Table des matières

Comparaison de deux approches de benchmark avec ubench.h

I. Objectif

Ce document présente une comparaison entre deux implémentations de benchmark pour un projet de réseau de neurones en C utilisant la bibliothèque ubench.h. Les deux versions, V1 et V2, mesurent les performances de la propagation et de la rétropropagation sur différentes architectures de réseaux de neurones, ainsi que la génération de données.

II. Description des deux versions

2.1 Version 1 : avec FIXTURE (UBENCH_F)

La version V1 utilise la fonctionnalité FIXTURE de ubench.h pour initialiser et nettoyer les structures du réseau avant et après chaque test. Cela permet une séparation claire entre la phase de setup, le test, et le teardown. Un fichier benchmark.h définit une structure de fixture même si elle est vide.

- Avantages :
  - Structure modulaire, facile à étendre
  - Permet de gérer des ressources complexes
- Inconvénients :
  - Syntaxe plus verbeuse
  - Peut paraître plus complexe si les fixtures ne sont pas nécessaires

2.2 Version 2 : approche simple (UBENCH_EX)

La version V2 repose uniquement sur des benchmarks simples (UBENCH_EX), sans gestion de fixture. Chaque test crée et détruit localement son réseau de neurones. C'est une approche plus directe et compacte.

- Avantages :
  - Plus simple à lire et à écrire
  - Suffisant si les tests sont autonomes
- Inconvénients :
  - Moins flexible pour la réutilisation d'état entre les tests
  - Risque de duplication de code (création/destruction répétées)

III. Résultats des benchmarks

Les tests ont été exécutés sur la même machine. Voici les moyennes obtenues :

Nom du test	Résultat V1 (µs)	Résultat V2 (µs)	Différence
propagation_2_2	0.137	0.116	-0.021
backpropagation_2_2	0.152	0.140	-0.012
spirales_archimede	11.119	11.441	+0.322
spirales_3_classes	20.261	19.981	-0.280
creer_reseau_2_3	1.895	1.896	+0.001
propagation_3->4	0.446	0.482	+0.036
backpropagation_3->4	0.415	0.432	+0.017

IV. Analyse

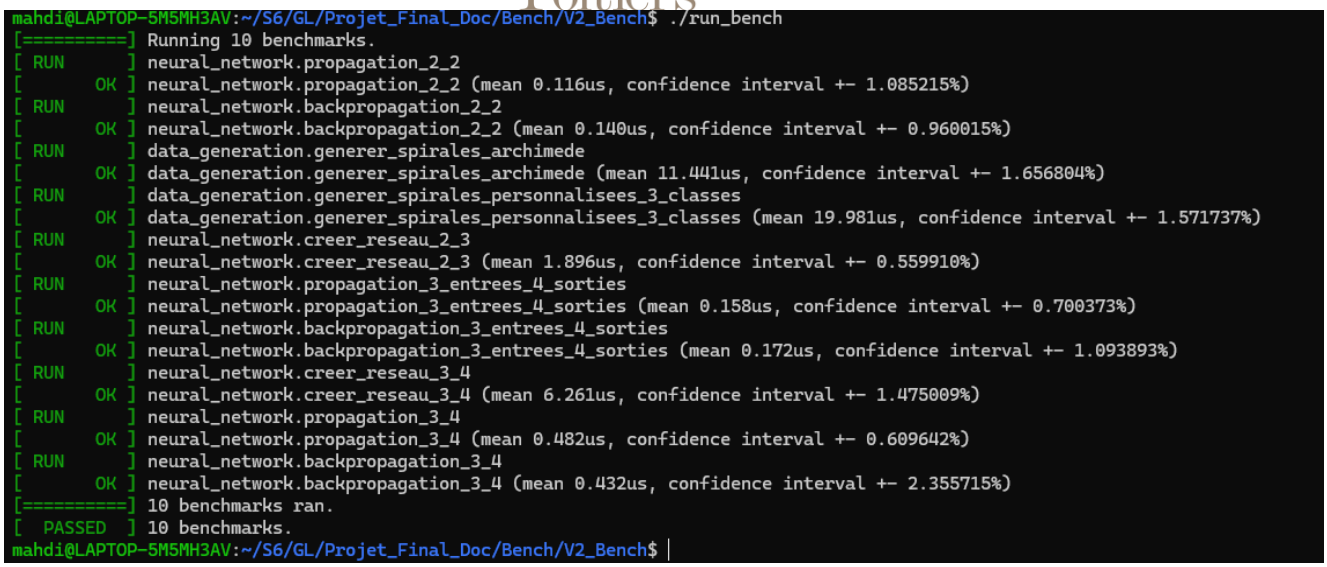
Les résultats montrent que les deux versions sont très proches en performance. Quelques variations peuvent s'expliquer par le bruit d'exécution ou des allocations mémoire locales répétées dans la version V2. Toutefois, l'écart reste négligeable dans tous les cas.

V. Conclusion

Les deux approches sont valides. La V1 est plus rigoureuse et adaptée à des cas complexes avec beaucoup d'état partagé. La V2 est plus rapide à écrire et lisible. Pour ce projet, la V2 suffit largement, mais la V1 pourrait être utile pour des tests plus complexes.

VI. Captures d'écran

```
mahdi@LAPTOP-5M5MH3AV:~/S6/GL/Projet_Final_Doc/Bench/V1_Bench$ ./run_bench
[==========] Running 10 benchmarks.
[ RUN      ] neural_network_propagation_2_2.propagation_only
[       OK ] neural_network_propagation_2_2.propagation_only (mean 0.137us, confidence interval +- 2.027209%)
[ RUN      ] neural_network_propagation_2_2.backpropagation
[       OK ] neural_network_propagation_2_2.backpropagation (mean 0.152us, confidence interval +- 0.826483%)
[ RUN      ] data_generation.generer_spirales_archimede
[       OK ] data_generation.generer_spirales_archimede (mean 11.119us, confidence interval +- 0.676373%)
[ RUN      ] data_generation.generer_spirales_personnalisees_3_classes
[       OK ] data_generation.generer_spirales_personnalisees_3_classes (mean 20.261us, confidence interval +- 1.723019%)
[ RUN      ] neural_network.creer_reseau_2_3
[       OK ] neural_network.creer_reseau_2_3 (mean 1.895us, confidence interval +- 0.523525%)
[ RUN      ] neural_network.propagation_3_entrees_4_sorties
[       OK ] neural_network.propagation_3_entrees_4_sorties (mean 0.142us, confidence interval +- 1.068513%)
[ RUN      ] neural_network.backpropagation_3_entrees_4_sorties
[       OK ] neural_network.backpropagation_3_entrees_4_sorties (mean 0.179us, confidence interval +- 1.028388%)
[ RUN      ] neural_network.creer_reseau_3_4
[       OK ] neural_network.creer_reseau_3_4 (mean 6.286us, confidence interval +- 1.196457%)
[ RUN      ] neural_network.propagation_3_4
[       OK ] neural_network.propagation_3_4 (mean 0.446us, confidence interval +- 1.508225%)
[ RUN      ] neural_network.backpropagation_3_4
[       OK ] neural_network.backpropagation_3_4 (mean 0.415us, confidence interval +- 1.997341%)
[==========] 10 benchmarks ran.
[  PASSED  ] 10 benchmarks.
mahdi@LAPTOP-5M5MH3AV:~/S6/GL/Projet_Final_Doc/Bench/V1_Bench$
```



Projet encadré par M. Nicolas Courilleau
Réalisé par El Mahdi Benfdal et Ewen Croizier
Licence 3 Informatique – Génie Logiciel
Année universitaire 2024-2025