Démarche

Mehdi Mounsif

28 février 2018

1 Récap : The day before

- Après des essais fructueux, échec cuisant de PPO. Tester avec convolutions
- Attention PPO : Le ratio est constamment 1. Les deux politiques estiment la même chose. Trouver le problème
- Lectures sur GANs. Possibilités de générations de trajectoires. [2] à pousuivre
- Idée RL : HER - tirer des leçons des échecs (mise en abîme ?). Voir [1]
- Début de la rédaction sur HTM. Poursuivre

2 Implémentation en cours de DPPG

Basée sur la méthode de A chaque fois que j'utilise Experience Replay, l'algorithme est beaucoup plus long. J'ai testé différentes implémentations pour vérifier la vitesse lorsqu'on tire des expériences de la mémoire. Utiliser **itemgetter**. Penser à **deque**.

Dans DDPG, on montre que le gradient de la politique stochastique $\nabla_\theta \mu(a, |\theta, s)$ est équivalent au gradient de la politique déterministe obtenu par :

$$\nabla_\theta \mu(a, |\theta, s) \sim \mathbb{E}_{\mu'}[\nabla_a Q(s, a|\theta))\nabla_\theta \mu(s|\theta)]$$

D'après [3], il est intéressant d'utiliser du bruit adaptatif dans l'espace des paramètres. C'est une méthode d'exploration simple à implémenter qui n'impacte négativement la performance que très rarement. En général, on ajoute du bruit à l'espace des actions, pour altérer les probabilités associées aux actions (ou tout simplement choisir l'action que l'on désire). L'article souligne qu'agir de la sorte provoque une exploration quelque peu chaotique, puisque que l'action choisie peut, à cause du bruit, être extrêmement loin des *croyances* de l'agent. A contrario, le bruit dans les paramètres de l'agent encourage l'exploration tout en maintenant une cohérence avec la logique de l'agent (les actions changent plus progressivement). Quelques subtilités néanmoins :

- Les différentes couches peuvent avoir différentes sensibilités...
- ... Qui peuvent évoluer au cours du temps
- De plus, il est difficile de choisir intuitivement une magnitude de bruit adaptée

Les réponses suivantes on été proposées :

- Pour la sensibilité, on utilise une couche de normalisation pour s'assurer que la sortie de la couche $i-1$ est dans la même dimension que l'entrée de la couche i
- Pour les deux remarques suivantes, on utilise un schéma d'adaptation.

La magnitude est adapté à chaque itération de la manière suivante :

Concrètement, on détermine des politiques $\tilde{\pi} = \pi_{\tilde{\theta}}$ où à partir des paramètres $\tilde{\theta} = \theta + \mathcal{N}(0, \sigma^2 I)$

3 GANs : Suite

Rappel de la fonction de coût de la discriminatrice :

$$J_D(\theta_D, \theta_G) = -\frac{1}{2}\mathbb{E}_{x\sim p_data} \log D(x) - \frac{1}{2}\mathbb{E}_z \log(1 - D(G(z)))$$

L'entrainement de la fonction discriminatrice permet à terme d'obtenir le ratio :

$$\frac{p_{data}(x)}{p_{model}(x)} \quad \forall x$$

Cette estimation est à la base du calcul des divergences et de leurs gradients. Pour la génératrice, considérer les GANs comme un jeu d'opposition à somme nulle pointe naturellement vers la fonction de coût opposée :

$$J_D = -J_G$$

En théorie, on peut alors résumer l'ensemble par une **value fonction** minimax :

$$V(\theta_D, \theta_G) = -J_D(\theta_D, \theta_G)$$

Cependant, en pratique, ceci est peu utile. On préfèrera une fonction de coût heuristique :

$$J_G = -\frac{1}{2}\mathbb{E}_z \log D(G(Z))$$

Ou correspondant au maximum de la vraisemblance, où on vise à minimiser la différence (KL divergence) entre la distribution du training set et celle du modèle. :

$$J_G = -\frac{1}{2}\mathbb{E}_z \exp(\sigma^{-1}(D(G(Z))))$$

Où σ est la fonction sigmoide.

3.1 Comparaison des fonctions de coût

Lorsqu'on chercher à minimiser l'écart entre les distributions de probabilité, notamment avec KL divergence, il faut noter que cette divergence n'est pas symétrique. Ainsi, $D_{KL}(p_{model}|p_{data}) \neq D_{KL}(p_{data}|p_{model})$,dites respectivement **Jensen-Shannon divergence** et **maximum likelihood divergence**.

3.2 Architecture classique : DCGAN

Penser à utiliser BatchNorm.

Références

[1] Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. Hindsight Experience Replay. *ArXiv e-prints* (jul 2017).

[2] Goodfellow, I. NIPS 2016 Tutorial : Generative Adversarial Networks. *ArXiv e-prints* (Dec. 2017).

[3] Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. Parameter Space Noise for Exploration. *ArXiv e-prints* (June 2017).

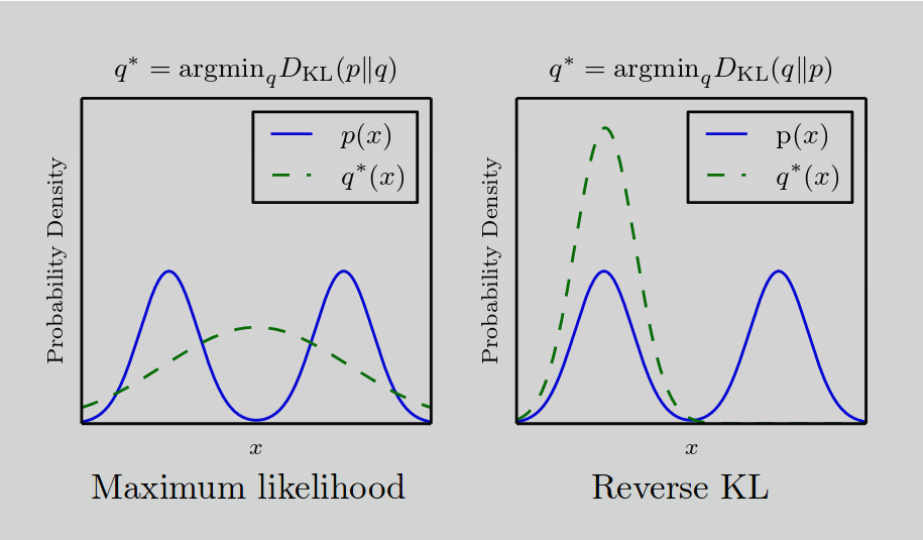

FIGURE 1 – Exemples d'approximation avec critère maximum likelihood (gauche) et Jensen-Shannon (droite)

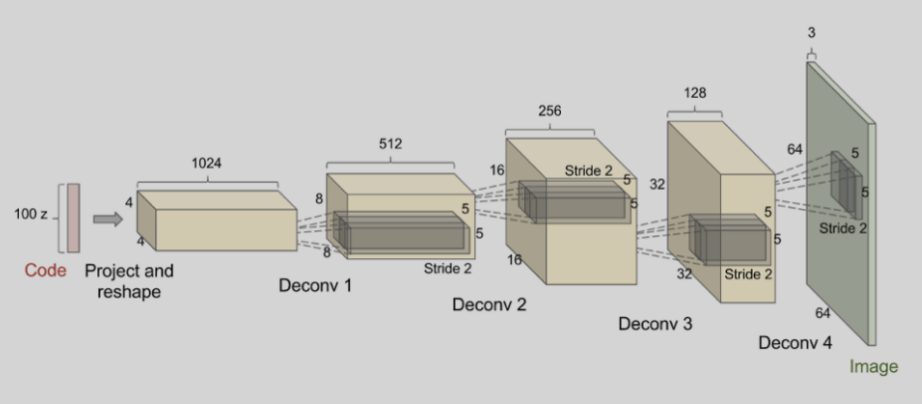

FIGURE 2 – Architecture du générateur DCGAN