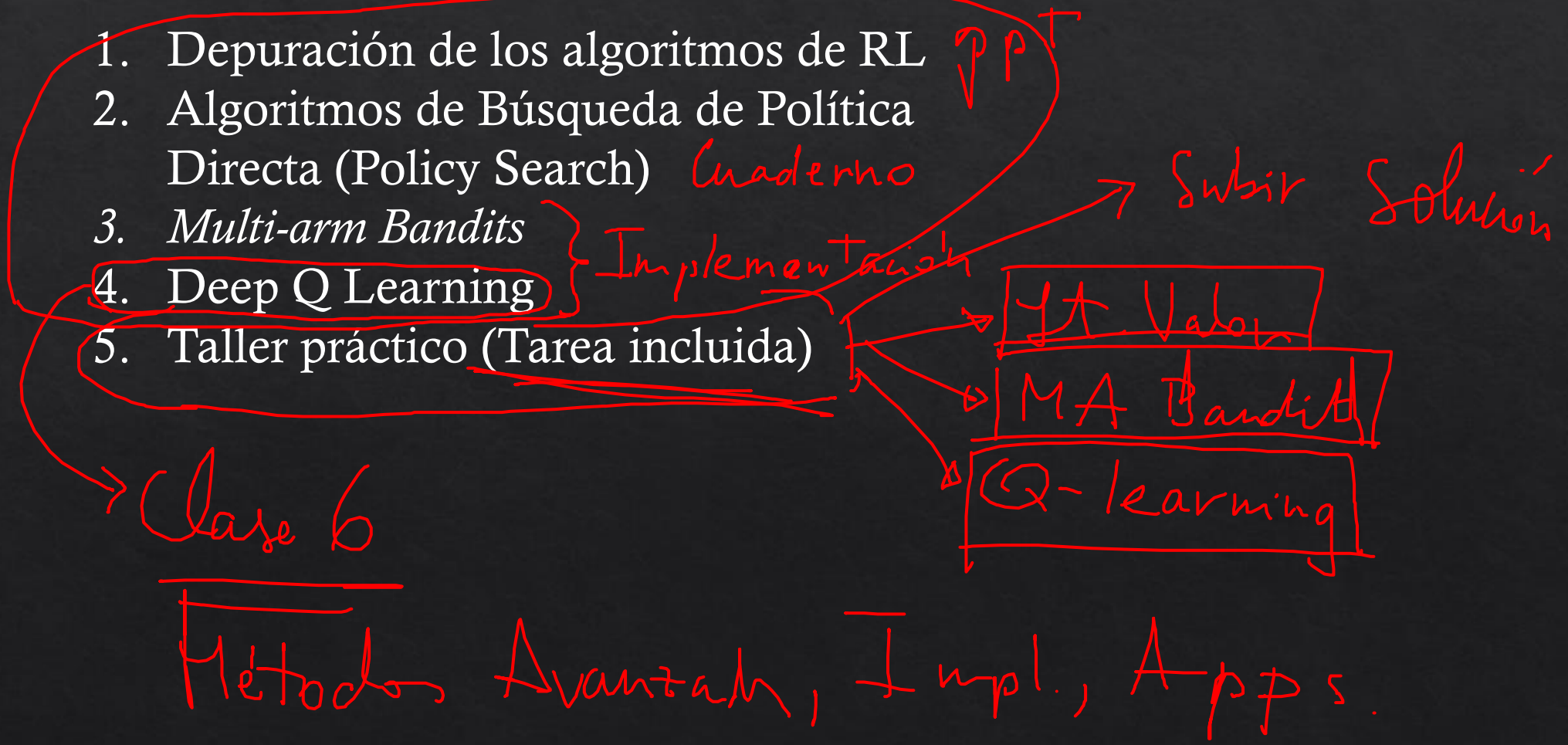
Agenda

1. Depuración de los algoritmos de RL
2. Algoritmos de Búsqueda de Política Directa (Policy Search)
3. *Multi-arm Bandits*
4. Deep Q Learning
5. Taller práctico (Tarea incluida)



Depuración de Algoritmos de RL

Clase 5.1 Jorge Vásquez

Algoritmo de ML

1. Construir un modelo/simulador de un helicóptero
2. Escoger una función de recompensas, por ej. $R(s) = -|S - S_{desired}|^2$
3. Hacer correr un algoritmo de RL para volver el helicóptero en un simulador y maximizar $E[R(So) + R(S1) + \cdots + R(St)]$ y luego obtener una política π

Algoritmo de ML

1. Supongamos que hacemos eso y el controlador resultante π entrega un rendimiento mucho peor que un piloto humano. ¿Qué hacemos?
   - ¿Mejorar el modelo o simulador?
   - ¿Modificar la función de recompensas R?
   - ¿Modificar el algoritmo de RL?

Depurar un Algoritmo de RL

- El controlador dado π rinde mal
- Asumamos :
  - El simulador del helicóptero es preciso
  - El algoritmo de RL controlar correctamente el helicóptero en el simulador y también maximiza ganancia esperada

$$\text{payoff } V^{\pi}{}_{R}(s_0) = E[R(s_0) + R(s_1) + \ldots + R(s_T) \mid \pi_{RL}, s_0].$$

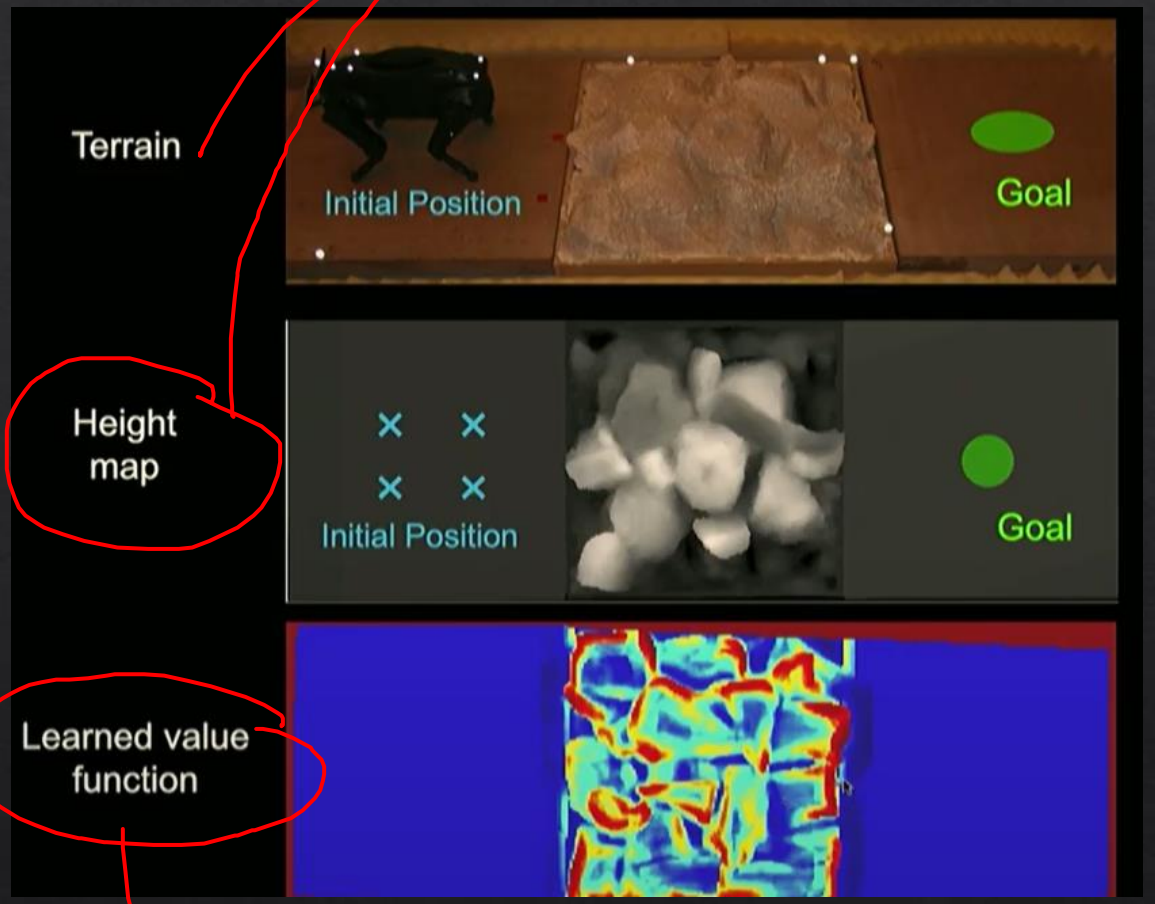
  - Maximizar ganancia esperada corresponde a un vuelo autónomo correcto.
- Luego, el controlador π debería hacer volar bien el helo

Depurar un Algoritmo de RL

- Diagnosticar:

1. Si la π_{RL} vuela bien en el simulador, pero no en el mundo real, entonces el problema es el simulador. $V^{\pi_{RL}}(s_0) < V^{\pi_{humano}}(s_0)$
2. Dejemos que π_{humano} sea la política de control y si entonces el problema es el algoritmo de RL (no maximiza la ganancia esperada)
3. De lo contrario, el problema es en la función de costo (la maximización no corresponde a un buen vuelo autónomo)

Perro Robótico



Real → Modelo

II. fn. de pérdida

Algo RL → evaluar