CYCLE 5

RESOLUTION DE PROBLEMES PAR UTILISATION DE L'INGENIERIE NUMERIQUE OU L'APPRENTISSAGE AUTOMATISE

TP1 - PSI

TP 1.1

DECOUVERTE DES ROBOTS ALPHA I

L'objectif de ce TP discuter de ce qu'on entend par « Intelligence Artificielle » (IA) à travers la présentation de « 4 niveaux croissants d'autonomie » du robot AlphAI : deux « sans » et deux « avec » de l'IA (mais nous verrons aussi que la notion d'IA elle-même n'est pas bien définie !!).

1 CONNEXION

Sur les ordinateurs, lancer le logiciel AlphAI et se connecter au robot correspondant :

→ Allumez le robot (l'interrupteur se trouve en dessous). Il effectue un petit mouvement puis clignote en blanc une fois qu'il est prêt à recevoir une connexion.

→ Notez le numéro du robot inscrit sur sa plaque du dessous.

→ Choisissez si vous voulez vous connecter en Wi-Fi ou en Bluetooth (le Bluetooth est préférable lorsqu'il y a plus de 4 robots, ou si de nombreux réseaux Wi-Fi sont déjà présents dans la salle).

Connexion Wi-Fi	Connexion Bluetooth
- Connectez l'ordinateur au Wi-Fi du robot : cherchez le réseau Wi-Fi qui commence par ALPHAI et se termine par le numéro de votre robot : le mot de passe est identique au nom du wifi (attention aux majuscules). - Dans le menu Outils, choisissez « wifi ».	- Dans le menu Connexions > Bluetooth, choisissez dans la liste le nom (numéro) de votre robot. - Si votre robot n'est pas dans la liste, cliquez sur « Mon robot n'est pas dans la liste » et suivez les instructions. Votre robot est alors ajouté à la liste où vous pouvez le sélectionner.

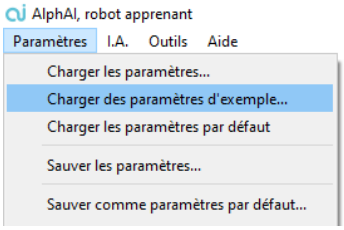
→ Cliquez sur le bouton « connexion » 

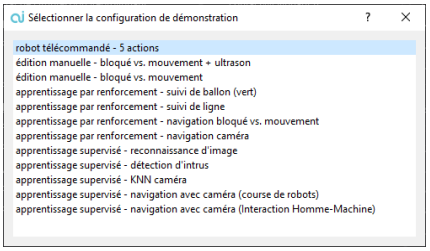
pour vous connecter au robot. Vous voyez apparaître son niveau de batterie  en bas à droite.

2 NIVEAU 1 : TELEGUIDAGE

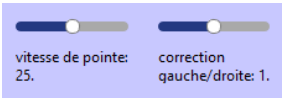
Manipulation

Dans le menu « Paramètres », choisissez « Charger des paramètres d'exemple… » puis sélectionnez (double-cliquez) la configuration « robot télécommandé – 5 actions ».





Vérifiez que vous êtes bien connecté au robot (le niveau de batterie  apparaît en bas à droite).
Vous pouvez commencer à piloter le robot AlphAI, soit en cliquant sur les 5 flèches à droite, soit en utilisant les flèches du clavier de l'ordinateur.
Vous pouvez augmenter la vitesse du robot avec le curseur « vitesse de pointe », et s'il ne va pas tout à fait droit vous pouvez corriger cela avec le curseur « correction gauche/droite » (par exemple, s'il va trop à gauche, déplacez le curseur vers la droite).



Discussion

Discutez ensemble de ce qu'il peut y avoir derrière ce robot télécommandé en terme d'ingénierie : le robot n'est pas autonome puisqu'il est dirigé par l'humain, néanmoins beaucoup de principes sont déjà en œuvre.

Niveau 1 : Téléguidage

Qualités du robot
- Moteurs
- Carte électronique
- Connection sans fil avec l'ordinateur
- Programme sur l'ordinateur pour recevoir les ordres

Similarités avec le vivant
- Moteurs ↔ Muscles
- Carte électronique, communication, etc. ↔ Nerfs

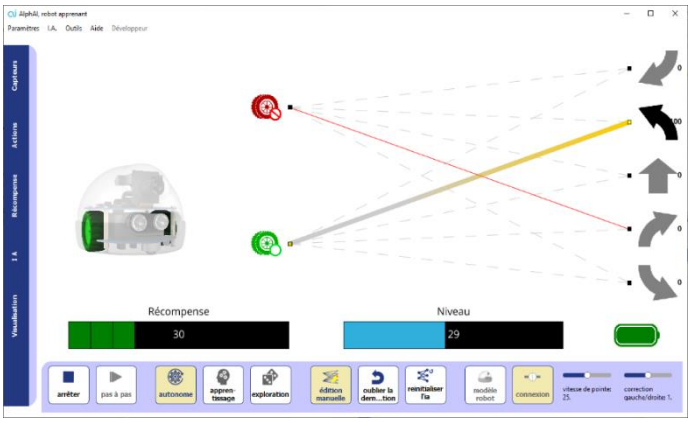
Qualités que le robot n'a pas
Le robot n'a aucune autonomie
L'humain décide chaque action du robot (« ordres »)

3 NIVEAU 2 : PROGRAMMATION

Nous n'abordons pas encore ici l'apprentissage automatique, mais commençons à nous familiariser avec un petit réseau de neurones artificiels en le « programmant » à la main.

Manipulation

Dans le menu « Paramètres », choisissez « Charger des paramètres d'exemple… » puis sélectionnez (double-cliquez) la configuration « édition manuelle – bloqué vs. mouvement ».



Vous voyez apparaître un mini « réseau de neurones » dans lequel c'est à vous de tracer des connexions !

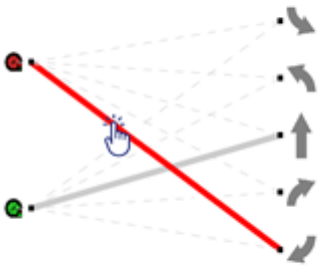
A gauche de ce réseau, les symboles « roue rouge » et « roue verte » représentent l'état du capteur de mouvement du robot. Les petits carrés noirs ou jaunes sont des « neurones artificiels ». À tout moment, si le robot n'est pas bloqué par un obstacle, c'est le neurone « roue verte » qui s'allume :  . Si au contraire il est bloqué par un obstacle, c'est le neurone « roue bloquée » qui s'allume :  . En cliquant sur les traits qui apparaissent en pointillé, créez des connexions entre ces 2 neurones d'*entrée* et les 5 neurones de *sortie* correspondant aux 5 actions disponibles que vous connaissez déjà. Grâce à ces connexions, vous pourrez faire choisir au robot une action différente selon qu'il est bloqué ou non.

Observez en bas la *Récompense* et le *Niveau* indiqués : plus le robot va vers l'avant plus il reçoit des récompenses élevées et plus son niveau (qui est la moyenne des récompenses obtenues en une minute) augmente. Essayez de mettre en place les 2 connexions qui vont permettre au robot d'atteindre le meilleur niveau.

Discussion

Nous avons fait se déplacer AlphAI de manière autonome. Ce n'était plus un humain qui prenait les décisions au fur et à mesure, mais un *programme informatique*, représenté sous la forme graphique de connexions entre entrées « capteurs » et sorties « actions ». En Scratch, Python ou tout autre langage usuel, le programme aurait été de la forme *si* je suis bloqué, *alors* le robot se tourne en arrière, *sinon* il va tout droit. Récapituler dans la discussion les éléments supplémentaires qui interviennent, les similarités et les différences avec le vivant.

Niveau 2 : Programmation



Qualités du robot (en plus des précédentes)
- Capteurs
- Programme de décisions des actions en fonction de l'état des capteurs permet un déplacement autonome

Similarités avec le vivant
- Capteurs ↔ Organes des sens
- Programme de décisions ↔ Cerveau = centre de commandes

Qualités que le robot n'a pas
Le robot ne fait qu'exécuter ce que l'humain a prévu.
L'humain a prévu à l'avance tous les cas de figure et ce qu'il faut faire.
Le robot exécute toujours la même chose, ne s'adapte pas, n'a aucune initiative, etc.

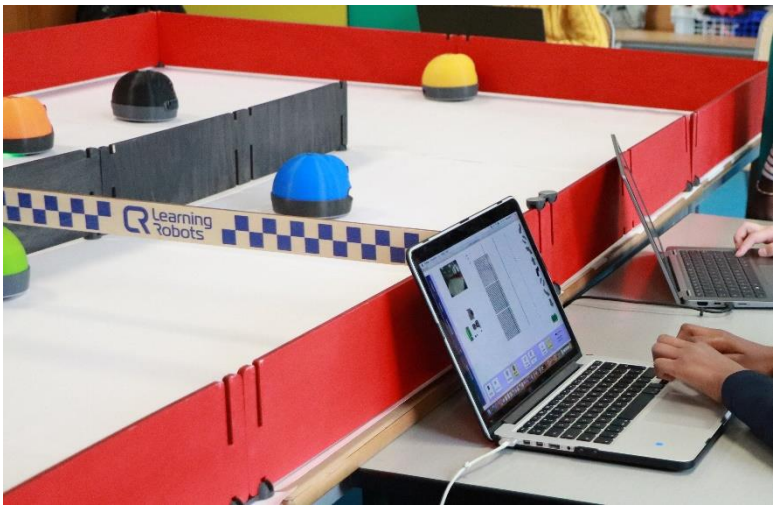
Seconde manipulation

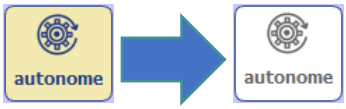
Répéter la manipulation avec la configuration « édition manuelle – bloqué vs. mouvement + ultrason ». Il faudra alors décider quelle action faire dans 4 cas de figure : robot bloqué et obstacle détecté devant, robot bloqué et pas d'obstacle détecté, robot non bloqué mais obstacle détecté devant, robot not bloqué et pas d'obstacle. Attention, l'ultra-son ne « voit » pas les murs sur lesquels il arrive en biais.

4 NIVEAU 3 : APPRENTISSAGE SUPERVISE (« PAR IMITATION »)

Ça y est, nous entrons dans le vif du sujet, avec les notions les plus importantes à apprendre à propos de l'IA ! Le principe de l'Apprentissage Supervisé est que pendant une première phase d'*entraînement* on fournit à l'algorithme d'IA des exemples de décisions qu'il doit prendre en fonction des entrées qu'il reçoit. L'algorithme apprend grâce à ces exemples et devient capable dans une deuxième phase d'*utilisation* de prendre lui-même les bonnes décisions sur de nouvelles données d'entrée.

Entraînement



Chargez la configuration « apprentissage supervisé – navigation avec caméra (course de robots) ». Décochez le bouton « autonomie » 

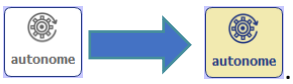
pour que le robot ne se déplace pas par lui-même. Appuyez sur « démarrer » puis pilotez le robot pour lui apprendre les bonnes décisions, en cliquant sur les flèches à l'écran ou en utilisant les flèches du clavier. La différence avec le pilotage de tout à l'heure (Niveau 1 : Téléguidage), c'est que maintenant l'IA mémorise les actions que vous faites faire au robot !

Utilisation : Course de robot(s) autonome(s)

N'essayez pas d'aller trop vite, prenez le temps de bien choisir l'action adéquate à chaque fois. Faites attention à **ne pas confondre la gauche et la droite du robot** !

Apprenez au robot à parcourir le circuit dans le bon sens, sans entrer en collision avec les murs.

Vous pouvez également déplacer le robot à la main pour le mettre face à des situations qu'il n'a pas encore rencontrées.

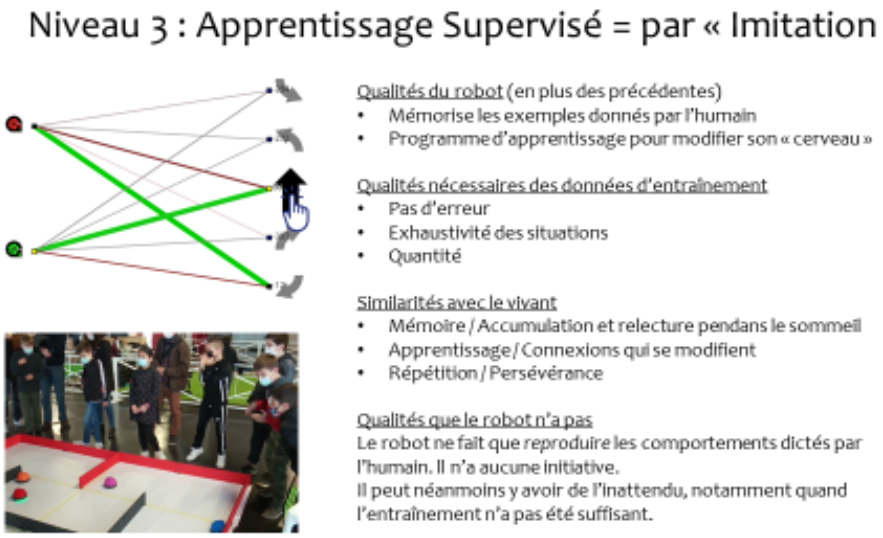
Repassez le(s) robot(s) en mode autonome 

. Les robots se conduisent tout seuls !
Si vous en avez le temps, vous pouvez organiser une première **course d'essai** entre les robots. Pendant cette course, il est possible de **continuer d'améliorer l'entraînement**. En effet, les robots vont se mettre tout seuls dans des situations non rencontrées pendant l'entraînement initial (par exemple, se bloquer sur un bord de l'arène, ou sur un autre robot). Vous pouvez à ce moment-là en *reprendre le contrôle* avec les flèches à l'écran ou avec le clavier *sans avoir besoin de désactiver le mode autonome*, pour lui faire faire la bonne action (typiquement, se dégager dans la bonne direction) : cela ajoute de nouvelles données d'entraînement grâce auxquelles l'IA apprend désormais comment réagir également à ces nouvelles situations.

Dans tous les cas, vous pourrez terminer avec une **course « officielle »**. Au top départ, tous les robots alignés sur la ligne de départ sont démarrés en mode autonome par leurs utilisateurs, qui n'ont ensuite plus le droit d'aider les robots en les pilotant. Si les robots se bloquent sur des obstacles, l'enseignant les remet dans la course à la main au bout de 5 secondes bloqué. Une telle course est très stimulante ; bien entendu ce sont les robots les mieux entraînés qui ont le plus de chances de gagner la course.
Dans le cadre d'un cours ou d'une conférence avec un seul robot, bien entendu on n'organise pas de course mais on fait la démonstration que le robot a appris à parcourir le circuit de manière autonome.

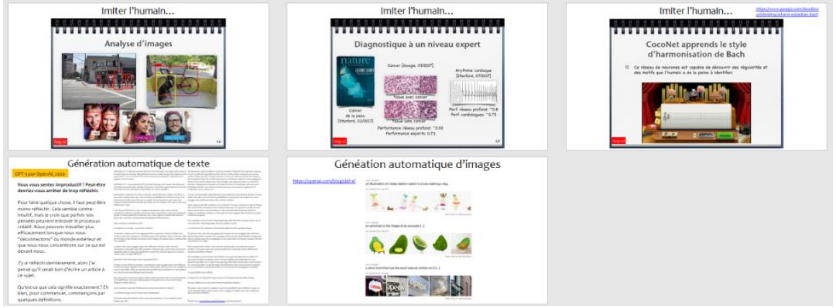
Discussion

Nous avons utilisé ici un algorithme d'**apprentissage supervisé**. Cet algorithme est basé sur le **réseau de neurones artificiels** qui est affiché à l'écran (mais il existe d'autres types d'algorithmes qui ne sont pas basés sur des réseaux de neurones, par exemple l'algorithme des K plus proches voisins). Tous les algorithmes d'apprentissage supervisé fonctionnent de la manière suivante.
Dans un premier temps appelé **entraînement**, nous avons fourni des **exemples de décisions** à l'algorithme, c'est-à-dire à la fois des images de ce que voit la caméra du robot et les actions qu'il faut choisir lorsque la caméra voit ces images. Pendant l'entraînement, le réseau de neurones artificiels a été *modifié* par l'algorithme d'apprentissage (plus spécifiquement, certaines connexions ont été augmentées, d'autres diminuées, de sorte que le réseau de neurones devienne capable d'imiter les mêmes prises de décision).
Dans un second temps appelé **utilisation**, l'algorithme est capable de prendre de bonnes décisions sur de nouvelles images.
Nous avons vu que tous les robots n'ont pas appris la même chose, certains se sont mieux comportés pendant la course, d'autres moins bien. Comme tous les participants ont utilisé le même algorithme, nous voyons bien que **la qualité des prises de décisions autonomes dépend de la qualité des données d'entraînement**.
Nous pouvons discuter ensemble notamment de ce qui fait la qualité des données d'entraînement.



Exemples d'utilisation de l'apprentissage supervisé

Quelques exemples de l'apprentissage supervisé :



5 Niveau 4 : Apprentissage par Renforcement (« par essais et erreurs »)

Faire apprendre avec des récompenses

Avec l'**apprentissage par renforcement**, l'IA n'apprend plus à partir d'exemples fournis par l'utilisateur, mais en menant ses propres essais et erreurs. Bien entendu, l'utilisateur continue de lui fixer un but, sous la forme d'une **récompense** (penser au score dans un jeu vidéo, à la victoire ou défaite dans un jeu – d'ailleurs c'est une IA basée sur de l'apprentissage par renforcement qui a battu le champion du monde du jeu de Go en 2016…, ou encore au sucre qu'on donne à un animal pour lui apprendre des tours !). Ainsi par essais et erreurs l'IA va découvrir par elle-même quelles sont les meilleures actions à mener dans telle ou telle situation. La récompense permettra notamment de **renforcer** les « bonnes actions ».

Dans les activités présentées ci-dessous, la récompense a été conçue pour « inciter » le robot à se déplacer dans l'arène en faisant le maximum de lignes droites, mais en évitant les murs à l'avance grâce à des virages. Voici comment elle est calculée (vous pourriez discuter avec l'audience pour faire deviner quel mode de calcul serait une bonne incitation pour obtenir ce comportement) : Le robot reçoit le maximum de récompense à chaque fois qu'il va tout droit (100), mais reçoit des récompenses négatives (-50, « punitions ») lorsqu'il va en arrière ou s'il essaie d'avancer alors qu'il est bloqué par un obstacle. Il reçoit aussi une récompense intermédiaire lorsqu'il effectue un virage avant gauche ou avant droite (80 dans le cas de la première activité, 30 dans le cas de la seconde).

Nous proposons deux configurations pour introduire l'apprentissage par renforcement. Si l'on veut privilégier la manipulation par les élèves, nous conseillons « apprentissage par renforcement – navigation caméra ». Si l'on veut aller plus loin en privilégiant les explications de « comment ça fonctionne », sous la forme d'une démonstration par le professeur ou le conférencier, nous conseillons « apprentissage par renforcement – navigation bloqué vs. mouvement ». On pourra bien sûr faire les deux si l'on dispose de suffisamment de temps, dans l'ordre que l'on voudra.

Chargez l'une ou l'autre de ces deux configurations. Nous retrouvons les indicateurs *Récompense* et *Niveau* rencontrés dans la partie « Niveau 2 : Programmation ». L'indicateur *Récompense* indique les récompenses obtenues après chaque nouvelle action, tandis que l'indicateur *Niveau* indique la moyenne des récompenses obtenues au cours de la dernière minute écoulée. Naturellement, ce niveau va augmenter au cours de l'apprentissage.

Manipulation : navigation avec caméra

Chargez la configuration « apprentissage par renforcement – navigation caméra » et démarrez l'apprentissage 

. Le robot démarre, vous observerez que ses premiers mouvements sont hésitants mais qu'il découvre rapidement et favorise les actions qui donnent des points, à savoir aller tout droit et les deux virages avant. Assez rapidement également il apprend à effectuer un virage arrière pour se débloquer des murs. Ce qui lui prend plus de temps à apprendre, c'est la distinction entre les moments où il peut aller tout droit pour engranger un maximum de récompenses, et ceux où il vaut mieux tourner avant un mur pour ne pas se prendre de punitions liées au blocage.

Pour apprendre, l'IA effectue de temps en temps des « explorations » aléatoires, c'est-à-dire qu'elle choisit d'effectuer non pas l'action qu'elle pense être la meilleure, mais une autre au hasard, de manière à peut-être découvrir de meilleures manières de faire. Nous pouvons faire l'analogie avec, en biologie, les mutations aléatoires pendant l'évolution. La plupart de ces explorations se révèlent mauvaises, mais quelques-unes conduisent à découvrir vraiment les meilleures actions à effectuer.

Pour un apprentissage plus efficace, il est possible de « guider » le robot, c'est-à-dire de lui commander quelles explorations effectuer. Pour cela, on prend le contrôle du robot avec les flèches à l'écran ou avec le clavier. Ainsi les élèves vont pouvoir interagir avec l'apprentissage en cours et l'accélérer.

Au bout de 10-15 minutes si on ne l'a pas aidé, ou moins si on l'a aidé en prenant le contrôle, le robot navigue de manière satisfaisante dans l'arène, tournant avant les murs, allant tout droit lorsqu'il y a la place. Pour que ce comportement cesse d'être interrompu par les explorations, désactivez les explorations 


: le comportement devient encore plus satisfaisant et le niveau continue d'augmenter.

Pour aller plus loin : Explication détaillée sur un exemple simple

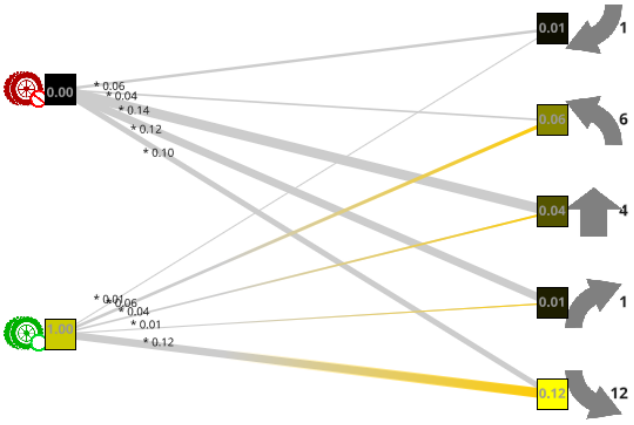
La démonstration qui suit permet d'entrer dans les détails d'un premier algorithme d'apprentissage par renforcement, le Q-learning. La visualisation de cet algorithme sous la forme d'un réseau de neurones artificiel permettra également de se familiariser avec les apprentissages de ces derniers. Au lieu de démarrer l'apprentissage avec le bouton « démarrer », nous allons le décortiquer avec le bouton « pas à pas » 

. Nous recommandons à l'enseignant de les pratiquer dans un premier temps avec le robot simulé qui apparaît en bas à droite lorsqu'on n'est pas connecté au vrai robot.

Chargez la configuration « apprentissage par renforcement – navigation bloqué vs. mouvement ». Comme dans « Niveau 2 : Programmation », l'IA ne dispose comme seule information sensorielle que le fait de savoir si le robot est bloqué (neurone « roue rouge ») ou en mouvement (neurone « roue verte »). Pour obtenir le maximum de récompenses elle devra apprendre à aller tout droit lorsque le robot n'est pas bloqué, et à tourner en arrière lorsqu'il est bloqué.

Comme le réseau a peu de connexions, nous voyons distinctement chacune d'entre elles, qui apparaissent plus ou moins fortes (épaisseur des traits et nombres indiquant cette force ; on parle encore de « poids » des connexions). En règle générale, les connexions dans un réseau de neurones sont initialisées aléatoirement ; dans cette configuration précise cependant, à des fins pédagogiques, elles sont toujours initialisées de la même manière.

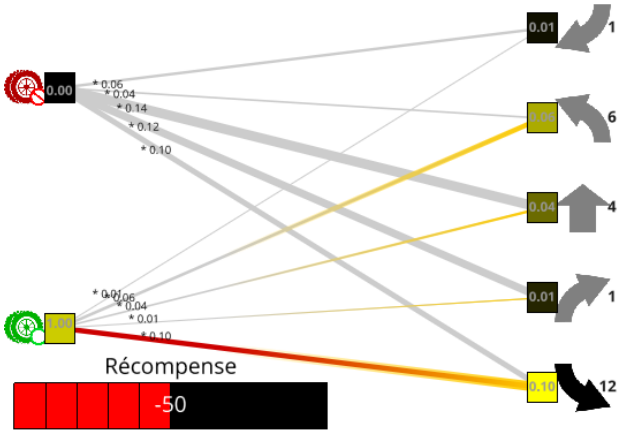
Lorsque le robot n'est pas bloqué (le neurone « roue verte » est allumé avec la valeur 1) les 5 neurones de sortie sont « excités » par ce neurone roue verte en proportion de comment ils y sont connectés : on observe les valeurs 1, 6, 4, 1 et 12. Au cours de l'**apprentissage**, l'algorithme du Q-learning va *modifier les connexions* de telle sorte que ces valeurs de sorties prédisent au mieux les récompenses qui seront reçues après avoir choisi telle ou telle action.



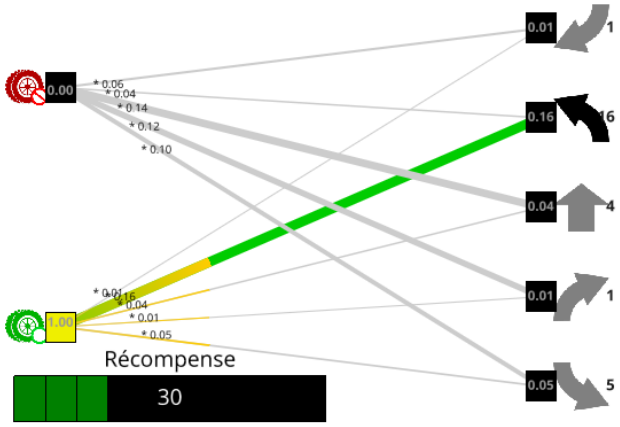
Appuyons une fois sur « pas à pas » 

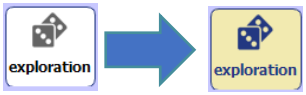
. Le robot choisit l'action avec la plus forte valeur, c'est-à-dire la 5ème : il recule à droite. La récompense reçue est négative : -50, c'est-à-dire bien inférieure à la valeur 12. L'algorithme du Q-learning va alors « corriger » cette erreur en *diminuant* la connexion reliant la roue verte à l'action reculer à droite. La connexion est colorée en rouge pour mettre en évidence cette diminution. A présent l'action « reculer à droite » ne sera plus valorisée qu'à 10.

Appuyons à nouveau sur « pas à pas » : c'est encore l'action « reculer à droite » qui est choisie, mais comme la récompense est à nouveau de -50, la connexion est à nouveau diminuée. Après plusieurs



A présent sur les pas à pas suivants, c'est par conséquence l'action « tourner à gauche » qui est sélectionnée. La récompense reçue est à présent de 30, ce qui est supérieur à la valeur prédite de 6 : l'algorithme va cette fois corriger l'erreur en *augmentant* la valeur de la connexion. La connexion est colorée en vert pour mettre en évidence cette augmentation.

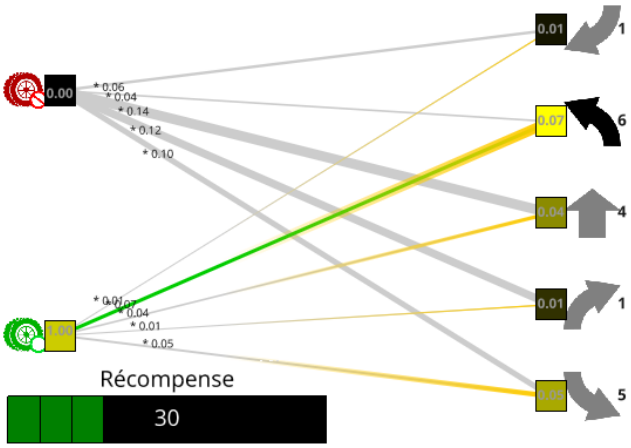


Activez l'exploration 

:

A présent le robot essaie de temps en temps une autre action (la flèche sélectionnée est colorée en *bleu*) et l'algorithme met à jour la valeur de la connexion correspondante. Cela va conduire notamment à une augmentation de la valeur de « aller tout droit » lorsque le robot n'est pas bloqué.

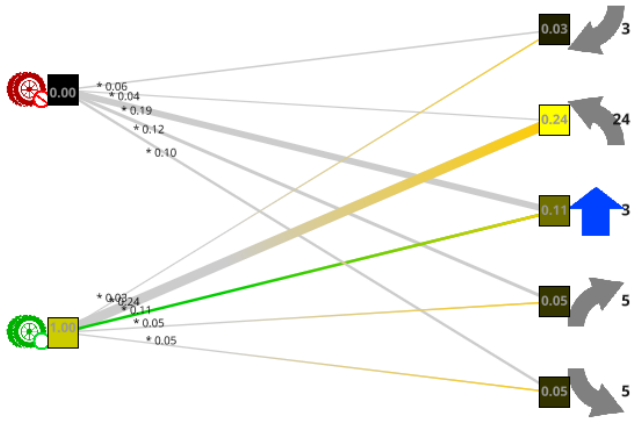
pas à pas, cette connexion devient plus faible que la connexion avec l'action « tourner à gauche ».



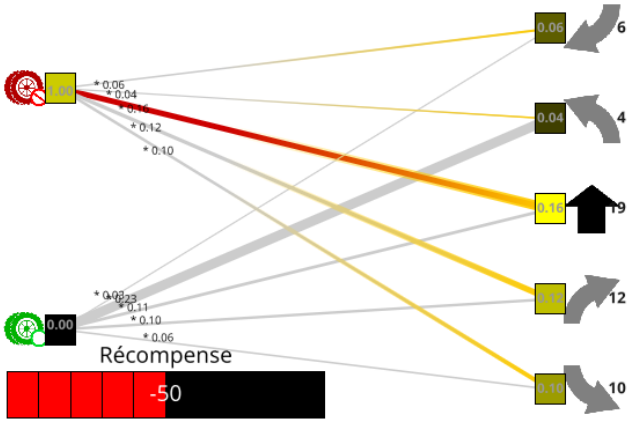
Tous les pas à pas suivants se répètent ensuite à l'identique : le robot choisit de tourner à gauche et la connexion avec « tourner à gauche » est renforcée. Nous comprenons qu'elle va augmenter jusqu'à atteindre la valeur 30 où elle se stabilisera : appuyez à présent sur « démarrer » 

pour ne plus marquer de pause. On constate que l'apprentissage est imparfait puisque le robot tourne sur lui-même indéfiniment et que l'IA n'a pas appris que l'action « tout droit » a en réalité une valeur plus grande que la valeur 4 affichée pour l'instant !

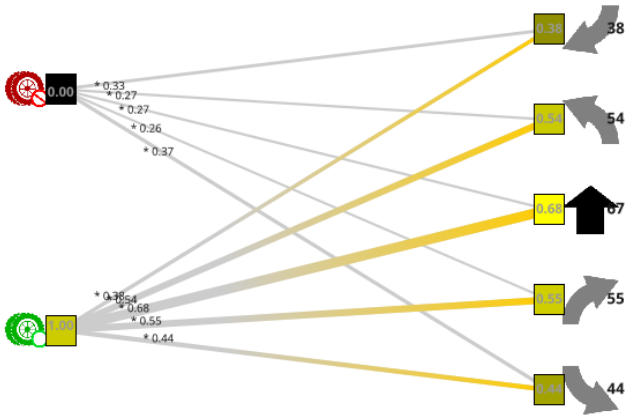
Pour remédier à cela, il faut également essayer de temps en temps les autres actions ! Nous allons à présent activer une nouvelle option qui fait cela, c'est l'**exploration** !



A un moment donné également, le robot va se bloquer sur un obstacle ; c'est alors le neurone « roue rouge » qui s'allumera avec la valeur 1 et qui excitera les 5 neurones d'action selon les 5 connexions correspondantes. Le robot commencera par aller tout droit étant donné la connexion prépondérante avec cette action, mais vu la



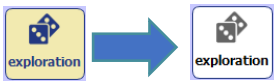
récompense reçue de -50, cette connexion sera vite diminuée.



Laissez le robot continuer de tourner pendant un certain temps (si vous utilisez le robot simulé sans vous être connecté au vrai robot, vous pouvez activer la simulation accélérée 

pendant quelques dizaines de secondes puis la désactiver). Les connexions dans le réseau de neurones finiront par ressembler à l'image ci-contre : lorsque le robot n'est pas bloqué (roue verte) la connexion la plus forte est avec l'action aller tout droit ; lorsqu'il est bloqué (roue rouge) la connexion la plus forte est avec une des deux actions virage arrière (ici arrière droit, à 0.37).

Nous avons bien vu comment l'**apprentissage** a consisté en une **reconfiguration du réseau de neurone**, qui a conduit à l'obtention de plus de **récompenses** et l'augmentation du **niveau**.

Nous avons vu en particulier l'importance de l'**exploration** au cours de l'apprentissage pour vraiment découvrir la ou les actions optimales. Cette exploration devient néanmoins inutile au bout d'un certain moment, et la désactiver en fin d'apprentissage 

permet d'éviter les actions aléatoires et d'augmenter encore le niveau. L'exploration nécessite donc un bon *dosage* ; la question du *compromis entre exploration et exploitation* est un grand enjeu de la recherche en apprentissage par renforcement, et n'est pas sans lien avec nos apprentissages humains (voir discussion).

Pour être complet, il faut mentionner également une subtilité particulière lorsque le robot est bloqué après avoir foncé droit dans un mur : on voit bien que s'il choisit de tourner à gauche, aller tout droit ou tourner à droite, il ne parviendra pas à avancer (restera bloqué) et donc recevra une récompense/punition de -50. Si au contraire il choisit de reculer à gauche ou à droite, il parviendra à se débloquer, mais recevra néanmoins la même récompense/punition de -50. S'il s'en était tenu à la valeur de la récompense reçue, l'algorithme n'aurait donc pas su privilégier les virages arrières par rapport aux autres actions. En réalité il est capable de valoriser plus les virages arrière car la mise à jour des connexions pendant l'apprentissage prend en compte non seulement la récompense reçue immédiatement, mais également le nouvel état (bloqué ou non bloqué) dans lequel le robot parvient. Nous n'entrons pas plus dans le détail mais retenons l'importance de ne pas considérer seulement la récompense immédiate, mais également d'**anticiper** des récompenses pouvant intervenir plus tard.

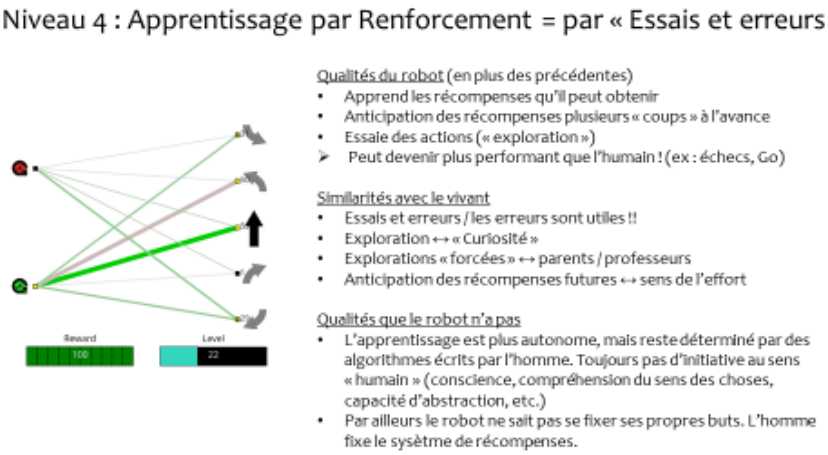
Discussion

Si l'apprentissage supervisé est ce qui est le plus utilisé dans un très grand nombre de domaines, l'apprentissage par renforcement est particulièrement fascinant puisqu'il donne à l'IA la capacité d'apprendre en très grande autonomie. L'apprentissage par renforcement est utilisé notamment pour les programmes et robots « en interaction » avec le monde et avec les personnes : victoire au jeu de Go contre le champion du monde Lee Sedol en 2016, victoire de bots sur les meilleures équipes de jeux comme Starcraft, applications en robotique, etc.

Nous recommandons de discuter les nombreuses nouvelles similarités découvertes entre apprentissages artificiel et humain :

- L'exploration fait directement référence à la curiosité. La curiosité est nécessaire, autrement on reste enfermé dans des manières de faire qui marchent mais ne sont pas les meilleures. D'un autre côté, trop de curiosité peut être du papillonnage en empêchant d'exploiter ce qu'on a déjà appris.
- A côté de la curiosité naïve, l'aide d'un maître peut aider à faire découvrir plus vite les bons comportements (cf. lorsque l'utilisateur reprend le contrôle dans l'activité de manipulation ci-dessus).
- Nous avions insisté à propos de l'apprentissage supervisé sur le fait que les données d'apprentissage ne doivent pas comporter d'erreur. Mais ici faire des erreurs ne pose pas de problème, au contraire : le robot apprend de ses erreurs ! (il apprend que telle ou telle action à tel moment donnera moins de récompense, et ne la fera plus)
- Dans la configuration « navigation caméra », l'IA emmagasine ses expériences dans une mémoire et est capable de continuer d'apprendre même à l'arrêt, ce qui fait écho à nos apprentissages pendant le sommeil ! (on peut en faire l'expérience ainsi : faire tourner l'apprentissage 3 minutes ; puis désactiver l'autonomie pendant 3 minutes : le robot stoppe mais l'apprentissage sur ses expériences continue ; réactiver l'autonomie : le robot se comporte mieux qu'avant !)

Mais il reste aussi des différences très fortes entre IA et apprentissage humain, comme détaillé dans la diapositive.



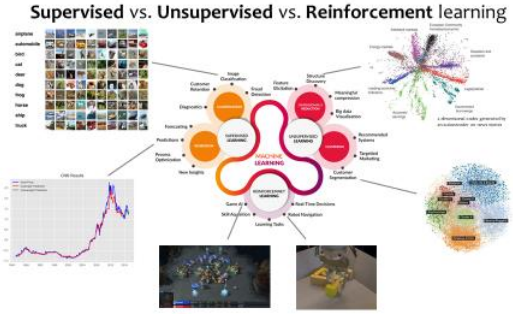
Exemples d'utilisation de l'apprentissage par renforcement

Là encore quelques exemples.

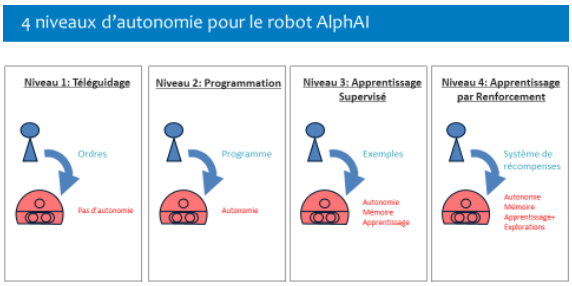
6 CONCLUSION

Récapitulons quelques notions apprises au cours de cette activité :

- L'**Intelligence Artificielle** est un projet dont nous sommes encore loin de l'accomplissement, nous appelons néanmoins Intelligence Artificielle certaines facultés de nos programmes, notamment ceux faisant appel à du « **machine learning** ».
- Nous avons présenté deux des trois grandes familles du *machine learning* : **apprentissage supervisé** et **apprentissage par renforcement**. La troisième catégorie est **l'apprentissage non supervisé**, que nous n'avons pas présenté, mais nous y viendrons à l'avenir !



- L'apprentissage a lieu pendant une phase dite d'**entraînement**, puis on peut **utiliser** l'IA qui a appris.
- Le principe de l'**apprentissage supervisé** est d'apprendre sur des **données d'entraînement** (c'est à dire sur des exemples) pour ensuite pouvoir **généraliser** à d'autres données. Nous avons discuté notamment des **qualités** nécessaires aux données d'entraînement pour de bonnes généralisations.
- Le principe de l'**apprentissage par renforcement** est d'apprendre **par essais et erreurs**, guidé par un mécanisme de **récompenses**.
- Nous nous sommes également familiarisés avec les **réseaux de neurones artificiels**, algorithme particulièrement populaire.
- Nous avons discuté tout au long de l'activité des **similarités** entre IA et intelligence humaine, qui vont croissantes au long des 4 niveaux d'autonomie identifiés. Mais les **différences** restent fortes. Notamment dans tous les cas c'est l'**humain qui contrôle le système et/ou ses apprentissages**.



- Il n'en demeure pas moins que tout cela bouge à grande vitesse, aussi est-il important de s'approprier aussi bien les aspects techniques que éthiques de l'IA.

Alors, l'Intelligence Artificielle ?

Un nouvel outil formidable !!!

L'humain en question !!?

Le maîtriser pour ne pas être maîtrisé

Comprendre pour réagir

➡ Ouvrir toutes les boîtes noires de l'Intelligence Artificielle avec le robot **AlphAi**

➡ S'intéresser aux questions sociétales et philosophiques que pose l'I.A.