

L’EDUCATION À L’INTELLIGENCE ARTIFICIELLE

Cours Découverte de l’IA

TP6 ÉvaluÉ

Cours Découverte de l’IA – 7 novembre 2024

Pour cette évaluation, il est demandé de compléter ce document, et de créer plusieurs documents python. À la fin de la séance, vous devrez envoyer un mail à simon.collet@learningrobots.ai en indiquant bien vos nom et prénom et en joignant l’ensemble des documents réponses. Merci d’écrire vos réponses en utilisant une couleur différente afin de faciliter la correction.

**Nom :**

**Prénom :**

## TP6 évalué

Les compétences évaluées dans ce TP sont :

* Compréhension générale des principes du machine learning et de l’apprentissage supervisé.
* Interprétation de l’espace d’états dans le logiciel AlphAI.
* Compréhension de l’algorithme des k plus proches voisins (KNN).
* Programmation du robot en python.

Dans ce TP, la tâche du robot sera simplement de suivre une ligne noire tracée au sol. Nous allons tester différentes configurations et différentes approches pour réaliser cette même tâche.

**Astuce : vous pouvez déplacer le robot simulé avec la souris en cliquant dessus.**

### Préparation : changer la couleur du robot simulé.

1. Démarrez l’ordinateur sous Windows. Installez le logiciel AlphAI si nécessaire (<https://learningrobots.ai/les-ressources/#download_software>, **version 1.9.0-beta18**) ou mettez à jour le logiciel installé.
2. Lancez le logiciel et connectez-vous au robot simulé.
3. Dans le menu **Fichier**, choisissez **Affichage des paramètres > Expert**, pour voir tous les paramètres de configuration nécessaire pour ce TP.
4. Dans l’onglet **Visualisation**, tout en bas, changez la **couleur de la coque** de votre robot pour correspondre aux instructions suivantes :

Rangées 1, 3, 5 : de la gauche vers la droite : rouge, vert, bleu, jaune, magenta, noir, orange, blanc.

Rangées 2, 4, 6 : de la gauche vers la droite : magenta, noir, orange, blanc, rouge, vert, bleu, jaune.

1. Faites une copie d’écran de la fenêtre du logiciel (utiliser le raccourci **Fn + Impr écran** ou **Windows + Maj + S**) et copiez-là ci-dessous.

### Une image contenant clipart, dessin humoristique  Description générée automatiquementPartie A : suivi de ligne avec un unique capteur.

**Configuration**

1. Rendez-vous dans le menu **Fichier > Configurations d’exemples**. Dans la fenêtre qui s’ouvre, sélectionnez sur la troisième ligne la configuration **Apprentissage supervisé – suivi de ligne** (en rouge).
2. Dans l’onglet **Capteurs**, vérifiez que seuls les capteurs à infrarouges (capteurs IR) sont actifs, et sélectionnez le mode **capteur central**. La couche d’entrée du réseau de neurones doit alors comporter un seul neurone, correspondant au capteur IR central.
3. Dans l’onglet **Actions**, sélectionnez uniquement les actions **avancer** et **pivoter un peu**. Au-dessus, diminuer la durée des actions jusqu’à la valeur minimale : 0,1 seconde.
4. Dans l’onglet IA, vérifiez que le **type d’apprentissage** est bien **apprentissage supervisé**. Pour le paramètre **algorithme**, choisissez **K plus proches voisins**. Le paramètre K doit valoir 1 pour l’instant.
5. Dans l’onglet Visualisation, tout en haut, réglez le paramètre **affichage** sur le mode **espace d’états**.
6. Dans la barre de contrôle en bas, fixez la **vitesse** du robot à sa valeur minimale : 15.
7. Le robot simulé peut être déplacé à la souris. Le point principal de l’espace d’états se déplace en conséquence. À quelle valeur correspond l’abscisse de ce point ? Comment cette valeur est-elle déterminée ?

**Premier apprentissage**

1. Lorsque l’on ajoute des données d’entraînement dans la mémoire du robot, des zones de couleur apparaissent dans l’arrière-plan. Puisqu’il n’y a qu’un seul capteur actif, le graphique est ici en une dimension, et les zones de couleur sont des bandes verticales. Réalisez un **premier apprentissage** selon le principe suivant : lorsque le robot voit la ligne, il va tout droit ; et lorsqu’il ne voit plus la ligne, il pivote vers la droite. Ajustez la valeur du paramètre k si nécessaire. Faites une capture d’écran de la fenêtre complète après apprentissage, et insérez-là ci-dessous.
2. Parmi les trois images ci-dessous, quel est l’apprentissage qui donnera le meilleur résultat ? Expliquez pourquoi en quelques mots.

|  |  |  |
| --- | --- | --- |
| Apprentissage A | Apprentissage B | Apprentissage C |
| Une image contenant texte, capture d’écran, Rectangle, ligne  Description générée automatiquement |  |  |

1. Que se passe-t-il si on place le robot sur la ligne en sens antihoraire (sens inverse des aiguilles d’une montre) ?
2. Que se passe-t-il si on place le robot loin de la ligne ?

**Second apprentissage**

1. **Réinitialisez l’apprentissage et réalisez un second apprentissage** selon le principe suivant : lorsque le robot voit la ligne, il pivote à droite ; et lorsqu’il ne voit plus la ligne, il va tout droit. Ajustez la valeur du paramètre k si nécessaire. Faites une capture d’écran de la fenêtre complète après apprentissage, et insérez-là ci-dessous.
2. Que se passe-t-il si on place le robot loin de la ligne à l’**intérieur** du tracé ?
3. Que se passe-t-il si on place le robot loin de la ligne à l’**extérieur** du tracé ?
4. Quel principe d’apprentissage vous semble le plus adapté entre le principe décrit à la question 8 et celui de la question 12 ? Justifiez en quelques mots.

**Question de programmation.**

1. Réinitialisez de nouveau l’apprentissage. Puis, dans l’onglet **IA**, activez le mode **code python** pour le paramètre **algorithme**. Sélectionnez **créer un nouveau fichier** et appelez-le **suivi\_ligne\_1\_capteur.py**. Ouvrez ce fichier avec un éditeur de texte, et modifiez la fonction **take\_decision**. Votre programme devra reproduire le comportement du principe d’apprentissage choisi à la question précédente et ainsi permettre au robot simulé de suivre la ligne au sol. N’oubliez pas de tester votre programme et de joindre le fichier ainsi créé au moment de rendre votre travail. Copiez ci-dessous une capture d’écran de la fenêtre du logiciel permettant de voir la coloration de l’espace d’états correspondant à l’exécution de votre programme.

### Une image contenant clipart, dessin humoristique  Description générée automatiquementPartie B : suivi de ligne avec 2 capteurs.

**Configuration**

1. Rendez-vous dans le menu **Fichier > Configurations d’exemples**. Dans la fenêtre qui s’ouvre, sélectionnez sur la troisième ligne la configuration **Apprentissage supervisé – suivi de ligne** (en rouge).
2. Dans l’onglet **Capteurs**, vérifiez que seuls les capteurs à infrarouges (capteurs IR) sont actifs, et sélectionnez le mode **capteurs intérieurs**. La couche d’entrée du réseau de neurones doit alors comporter deux neurones, correspondant aux capteurs IR droite et gauche.
3. Dans l’onglet **Actions**, sélectionnez uniquement les actions **avancer**, **tourner**, et **reculer**. Au-dessus, diminuer la durée des actions jusqu’à la valeur minimale : 0,1 seconde.
4. Dans l’onglet IA, vérifiez que le **type d’apprentissage** est bien **apprentissage supervisé**. Pour le paramètre **algorithme**, choisissez **K plus proches voisins**. Le paramètre K doit valoir 1 pour le moment.
5. Dans l’onglet Visualisation, tout en haut, réglez le paramètre **affichage** sur le mode **espace d’états**.
6. Dans la barre de contrôle en bas, fixez la **vitesse** du robot à sa valeur minimale : 15.

**Premier apprentissage**

1. Réalisez un premier apprentissage en utilisant **uniquement les actions tourner à droite et tourner à gauche**. Assurez-vous que cet apprentissage permet au robot de suivre la ligne dans le sens horaire (sens des aiguilles d’une montre). Ajustez la valeur du paramètre k si nécessaire. Insérez ci-dessous une capture d’écran de la fenêtre du logiciel permettant de voir l’espace d’états correspondant à cet apprentissage.
2. Que se passe-t-il si on place le robot sur la ligne en sens antihoraire (sens inverse des aiguilles d’une montre) ?

**Questions sur l’espace d’états**

**Voici ci-dessous une représentation de l’espace d’états à 2 dimensions qui correspond à la configuration choisie avec 2 capteurs infrarouges. Dans cet espace ont été représentés 5 points appelés A, B, C, D et E, ainsi que des séparations entre les zones d’influence de ces points selon l’algorithme des k plus proches voisins. Les questions de cette partie portent sur ce graphique.**

****

1. Trouvez le point correspondant à chacune des situations représentées ci-dessous et complétez la colonne **point correspondant**.

|  |  |  |  |
| --- | --- | --- | --- |
| **Schéma** | **Description** | **Point correspondant** | **Meilleure action** |
|   | Loin de la ligne. |  |  |
|   | Ligne du côté gauche. |  |  |
|   | Ligne au milieu. |  |  |
|  | Ligne du côté droit. |  |  |
|   | Perpendiculaire à la ligne. |  |  |

1. L’objectif du robot est de suivre la ligne du mieux possible, dans le sens horaire (sens des aiguilles d’une montre). Les actions disponibles pour le robot sont : avancer, reculer, tourner à droite, tourer à gauche. Choisissez la meilleure action à réaliser dans chacune des situations ci-dessus et complétez la colonne **meilleure action**.

**Deuxième apprentissage**

1. Réinitialisez l’apprentissage, puis réalisez un second apprentissage en utilisant les 4 actions disponibles : avancer, reculer, tourner à droite ou à gauche. Vous pouvez vous inspirer du tableau complété ci-dessus. Assurez-vous que cet apprentissage permet au robot de suivre la ligne. Ajustez la valeur du paramètre k si nécessaire. Insérez ci-dessous une capture d’écran de la fenêtre du logiciel permettant de voir l’espace d’états correspondant à cet apprentissage.
2. Que se passe-t-il si on place le robot sur la ligne en sens antihoraire (sens inverse des aiguilles d’une montre) ?

**Question de programmation.**

1. Réinitialisez de nouveau l’apprentissage. Puis, dans l’onglet **IA**, activez le mode **code python** pour le paramètre **algorithme**. Sélectionnez **créer un nouveau fichier** et appelez-le **suivi\_ligne\_2\_capteurs.py**. Ouvrez ce fichier avec un éditeur de texte, et modifiez la fonction **take\_decision**. Votre programme devra permettre au robot simulé de suivre la ligne au sol **dans les deux sens**. Vous êtes libres d’utiliser les 4 actions suivantes : avancer, reculer, tourner à droite et tourner à gauche. N’oubliez pas de tester votre programme et de joindre le fichier ainsi créé au moment de rendre votre travail. Copiez ci-dessous une capture d’écran de la fenêtre du logiciel permettant de voir la coloration de l’espace d’états correspondant à l’exécution de votre programme.

### Partie C : questions de programmation.

1. **Question de cours.** Créez un fichier python appelé **argmax.py**. Définissez une fonction **argmax** prenant en paramètre une liste de nombres et renvoyant la valeur de **l’indice du plus grand élément** de cette liste. Dans le cas où le plus grand élément est présent plusieurs fois, on pourra renvoyer l’indice de n’importe laquelle de ses positions. Inclure le fichier **argmax.py** dans votre e-mail de dépôt.
2. **Question plus difficile.** Dans l’onglet **Capteurs**, activez les **capteurs IR** avec le mode **5 capteurs**. Dans l’onglet IA, activez le mode **code python**, et créez un nouveau fichier appelé **suivi\_ligne\_5\_capteurs.py**. Ouvrez ce fichier avec un éditeur de texte, et modifiez la fonction **take\_decision**. Votre programme devra permettre au robot simulé de suivre la ligne au sol **dans les deux sens**. Vous êtes libres d’utiliser les **actions** que vous souhaitez. N’oubliez pas de tester votre programme et de joindre le fichier ainsi créé au moment de rendre votre travail.

### Rendu de votre travail par e-mail.

Pour déposer votre travail, merci d’envoyer un e-mail à simon.collet@learningrobots.ai. **Merci d’indiquer vos nom et prénom dans l’objet du message**. L’heure d’envoi du message sera contrôlée et les retards seront pénalisés.

Merci d’envoyer en pièce-jointe un **dossier compressé** contenant les documents suivants :

* Ce document complété avec vos réponses au format docx.
* Le fichier **suivi\_ligne\_1\_capteur.py**.
* Le fichier **suivi\_ligne\_2\_capteurs.py**.
* Le fichier **argmax.py**.
* Le fichier **suivi\_ligne\_5\_capteurs.py**.