alphai

L'ÉDUCATION À L'INTELLIGENCE ARTIFICIELLE

Apprentissage par renforcement

Algorithme Q-learning

Corrigé

Introduction

Rappel : en apprentissage supervisé, le réseau de neurone apprend à accomplir une tâche en se basant sur des **données étiquetées** fournies par l'utilisateur. Dans le cas des robots AlphAI, ces données sont transmises au robot lorsqu'on le pilote. Lorsque la tâche à accomplir est complexe, la quantité de données nécessaire peut être très importante, ce qui est une limitation dans certaines applications.

À l'inverse, l'apprentissage par renforcement est une méthode d'apprentissage machine (*machine learning*) qui ne nécessite pas de données étiquetées. On fournit simplement à l'IA une **fonction de récompense** qui calcule un score, positif ou négatif, en fonction des actions choisies par le réseau. L'IA va ensuite essayer des actions plus ou moins au hasard, et mettre à jour les valeurs des connexions neuronales pour essayer de maximiser son score.

Cette technique est notamment utilisée pour entraîner des IA à jouer à des jeux comme les échecs, le jeu de Go ou divers jeux vidéos. Puisque l'apprentissage ne se base pas (ou pas uniquement) sur des données de jeu entre humains, il est possible d'observer des stratégies inattendues, ou qui dépassent le niveau de jeu des meilleurs joueurs humains.

Au cours de cette activité, nous allons étudier la méthode d'apprentissage par renforcement appelée **Q-learning**. Grâce à cette méthode, nous allons entraîner un robot AlphAI à se déplacer dans une arène sans rester bloqué contre les parois. Cette tâche simple nous permettra de bien comprendre les mécanismes qui interviennent au niveau du réseau de neurones.

1 Récompense et édition manuelle

1.1 Configuration du logiciel

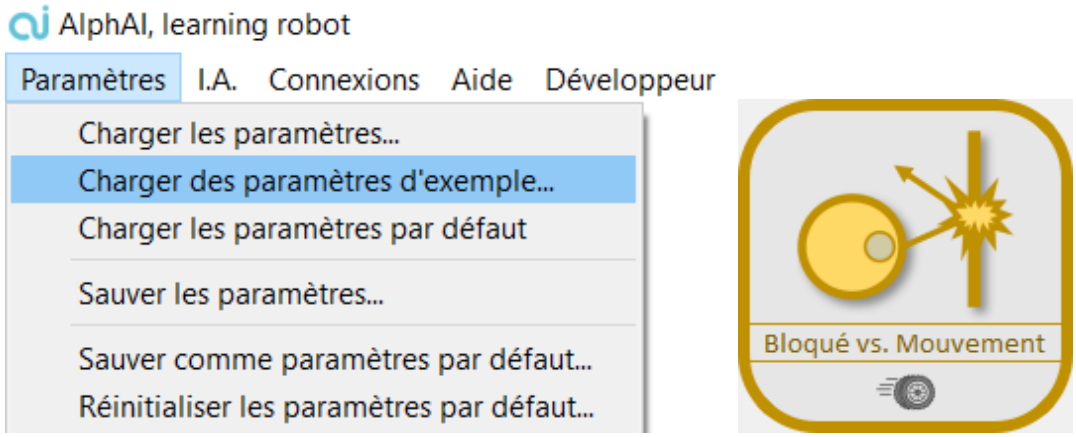

Figure 1 – Charger la configuration *Édition manuelle - Bloqué vs. Mouvement.*

1. Lancez le logiciel AlphAI et connectez-vous à un robot.

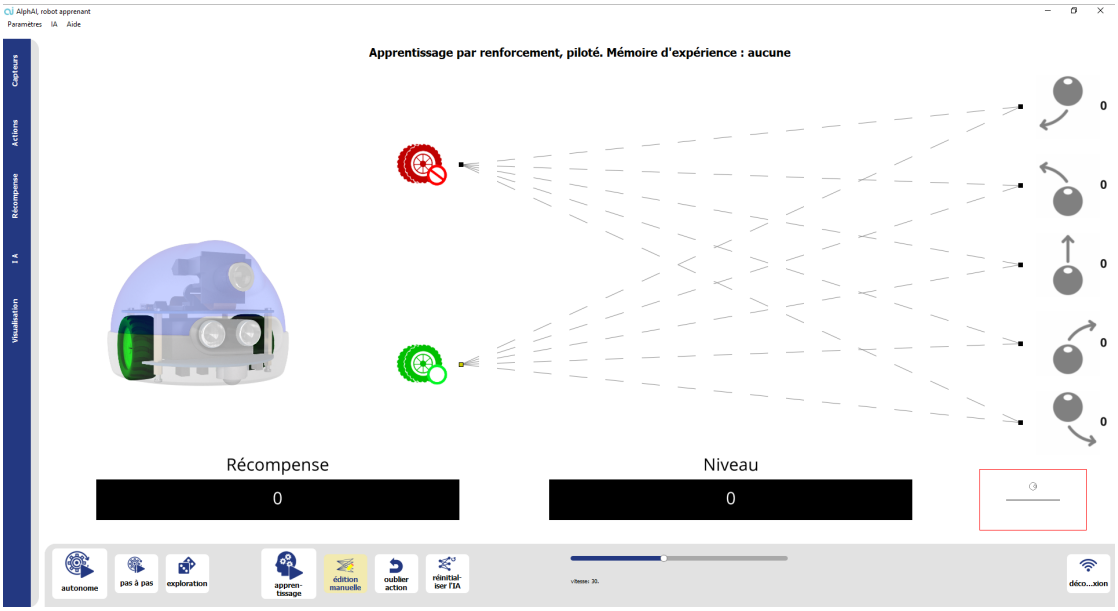

FIGURE 2 – Écran de démarrage pour le scénario *Édition manuelle - Bloqué vs. Mouvement*.

2. Placez le robot dans une petite arène.
3. Chargez les paramètres d'exemple correspondant au scénario **Édition manuelle - Bloqué vs. Mouvement** (voir figure 1).
4. Si la configuration est effectuée correctement, votre écran doit ressembler à la figure **??**.

1.2 ÉDITION MANUELLE DU RÉSEAU

1. Pilotez votre robot à l'aide des actions disponibles et observez la jauge **Récompense**. La valeur de la récompense peut prendre 3 valeurs différentes. *Quelles sont ces valeurs et à quelles situations (état des capteurs et action choisie) correspondent-elles ?* La récompense vaut 100 lorsque le robot avance en ligne droite, -50 lorsqu'il est bloqué contre un obstacle ou lorsqu'il recule, et environ 35 lorsqu'il avance en virage.
2. Les connexions en pointillés ont un poids de 0. En cliquant dessus, vous pouvez leur attribuer un poids de 1, et réciproquement. Dans ce mode, une seule connexion peut être active pour chaque neurone d'entrée. *Trouvez une configuration des connexions qui permet de maximiser la moyenne des récompenses reçues.* Relier le neurone *bloqué* à une action *reculer en virage* et le neurone *pas bloqué* à l'action *tout droit*. Voir figure 3.
3. Activez le mode autonome et observez la jauge **Niveau**. Elle correspond à la valeur moyenne de la récompense sur la dernière minute. *Notez la valeur du niveau moyen atteint après une minute de mode autonome.* Le niveau exact

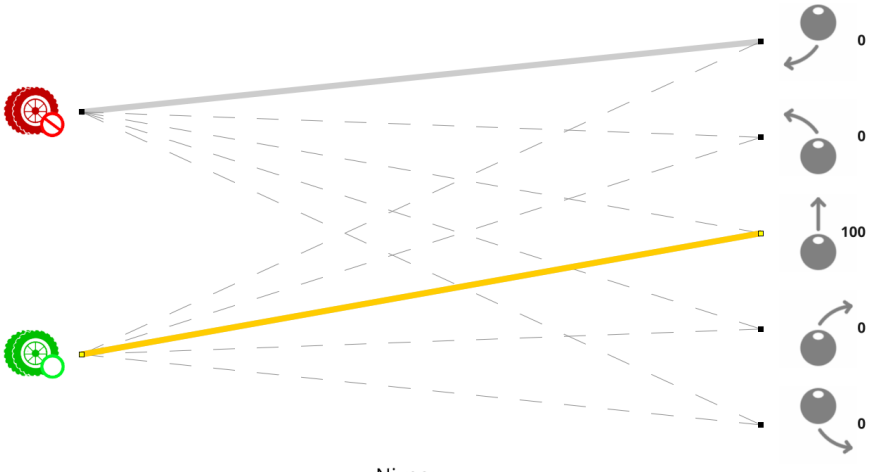

FIGURE 3 – Une solution pour maximiser la moyenne des récompenses reçues.

dépend de la taille et la forme de l'arène, ainsi que la vitesse du robot, mais devrait généralement être supérieur à 50.

4. Astuce : vous pouvez adapter la valeur de la vitesse du robot en fonction de la taille de votre arène. *Quelle valeur de vitesse permet d'atteindre le meilleur niveau ? Pourquoi ?* Les vitesses plus faibles permettent généralement d'atteindre un meilleur niveau, car le robot passe une plus grande proportion du temps à avancer en ligne droite.

2 APPRENTISSAGE PAR RENFORCEMENT

2.1 CONFIGURATION DU LOGICIEL

1. Activez le mode **apprentissage** en cliquant sur le bouton correspondant, et **réinitialisez l'IA**.
2. Dans le menu **Paramètres**, sélectionnez **Affichage des paramètres** → **Avancé**.
3. Dans l'onglet **Visualisation**, activez l'affichage des **valeurs des connexions**.
4. Comparez votre écran à la figure 4.
5. Placez le robot au centre de l'arène afin qu'il puisse tourner en rond en évitant les murs.

2.2 APPRENTISSAGE PAS À PAS

1. Notez les valeurs des connexions et des neurones. *Pouvez-vous prédire la prochaine action effectuée par le robot ?* La prochaine action effectuée par le robot correspond au neurone de sortie dont le niveau d'activation est le plus élevé.

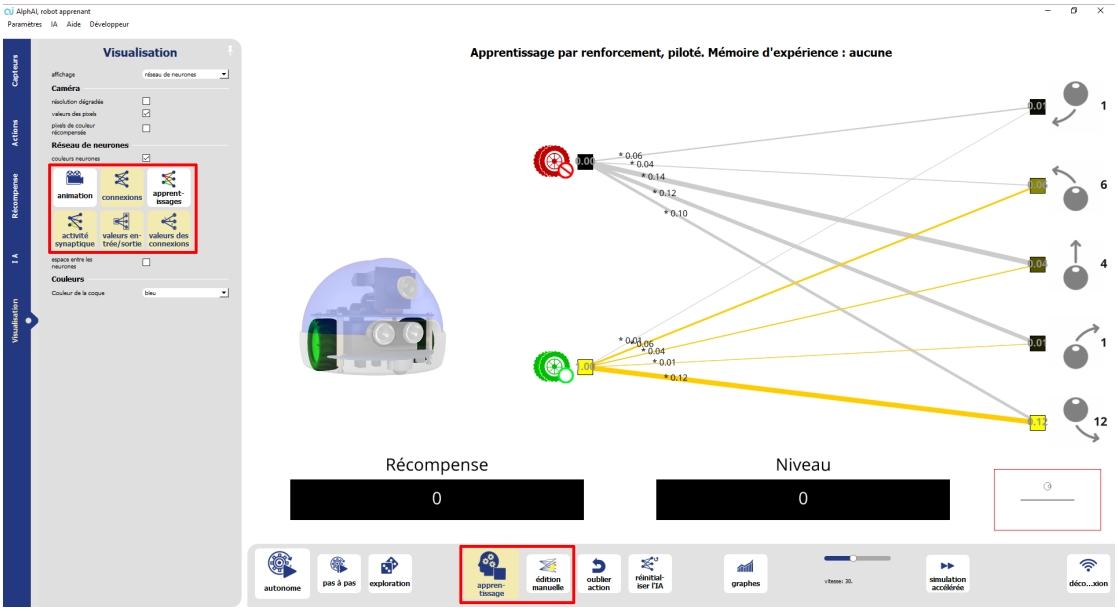

FIGURE 4 – Écran de démarrage pour l'apprentissage par renforcement.

2. Cliquez une fois sur le bouton **pas à pas** pour effectuer une seule action autonome. *Quelle connexion a vu son poids modifié ? A-t-il augmenté ou diminué ? Faire le lien avec la valeur de la récompense reçue.* C'est la connexion *active*, reliant le neurone d'entrée qui est actif au neurone de sortie ayant la valeur la plus élevée, qui est modifiée. Sa valeur augmente lorsque la récompense est positive et diminue lorsqu'elle est négative.
3. Appuyez de nouveau sur **pas à pas** jusqu'à ce que l'action choisie par le robot change. *Aviez-vous anticipé ce changement ? Quelle est la nouvelle action choisie par le robot ?* Normalement, le robot devrait se mettre à tourner en rond (sauf s'il entre en contact avec un mur).
4. Activez le mode **autonome** et observez les valeurs des connexions. *Expliquez pourquoi le robot choisit toujours la même action. Vers quelle valeur semble converger la connexion correspondant à cette action ?* Le robot choisit systématiquement l'action qui correspond au neurone de sortie le plus activé (avancer en virage). La valeur de la connexion correspondante devrait converger vers la valeur de la récompense (environ 35). Cette valeur étant toujours la plus élevée, le robot ne peut pas sélectionner d'autre action.
5. *Le comportement appris par le robot est-il optimal ? Quel niveau atteint-il ?* Le comportement n'est pas optimal. Le robot atteint un niveau d'environ 35, alors qu'il est possible de faire mieux, comme c'était le cas dans la partie précédente.
6. Activez maintenant le mode **exploration** en cliquant sur le bouton correspondant. *Décrire l'effet de cette option sur l'apprentissage. Quel niveau le robot atteint-il après quelques minutes ?* Le robot se met à choisir certaines actions au hasard. Ces actions sont indiquées en bleu dans l'interface.

2.3 Importance de l'exploration

À retenir :

En **apprentissage par renforcement**, le robot est autonome dès le début de son apprentissage, et doit donc choisir tout seul les actions à effectuer. Il se base ensuite sur la **récompense** reçue afin de mettre à jour les valeurs de ses connexions pour améliorer son score.

Pendant l'apprentissage, il y a deux façons de sélectionner la prochaine action à réaliser : soit le robot choisit l'action correspondant au neurone de sortie le plus activé, qui a théoriquement la meilleure chance de donner la récompense la plus élevée ; soit il choisit une action au hasard. Choisir toujours la *meilleure* action permet de maximiser les récompenses à court terme mais ne permet pas de trouver de nouvelles stratégies pour augmenter son niveau à long terme. Ce compromis porte le nom de **dilemme exploration / exploitation**.

Dans **AlphAI**, le robot choisit par défaut la meilleure action, mais en activant le mode **exploration**, on peut choisir la proportion de ses actions qui sont choisies au hasard. Le paramètre **fréquence d'exploration** apparaît dans l'onglet **IA**. Pour un apprentissage rapide, il faut une fréquence d'exploration élevée en début d'apprentissage, puis la réduire petit à petit, une fois que toutes les actions ont été explorées. Une fois l'apprentissage terminé, on peut désactiver le mode **exploration**.

3 Q-learning

Dans cette partie, nous allons étudier l'algorithme utilisé pour calculer la nouvelle valeur d'une connexion après chaque action. Cet algorithme s'appelle Q-learning et consiste principalement en l'application d'une formule. Cette formule comporte deux paramètres qui prennent des valeurs entre 0 et 1 :

- La **vitesse d'apprentissage**, notée α (alpha).
- Le **facteur d'actualisation**, noté γ (gamma).

On notera également :

- a l'action effectuée à l'instant t.
- r la récompense reçue après cette action.
- s l'état dans lequel se trouve le robot juste avant cette action. Dans le cadre de ce TP, cela correspond à l'état des capteurs, c'est-à-dire les niveaux d'activation des neurones d'entrée.
- $Q_t(s, a)$ la valeur à l'instant t (avant l'action a) de la connexion reliant l'état s à l'action a.

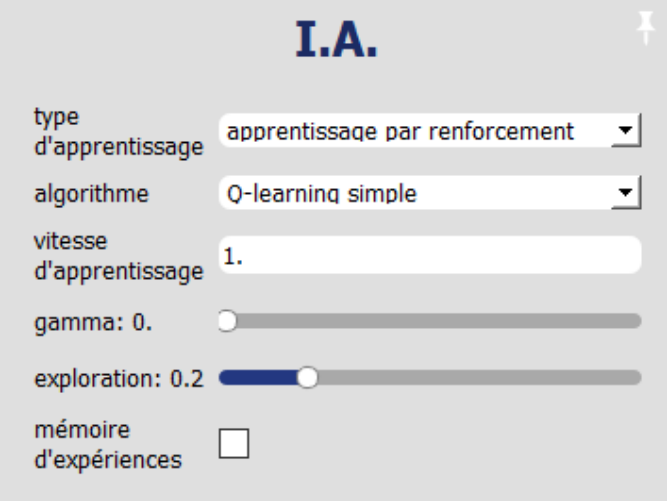

FIGURE 5 – Valeurs des paramètres d'IA pour le mode Q-learning.

3.1 Mise à jour des connexions

1. **Réinitialisez l'IA** en cliquant sur le bouton correspondant. Puis, dans l'onglet **IA**, réglez la **vitesse d'apprentissage** à **1**, le **facteur d'actualisation** (gamma) à **0** et la **fréquence d'exploration** à **0.2**. Lancez un apprentissage pas à pas avec ces paramètres et observez les valeurs des connexions. *Déterminez la formule de mise à jour des connexions.* **Indice** : dans cette configuration, la nouvelle valeur de la connexion $Q_{t+1}(s,a)$ dépend uniquement de la valeur de la récompense r. Dans ce cas, la formule de mise à jour est simplement : $Q_{t+1}(s,a) = r$.
2. *L'apprentissage est-il efficace dans cette configuration ? Pourquoi ?* L'apprentissage n'est pas efficace, car les valeurs des connexions alternent entre les différentes valeurs de récompense, qui dépendent aussi du contexte dans lequel les actions ont été choisies.
3. Réinitialisez de nouveau l'IA, et réglez la vitesse d'apprentissage à **0.5** en conservant le paramètre gamma à **0**. Lancez un nouvel apprentissage pas à pas avec ces paramètres, et observez les valeurs des connexions. *Déterminez la formule de mise à jour des connexions.* **Indice** : dans cette configuration, la nouvelle valeur de la connexion $Q_{t+1}(s,a)$ dépend de la valeur de la récompense r et de la précédente valeur de la connexion $Q_t(s,a)$. Dans ce cas particulier, la formule de mise à jour est : $Q_{t+1}(s,a) = \frac{Q_t(s,a)+r}{2}$.
4. Essayez de nouveau avec une valeur de vitesse d'apprentissage $\alpha = 0.1$ afin de généraliser la formule précédente à toutes les valeurs possibles de α. On rappelle que α est une valeur comprise entre 0 et 1. **Indice** : votre formule doit maintenant faire intervenir la vitesse d'apprentissage α. Dans ce cas plus général, la formule de mise à jour est : $Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha r$.
5. *Quelle valeur de la vitesse d'apprentissage donne les meilleurs résultats ? Le robot parvient-il à trouver la meilleure action dans toutes les circonstances ?*

Les valeurs plutôt faibles donnent en général de meilleurs résultats : un apprentissage plus lent mais moins de fluctuations. Dans cette configuration, le robot ne parvient pas à déterminer la meilleure action à prendre lorsqu'il est bloqué, puisqu'il reçoit alors une pénalité de -50 quoi qu'il fasse.

6. Réinitialisez de nouveau l'IA. Réglez le paramètre gamma à **0.5** et conservez votre valeur de vitesse d'apprentissage. Lancez un nouvel apprentissage pas à pas avec ces paramètres, et observez les valeurs des connexions. *Quelles sont les différences avec l'apprentissage précédent ? Expliquez intuitivement le rôle du paramètre γ.* Le robot parvient désormais (difficilement) à apprendre qu'il vaut mieux se retourner lorsqu'il est bloqué. Le paramètre γ permet de prendre en compte les **récompenses futures** reçues par le robot.

3.2 Résumé Q-learning

À retenir :

L'algorithme du **Q-learning** permet de mettre à jour les valeurs des connexions après chaque récompense reçue. Lorsque le **facteur d'actualisation** γ est nul, la formule du Q-learning tient compte uniquement des récompenses reçues. Dans ce cas, il est impossible d'apprendre que certaines actions qui donnent lieu à une récompense négative (comme se retourner face à un obstacle), sont pourtant nécessaires afin d'obtenir ensuite de meilleures récompenses.

Pour obtenir un meilleur apprentissage, il ne faut pas se contenter d'observer la valeur de la récompense immédiate, mais aussi de l'état dans lequel se trouve le robot après avoir agi. Le paramètre γ permet de choisir l'importance relative à accorder aux récompenses immédiates et aux valeurs de l'état dans lequel se trouve le robot.

La formule complète de mise à jour d'une connexion est :

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha((1-\gamma)r + \gamma Q_t(s',a*))$$

où s' représente l'état dans lequel se trouve le robot **après** avoir effectué l'action a, et $a*$ est l'action qui, à partir de l'état s', permet d'obtenir la meilleure récompense. Autrement dit, le terme $Q_t(s',a*)$ tente d'estimer la *valeur* de l'état d'arrivée s' en supposant que la prochaine action sera optimale.

4 Deep Q-learning : apprentissage avec caméra

Dans cette partie, nous allons voir les modifications à apporter afin de pouvoir utiliser l'algorithme du **Q-learning** avec la caméra du robot.

1. Dans le menu **Paramètres** → **charger des paramètres d'exemple**, char-

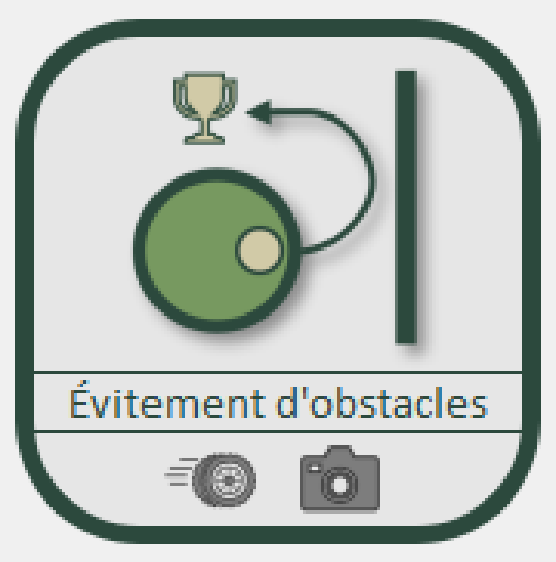

FIGURE 6 – Icône de la configuration : *Apprentissage par renforcement - Évitement d'ostacles*.

gez la configuration **Apprentissage par renforcement - Évitement d'obstacles** (voir figure 6).

2. Parcourez les onglets **Récompense** et **IA** et notez les différences avec les paramètres précédents. *Comprenez-vous l'effet de chacun de ces paramètres ?* Dans l'onglet **Récompense**, on peut voir que la valeur de la récompense pour les virages a été augmentée à environ 82. Dans l'onglet **IA**, on peut voir que l'algorithme sélectionné est maintenant *deep Q-learning*, les valeurs des paramètres **vitesse d'apprentissage**, γ et **exploration** ont été ajustées. Trois couches de neurones intermédiaires ont été ajoutées, avec une fonction d'activation de type *Leaky ReLu*. De plus, les options *biais neuronal* et *mémoire d'expériences* ont été sélectionnées.
3. Lancez l'apprentissage en mode autonome. En utilisant vos connaissances, ajuster les valeurs des paramètres **vitesse d'apprentissage**, γ et **exploration** pour améliorer les performances de l'apprentissage. L'apprentissage nécessite environ 10 minutes. *Quel niveau le robot atteint-il ?* Le niveau maximal dépend de l'arène et de la vitesse du robot, mais devrait être supérieur aux niveaux atteints sans la caméra.