

Projet en Deep Learning

Master Maths & IA, Université Paris-Saclay

Responsable : Guillermo Durand

Table des matières

1. Description de l'UE

- Plan
- Sujets

2. Streamlit

3. Bonnes pratiques

4. Pytorch

Plan

- Des projets à mener en groupe de 2 (ou seul.e, ou 3 à discuter)
- 1 projet = 1 ou plusieurs articles de deep learning
- Livrables :
 - ▶ 1 exposé (20-30 minutes) sur les articles
 - ▶ 1 mini-rapport en \LaTeX (≤ 10 pages sans compter table des matières et bibliographie, surtout pas de code) qui démontre compréhension, ré-appropriation des concepts et capacité d'exposition et de rédaction
 - ▶ 1 implémentation sous forme d'application Streamlit
- ATTENTION, les 03/02 et 10/02 : la séance est à 14h
- La salle 0E5 est disponible l'après-midi pour travailler en autonomie (le matin les 03/02 et 10/02)

Exposés

- 1 ou 2 à chaque séance
- À penser comme un mini-cours donné au reste de la promotion
- Objectif : tout le monde doit comprendre de quoi parle l'article et les méthodes proposées
- Support au choix (tableau, slides, hybride)
- Participation de tous les membres du groupe à part égale

Calendrier

- 8 séances principales
- 20 jan, 27 jan : pas d'exposé
- 3 fév, 10 fév, 17 fév, 3 mars : 1 exposé en début de séance
- 10 mars, 17 mars : 2 exposés

Residual Networks

1 personne

- <https://arxiv.org/pdf/1512.03385.pdf>

Étude et comparaison de fonctions d'activation

1-2 personnes

- <https://arxiv.org/pdf/2109.14545.pdf>

Batch Normalization, Layer Normalization

2 personnes

- <https://arxiv.org/pdf/1502.03167.pdf>
- <https://arxiv.org/pdf/1607.06450.pdf>

VAEs et extensions

2 personnes

- <https://arxiv.org/pdf/1312.6114.pdf>
- <https://arxiv.org/pdf/1803.02991.pdf>

PixelCNN, PixelRNN, Gated Pixel CNN

2 personnes

- <https://arxiv.org/pdf/1601.06759.pdf>
- <https://arxiv.org/pdf/1606.05328.pdf>

Transformer, attention

2 personnes

- <https://arxiv.org/pdf/1706.03762.pdf>

Feature-wise transformations

2-3 personnes

- <https://distill.pub/2018/feature-wise-transformations/>
- <https://arxiv.org/pdf/1709.07871.pdf>
- <https://github.com/ethanjperetz/film>

Neural ODEs, Latent ODEs

2-3 personnes

- <https://arxiv.org/pdf/1806.07366.pdf>
- <https://arxiv.org/pdf/2202.02435.pdf>
- [https://www.sciencedirect.com/\[...\]](https://www.sciencedirect.com/[...])

Deep Koopman

2-3 personnes

- <https://www.nature.com/articles/s41467-018-07210-0>
- [https://proceedings.neurips.cc/\[...\]](https://proceedings.neurips.cc/[...])

Structured Inference Networks

2-3 personnes

- <https://arxiv.org/pdf/1609.09869.pdf>

Generative Adversarial Networks (GANs)

2-3 personnes

- <https://arxiv.org/pdf/1406.2661>
- <https://arxiv.org/pdf/1710.07035>

Normalizing Flow

2-3 personnes

- https://en.wikipedia.org/wiki/Flow-based_generative_model
- <https://arxiv.org/pdf/1410.8516>
- <https://arxiv.org/pdf/1605.08803>
- <https://arxiv.org/pdf/1807.03039>

Implicit Neural Representations with Periodic Activation Functions

2-3 personnes

- <https://arxiv.org/pdf/2006.09661.pdf>

Table des matières

1. Description de l'UE

2. Streamlit

3. Bonnes pratiques

4. Pytorch

Streamlit

- Module Python qui produit des applications interactives sur navigateur
- Similaire à shiny pour R
- Bon outil de communication et de visualisation
- Exemples d'applications sur <https://streamlit.io/>

Table des matières

1. Description de l'UE

2. Streamlit

3. Bonnes pratiques

- git
- Documentation
- Autres

4. Pytorch

- Logiciel de contrôle de versions
- Outil très efficace pour collaborer à plusieurs sur du code
- 1 projet = 1 repository git
- Un repo a plusieurs branches parallèles qui conservent l'historique des modifications
- 1 sauvegarde sur une branche = 1 commit
- Plusieurs sites d'hébergement de projet fonctionnent avec git, notamment :
 - ▶ <https://github.com/>
 - ▶ <https://gitlab.com/>
- De nombreux IDE ont des fonctionnalités git intégrées (VSCode, Rstudio...)

- Aujourd'hui : création de comptes github, création de repo, ajout du reste du groupe comme collaborateurs, accès à github en ssh pour ne pas retaper son mot de passe à chaque fois, lien pour le faire
- Commandes les plus utiles :
 - ▶ clone
 - ▶ commit
 - ▶ push
 - ▶ pull
 - ▶ branch
 - ▶ checkout
 - ▶ merge
- formation en ligne sur Openclassrooms
- Site utile pour ces commandes de base :
[https://www.freecodecamp.org/news/10-important-git-commands\[...\]](https://www.freecodecamp.org/news/10-important-git-commands[...])

- Utiliser les branches ! Une branche “main” plus une branche par membre du groupe
- La branche “main” ne devrait comporter que des merge d’autres branches de développement (ne pas suivre mon mauvais exemple)
- Bien préparer son .gitignore
- Github permet de fork un repo d’un autre auteur

Documentation

- Bien documenter son code :
 - ▶ Noms d'objets significatifs
 - ▶ Commentaires si nécessaire (sans en abuser)
 - ▶ Et surtout docstrings!
<https://numpydoc.readthedocs.io/en/latest/format.html> ou
<https://google.github.io/styleguide/pyguide.html#38-comments-and-docstrings>
 - ▶ Ne pas laisser de code inutile
- sphinx lit automatiquement les docstrings et crée une documentation (html ou pdf)

- Linting :
 - ▶ Améliorer la lisibilité du code en se conformant à des standards
 - ▶ 1 ou 2 outils possibles : flake8, black
 - ▶ Se combine avec git grâce à pre-commit
- Utiliser `if __name__ == "__main__":` pour les scripts
- Faire plein de petites fonctions avec un but précis (et les documenter avec des docstrings)
- Pas de code dupliqué (les fonctions aident beaucoup à l'éviter)
- Environnement virtuel

Table des matières

1. Description de l'UE
2. Streamlit
3. Bonnes pratiques
4. Pytorch

- La bibliothèque de deep learning de référence actuelle
- Abandon progressif de Tensorflow, émergence de JAX
- Permet de manipuler plusieurs jeux de données grâce à :
 - ▶ `torch.utils.data.DataLoader`
 - ▶ `torchtext.datasets`
 - ▶ `torchvision.datasets`
 - ▶ `torchaudio.datasets`