

# Apprentissage profond et ses applications

## Devoir 1

Date de remise : 7 octobre 2024

## 1 Description

En classe, nous avons appris à utiliser la librairie PyTorch, ses modules de base, les opérations sur les tenseurs de même que la création de nos propres modèles.

Dans ce devoir, nous passerons en revue certaines opérations tensorielles et construirons des modèles avancés d'apprentissage en profond pour deux problèmes réels, notamment la classification d'images et la classification de phrases.

Ce devoir se divise en trois parties:

- **Partie 1: Opérations sur les tenseurs.** Comme nous l'avons appris en classe, les tenseurs sont la structure de données fondamentale utilisée dans PyTorch. Une variété d'opérations sur les tenseurs a été implémentée dans cette librairie, et ces opérations servent de blocs de construction de nos modèles d'apprentissage en profond. Pour vous permettre de vous familiariser avec les opérations de base sur les tenseurs, nous passerons en revue certaines opérations. Si vous n'êtes pas familier.ère.s avec certaines opérations, vous pouvez vous référer au [document officiel de PyTorch](#), où certaines explications plus détaillées sont disponibles.
- **Partie 2: Réseaux de neurones convolutifs.** En vision par ordinateur, les réseaux de neurones convolutifs (CNNs) forment une famille de modèles des plus importants, ne serait-ce que parce que plusieurs modèles plus avancés mettent à profit pareille architecture. En outre, les CNNs se sont avérés efficaces dans une variété d'applications, telles que la classification d'images et la détection d'objets. Dans cette partie du devoir, nous allons implémenter un réseau de neurones convolutifs et l'appliquer pour la classification d'images sur l'ensemble de données CIFAR10.
- **Partie 3: Réseaux de neurones récurrents.** Dans le traitement du langage naturel, les réseaux de neurones récurrents (RNNs) sont, à ce jour, encore mis à profit par plusieurs architectures faisant partie de l'état de l'art dans le domaine. Pour se familiariser davantage avec les RNNs, nous allons construire un RNN afin d'analyser des sentiments, c'est-à-dire classer une phrase comme étant associée à un sentiment positif, négatif ou neutre. Pour voir un exemple ludique, vous pouvez visiter cette [page web](#).

## 2 Instructions

Plus de détails sur les instruction du devoir se trouve sur le [Colab](#). Comme nous l'avons vu dans le tutoriel lors de la quatrième semaine, vous pouvez directement travailler à même l'environnement Colab, où des GPUs sont mis à votre disposition.

**Veillez ne pas enregistrer sur le fichier d'origine. Au lieu de cela, vous devriez faire une copie du fichier sur votre propre Google Drive et faire vos devoirs sur votre propre copie.**

## 3 Soumission du devoir

Pour soumettre votre devoir, remplir [le document suivant](#).

## 4 Évaluation

Pour les trois parties du devoir, nous vérifierons votre code et vos résultats pour voir si les modèles ont été implémentés de manière correcte. De plus, pour les Parties 2 et 3, il y a des tableaux à remplir lesquels seront évalués.