

## **Master 1 - ANDROIDE**

# **Carnet de bord : les coulisses de la recherche documentaire**

*Apprentissage par renforcement profond : analyse  
détaillée de SVPG*

<b>1. Introduction</b>	<b>2</b>
<b>2. Mots clés retenus</b>	<b>2</b>
<b>3. Recherche documentaire</b>	<b>3</b>
3.1. Cheminement de recherche documentaire:	3
3.2. Description des outils de recherche :	3
<b>4. Bibliographie</b>	<b>4</b>
<b>5. Évaluation des sources</b>	<b>5</b>

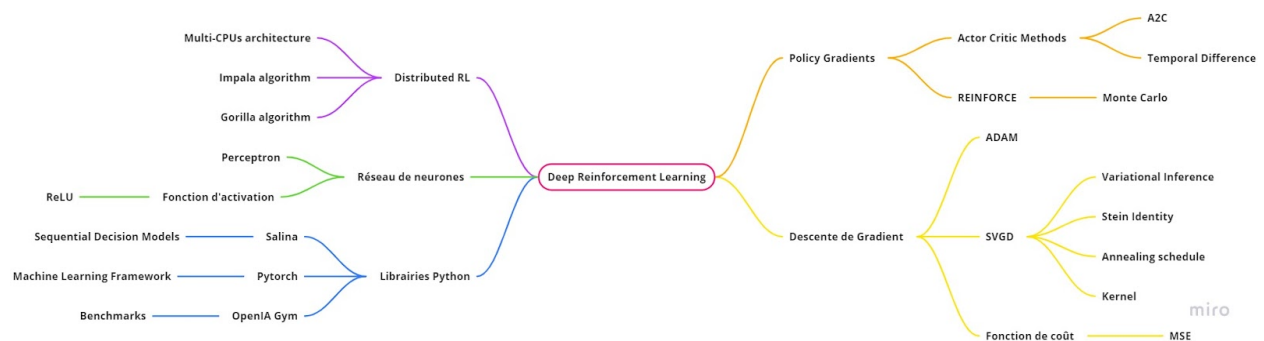
# 1. Introduction

Stein Variational Policy Gradient [9] (ou SVPG) est une méthode d'apprentissage par renforcement (reinforcement learning ou RL) [14] qui permet l'apprentissage et l'exploitation de plusieurs politiques dans une architecture multi-CPU. Plusieurs agents ("workers") travaillent en parallèle, ce qui accélère l'exploration. Le but de SVPG est d'empêcher ces travailleurs d'apprendre la même solution en les éloignant les uns des autres, ce qui permet une plus grande diversité d'exploration.

Ainsi, le but de notre projet est de moderniser cet algorithme en utilisant des outils plus modernes tels que PyTorch [12], SaLiNa [4] et OpenIA Gym [2], et de le comparer à d'autres (Impala [5], Gorilla [13]...) afin mettre en avant les cas d'utilisation pertinents.

## 2. Mots clés retenus

Deep Reinforcement Learning, Policy Gradients, Actor Critic Methods, A2C, Temporal Difference, REINFORCE, Monte Carlo, Descente de Gradient, ADAM, SVGD, Variational Inference, Stein Identity, Annealing Schedule, Kernel, Fonction de coût, MSE, Distributed RL, Multi-CPU architecture, Impala algorithm, Gorilla algorithm, Réseau de neurones, Perceptron, Fonction d'activation, ReLU, Sequential Decision Models, Salina, Machine Learning Framework, Pytorch, OpenIA Gym, Benchmarks.



Carte heuristique des mots clés retenus (<https://miro.com/>).

## 3. Recherche documentaire

### 3.1. Cheminement de recherche documentaire:

Dans un premier temps notre encadrant nous a fait un cours d'introduction sur les différents points qui risqueraient d'être abordés durant le projet. Il nous a aussi présenté l'algorithme à implémenter (SVPG) et nous a conseillé de lire l'article correspondant [9], ainsi que certains papiers associés [7,10].

A partir de là, pour chaque nouvel algorithme ou concept mathématique, nous cherchons premièrement sur **Google** ce qui nous amène régulièrement à **Wikipédia**, ou d'autres catalogues plus ou moins spécialisés dans le domaine comme **Towards Data Science** (*sous-section de MEDIUM*). Cela nous offre une vue générale du concept, permettant de nous familiariser avec les différents termes. Ces sources ont l'avantage d'être accessibles sans avoir besoin de beaucoup de compétences dans le domaine.

Pour affiner notre compréhension, nous cherchons des articles scientifiques associés à ces concepts sur **Arxiv**, **Google Scholar** et le **Catalogue en ligne de SU**. Nous considérons un article pertinent si : il a été publié récemment, son auteur est le créateur de l'algorithme (ou de la librairie), son nombre de citations est élevé, notre encadrant nous l'a recommandé, il traite bien du sujet concerné...

Ces papiers peuvent être très riches en information et utiliser des concepts complexes que nous n'avons jamais manipulés. Pour les comprendre nous sommes obligés de repartir sur des sources moins formelles où l'information sera moins condensée et les exemples plus nombreux. Nous pouvons alors revenir aux étapes précédentes.

Enfin, nous cherchons les implémentations des algorithmes sur **Paper With Code** et **Github**. Elles peuvent être des références qui nous permettent de continuer nos travaux, ou du moins d'éclairer la voie sur comment initier nos implémentations.

### 3.2. Description des outils de recherche :

- **Google** : outil très rapide, qui nous permet d'accéder rapidement aux différentes ressources.
- **Wikipédia** : ressources sur les algorithmes et concepts mathématiques connus. Pas forcément très spécialisé mais permet de donner une image d'ensemble du concept. De plus s'il y a un terme avec lequel nous ne sommes pas familier il suffit de cliquer sur les liens pour savoir à quoi ce terme fait référence, la recherche par rebond est particulièrement facilitée. Cite les sources de première qualité.
- **Towards Data Science (MEDIUM)** : des spécialistes des datas sciences nous expliquent un concept dans un article court en multipliant les exemples. On peut très facilement faire de la recherche par rebond et donc comprendre tout ce qui est nécessaire pour la compréhension de cet article.
- **Arxiv**: une archive ouverte de publications d'articles scientifiques hébergée par l'Université de Cornell, qui donne l'accès gratuitement et simplement aux prépublications des articles. Très utilisée dans le monde de la recherche.
- **Google Scholar** : permet d'avoir accès rapidement aux articles scientifiques.
- **Catalogue en ligne de SU** : outil simple, sûr, complet et rapide. Permet de rechercher des articles, des livres avec leur référence dans Zotero.
- **Paper With Code**: permet de trouver les articles scientifiques (notamment en Machine Learning et en Deep Learning) avec le code source associé. Ce dernier peut être écrit par les auteurs eux-mêmes ou par la communauté.
- **Github**: permet de trouver les dépôts et les projets intéressants qui peuvent être utiles pour notre projet.

## 4. Bibliographie

[1,3,6–8,10,11,15,16]

- [1] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. 2017. Variational Inference: A Review for Statisticians. *J. Am. Stat. Assoc.* 112, 518 (April 2017), 859–877. DOI:<https://doi.org/10.1080/01621459.2017.1285773>
- [2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *ArXiv160601540 Cs* (June 2016). Retrieved April 3, 2022 from <http://arxiv.org/abs/1606.01540>
- [3] Francesco D’Angelo and Vincent Fortuin. 2021. Annealed Stein Variational Gradient Descent. *ArXiv210109815 Cs* (March 2021). Retrieved April 3, 2022 from <http://arxiv.org/abs/2101.09815>
- [4] Ludovic Denoyer, Alfredo de la Fuente, Song Duong, Jean-Baptiste Gaya, Pierre-Alexandre Kamienny, and Daniel H. Thompson. 2021. SaLinA: Sequential Learning of Agents. (October 2021). Retrieved February 26, 2022 from <http://arxiv.org/abs/2110.07910>
- [5] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. 2018. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. *ArXiv180201561 Cs* (June 2018). Retrieved April 3, 2022 from <http://arxiv.org/abs/1802.01561>
- [6] Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. *ArXiv14126980 Cs* (January 2017). Retrieved April 3, 2022 from <http://arxiv.org/abs/1412.6980>
- [7] Qiang Liu and Dilin Wang. 2019. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. *ArXiv160804471 Cs Stat* (September 2019). Retrieved April 3, 2022 from <http://arxiv.org/abs/1608.04471>
- [8] Yang Liu. 2021. *svpg\_REINFORCE*. Retrieved April 3, 2022 from [https://github.com/largelymfs/svpg\\_REINFORCE](https://github.com/largelymfs/svpg_REINFORCE)
- [9] Yang Liu, Prajit Ramachandran, Qiang Liu, and Jian Peng. 2017. Stein Variational Policy Gradient. (April 2017). Retrieved February 22, 2022 from <https://arxiv.org/abs/1704.02399v1>
- [10] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. *ArXiv160201783 Cs* (June 2016). Retrieved February 26, 2022 from <http://arxiv.org/abs/1602.01783>
- [11] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. 4.
- [12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *ArXiv191201703 Cs Stat* (December 2019). Retrieved April 3, 2022 from <http://arxiv.org/abs/1912.01703>
- [13] Tuomas Pelkonen, Scott Franklin, Justin Teller, Paul Cavallaro, Qi Huang, Justin Meza, and Kaushik Veeraraghavan. 2015. Gorilla: a fast, scalable, in-memory time series database. *Proc. VLDB Endow.* 8, 12 (August 2015), 1816–1827. DOI:<https://doi.org/10.14778/2824032.2824078>
- [14] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement learning: an introduction* (Second edition ed.). The MIT Press, Cambridge, Massachusetts.
- [15] Jordi TORRES.AI. 2021. Policy-Gradient Methods. *Medium*. Retrieved April 3, 2022 from <https://towardsdatascience.com/policy-gradient-methods-104c783251e0>
- [16] Chris Yoon. 2019. Understanding Actor Critic Methods. *Medium*. Retrieved April 3, 2022 from <https://towardsdatascience.com/understanding-actor-critic-methods-931b97b6df3f>

## 5. Évaluation des sources

**SaLiNa [4]** : L'article et la librairie sont très récents (2021). Des mises à jour sont faites régulièrement.

Une de nos tâches principales est de coder l'algorithme SVPG avec cette librairie. Elle semble avoir plein d'aspects intéressants pour le Machine Learning puisque c'est une librairie qui a beaucoup de potentiel c'est pour cette raison que nous l'utilisons. Elle est essentielle pour notre travail.

C'est une source recommandée par notre professeur, écrite par Nicolas Denoyer, un des auteurs de la librairie. C'est un ancien membre du LIP6, actuellement chercheur au centre de recherche en IA de Facebook. C'est donc l'une des personnes les plus compétentes pour traiter le sujet. L'article a été mis en ligne sur Arxiv (cf. Description des outils de recherche).

Il décrit précisément les différents outils offerts par la librairie et donne de nombreux exemples sur les cas d'utilisation. Le tout est vérifiable en testant nous même les exemples proposés par la librairie.

Le document est un article de recherche décrivant l'utilisation d'une librairie python d'apprentissage par renforcement (Reinforcement Learning RL). Il a pour but de nous familiariser avec. N'étant pas le meilleur format pour décrire une librairie, il est accompagné d'une série de vidéos explicatives, et d'une documentation pour en faciliter la compréhension.

**SVPG [9]** : Cet article a été écrit en juin 2016. Il n'y a pas eu de mise à jour depuis, c'est donc un travail que l'on peut considéré comme plutôt daté dans le domaine de l'IA où les outils et les recherches avancent très vite.

Il présente l'algorithme SVPG en apprentissage par renforcement profond (Deep Reinforcement Learning DRL). C'est un des algorithmes cœur de notre projet qui est présenté dans cet article. Il est donc essentiel pour notre projet malgré que les bibliothèques utilisées ne soient plus d'actualité.

C'est un article de recherche, écrit par Yang Liu, Prajit Ramachandran, Qiang Liu Dartmouth et Jian Peng, les auteurs de l'article sont les créateurs des algorithmes SVPG et SVGD. Ce sont des personnes très compétentes dans le domaine du Machine Learning (5142 et 1288 citations-Google Scholar). Cette source nous a été recommandée par notre professeur lors de la présentation de notre projet.

Le contenu est précis et nous permet d'implémenter les algorithmes. Cependant les résultats sont difficilement reproductibles car les outils utilisés à l'époque sont maintenant obsolètes. Il est donc obligatoire de modifier grandement le code afin de vérifier leurs résultats, ce qui est le but de notre projet.

L'article a pour but d'introduire un nouvel algorithme d'apprentissage par renforcement qui donnerait de meilleurs résultats dans certaines situations. Mais pour le comprendre il faut avoir des connaissances à priori en apprentissage par renforcement profond (DRL) et connaître l'algorithme SVGD.

**Reinforcement Learning [14]** : deuxième édition, publiée en 2018 par MIT press (maison d'édition universitaire américaine affiliée au Massachusetts Institute of Technology à Cambridge, Massachusetts). Comme c'est un ouvrage qui décrit les bases de l'apprentissage par renforcement (RL), le travail est plutôt d'actualité, d'autant plus qu'il a été mis à jour depuis sa version de 1992).

Ce livre est considéré comme la "bible pour ceux qui souhaitent apprendre le l'apprentissage par renforcement (RL)" (Professor of Computer Science, University of Washington, and author of The Master Algorithm), contient des mises à jour sur les algorithmes les plus récents. C'est un contenu un peu trop général, mais il apporte toutes les bases nécessaires pour commencer dans le RL.

Cet ouvrage est souvent revenu dans nos recherches. Il est écrit par Richard S. Sutton and Andrew G. Barto, 2 pionniers dans le domaine. Richard S. Sutton est professeur de sciences informatiques et titulaire de la Chaire AITF en apprentissage par renforcement et en intelligence artificielle à l'Université de l'Alberta. Il est également chercheur scientifique émérite à DeepMind (Google). Andrew

G. Barto est professeur émérite au *College of Computer and Information Sciences* de l'Université du Massachusetts à Amherst.

Le contenu est organisé de manière appropriée et accessible aux nouveaux apprenants. Il a véritablement un objectif “grand public” dans son approche dans le sens où on a pas besoin de connaissance particulière en RL avant d'aborder ce livre. Les auteurs s'attachent donc à donner les exemples pratiques pour aider les lecteurs à comprendre l'intuition derrière des algorithmes complexes, plutôt que de rentrer dans une rigueur mathématique qui risquerait de perdre la plupart des lecteurs.