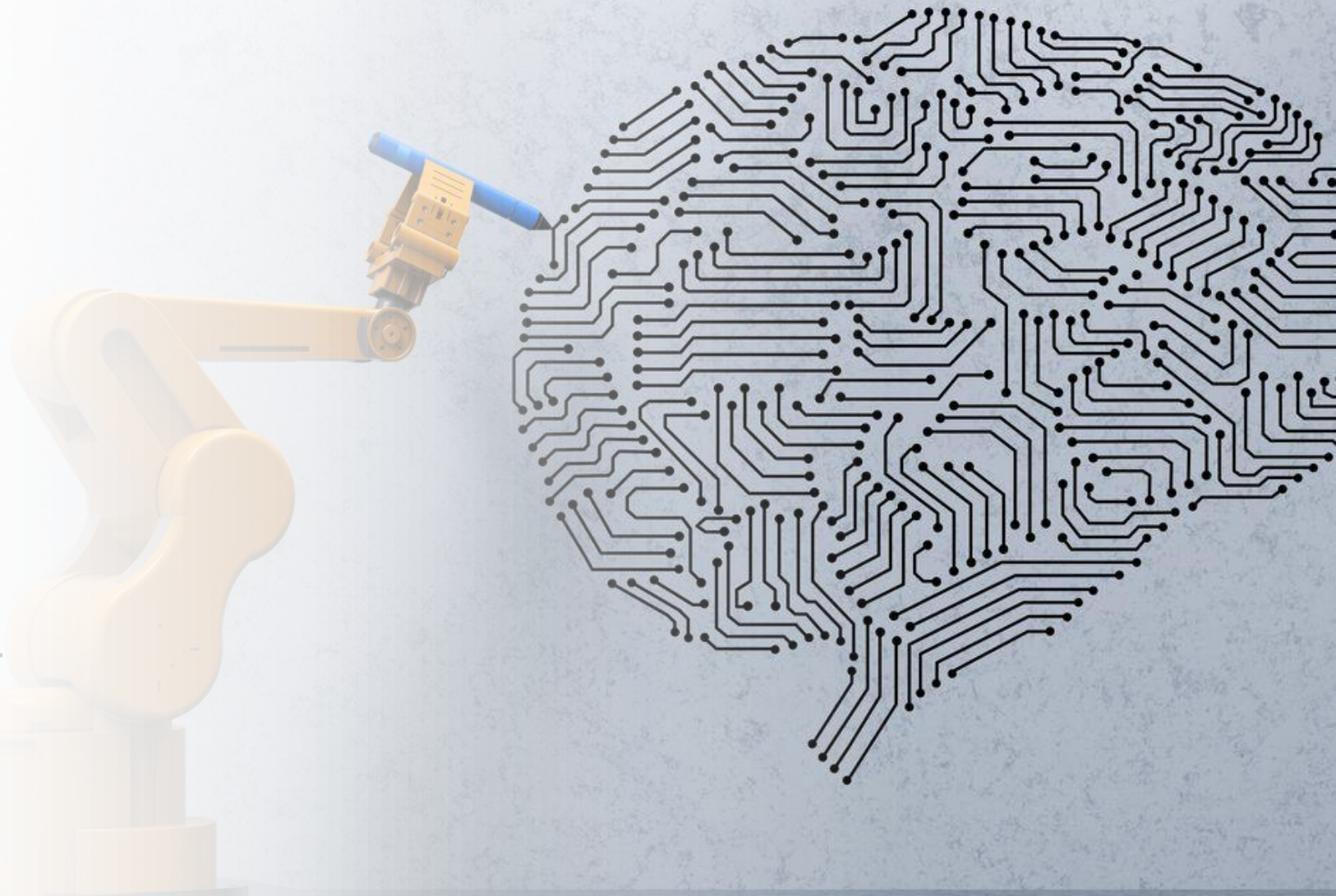
Aprendizaje por refuerzo

Clase 8: gradiente de política



Para el día de hoy…

- Gradiente de política
- REINFORCE
- Actor-critico

Aprendizaje por refuerzo basado en política

- En las ultimas clases, hemos aproximado la función valor o de acción utilizando los parámetros w

$$v_w(s) \approx v^\pi(s)$$
$$q_w(s,a) \approx q^\pi(s,a)$$

- A partir de esto, generamos la política siguiendo una política (ϵ)-voraz
- Ahora, parametrizaremos directamente la política

$$\pi_\theta(s,a) = \mathbb{P}[a|s,\theta]$$

RL basado en valor y en política

Basado en valor

Aprende la función de valor

Política implícita

Basado en política

No existe función de valor

Aprender la politica

Ventas de RL basado en política

Ventajas

Mejor convergencia

Efectivo en alta dimensionalidad y/o espacios continuos

Puede aprender políticas estocásticas

Desventajas

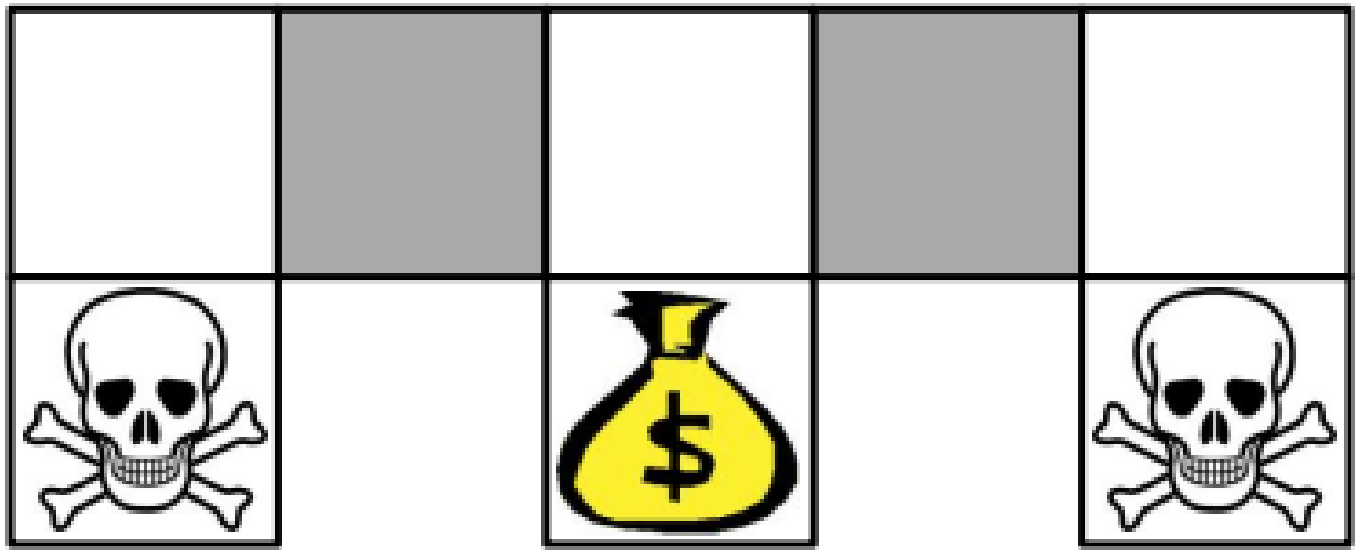
Típicamente converge a un óptimo local

Evaluar una política puede ser ineficiente y tener alta varianza

¿Políticas estocásticas?

- En MDPs siempre existe una política determinista óptima
- Pero… la mayoría de los problemas no son completamente observables
  - Especialmente común con funciones de aproximación
  - La política entonces puede ser estocástica
- El espacio de búsqueda es más suave con políticas estocásticas
- Provee exploración durante el aprendizaje

Un ejemplo



- El agente no puede diferenciar los estados grises
- Consideremos las características ($A = \{N, E, S, W\}$)
- $\phi(s, a) = 1(wall\ to\ N, a = move\ E)$
- Comparar RL basada en valor, usando una aproximación a la función valor $Q_w(s, a) = f(\phi(s, a), w)$
- A RL basado en política usando $\pi_\theta(s, a) = g(\phi(s, a), \theta)$

Funciones objetivo

- Objetivo: dada una política $\pi_\theta(s, a)$ con parámetros θ, encontrar la mejor θ
- Pero… ¿Cómo medimos la calidad de la política π_θ?
- En ambientes de episodios, podemos usar el retorno promedio por episodio
- En ambientes continuos, podemos usar la recompensa promedio por paso

Función objetivo: episodios

- $J_G(\theta) = \mathbb{E}_{s_0 \sim d_0, \pi_\theta}[\sum_{t=0}^{\infty} \gamma^t R_{t+1}]$
  - $= \mathbb{E}_{s_0 \sim d_0, \pi_\theta}[G_0]$
  - $= \mathbb{E}_{s_0 \sim d_0}[\mathbb{E}_{\pi_\theta}[G_t | S_t = S_0]]$
  - $= \mathbb{E}_{s_0 \sim d_0}[v_{\pi_\theta}(S_0)]$
- Donde d_0 es la distribución del estado inicial

Función objetivo: recompensa promedio

- $J_R(\theta) = \mathbb{E}_{\pi_\theta}[R_{t+1}]$
- $= \mathbb{E}_{S_t \sim d_{\pi_\theta}}[\mathbb{E}_{A_t \sim \pi_\theta(S_t)}[R_{t+1}|S_t]]$
- $= \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s,a) \sum_r p(r|s,a) r$
- Donde $d_\pi = p(S_t = s|\pi)$ es la probabilidad de estar en el estado s en el largo plazo

Optimización de política

- RL basado en política es un problema de optimización
- Encontrar θ que maximice $J(\theta)$
- Algunos enfoques no usan gradiente
  - Hill climber
  - Simplex/Nelder Mead
  - Algoritmos evolutivos
- Otros usan gradiente
  - Gradiente descendente
  - Gradiente conjugado
  - Regiones de confianza
  - Quasi-Newton

Pero…

- ¿Cómo calculamos el gradiente $\nabla_\theta J(\theta)$
- Supongamos que la política es diferenciable casi en todos el dominio
- Para la recompensa promedio

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\pi_\theta}[R]$$

- ¿Cómo hacemos que $\mathbb{E}[R]$ dependa de θ?

MDP de un estado

- Consideremos un MDP de un estado
  - El estado inicial $s \sim d(s)$
  - $J(\theta) = \mathbb{E}_{\pi_\theta}[R(S, A)]$
- No podemos muestrear R_{t+1} y tomar el gradiente
- Entonces debemos hacer un poco de manipulación

Un poco de manipulación

- Sea $r_{sa} = \mathbb{E}[R(S,A)|S = s, A = a]$
- $\nabla_\theta \mathbb{E}_{\pi_\theta}[R(S,A)] = \nabla_\theta \sum_s d(s) \sum_a \pi_\theta(a|s) r_{sa}$
  - $= \sum_s d(s) \sum_a r_{sa} \nabla_\theta \pi_\theta(a|s)$
  - $= \sum_s d(s) \sum_a r_{sa} \pi_\theta(a|s) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}$
  - $= \sum_s d(s) \sum_a \pi_\theta(a|s) r_{sa} \nabla_\theta \log \pi_\theta(a|s)$
  - $= \mathbb{E}_{d,\pi_\theta}[R(S,A) \nabla_\theta \log \pi_\theta(a|s)]$

Entonces…

$$\nabla_\theta \mathbb{E}_{\pi_\theta}[R(S,A)] = \mathbb{E}[R(S,A)\nabla_\theta \log \pi_\theta(a|s)]$$

- Ahora podemos muestrear
- La actualización del gradiente ascendente es

$$\theta_{t+1} = \theta_t + \alpha R_{t+1} \nabla_\theta \log \pi_\theta(a|s)$$

- Esto es un algoritmo de gradiente estocástico

Teorema de gradiente de política

- El gradiente política también aplica a MDPs
- Remplaza la recompensa R con el retorno G_t o $q_\pi(s,a)$
- Teorema: para cualquier política diferenciable $\pi_\theta(s,a)$, el gradiente de política de $J(\theta) = \mathbb{E}[R|\pi]$ es

$$\nabla_\theta J(\theta) = \mathbb{E}_\pi[q_{\pi_\theta}(S_t, A_t)\nabla_\theta \log \pi_\theta(A_t|S_t)]$$

- Donde
  - $q_\pi(s,a) = \mathbb{E}_\pi[R_{t+1} - \rho + q_\pi(S_{t+1}, A_{t+1})|S_t = s, A_t = a]$
  - $\rho = \mathbb{E}_\pi[R_{t+1}]$

Gradiente de política de Monte-Carlo (REINFORCE)

- Actualizar los parámetros por medio de gradiente ascendente estocástico
- Usa el teorema de gradiente de política
- Usa el retorno v_t

$$\Delta\theta_t = \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$$

REINFORCE: Monte-Carlo Policy-Gradient Control (episodic) for π_*

Input: a differentiable policy parameterization $\pi(a|s, \boldsymbol{\theta})$
Algorithm parameter: step size $\alpha > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ (e.g., to $\mathbf{0}$)

Loop forever (for each episode):
  Generate an episode $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$, following $\pi(\cdot|\cdot, \boldsymbol{\theta})$
  Loop for each step of the episode $t = 0, 1, \ldots, T-1$:
    $G \leftarrow \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ (G_t)
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \gamma^t G \nabla \ln \pi(A_t|S_t, \boldsymbol{\theta})$

Política softmax

- Consideremos una política softmax en la preferencia de acciones $h(s,a)$
- La probabilidad de las acciones es proporcional a la exponencial de su peso

$$\pi_\theta(a|s) = \frac{e^{h(s,a)}}{\sum_b e^{h(s,b)}}$$

- El gradiente de la log probabilidad es

$$\nabla \log \pi_\theta(A_t, S_t) = \nabla_\theta h(S_t, A_t) - \sum_a \pi_\theta(a|S_t) \nabla_\theta \mathrm{h}(\mathrm{S_t}, \mathrm{a})$$

Acciones continuas

- RL basado en valor puede extenderse a espacios continuos de forma no trivial
  - ¿Cómo aproximamos $q(s, a)$?
  - ¿Cómo calculamos $\max_a q(s, a)$?
- Cuando actualizamos los parámetros de la política esto se vuelve más sencillo

Política Gaussiana

- Consideremos $\mu_\theta(s)$ y σ^2
- La política Gaussiana $A_t \sim \mathcal{N}(\mu_\theta(S_t), \sigma^2)$
- El gradiente de la log política es

$$\nabla_\theta \log \pi_\theta(s,a) = \frac{A_t - \mu_\theta(S_t)}{\sigma^2} \nabla \mu_\theta(s)$$

- Esto se puede usar con REINFORCE ;)

Reduciendo la varianza

- La aproximación de valor puede ser complicada…
- El gradiente de política de Monte-Carlo tiene alta varianza
- A esto le llamamos actor-critico

El critico

- El critico es resolver la evaluación de política
  - ¿Cuál es el valor de v_{π_θ} de la política π_θ para los parámetros actuales θ?
- Esto se puede resolver con
  - Evaluación de política de Monte-Carlo
  - $TD(0)$
  - $TD(n)$
  - $TD(\lambda)$
  - Mínimos cuadrados

El actor

- Es encontrar la política $\pi_\theta(s, a)$
  - ¿Cuáles son los parámetros θ?
- Esto se puede hacer
  - Usando métodos de gradiente de política
  - Usando métodos libres de gradiente

Actor-critico

- Ahora tendremos dos conjuntos de parámetros
  - Critico: actualizar la función de valor con los parámetros w
  - Actor: Actualizar los parámetros de la política θ en la dirección del critico
- Estos algoritmos siguen una aproximación del gradiente de la política

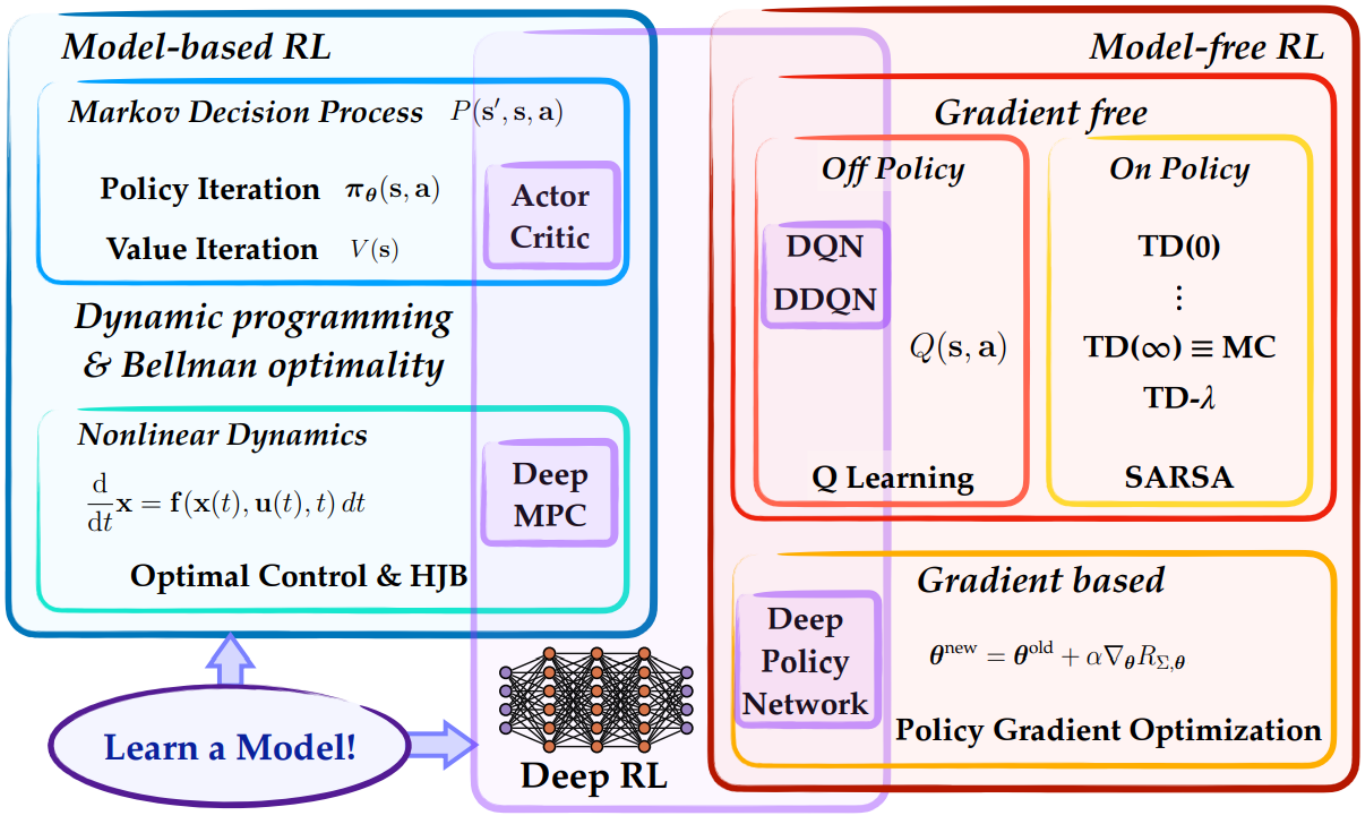
$$\nabla_\theta J(\theta) \approx \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(s,a) q_w(s,a)\right]$$
$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s,a) q_w(s,a)$$

El algoritmo simple

One-step Actor–Critic (episodic), for estimating $\pi_{\boldsymbol{\theta}} \approx \pi_*$

Input: a differentiable policy parameterization $\pi(a|s,\boldsymbol{\theta})$
Input: a differentiable state-value function parameterization $\hat{v}(s,\mathbf{w})$
Parameters: step sizes $\alpha^{\boldsymbol{\theta}} > 0$, $\alpha^{\mathbf{w}} > 0$
Initialize policy parameter $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ and state-value weights $\mathbf{w} \in \mathbb{R}^{d}$ (e.g., to $\mathbf{0}$)
Loop forever (for each episode):
  Initialize S (first state of episode)
  $I \leftarrow 1$
  Loop while S is not terminal (for each time step):
    $A \sim \pi(\cdot|S,\boldsymbol{\theta})$
    Take action A, observe S', R
    $\delta \leftarrow R + \gamma\hat{v}(S',\mathbf{w}) - \hat{v}(S,\mathbf{w})$ (if S' is terminal, then $\hat{v}(S',\mathbf{w}) \doteq 0$)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha^{\mathbf{w}}\delta\nabla\hat{v}(S,\mathbf{w})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha^{\boldsymbol{\theta}} I\delta\nabla \ln\pi(A|S,\boldsymbol{\theta})$
    $I \leftarrow \gamma I$
    $S \leftarrow S'$

Nuestro mapa…



- A partir de la experiencia ahora sabemos como
  - Aprender la política
  - Aprender la función de valor
  - Combinar ambas cosas
- Pero…
  - Aún podemos hacer algo más
  - Podemos darle a nuestros agentes el don de simular
  - ¡Y juntar todo!

Para la otra vez…

- Métodos basados en modelos