

TP1 : Apprentissage par un neurone d'un ensemble de points répartis en deux classes dans le plan

1 – Création du dataset pour l'entraînement du neurone

Créer deux ensembles de points de dimension 2, l'un de classe 0 et l'autre de classe 1.

Précision : utiliser la fonction python `Random()` qui génère des réels compris entre 0 et 1 ; puis arranger vous pour séparer géométriquement les deux ensembles de points. Exemple : le premier ensemble de points (classe 0) seront dans la zone : $0 \leq \text{abscisse} \leq 10$ et $0 \leq \text{ordonnée} \leq 10$ et le second ensemble de points (classe 1) seront dans la zone : $-10 \leq \text{abscisse} < 0$ et $0 \leq \text{ordonnée} \leq 10$ (ce n'est qu'une proposition ; vous pouvez découper autrement si vous le souhaitez).

. Garder environ 80% des points pour la phase d'entraînement : **dataset_entrainement**(phase2)

. Garder environ les 20% autres points pour la phase de test : **dataset_test** (Phase 3)

2 – Entraînez un neurone avec les 80% du dataset de points généré en phase 1 grâce à l'algorithme vu en cours.

Précision : A la fin de l'entraînement, les paramètres du modèle, à savoir : **w₁, w₂ et b** sont ainsi fixés.

3 – Mesurer la précision du modèle (neurone)

. **Rappel** : Une fois l'entraînement terminée (Phase 2), les paramètres **w₁, w₂ et b** du modèles sont fixés.

. Utilisez votre modèle(neuron)e ainsi obtenu pour prédire les données des points du dataset **dataset_test**.

. Calculer la précision du modèle par la formule :

Précision = (nombre de points bien classés / nombre total de points)

4 – Refaire l'expérience (phase 1, 2 et 3) avec de nouveaux dataset répartis différemment dans le plan.