

L’EDUCATION À L’INTELLIGENCE ARTIFICIELLE

Cours Découverte de l’IA

TP6 ÉvaluÉ

Cours Découverte de l’IA – 12 décembre 2024

Pour cette évaluation, il est demandé de compléter ce document avec vos réponses. À la fin de la séance, vous devrez déposer ce document complété sur la plateforme e-campus. Merci d’écrire vos réponses en utilisant une couleur différente afin de faciliter la correction.

**Nom :**

**Prénom :**

## TP11 évalué : réseaux de neurones et Q-learning

Les compétences évaluées dans ce TP sont :

* Compréhension des réseaux de neurones : connexions, biais, entraînement, etc.
* Compréhension des principes de l’apprentissage par renforcement en général, et de l’algorithme du Q-learning en particulier.
* Programmation d’une fonction de récompense 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. Un clic droit permet de le faire pivoter.**

### 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-beta23**) 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, logo, symbole  Description générée automatiquementPartie A : Suivi de ligne avec un réseau de neurones édité manuellement.

**Configuration**

1. Rendez-vous dans le menu **Fichier > Configurations d’exemples**. Dans la fenêtre qui s’ouvre, sélectionnez sur la deuxième ligne (**Édition manuelle**) la configuration **Suivi de ligne** (en jaune).
2. Dans l’onglet **Actions**, diminuez la durée des actions jusqu’à la valeur minimale : 0,1 seconde.
3. Dans la barre de contrôle en bas, fixez la **vitesse** du robot à sa valeur minimale : 15.

**Questions**

1. Voici ci-contre une représentation de la couche d’entrée du réseau de neurones que nous allons utiliser dans cette partie. Surlignez ci-dessous les affirmations vraies et barrez les affirmations fausses.
	1. Chaque neurone d’entrée correspond à un des 5 capteurs infra-rouges du robot.
	2. Dans cette configuration, les neurones d’entrée peuvent prendre n’importe quelle valeur entre 0 et 1.
	3. Dans cette configuration, il y a au maximum un seul neurone actif à la fois.
	4. Le capteur le plus à gauche correspond au neurone A.
	5. Le neurone A correspond au capteur le plus à droite.
	6. Le capteur central correspond au neurone C.
	7. Il est impossible de savoir à quel neurone correspond chaque capteur car cela est déterminé aléatoirement à l’initialisation du réseau.
2. Pour chacun des neurones de la couche d’entrée (A, B, C, D et E), sélectionnez l’action la plus adaptée pour le robot lorsque ce neurone est actif. Relier ce neurone à cette action en cliquant sur la connexion correspondante. Puis, prenez une capture d’écran et copiez-là ci-dessous.
3. En conservant les connexions établies à la question précédente, effectuez les tests ci-dessous. Pour chaque test, décrivez ce qu’il se passe en quelques mots.
	1. Le robot est placé sur la ligne dans le sens horaire, puis on active le mode autonome.
	2. Le robot est placé sur la ligne dans le sens antihoraire, puis on active le mode autonome.
	3. Le robot est placé loin de la ligne, puis on active le mode autonome.

### Partie B : Questions théoriques sur un réseau de neurones

**Cette partie peut être réalisée sans utiliser le logiciel AlphAI, mais vous pouvez aussi l’utiliser pour vous aider à répondre aux questions.**

**L’objectif est de paramétrer un réseau de neurone assez simple pour une tâche de suivi de ligne. Pour cela, nous allons utiliser le réseau de neurones représenté ci-dessous où chaque neurone est représenté par une lettre et chaque connexion par un nombre.**

**Pour afficher et modifier les paramètre du réseau, rendez-vous dans l’onglet Visualisation et assurez-vous que le bouton valeur des connexions est activé et que l’option édition manuelle avancée est également active.**

****

1. Les deux capteurs utilisés sont les capteurs à infra-rouges internes gauche et droit. Surlignez la bonne réponse ci-dessous.
	1. Le capteur de gauche correspond au neurone A et le capteur de droite au neurone B.
	2. Le capteur de droite correspond au neurone A et le capteur de gauche au neurone B.
	3. La correspondance entre les neurones et les capteurs est aléatoire.

**Dans cette configuration, on suppose que les neurones d’entrée peuvent prendre n’importe quelle valeur entre 0 et 1. On souhaite que le comportement du robot remplisse 3 conditions :**

* 1. **Il tourne à gauche lorsque le capteur de gauche possède une valeur plus élevée que le capteur de droite.**
	2. **Il tourne à droite lorsque le capteur de droite possède une valeur plus élevée que le capteur de gauche.**
	3. **Il recule lorsque les deux capteurs ont des valeurs faibles (il ne voit pas la ligne).**
1. Dans le tableau ci-dessous, proposez des valeurs **entières** pour les poids des connexions permettant de remplir les conditions **a** et **b**.

|  |  |
| --- | --- |
| **Connexion** | **Poids** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |

1. Dans le tableau ci-dessous, proposez des valeurs de biais **arrondis à une décimale** pour les neurones de sortie qui permettent de satisfaire la condition **c**. **Attention : les conditions a et b doivent toujours être satisfaites !**

|  |  |
| --- | --- |
| **Neurone** | **Biais** |
| C |  |
| D |  |
| E |  |

1. Parmi les 4 espaces d’états ci-dessous, indiquez lequel satisfait ces trois conditions (a, b et c).

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | D |
|  |  |  |  |

### Partie C : Suivi de ligne en apprentissage supervisé avec un réseau de neurones (5 capteurs, 6 actions).

**Configuration**

1. Rendez-vous dans le menu **Fichier > Configurations d’exemples**. Dans la fenêtre qui s’ouvre, sélectionnez sur la troisième ligne (**Apprentissage supervisé**) la configuration **Suivi de ligne** (en rouge).
2. Dans l’onglet **Actions**, diminuez la durée des actions jusqu’à la valeur minimale : 0,1 seconde.
3. Dans la barre de contrôle en bas, fixez la **vitesse** du robot à sa valeur minimale : 15.

**Premier entraînement**

1. Réalisez un entraînement avec ces paramètres. Assurez-vous que le robot est capable de suivre la ligne dans les deux sens. Faites une capture d’écran du réseau où l’on peut bien voir les valeurs de chaque connexion et copiez-là ci-dessous.

**Second entraînement**

1. Réinitialisez l’apprentissage, puis dans l’onglet **Visualisation**, tout en haut, réglez le paramètre **affichage** sur le mode ‘‘**données d’entraînement**’’. Réalisez un apprentissage en utilisant uniquement 6 points de données bien choisis. Assurez-vous que le robot est capable de suivre la ligne dans les deux sens. Faites une capture d’écran où l’on peut bien voir les 6 points de données utilisés pour l’apprentissage et copiez-là ci-dessous.

### Partie D : Apprentissage par renforcement du suivi de ligne avec l’algorithme du Q-learning.

**Configuration**

1. Rendez-vous dans le menu **Fichier > Configurations d’exemples**. Dans la fenêtre qui s’ouvre, sélectionnez sur la troisième ligne (**Apprentissage supervisé**) la configuration **Suivi de ligne** (en rouge).
2. Dans l’onglet **Capteurs**, assurez-vous que seuls les capteurs infrarouges **internes** sont activés et que le mode binaire (*binary mode*) est activé. Le réseau de neurones doit comporter 2 neurones d’entrée.
3. Dans l’onglet **Actions**, diminuez la durée des actions jusqu’à la valeur minimale : 0,1 seconde, et conservez uniquement les actions **tourner** et **reculer**. Le réseau de neurones doit comporter 3 neurones de sortie.
4. Dans l’onglet **Récompense**, sélectionnez le mode de récompense ‘‘**Suivi de ligne**’’.
5. Dans l’onglet **IA**, sélectionnez type d’apprentissage = apprentissage par renforcement, algorithme = Q-learning simple, vitesse d’apprentissage = 0.2, gamma = 0.8, exploration = 0.2 et mémoire d’expérience = désactivé.
6. Dans l’onglet **Visualisation**, vérifiez que les **valeurs des connexions** sont visibles.
7. Dans la barre de contrôle en bas, fixez la **vitesse** du robot à sa valeur minimale : 15, activez l’**apprentissage** avec le bouton correspondant, et **réinitialisez l’IA**.

**Questions**

1. Réalisez des tests avec le robot simulé et complétez le tableau ci-dessous avec la valeur de la récompense correspondant à chacune des situations décrites par le couple (état +

action).

|  |  |  |
| --- | --- | --- |
| **État** | **Action** | **Récompense** |
| Robot sur la ligne | Tourner |  |
| Robot sur la ligne | Reculer |  |
| Robot loin de la ligne | Tourner |  |
| Robot loin de la ligne | Reculer |  |

1. Voici ci-dessous une représentation du modèle qui va être entraîné par l’algorithme du Q-learning. Répondez aux questions ci-dessous à propos de ce modèle.
	1. Combien y a-t-il d’états possibles (*states*) dans ce modèle ? Décrivez chacun de ces états en quelques mots.
	2. Combien y a-t-il d’actions possibles dans ce modèle ?
	3. Combien de lignes comporte la table de Q-learning pour ce modèle ?
	4. Combien de colonnes comporte la table de Q-learning pour ce modèle ?



1. Dans l’onglet IA, sélectionnez **algorithme = code python**, et créez un nouveau fichier appelé **q\_learning.py**. Ouvrez ce fichier avec un éditeur de texte. Complétez la fonction **init** selon le modèle ci-dessous. Cette fonction doit initialiser la variable **q\_table** avec le bon nombre de lignes et de colonnes. Chaque valeur de la **q\_table** doit être une valeur décimale aléatoire comprise entre -1 et 1. N’oubliez pas de tester votre fonction, puis remplacez l’image ci-dessous par une capture d’écran de votre code.



1. Complétez maintenant la fonction **learn** selon le modèle ci-dessous de façon à appliquer les équations du Q-learning. La documentation de la fonction (texte entre triples guillemets) rappelle les valeurs de ses différents paramètres. N’oubliez pas de tester votre fonction, puis remplacez l’image ci-dessous par une capture d’écran de votre code.



On rappelle les équations du Q-learning :



**Récompense personnalisée**

1. On souhaite maintenant concevoir une fonction de récompense pour un robot suiveur de ligne à **5 capteurs**. Cette récompense devra satisfaire les conditions suivantes :
	1. Lorsque la ligne est détectée par au moins un capteur, la récompense est proportionnelle à la vitesse du robot (donc si le robot recule, la récompense est négative ; et s’il avance, elle est positive).
	2. La récompense est plus élevée lorsque la ligne est détectée par le capteur central, et plus faible lorsque le robot détecte la ligne avec ses capteurs latéraux (pour inciter le robot à bien se centrer sur la ligne).
	3. Lorsque la ligne n’est pas détectée, la récompense est négative (mais il faut trouver un moyen d’inciter le robot à retrouver la ligne malgré cela).

Avec un éditeur de texte, recopiez et compléter la fonction ci-dessous, puis remplacez l’image ci-dessous par une capture d’écran de votre solution.



### Rendu de votre travail sur e-campus.

Pour déposer votre travail, merci de vous rendre sur la page du cours et de déposer vos documents sur la plateforme prévue à cet effet.

Toutes vos réponses doivent être incluses dans ce document.

N’oubliez pas de sauvegarder convenablement votre travail.

Vérifiez que vous déposez bien le bon fichier et pas une copie non complétée.