

Exercice Python

Hadrien Bodin

November 2025

L'entraînement de modèle d'intelligence artificielle dits supervisés demande d'itérer un grand nombre de fois sur des objets. On dit qu'un modèle apprend en plusieurs époques. A chaque époque, le modèle doit voir tous les éléments d'un dataset.

On considère une classe dataset que l'on veut iterable. Comme indiqué précédemment, on souhaite itérer dessus pour chaque époque.

On considère ici que l'objet d'étude est "un passager du Titanic". Vous pouvez récupérer la liste des passagers depuis le dataset d'exemple.

Il est nécessaire que les objets du dataset soient :

- Présentés dans un ordre différent à chaque époque
- Présentés en "batch" (ou groupe) d'objets. L'itérateur doit donc renvoyer non pas un mais plusieurs objets à chaque appel.
 - Le nombre d'objet par batch doit être laissé comme paramètre à l'utilisateur
 - Les batch doivent tous avoir le même nombre d'objets à l'exception du dernier
 - L'utilisateur peut définir un "overlap", qui correspond au recouvrement entre deux batchs successifs. Par exemple un recouvrement de 2, signifie que les 2 derniers objets du n-ième batch sont les mêmes que les 2 premiers du (n+1)-ième batch
- Pour augmenter artificiellement la quantité de données, il peut être pertinent de passer par des fonctions d'augmentation.
Ces fonctions enrichissent la donnée. On peut par exemple penser à une fonction qui pivoterait une image, ou la recadrerait.
Il est crucial que ces fonctions soient appliquées aléatoirement et avec une probabilité p à chaque élément du dataset, et à chaque époque.
 - Permettre à l'utilisateur de définir des fonctions d'augmentations 'custom' et les probabilités d'application associées.
 - Gérer l'application de la fonction
 - Par ailleurs, proposer des fonctions d'augmentation par défaut
 - Tester le fonctionnement en donnant une fonction d'augmentation 'custom' au dataset

Construire la classe dataset