

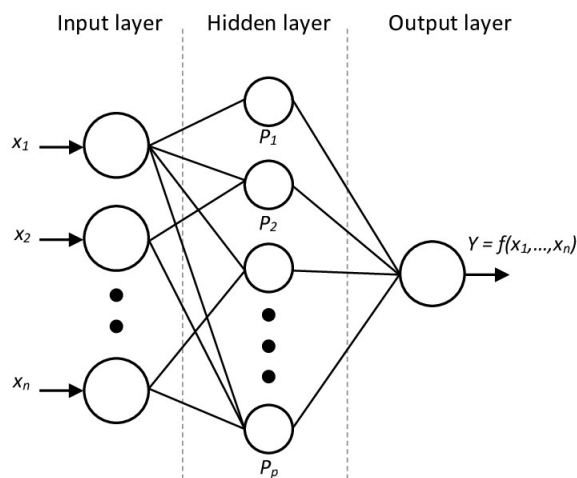
# Machine Learning

## TP3

### Perceptrons multi-couches et Arbres de Décision

#### Exercice 1.

Ce que vous avez réalisé dans la TP2 sera utile. Vous allez utiliser un Perceptron avec une couche supplémentaire (voir figure). Entre deux couches successives les neurones sont complètement connectés avec les précédents. Au début de l'apprentissage tous les poids seront initialisés aléatoirement dans l'intervalle  $-0.5 \dots 0.5$  ; ensuite il n'y aura que les poids entre la couche intermédiaire (« hidden layer ») et la couche de sortie (« output layer ») qui pourront varier selon la règle d'apprentissage du Perceptron utilisée dans le TP2.



Question :

Après la phase d'apprentissage, pour les données « student-data-train.csv » et « student-data-test.csv » mesurer le taux de classifications correctes. On essayera un nombre de neurones dans la couche cachée égal à : 10 ; 50 ; 100 ; 200. A vous de décider le nombre d'itérations et la valeur du paramètre d'apprentissage.

#### Exercice 2.

Dans cet exercice le but est de se familiariser avec la librairie Python *scikit-learn* (<https://scikit-learn.org/stable/install.html>) .

Questions :

- Construire un arbre de décision avec les données de l'exercice précédent, en jouant avec certaines valeurs des paramètres (*min\_samples\_leaf*, *max\_depth*) et mesurer le taux de classifications correctes pour les ensembles d'entraînement et de test.
- Visualiser les arbres de décision avec *plot\_tree* ([https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot\\_tree.html](https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html)),
- Même question avec les données « Iris ».

## Indications.

- La liste des classes de *Scikit* se trouve à la page : <http://scikit-learn.org/stable/modules/classes.html>.

## Rendu.

Ce travail n'est pas noté (mais apprécié) ; il faudra le rendre sur *Cyberlearn* au plus tard le **mercredi 15 novembre**.