldsymbol{\theta})}}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, A)\right].$

- Recall that $f(s, a; \boldsymbol{\theta}) = \underline{\ln \pi(a|s; \boldsymbol{\theta}) + \text{const.}}$

Policy Gradient

Policy gradient: $\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_A\left[\frac{\partial\, \boxed{\ln \pi(A \mid s; \boldsymbol{\theta})}}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, A)\right].$

- Recall that $f(s, a; \boldsymbol{\theta}) = \ln \pi(a|s; \boldsymbol{\theta}) + \text{const}.$
- Thus the policy gradient is equal to:

$$\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_A \underline{\left[\frac{\partial\, \boxed{f(s, A; \boldsymbol{\theta})}}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s, A)\right]}.$$

Policy Gradient

Policy gradient: $$\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_A\left[\boxed{\frac{\partial\, f(s,A;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,A)}\right].$$

- Given s and $\mathbf{a}$, we can differentiate the auxiliary network f to obtain $\frac{\partial\, f(s,\mathbf{a};\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}$.

Training (3/4): Algorithms

Monte Carlo Approximation

Policy gradient: $$\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \boxed{\mathbb{E}_A}\left[\frac{\partial\, f(s,A;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,A)\right].$$

- Randomly sample action **a** by:

$$a_i \sim N\left(\underline{\hat{\mu}_i, \hat{\sigma}_i^2}\right), \quad \text{for all } i = 1, \cdots, d.$$

Monte Carlo Approximation

Policy gradient: $$\frac{\partial\, V_\pi(s)}{\partial\, \boldsymbol{\theta}} = \mathbb{E}_A\left[\boxed{\frac{\partial\, f(s,A;\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,A)}\right].$$

- Randomly sample action **a** by:

$$a_i \sim N\left(\hat{\mu}_i, \hat{\sigma}_i^2\right), \quad \text{for all } i = 1, \cdots, d.$$

- Stochastic policy gradient:

$$\boxed{\mathbf{g}(\mathbf{a}) = \frac{\partial\, f(s,\mathbf{a};\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot Q_\pi(s,\mathbf{a}).}$$

Approximations to Action-Value

Stochastic policy gradient: $\mathbf{g}(\mathbf{a}) = \boxed{\frac{\partial\, f(s,\mathbf{a};\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}}} \cdot \boxed{Q_\pi(s,\mathbf{a})}.$

Approximations to Action-Value

Stochastic policy gradient: $\mathbf{g}(\mathbf{a}) = \frac{\partial\, f(s,\mathbf{a};\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \boxed{Q_\pi(s, \mathbf{a})}.$

- **Actor-critic** approximates Q_π by the value network, $q(s, \mathbf{a}; \mathbf{w})$.
- Update policy network by: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \boxed{\frac{\partial\, f(s,\mathbf{a};\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot q(s, \mathbf{a}; \mathbf{w})}.$
- Update value network by TD learning.

Approximations to Action-Value

Stochastic policy gradient: $\mathbf{g}(\mathbf{a}) = \frac{\partial\, f(s,\mathbf{a};\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot \boxed{Q_\pi(s, \mathbf{a})}.$

- **REINFORCE** approximates $Q_\pi(s_t, \mathbf{a}_t)$ by the observed return:

$$\boxed{u_t = r_t + \gamma \cdot r_{t+1} + \gamma^2 \cdot r_{t+2} + \gamma^3 \cdot r_{t+3} + \cdots}$$

- Update policy network by: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \beta \cdot \boxed{\frac{\partial\, f(s,\mathbf{a};\boldsymbol{\theta})}{\partial\, \boldsymbol{\theta}} \cdot u_t}.$

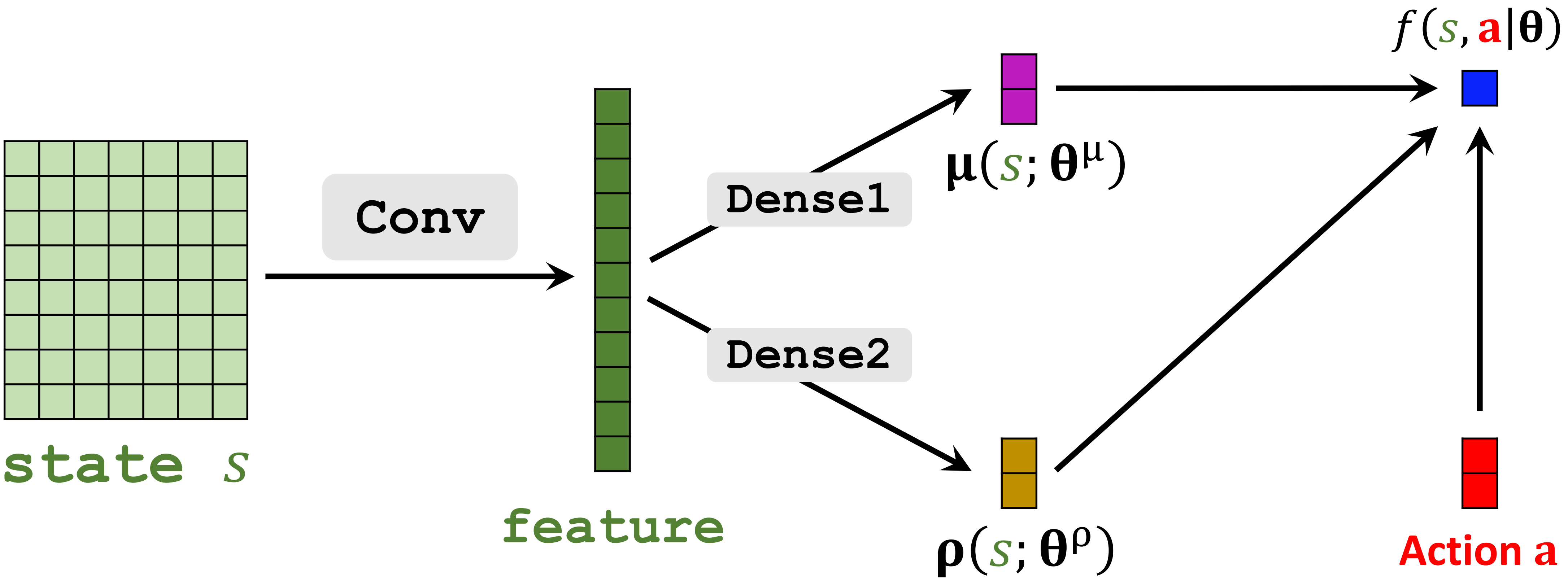
Summary

Continuous Control

- The number of actions is infinite.
- Approaches to continuous control:

  1. Discretize the action space and use standard DQN or policy network.
  2. Deterministic policy network (previous lecture).
  3. Stochastic policy network (this lecture).

Network Structure



Training

- Build auxiliary network, $f(s, \mathbf{a}; \boldsymbol{\theta})$, for computing policy gradient.
- Policy gradient algorithms: actor-critic and REINFORCE.

Training

- Build auxiliary network, $f(s, \mathbf{a}; \boldsymbol{\theta})$, for computing policy gradient.
- Policy gradient algorithms: actor-critic and REINFORCE.
- Improvement: Policy gradient with baseline.
  - Actor-critic ==> A2C.
  - REINFORCE ==> REINFORCE with baseline.

Thank you!