III. Limites

Règle de mise à jour des paramètres pour Policy Gradient

Cette règle fait l'hypothèse que la surface à optimiser est plate (méthode du 1er ordre) :

- α trop grand : on se déplace trop vite → risque d'effectuer des mouvements catastrophiques
- α trop petit → on risque d'apprendre trop lentement
    - si l'exploration nous emmène dans une zone plate avec politique fonctionnant mal localement, on risque d'avoir du mal à en sortir

ET très difficile de régler le learning rate sur des problèmes de RL ! Il n'est pas sensible au "terrain"

- Un changement mineur dans les paramètres peut modifier drastiquement la politique.
  ↳ limiter les déplacements de la politique pour qu'elle ne varie pas au delà d'un seuil à chaque étape
  Problèmes :
    - Comment régler le seuil ?
    - Comment transposer ce seuil dans l'espace des paramètres ?
- Forte variance
- On-Policy : même politique pour sampler et apprendre

$$\nabla_\theta J(\theta) = E_{\tau \sim \pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau) r(\tau)]$$

reward is calculated from the trajectory using the current policy

    - chaque trajectoire est utilisée une seule fois : à chaque fois que la politique est modifiée, on collecte de nouvelles trajectoires et les anciennes ne sont pas réutilisables.
      → faible efficacité d'apprentissage

Entropie

Souvent, la politique converge trop vite vers des situations sous-optimales : π tend rapidement vers une politique déterministe

→ Plus d'exploration

On peut rajouter un coût d'entropie qui permet de maintenir l'exploration tant qu'il reste de l'incertitude :

$$\Delta\theta = \alpha \sum_{t=0}^{T} [\nabla_\theta \log \pi_\theta(\mathbf{a}_t|s_t)(R_t - b_t(s_t)) + \beta \nabla_\theta H_\theta(s_t)]$$

$$H_\theta(s_t) := -\sum_{\mathbf{a}\in\mathcal{A}} \pi_\theta(\mathbf{a}|s_t) \log \pi_\theta(\mathbf{a}|s_t)$$