

TD : **keras** & validation croisée

Romain Tavenard

Dans cette séance, nous nous focaliserons sur la création et l'étude de modèles de type perceptron multi-couches à l'aide de la librairie **keras** et sur la sélection d'hyper-paramètres pertinents à l'aide des outils **sklearn**.

1 Modèles **keras** dans **sklearn**

Il est possible de transformer vos modèles **keras** (en tout cas, ceux qui sont de type **Sequential**) en modèles **sklearn**. Cela a notamment pour avantage de vous permettre d'utiliser les fonctionnalités de sélection de modèles vues lors du TD précédent.

Pour cela, vous devrez utiliser au choix l'une des classes **KerasClassifier** ou **KerasRegressor** (selon le problème de *machine learning* auquel vous êtes confronté) du module **keras.wrappers.scikit-learn**.

Le principe de fonctionnement de ces deux classes est le même :

```
clf = KerasClassifier(build_fn=ma_fonction, param1=12, param2="sgd", ...)
clf.fit(X, y)
clf.predict(X_test)
```

Une fois construit, l'objet **clf** s'utilise donc exactement comme un classifieur **sklearn**. L'attribut **build_fn** prend le nom d'une fonction qui retourne un modèle **keras**. Les autres paramètres passés lors de la construction du classifieur peuvent être :

- des paramètres de votre fonction **ma_fonction** ;
 - des paramètres passés au modèle lors de son apprentissage (appel à la méthode **fit()**).
1. Créez un réseau à deux couches cachées transformé en objet **sklearn** en spécifiant, lors de sa construction, le nombre d'itérations et la taille des *batches* de votre descente de gradient par *mini-batches*. Vous pourrez utiliser la méthode **score()** des objets **sklearn** pour évaluer ce modèle.
 2. Utilisez les outils de validation croisée de **sklearn** pour choisir entre les algorithmes d'optimisation **"rmsprop"** et **"sgd"**.

3. Mettez en place une validation croisée pour choisir la structure (nombre de couches, nombre de neurones par couche) et l'algorithme d'optimisation idoines pour le problème lié au jeu de données *Boston Housing* (pour lequel une fonction de préparation des données est fournie dans le module `dataset_utils`).