

LYCÉE LA MARTINIÈRE MONPLAISIR LYON

SCIENCES INDUSTRIELLES POUR L'INGÉNIEUR

CLASSE PRÉPARATOIRE P.S.I.

ANNÉE 2023 - 2024



C8 : RÉOLUTION DES MODÈLES PAR L'UTILISATION DE L'INTELLIGENCE ARTIFICIELLE

## TD 20 - Apprentissage supervisé et Machine Learning et apprentissage profond (C8-1)

### Compétences

- **Analyser**
  - Analyser un algorithme.
  - Analyser les principes d'intelligence artificielle.
  - Extraire un indicateur de performance pertinent à partir du cahier des charges ou de résultats issus de l'expérimentation ou de la simulation.
  - Caractériser les écarts entre les performances.
  - Interpréter et vérifier la cohérence des résultats obtenus expérimentalement, analytiquement ou numériquement.
  - Rechercher et proposer des causes aux écarts constatés.
- **Résoudre**
  - Choisir une démarche de résolution d'un problème d'ingénierie numérique ou d'intelligence artificielle.
  - Résoudre un problème en utilisant une solution d'intelligence artificielle.
- **Communiquer**
  - Trier les informations selon des critères.
- **Concevoir**
  - Modifier la commande pour faire évoluer le comportement du système.

## Exercice 1 : Apprentissage automatique du robot Alpha1

Source : Emilien DURIF

## 1 Présentation

On s'intéresse ici au robot **Alphai** développé pour la société **learnin-grobots** qui permet d'appréhender les enjeux de l'intelligence artificielle et du **Machine Learning** sur la robotique.

On peut utiliser ce robot pour lui faire apprendre par exemple à faire le plus grand parcours dans une enceinte sans toucher les murs par les méthode d'apprentissage supervisé selon :

- Recherche de plus proche voisin : machine learning
- Réseau de neurones : apprentissage profond



Le scénario d'apprentissage est le suivant :

On dispose d'une enceinte et d'un mur central. On souhaite que le robot alphai fasse le plus de tour consécutif du mur central en le moins de temps possible.



Pour cela on dispose de 3 actions :

Rotation sur place à gauche    Déplacement tout droit    Rotation sur place à droite



On dispose de données d'entrée provenant de capteurs :

- caméra : jusque 640x380 pixels
- capteur de robot bloqué

## 2 Apprentissage supervisé par recherche du plus proche voisin

**Q 1 : Mettre en oeuvre l'apprentissage du robot alphai par l'apprentissage supervisé par recherche du plus court chemin en suivant la documentation en annexe.**

Exporter les données d'apprentissage dans le menu IA/sauver la mémoire d'expérience. On pourra exporter les données sous le nom *alphai\_knn.buffer.npz*.

**Q 2 : Exploiter les données d'apprentissage à l'aide du script *traiter\_alphai.py* qui doit être sauvegardé dans le même répertoire que *alphai\_knn.buffer.npz*.**

**Q 3 : Faire varier le nombre de plus proches voisins et observer les résultats sur la matrice de confusion, le score d'apprentissage et la représentation graphique des plus proches voisins.**

**Q 4 : Mettre en oeuvre le robot en autonomie et vérifier ses performances.**

## 3 Apprentissage supervisé par réseau de neurones

**Q 5 : Mettre en oeuvre l'apprentissage du robot alphai par l'apprentissage supervisé réseau de neurones en suivant la documentation en annexe.**

**Q 6 : Faire varier le nombre de neurones, de couches et de données d'entrées.**

**Q 7 : Mettre en oeuvre le robot en autonomie et vérifier ses performances.**

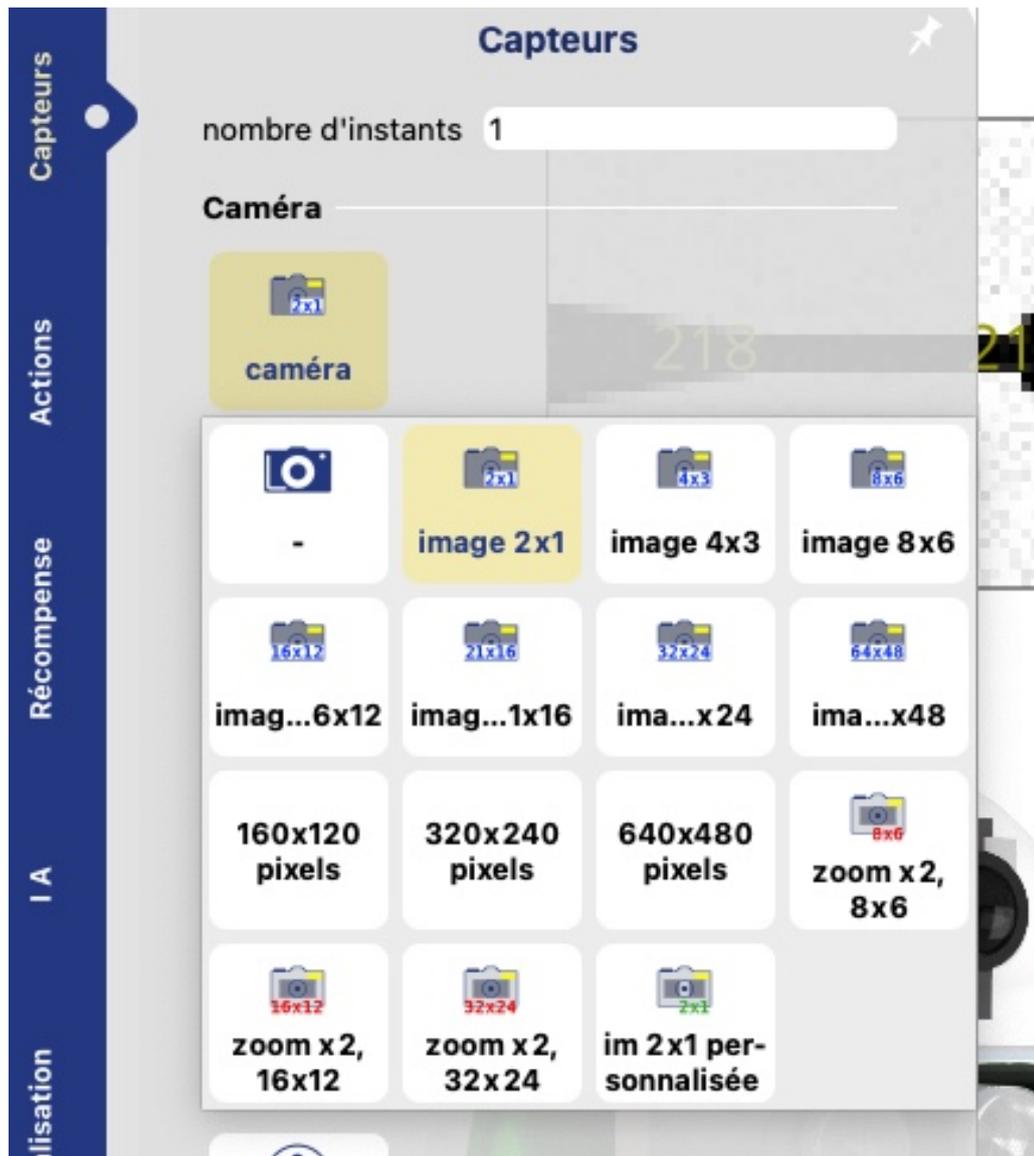
## 4 Annexe : utilisation du logiciel alphaI

Lancer le logiciel **alphaI** en mode simulation



### a) Apprentissage supervisé par recherche du plus proche voisin

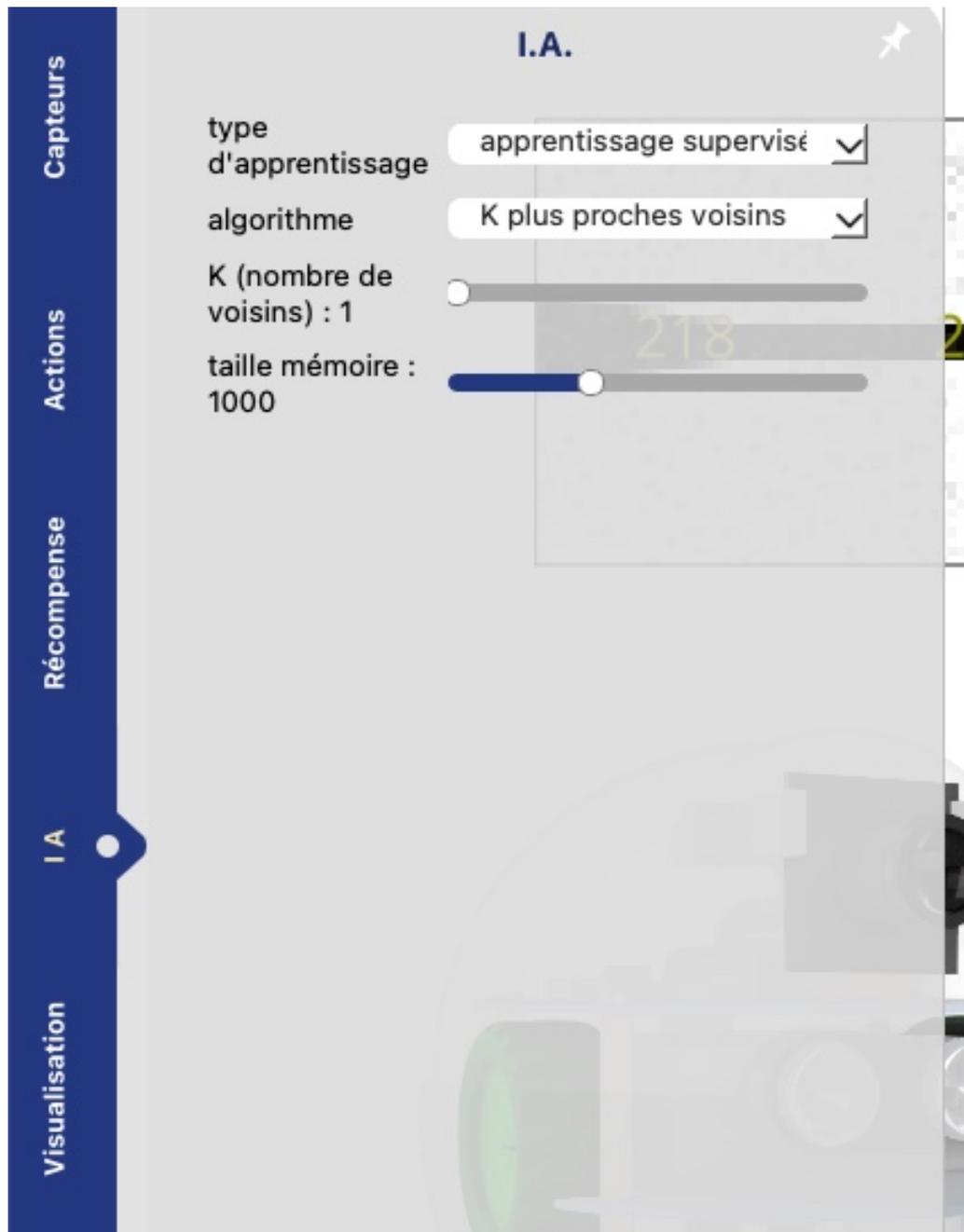
1. Dans l'onglet **capteur** choisir en premier lieu (pour le mode plus proche voisin) :
  - camera image : 2x1,
  - précalcul image : canal vert



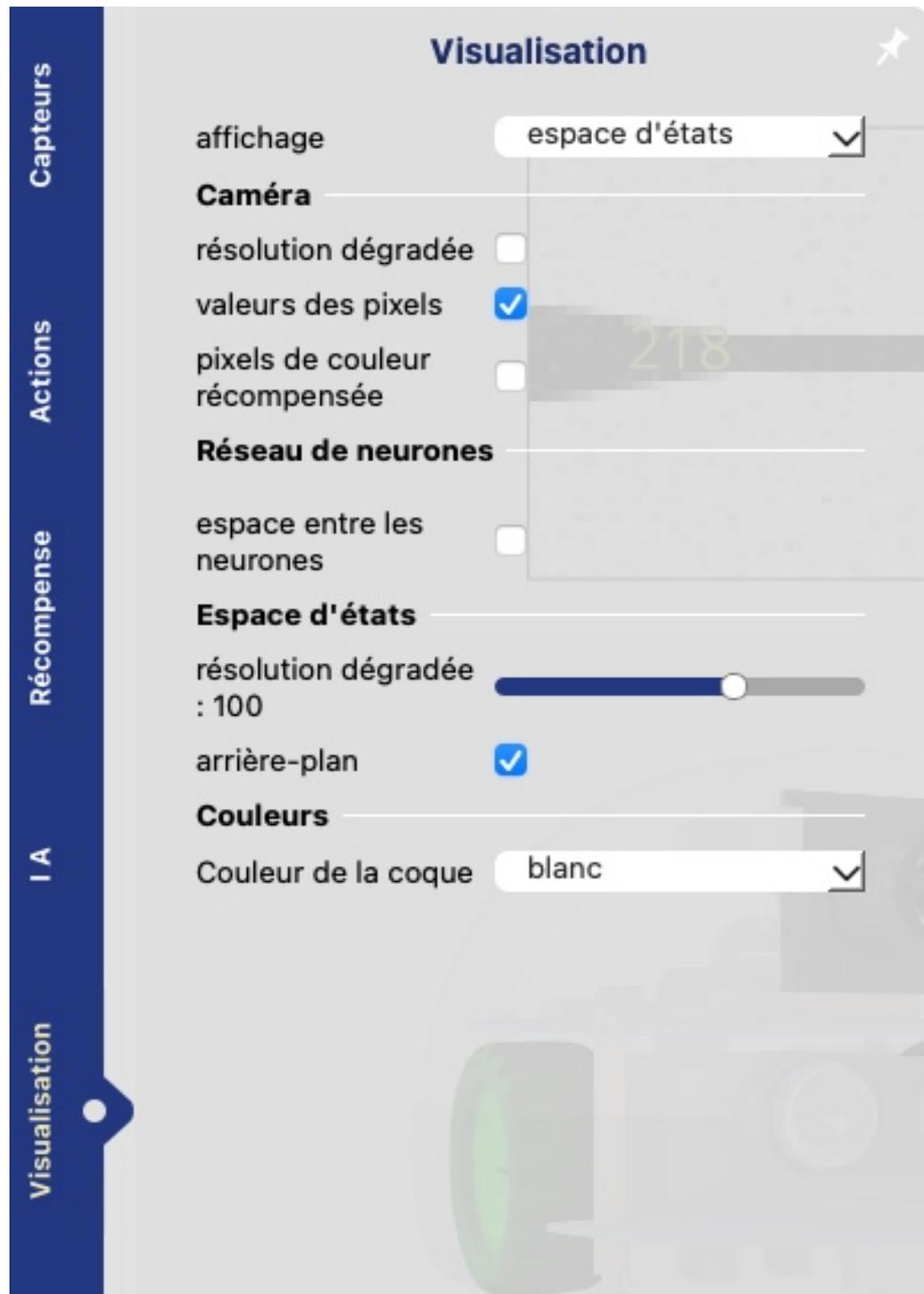
2. Dans l'onglet **actions** choisir la rotation sur place et la translation vers l'avant



3. Dans l'onglet **IA**
- type d'apprentissage : apprentissage supervisé
  - algorithme : K plus proche voisin



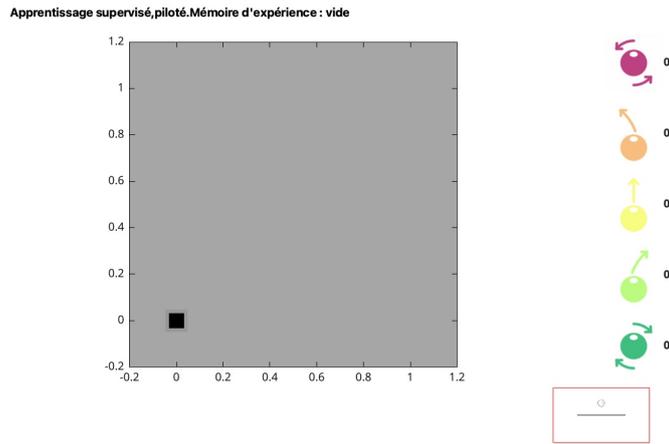
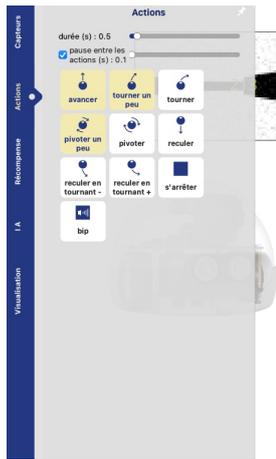
4. Dans l'onglet **Visualisation** cocher la case "arrière plan"



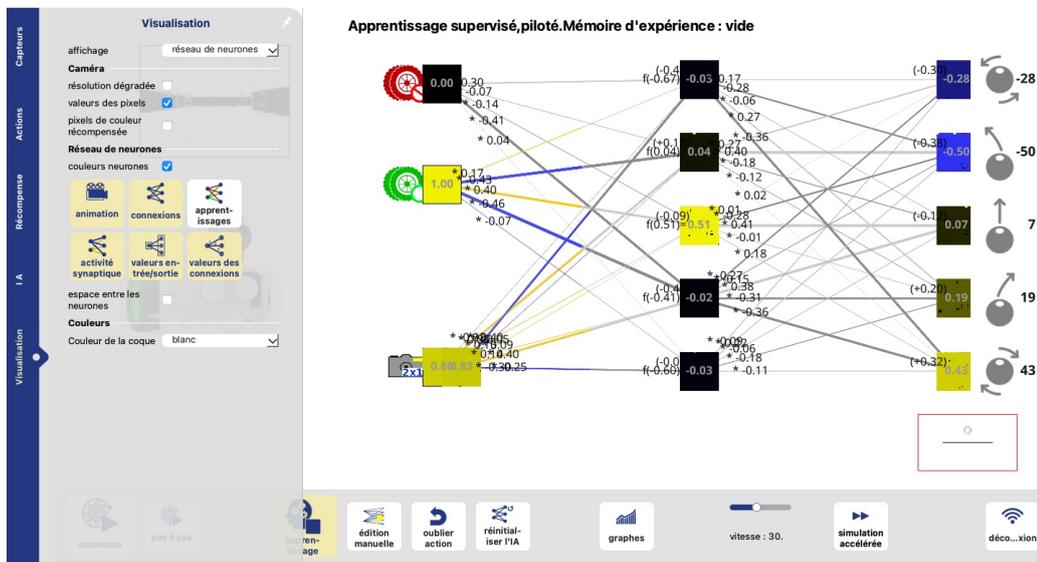
5. Vous pouvez alors activer la fonction apprentissage (en bas à gauche) puis diriger le robot pour lui faire apprendre à effectuer le circuit désiré.
6. Un fois l'apprentissage effectué, vous pouvez vérifier que votre apprentissage a bien fonctionné en faisant travailler le robot en autonomie en cliquant sur le mode **Autonome** en bas à gauche.

#### b) Apprentissage supervisé par réseau de neurones

1. Dans l'onglet **capteur** vous pouvez ajouter des données d'entrée : plus de pixels sur la caméra et ajouter le capteur de vitesse "bloqué/mouvement"
2. Dans l'onglet **actions** choisir la rotation sur place et la translation vers l'avant mais on peut également ajouter tourner un peu ce qui ajoute deux données de sortie.

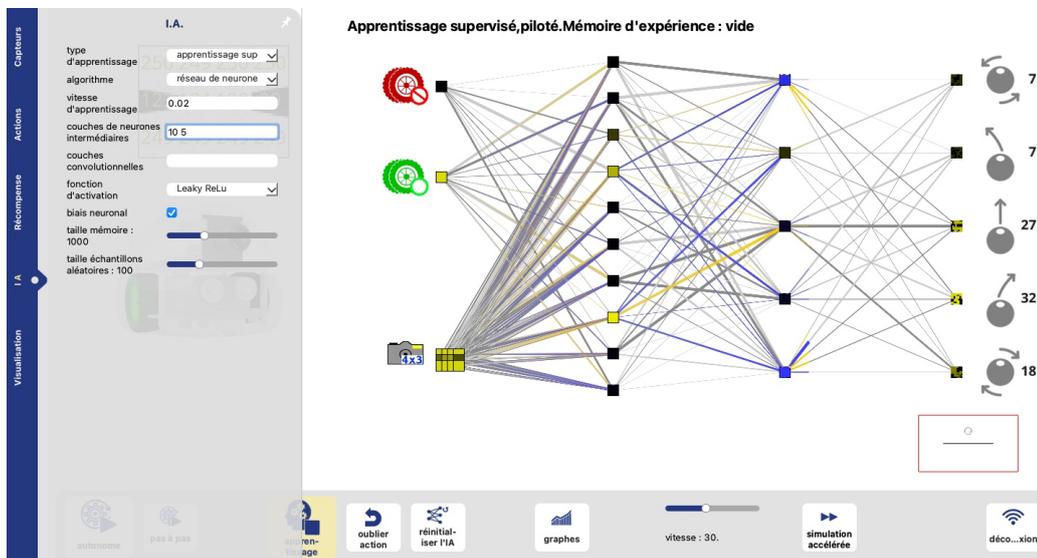


3. Dans l'onglet **Visualisation** choisir dans le menu déroulant affichage : réseau de neurones. On peut aussi ajouter les valeurs entrée/srtie et valeurs de connexion



4. Dans l'onglet **IA**

- type d'apprentissage : apprentissage supervisé
- algorithme : réseau de neurones
- Couches de neurones : séparés d'espace on précis par couche de neurones intermédiaires le nombre de neurones. Par exemple en mettant "10 5" on aura deux couches intermédiaires de respectivement 10 et 5 neurones.



5. Vous pouvez alors activer la fonction apprentissage (en bas à gauche) puis diriger le robot pour lui faire apprendre à effectuer le circuit désiré.

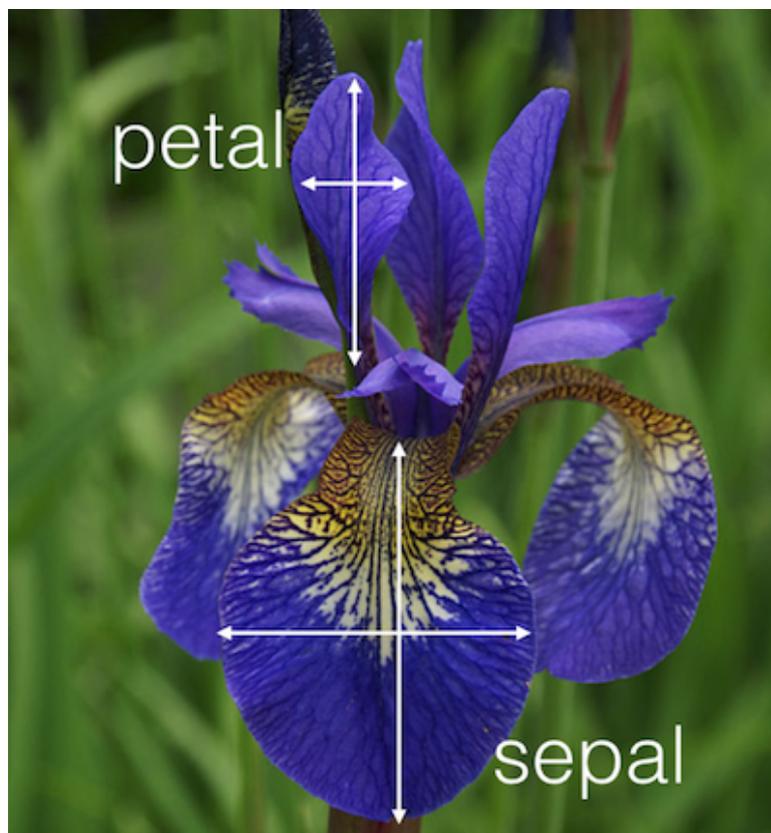
## Exercice 2 : Classification d'images d'IRIS

---

**Source :** Xavier Pessoles

### 1 Présentation

On s'intéresse ici à la base de donnée des Iris, présente par exemple dans `scikit-learn`, les données sont composées de 150 observations, chacune composée de 4 caractéristiques : longueur et largeur du sépale ainsi que longueur et largeur du pétale.



Pour la mise en oeuvre du TP on pourra utiliser un notebook développé sur Capytal. Avec le QR code ci-contre ou l'url ci-dessous, vous pourrez vous connecter via votre identifiant ENT.

<https://capytale2.ac-paris.fr/web/c/42ed-1417016>



## Corrigé

---

### Exercice 1 : Apprentissage automatique du robot Alphaï

---

#### 1 Apprentissage supervisé par recherche du plus proche voisin

**Q 1 : Mettre en oeuvre l'apprentissage du robot alphaï par l'apprentissage supervisé par recherche du plus court chemin en suivant la documentation en annexe.**

Exporter les données d'apprentissage dans le menu IA/sauver la mémoire d'expérience. On pourra exporter les données sous le nom *alphaï\_knn.buffer.npz*.

**Q 2 : Exploiter les données d'apprentissage à l'aide du script *traiter\_alphaï.py* qui doit être sauvegardé dans le même répertoire que *alphaï\_knn.buffer.npz*.**

**Q 3 : Faire varier le nombre de plus proches voisins et observer les résultats sur la matrice de confusion, le score d'apprentissage et la représentation graphique des plus proches voisins.**

**Q 4 : Mettre en oeuvre le robot en autonomie et vérifier ses performances.**

#### 2 Apprentissage supervisé par réseau de neurones

**Q 5 : Mettre en oeuvre l'apprentissage du robot alphaï par l'apprentissage supervisé réseau de neurones en suivant la documentation en annexe.**

**Q 6 : Faire varier le nombre de neurones, de couches et de données d'entrées.**

**Q 7 : Mettre en oeuvre le robot en autonomie et vérifier ses performances.**

### Exercice 2 : Classification d'images d'IRIS

---