

## Unité d'enseignement *Mathématiques et informatique*

### Apprentissage Par Projet (APP) : Entraîner un réseau de neurones dense à classifier des données issues d'un banc de perçage

#### Acquis d'apprentissage visés

À la fin de cet apprentissage par problème de  $3 \times 3h$ , chaque étudiant saura :

- ▷ Programmer en langage Python un *Réseau de Neurones Dense* (RND) dédié à la classification de données en utilisant les modules `tensorflow2` et `keras`.
- ▷ Expliquer la structure des données labellisées construites avec les données temporelles brutes issues du banc de perçage.
- ▷ Conduire l'entraînement supervisé du RND en utilisant les données labellisées disponibles dans un fichier CSV.
- ▷ Évaluer les performances du réseau entraîné en utilisant des données de test, en particulier afficher la matrice de confusion.

## → Planification suggérée des 3 séances d'APP

Au cours des 3 séances, les étudiants travaillent en autonomie par équipe de 3 ou 4 en utilisant l'**Environnement Virtuel Python** (EVP) `minfo_ml` spécialement créé pour cet APP sur leurs ordinateurs portables.

### → Travail préliminaire : à faire avant les 3 séances, avec un accès Internet haut-débit

- ▷ Télécharger les documents de travail depuis l'ENT SAVOIR.
- ▷ Créer l'EVP `minfo_ml` et installer les modules Python nécessaires au travail de *Machine Learning*, en suivant la procédure donnée sur page suivante.

### → Séance 1 & 2 : auto-formation, prise en main du problème

- ▷ travail personnel avec les trois *notebooks* `ML1_MNIST.ipynb`, `ML2_DNN.ipynb`, et `ML3_DNN_suite.ipynb` pour apprendre à charger les images MNIST, construire un réseau de neurones dense et l'entraîner à reconnaître les images du MNIST.
- ▷ **Prise en main du problème de l'APP** : entraînement supervisé d'un RND pour classifier les données labellisées obtenues par traitement des données brutes d'un banc de perçage instrumenté. Exploitation du RND entraîné, avec différents objectifs.

### → Séance 3 : fin du projet + rendu

- ▷ Première heure et demie ~ Fin du projet, finalisation du rendu ;
- ▷ Dernière heure et demie ~ Évaluation : présentation interactive du rendu à l'enseignant.

## → Préliminaire – Configurer l'EVP `minfo_ml`, auto-formation au *Machine Learning*

Télécharger depuis SAVOIR l'archive `APP-ML...zip` et **extraire** le dossier `APP-ML` sur ton ordinateur portable. Une fois extrait, le contenu du dossier `APP-ML` est le suivant :

 **Dossier Cours** : contient le fichier PDF de l'amphi "Comprendre et utiliser le *Machine Learning*".

**Dossier Notebook** : contient les *notebooks* d'auto-formation `ML...ipynb`.

**Dossier Drilling-Problem** : contient les fichiers pour traiter la partie "problème" de l'APP.

**Fichier Consignes.pdf** : le présent document.

**Fichier requirements.txt** : contient la liste des modules Python nécessaires au travail de l'APP.

### Créer et configurer l'Environnement Virtuel Python `minfo_ml`

Sur ton ordinateur portable avec `miniconda3` déjà installé :

- ▷ Ouvre une console **Anaconda Prompt** (Windows) ou un **terminal** (Mac ou Linux).
- ▷ Crée l'EVP `minfo_ml` en tapant la commande : `conda create -n minfo_ml pip python=3.8`  
puis réponds aux questions pour télécharger et installer les paquets Python...
- ▷ Active l'EVP `minfo_ml` en tapant la commande : `conda activate minfo_ml`

Les modules nécessaires à l'APP sont ensuite installés en utilisant le fichier `requirements.txt` :

- ▷ Positionne-toi dans le dossier `APP-ML` avec la commande :  
(Windows)  $\leadsto$  `cd C:\<chemin_du_dossier_APP-ML_copié_depuis_le_navigateur_de_fichiers>`  
(Mac/Linux)  $\leadsto$  `cd /home/.../APP-ML`
- ▷ Charge les modules Python nécessaires en tapant : `pip install -r requirements.txt`  
Les modules et leurs dépendances sont téléchargés et installés...
- ▷ Complète l'installation avec la commande : `conda install numpy pydot pydotplus`

## → Séance 1 & 2 – Auto-formation, prise en main du problème



### Auto-formation : travail personnel avec les *notebooks* de l'APP

Les trois *notebooks* de l'APP permettent d'acquérir les savoir-faire nécessaires pour construire un réseau de neurones dense et apprendre à l'entraîner avec des données labellisées.

Chaque étudiant doit s'auto-former en travaillant les *notebooks* chargés dans un processus *jupyter notebook* lancé dans l'EVP `minfo_ml`.



### Problème à traiter en équipe : entraîner un RND à classifier des données labellisées issues d'un banc de perçage

Le travail se fait avec le notebook `ML-drilling.ipynb` du dossier `Drilling-Problem`. Le plan de travail proposé est :

1. Comprendre comment est constitué le jeu de données labellisées qui va servir à entraîner le réseau de neurones.
2. Construire le RND en utilisant les acquis d'apprentissage de l'auto-formation.
3. Conduire l'entraînement supervisé du RND avec le jeu des données labellisées.
4. Évaluer les performances du RND entraîné en utilisant les données de test.
5. Rechercher parmi les données d'entraînement celles qui sont le plus pertinentes en considérant plus critères de pertinence.

Le projet est conduit par l'équipe qui doit répartir entre ses membres les tâches à réaliser pour le projet.

## → Séance 3 : rendu



### Préparation du notebook de rendu du projet

Le rendu de l'équipe est le *notebook* [ML-drilling.ipynb](#) complété et annoté par les membres de l'équipe.

Critères d'évaluation du rendu :

- aptitude à mettre en oeuvre les acquis d'apprentissage visés ;
- aptitude à présenter les différentes étapes de la résolution du problème projet en utilisant les possibilités d'un *notebook* Python ;
- aptitude à présenter et critiquer les résultats obtenus.
- aptitudes individuelles à répondre aux questions.