

DM-2 : Boosting et cartes de saillance

L'intention de ce DM est de :

1. Utiliser un algorithme de **boosting** sur de la reconnaissance de chiffres manuscrits
2. Réaliser des **cartes de saillance** à partir des modèles appris en faisant varier certains paramètres
3. Réaliser l'apprentissage d'un **enchâssement** (*'embedding'*) et refaire les expériences avec cet embedding comme espace d'entrée.

La suite de ce document fournit des indications pour réaliser les expériences. Vous êtes libres d'utiliser des bibliothèques différentes.

1. Utilisation d'un algorithme de boosting sur de la reconnaissance de chiffres manuscrits

Pour ces expériences, vous pourrez vous appuyer sur les ressources mentionnées dans <https://medium.com/@nik.avgeros/unleashing-the-power-of-ensemble-methods-a-deep-dive-into-classification-using-the-mnist-dataset-1aeefab4c6a6>.

Mais vous êtes libres d'utiliser d'autres ressources.

Notez qu'il ne faut pas utiliser de réduction de dimension par ACP dans les expériences qui suivent.

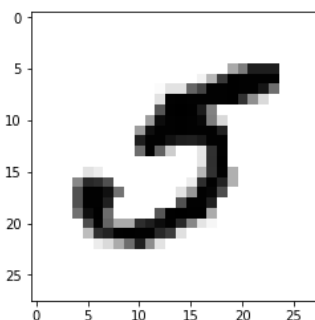


FIGURE 1 – Un exemple de '5' de la base MNIST.

L'idée est de tester l'algorithme AdaBoost sur les données MNIST (chiffres manuscrits décrits par $28 * 28$, soit 784, pixels en niveaux de gris) en faisant varier :

- les **classifieurs faibles** : décision stumps, mais aussi perceptron et arbres de décision de profondeurs différentes
- le **nombre d'étapes** de boosting
- les **classes binaires** étudiées (e.g. les '3' contre les '8', ou la classe de tous les chiffres entre '0' et '4' contre tous les chiffres entre '5' et '9')

1. Vous reporterez les **matrices de confusion** obtenues pour chaque combinaison de paramètres.
2. Vous en tirerez un **bilan** avec des **conclusions** sur les configurations les plus **favorables** et les plus **défavorables**.

2. Réalisation de cartes de saillance

Dans le contexte de la reconnaissance de chiffres manuscrits dans l'espace d'entrée des pixels, chaque “*decision stumps*” teste un *pixel*.

En examinant les décision stumps appris, et donc les pixels testés, il est possible de tracer une carte de saillance.

1. Donner les cartes de saillance obtenues pour diverses combinaisons testées dans la section 1.
Il est plus facile d'interpréter pour des discriminations binaires entre deux chiffres (e.g. les '3' contre les '8') et non entre groupes de chiffres (e.g. tous les chiffres entre '0' et '4' contre tous les chiffres entre '5' et '9').
2. Qu'en concluez-vous ?

3. Passage par un enchâssement (*embedding*)

Vous allez apprendre un espace de redescription en réalisant une auto-association des images en 28*28 pixels grâce à un Perceptron Multi-Couches à une couche cachée de N neurones avec $N \ll 784$ (e.g. 64).

Vous réaliserez alors l'apprentissage par boosting en partant de la description des exemples dans ce nouvel espace d'entrée.

1. Faites varier à nouveau des paramètres de l'algorithme AdaBoost, comme dans la section 1.
2. Donner les cartes de saillance obtenues pour diverses combinaisons testées dans la section 1 dans l'espace d'entrée des pixels.
Ici, on obtiendra les cartes de saillance dans l'espace d'entrée $\mathcal{X}$ en rétropropageant les activités mesurées dans les neurones identifiés dans l'espace de redescription, comme dans GradCAM.