Introducere în Reinforcement Learning

Cursul #8

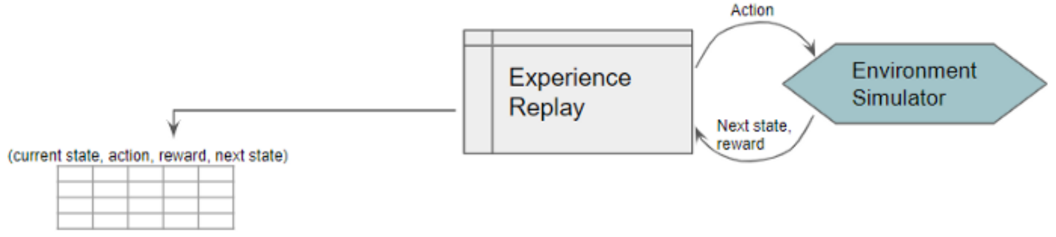
Ștefan Iordache, Cătălina Iordache, Ciprian Păduraru

Cuprins

01

DQN In-depth

DQN – Inițializarea

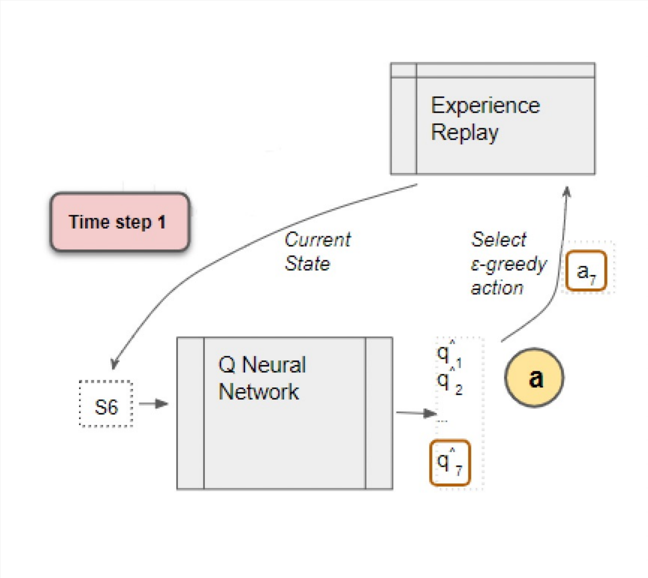


- Executăm un număr mare de pași pentru a construi setul de antrenare.

- Inițializăm Q Network cu ponderi aleatorii și le copiem către Target Network.

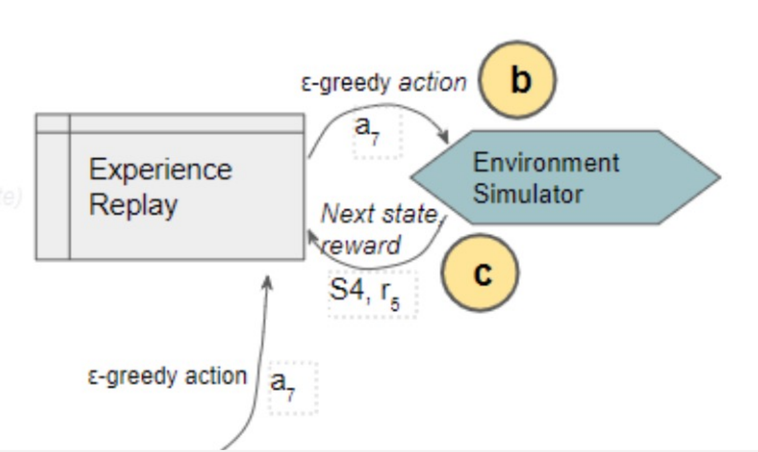
DQN – Experience Replay

- Q Network selectează o acțiune în mod **ε-**greedy, acționând precum agentul în timp ce interacționează cu mediul pentru generarea unui eșantion de antrenare.
- Nu se întâmplă partea de învățare aici!



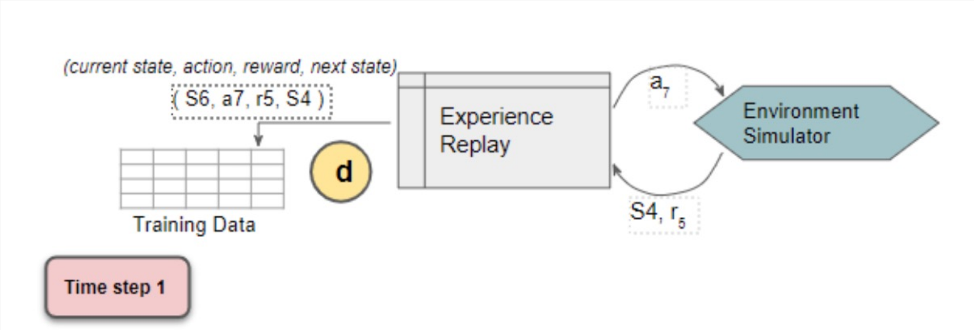
DQN – Experience Replay

- Executăm acțiunea **ε**-greedy și obținem perechea formată din (următoarea stare, reward).



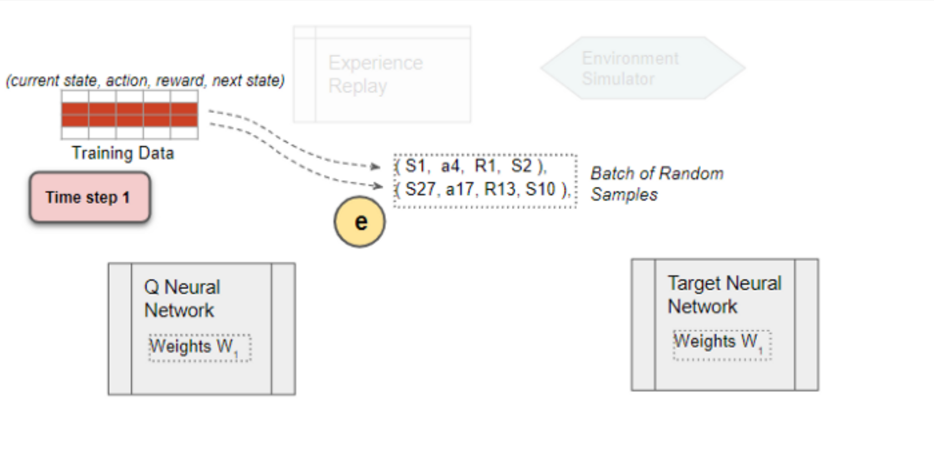
DQN – Experience Replay

- Stocăm tuplul obținut, fiind folosit mai târziu drept eșantion pentru partea de antrenare.



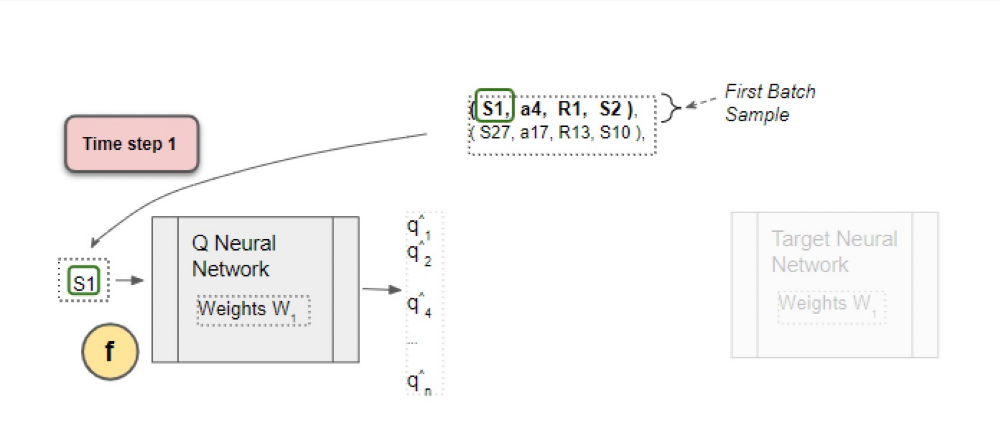
DQN – Q-Network

- Selectăm un batch aleatoriu drept input pentru cele două rețele.



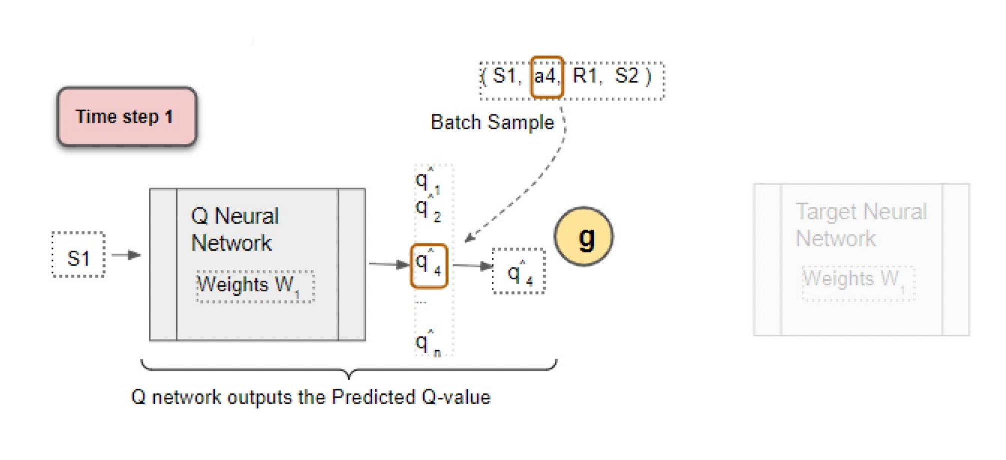
DQN – Q-Network

- Folosim starea curentă (S1, S27) din sample pentru a prezice valoarea Q pentru toate acțiunile care pot fi realizate din starea respectivă.



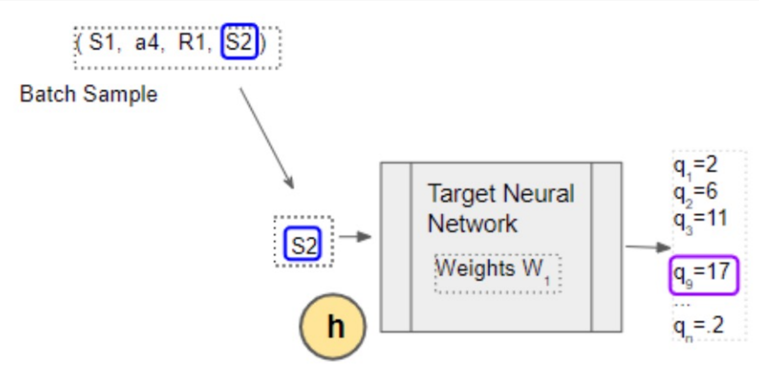
DQN – Q-Network

- Selectăm acțiunea prezisă cu ajutorul Q-value (în cazul expus a4, cu ajutorul q4).

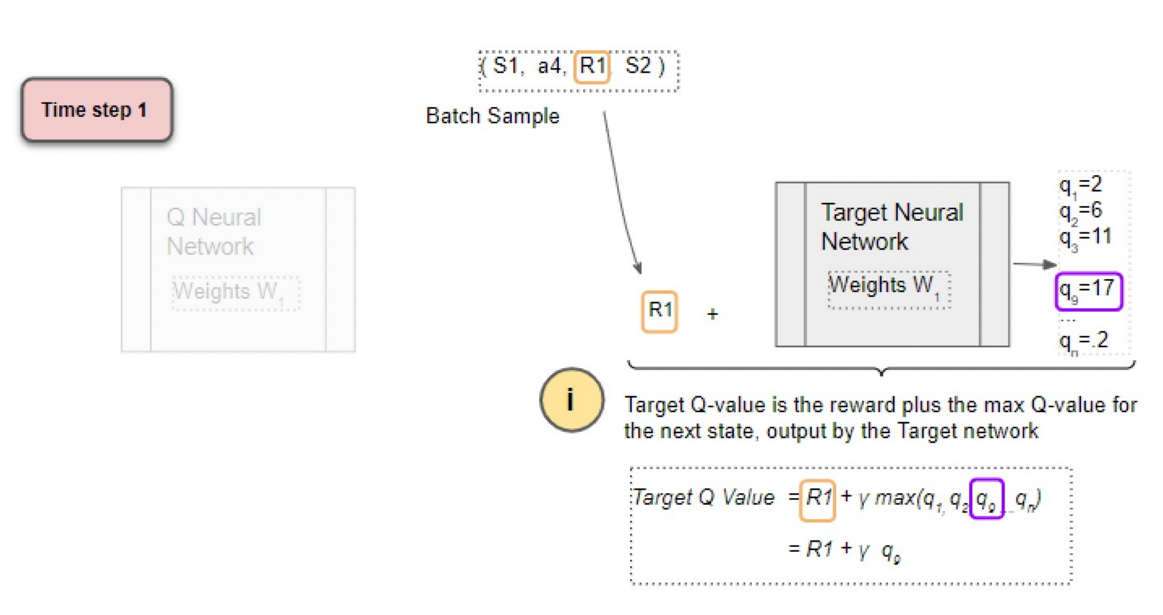


DQN – Target Network

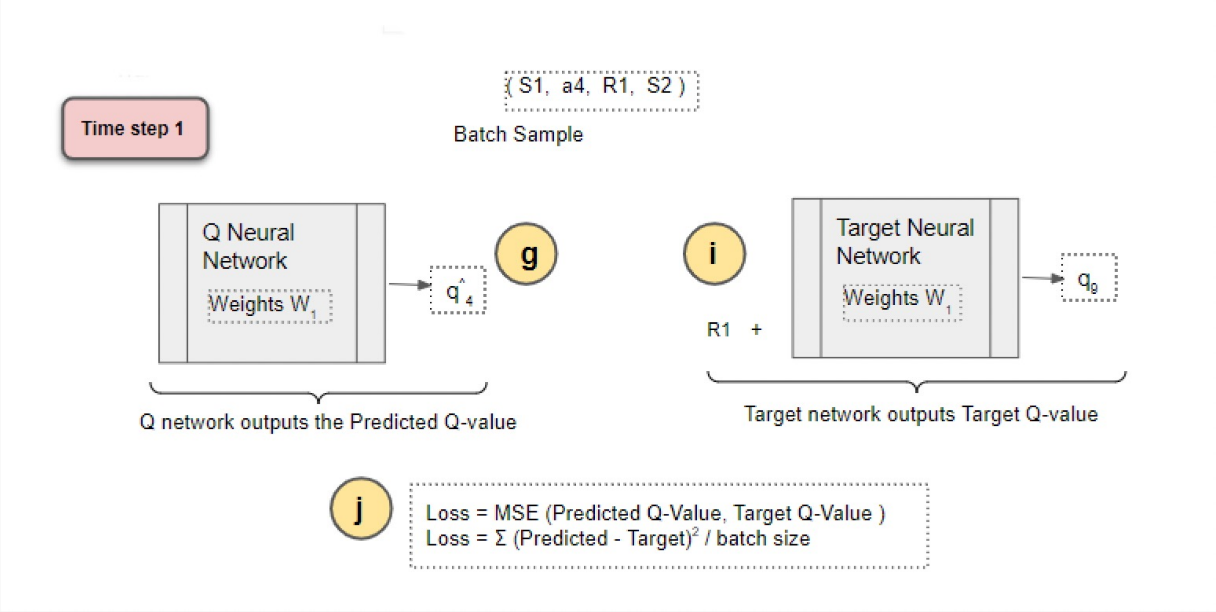
- Folosim starea următoare drept input pentru Target Network, astfel prezicând valorile Q pentru toate acțiunile care pot fi luate din starea următoare.
- Selectăm acțiunea cu Q-value maxim.



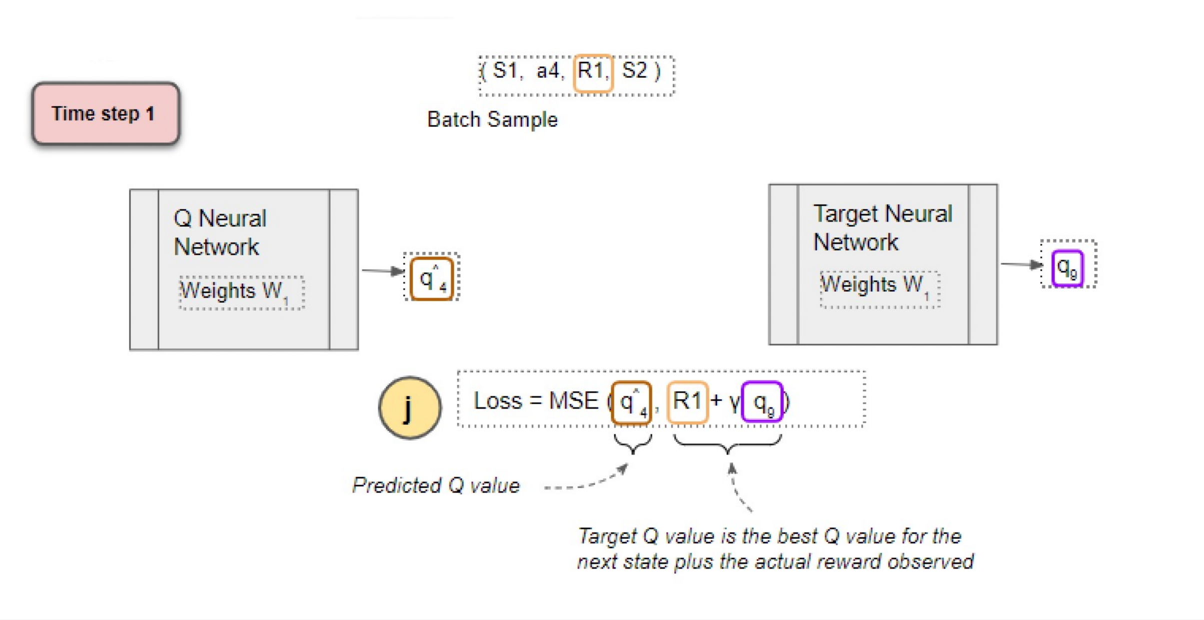
DQN – Target Q-Value



DQN – Loss


(S1, a4, R1, S2)
Batch Sample
Time step 1
Q Neural Network
Weights W_1
$\hat{q}_4$
g
i
R1 +
Target Neural Network
Weights W_1
q_9
Q network outputs the Predicted Q-value
Target network outputs Target Q-value
j
Loss = MSE (Predicted Q-Value, Target Q-Value)
Loss = Σ (Predicted - Target)^2 / batch size

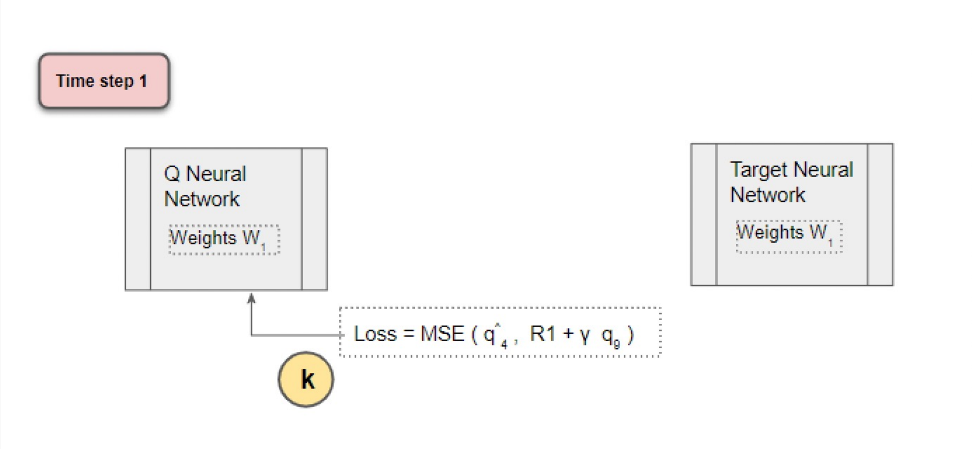
DQN – Loss



(S1, a4, R1, S2)
Batch Sample
Time step 1
Q Neural Network
Weights W1
q̂4
Target Neural Network
Weights W1
q9
j
Loss = MSE (q̂4 , R1 + γ q9)
Predicted Q value
Target Q value is the best Q value for the next state plus the actual reward observed

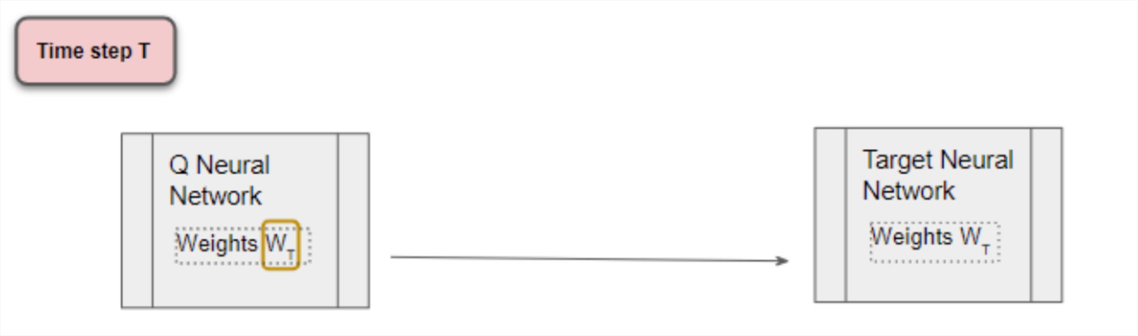
DQN – Back-Propagation – Q-Network

- Actualizăm ponderile pentru Q-Network prin procedeul denumit Back-Propagation, asistat de Gradient Descent.
- Nu sunt actualizate ponderile pentru Target Network!



DQN – Update Target Network

- Actualizarea Target Network se realizează prin copierea Q-Network după T momente de timp.

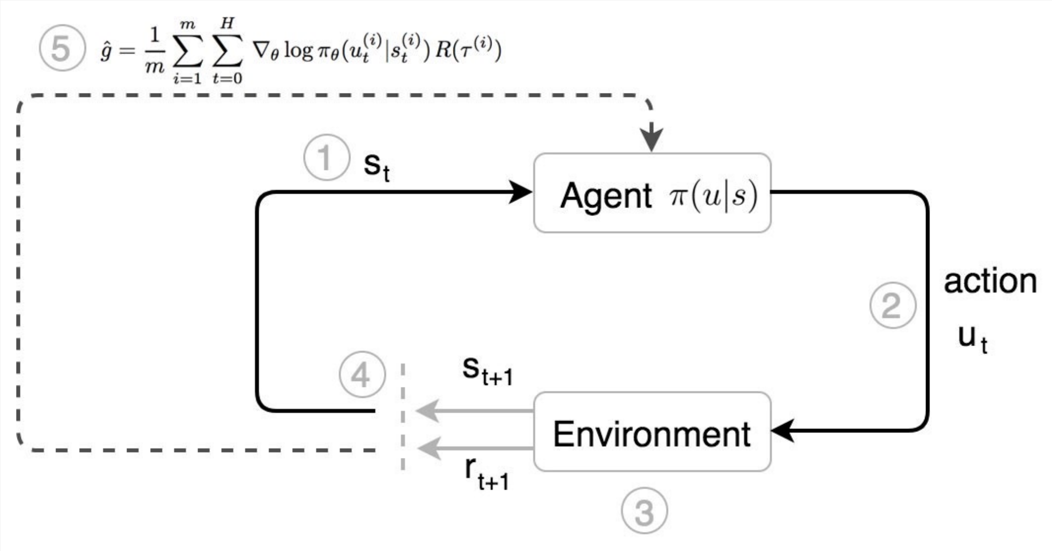


02

Policy Gradients

Cum funcționează? Ce e diferit?

- **Pasul 5 este cel mai important!**
- **Considerând traiectoria τ vom ajusta politica folosind totalul de recompense R(τ).**



$$(s_1, u_1, s_2, u_2, \ldots, s_H, u_H)$$

Policy Gradients?

Încercăm direct să maximizăm rezultatul prin pași mici în direcția gradientului.

Obiectiv!

- **Expected reward = suma probabilității unei anumite traiectorii x recompensă**

$$J(\theta) = \mathrm{E}\Big[\sum_{t=0}^{H} R(s_t, u_t); \pi_\theta\Big] = \sum_{\tau} P(\tau;\theta) R(\tau)$$

$$\max_{\theta} J(\theta) = \max_{\theta} \sum_{\tau} P(\tau;\theta) R(\tau)$$

Orizontul (H)

- **Introducem recompensele în calcul și exprimăm prin H numărul maxim de pași.**
- H poate tinde către infinit, până la o stare terminală, dar în practică nu folosim această ipoteză.

$$(s_1, u_1, r_1, s_2, u_2, r_2, \ldots, s_H, u_H, r_H)$$

Optimizări!

- **Folosim derivata parțială a unei funcții f(x)!!!**

$$f(x)\,\nabla_\theta \log f(x) = f(x)\,\frac{\nabla_\theta f(x)}{f(x)} = \nabla_\theta f(x)$$

- **Înlocuim f(x) cu politica π!!!**

$$\pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau) = \nabla_\theta \pi_\theta(\tau)$$

Multe calcule mai târziu...

- **Formula finală pentru Policy Gradient!**

$$\boxed{\begin{aligned}&\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})\right)\left(\sum_{t=1}^{T} r(\mathbf{s}_{i,t},\mathbf{a}_{i,t})\right)\\ &\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)\end{aligned}}$$

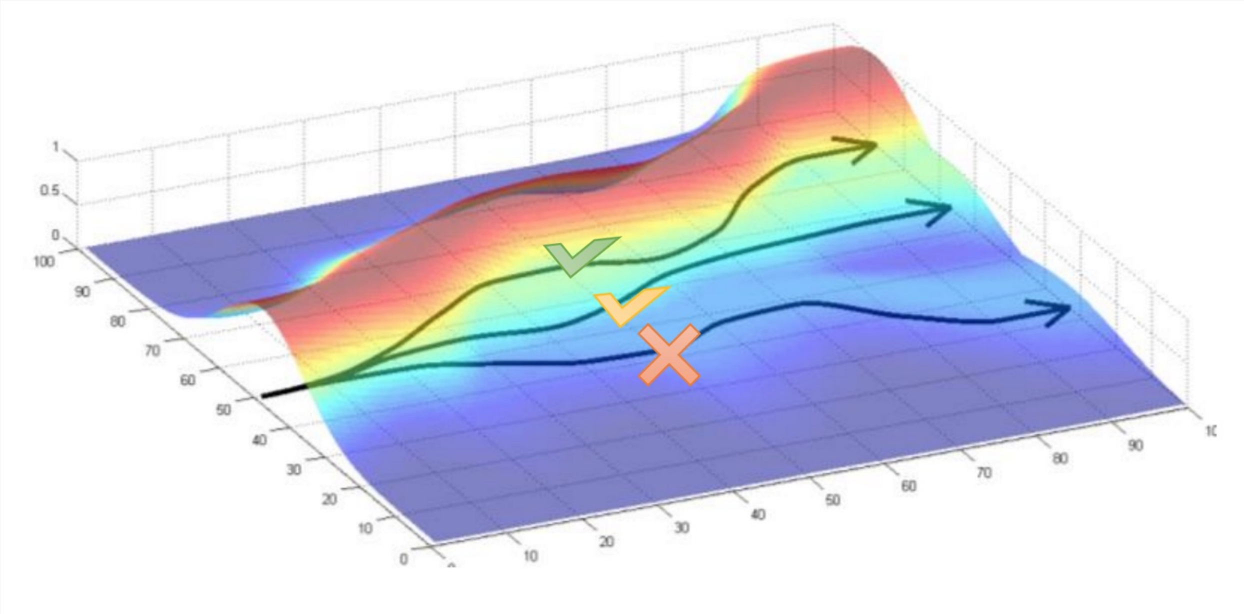
Dar...nu are logică!!!

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \underline{\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})} \right) \left(\sum_{t=1}^{T} r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \right)$$

Termenul subliniat este numit drept “maximum log likelihood”. În contextul nostru măsurăm în ce grad traiectoria aleasă se află sub politica curentă. Multiplicând cu reward-ul dorim să creștem acest grad daca traiectoria rezultă în reward-uri bune și contrar dacă scade recompensa.

Pe scurt: **Păstrăm ceea ce funcționează, aruncăm tot ce nu este bun!**

Să vizualizăm!



Un nou algoritm: REINFORCE!

REINFORCE algorithm:

1. sample $\{\tau^i\}$ from $\pi_\theta(\mathbf{a}_t|\mathbf{s}_t)$ (run the policy)
2. $\nabla_\theta J(\theta) \approx \sum_i \left(\sum_t \nabla_\theta \log \pi_\theta(\mathbf{a}_t^i|\mathbf{s}_t^i)\right)\left(\sum_t r(\mathbf{s}_t^i, \mathbf{a}_t^i)\right)$
3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Defectul Policy Gradients

- **Varianță mare!**
- **Covergență scăzută!**

- Varianța crescută provoacă o coborâre anormală pe gradient, astfel modelul fiind incapabil să învețe pe deplin și să atingă convergența.
- Cum rezolvăm???

Baseline

- **Introducem V(s), denumit și baseline.**

- **Continuăm și introducem ceva special pentru Policy Gradients:** **Advantage**

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \underbrace{\left(\sum_{t'=1}^{T} r(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'})\right)}_{Q(s,\, a)}$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \left(Q(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) - V(\mathbf{s}_{i,t})\right)$$

$$A^\pi(\mathbf{s}_t, \mathbf{a}_t) = Q^\pi(\mathbf{s}_t, \mathbf{a}_t) - V^\pi(\mathbf{s}_t)$$

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) A^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$$

Cum explicăm mai departe?

- **Exemple:**
  - Situația I: Traiectoria A primește recompensa +20 și traiectoria B primește recompensa -10. => **Creștem probabilitatea pentru A, o scădem pentru B.**
  - Situația II: Traiectoria A primește recompensa +20 și traiectoria B primește recompensa +5. => **Creștem probabilitatea atât pentru A, cât și pentru B.**

Thanks!

Este timpul pentru întrebări!!!

stefan.iordache10@s.unibuc.ro
catalina.patilea@s.unibuc.ro
ciprian.paduraru@fmi.unibuc.ro

+40 7.. ... ...