

Deep learning - Evaluation

L'objectif de ce devoir est d'analyser et de comparer deux méthodes de deep learning pour une même tâche. Vous choisirez entre **la génération de données, la classification ou l'apprentissage par renforcement**. Une fois le thème choisi, vous mènerez des expériences afin de déterminer quelle méthode est la plus performante selon différents critères. Une partie essentielle de ce devoir est la conception de datasets (ou environnement dans le cas du RL) mettant en évidence les différences entre les méthodes. Vous pouvez par exemple créer deux datasets de sorte à ce qu'une méthode performe mieux sur l'un et inversement sur l'autre et expliquer pourquoi. Le but est d'utiliser des données **simples** pour pouvoir bien identifier les faiblesses donc privilégier les données de petite dimensions.

1 Choix de tâches

1.1 Génération de données

Comparer la capacité des WGANs ([Arjovsky et al., 2017](#)) et des SGMs ([Song et al., 2020](#)) à générer des échantillons à partir des jeux de données. Vous devez concevoir des jeux de données qui mettent en évidence les différences de performance. Exemples :

- Un jeu de données multimodal (les GANs peuvent souffrir de mode collapse).
- Un jeu de données avec des détails fins (les SGMs peuvent mieux capturer la structure).
- Un jeu de données sur une sous-variété (les SMGs peuvent souffrir d'explosion de la fonction de score).

1.2 Classification

Comparer la performance en classification d'images des CNNs ([Krizhevsky et al., 2012](#)) et des Vision Transformers ([Lu et al., 2022](#)). Vous devez concevoir des jeux de données qui mettent en évidence les différences de performance. Exemples :

- Classification basée sur la texture (les CNNs devraient exceller).
- Objets nécessitant une dépendance à long terme (les Transformeurs peuvent être plus performants).
- Images endommagées ou partiellement cachées

Si vous avez du mal à concevoir un jeu de données, vous pouvez modifier des datasets déjà existants.

1.3 Apprentissage par Renforcement (RL)

Comparer l'efficacité et la stabilité de l'apprentissage par renforcement avec et sans modèle: MuZero ([Schrittwieser et al., 2020](#)) vs PPO ([Schulman et al., 2017](#)). Vous devez concevoir un environnement RL simple où les différences sont évidentes. Exemples :

- Une tâche où la planification est cruciale (le RL basé sur un modèle devrait être plus performant).
- Une tâche nécessitant une adaptation rapide (PPO peut être plus robuste).

2 Directives Générales

1. Conception du jeu de données/environnement :

- Votre jeu de données ou environnement RL doit mettre en évidence les différences entre les méthodes.
- Justifiez pourquoi le jeu de données choisi est pertinent pour la comparaison.
- La dernière séance de TP sera dédiée à vous guider dans votre projet. Réfléchissez en amont à ce que vous allez faire et on en discutera.

2. Implémentation :

- Vous pouvez utiliser des bibliothèques existantes (PyTorch, TensorFlow...) mais vous devez écrire votre propre pipeline d'entraînement.
- Fournissez un code lisible (**et qui tourne!**).

3. Évaluation des méthodes:

- Choisissez des métriques quantitatives et qualitatives appropriées.
- Fournissez des visualisations et des explications sur les comportements observés.

4. Rapport à soumettre en latex (8-15 pages):

- **Explications du fonctionnement des méthodes :** Présentation de la tâche et des méthodes choisies.
- **Conception du jeu de données/environnement :** Explication de votre choix.
- **Détails de l'implémentation :** Déroulement de l'entraînement. Ne mettez pas le code dans le rapport, fournissez un autre fichier.
- **Résultats et analyse :** Comparaison des performances et interprétation.

5. **Presentation orale:** Vous présenterez votre travail en 7 mins + 3 mins questions.

6. **Travail en équipe :** Vous pouvez travailler seul ou en binôme (le mieux) ou en trinôme.

7. **Date de remise du rapport:** 05/05/2025.

References

- M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y.W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223. PMLR, 2017.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- Zhiying Lu, Hongtao Xie, Chuanbin Liu, and Yongdong Zhang. Bridging the gap between vision transformers and convolutional neural networks on small datasets. *Advances in Neural Information Processing Systems*, 35:14663–14677, 2022.
- Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.