UNIVERSITE CLERMONT AUVERGNE
Clermont-Ferrand

École Universitaire de Physique et d'Ingénierie

MEMOIRE DE STAGE
1ière année de Master

Spécialité : Automatisme et Robotique

CHUPIN Clément

Automatic Design of Neural Network for Mobile Robot Control

Date de soutenance : 29/08/2022 14h - 15h
Entreprise : INRAE Clermont-Ferrand

Remerciements :

J’ai pu effectuer mon stage dans les meilleures conditions, j’y ai beaucoup appris, et cela a été possible grâce à l’accueil qu’on m’a fait au sein de l’INRAE.
Je tiens à remercier M ZHONG Zhangkai de m’avoir encadré et guidé durant mon stage, ainsi que son aide dans ma compréhension et ma communication avec le monde scientifique
Je remercie également M LENAIN Roland pour la direction administrative et son soutien lors de mon stage, il m’a permis d’avoir une bonne cohésion avec les différentes personnes avec lequel j’ai interagi.
Ainsi que toute l’équipe de travail au sein de l’équipe de recherche ROMEA, où il y règne une bonne cohésion d’équipe et une forte entraide permettant d’avoir, à la fois un environnement de travail agréable et efficace.
Et enfin mes professeurs encadrants, je remercie M THUILOT Benoit pour m’avoir conseillé et mis en lien sur ce stage de recherche, ainsi que Mm TEULIÈRE pour mon suivi lors de ce stage, et mon évaluation finale.
Concernant la partie technique de mon stage, je remercie la ferme de calcul du Mésocentre de l’université de Clermont-Ferrand de m’avoir accueilli pour que je puisse mener à bien mes expérimentations.

Le sujet de ce stage portait sur l'utilisation de réseaux de neurones auto-générés dans le cadre de contrôle direct des actions d'engins agricoles. En effet, ces dernières années, les recherches autour de l'intelligence artificielle ont amené deux axes très distincts mais tout deux intéressants dans le cadre de la robotique agricole, le premier est la prise de décision automatique d'un model neuronal, permettant ainsi la détermination automatique d'un comportement à adopter afin d'accomplir une tâche, ce qui permet, en théorie, d'obtenir en finalité des lois de commandes plus adaptés et plus autonome que ce que nous permet le domaine de l'automatisme.
Et la deuxième, dans le cadre de la création d'un réseau de neurones, est la recherche automatique d'architecture pour répondre au mieux à une tache donnée. Via ces deux nouveaux outils, le but de mon stage était de combiner ces deux technologies dans le but de trouver une manière de contrôler un robot agricole de manière robuste, autonome et adaptable, le but étant de surpasser les paradigmes de contrôle préexistant.
Mon objectif durant ce stage était de comprendre la génération de réseaux de neurones de manière automatique, comprendre la problématique de la création de l'architecture d'un réseau de neurones, ainsi que les avantages et contraintes de la solution qui est apporté par les réseaux auto-générés, puis de les combiner pour apporter une nouvelle solution plus adaptée dans le contrôle d'engins agricole à l'aide d'un modèle d'intelligence artificielle.
Pour parvenir à ce but, l'objectif était d'implémenter en simulation, un engins agricole capable de capter des informations afin de prendre des décisions de contrôle, et évaluer la qualité et la précision de ces prises de décisions, entre un réseau de neurones auto-généré et un construit par un humain.
La finalité idéale de ces expérimentations est l'évaluation du model sur des données réelles afin de déterminer si la solution des réseaux de neurones auto générés est viable pour le contrôle direct de robot, notamment dans le domaine agricole où l'espace de travail est aléatoire et en évolution permanente.

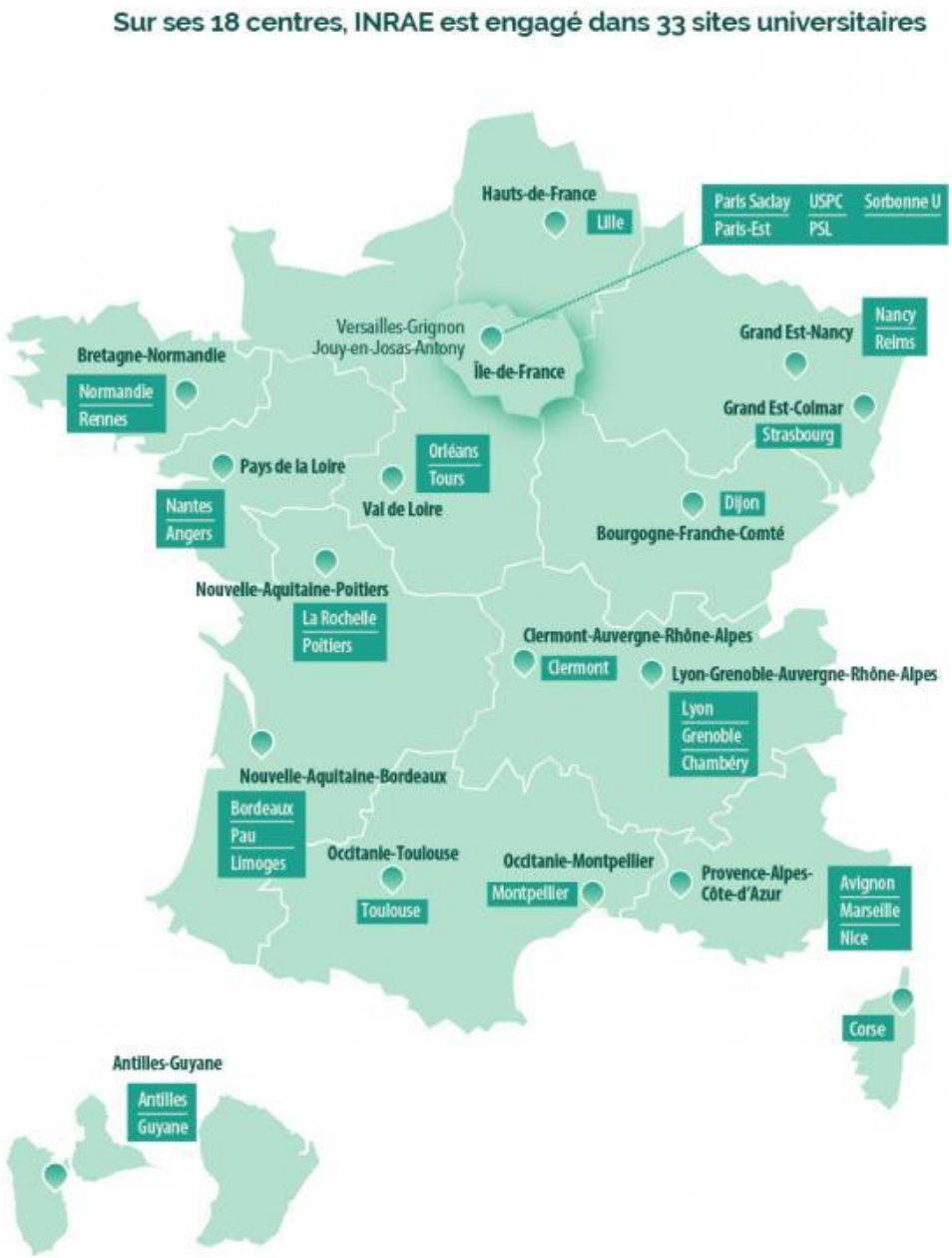
Sommaire :

Introduction :

La problématique du contrôle robotique est un vaste domaine. Le champ d'application de l'automatisme, par sa nature déterministe, est bien adapté au domaine industriel et plus globalement, dès que l'on veut contrôler un actionneur via une commande déterminée. L'automatisme trouve ses limites dans la détermination d'un comportement de contrôle d'un robot dans un milieu indéterminé avec une commande qualitative du résultat obtenu (réaliser une tâche sans forcément donner une commande déterminée aux actionneurs).
On va vouloir accomplir une tâche qui possède de nombreux degrés la liberté de décision, avec un environnement qui peut perturber de manière indéterminée les actions qu'on y effectue, par exemple le suivi d'une trajectoire sur un sol instable, ou la stabilisation d'un drone avec des perturbations atmosphériques.
L'automatisme amène déjà des solutions à cette problématique, on peut, grâce à cet outil avoir un contrôle adaptatif des actionneurs en fonction de la réaction de l'environnement, on peut même définir manuellement de nouveaux comportements pour pallier des situations bien précises. Mais le modèle de contrôle sera incapable de prendre en compte des situations nouvelles qui nécessitent une autonomie dans la prise de décision.
Ce stage avait pour but d'amener une solution technique afin d'améliorer les paradigmes de contrôle préexistant, via l'implémentation de l'auto-génération de réseaux de neurones pour améliorer la détermination d'un comportement face à une tâche donnée.
Le travail demandé mettait en avant l'autonomie, la recherche bibliographique, l'expérimentation et la prise de décision. La tâche se voulait un maximum libre pour permettre à l'étudiant d'apporter une solution originale et innovante. Par l'exploration de mon sujet et nos méthodes de travail, j'ai été amené à changer la méthode d'approche de la problématique de mon stage, en effet, la première solution comportait des contraintes qui ne la désignait a priori pas comme une solution viable pour le contrôle de robots, et cette solution technique était déjà expérimentée par mon binôme. J'ai été poussé à m'éloigner du traitement d'image et de l'entraînement supervisé qui est déjà un domaine très étudié, afin de me concentrer plus précisément sur le contrôle non-supervisé de robots multi-articulations.
J'ai par conséquent étudié une autre solution qui permet, si elle est correctement utilisée, d'améliorer la compréhension des données réelles par un réseau de neurones, sans affecter significativement le temps de calcul. On ne va pas améliorer la compréhension du réseau de neurones en modifiant son architecture, cependant on peut améliorer sa compréhension en modifiant la manière d'interpréter l'entrée. Pour être plus précis, la compréhension de l'entrée et la détermination d'un meilleur comportement sont deux choses différentes. Nous nous concentrerons sur la compréhension qui amène une meilleure détermination du comportement. La recherche se concentre sur ces deux points indépendamment (l'extraction d'information importante, et la politique de prise de décision).
Si mon étude aboutit, elle pourra alors apporter un guide dans l'implémentation de l'extraction de données via une transformée de Fourier pour améliorer la convergence et le résultat final d'un modèle neuronal (l'article scientifique est jugé correct mathématiquement, pertinent dans le sujet et la problématique traitée et singulier dans les différentes techniques étudiés).

Présentation de l'INRAE

L'INRAE, est un centre de recherche publique en agroécologie, mais plus précisément :
« INRAE est un EPST, Etablissement public à caractère scientifique et technologique créé le 1er janvier 2020 et issu de la fusion de l'Inra, Institut national de la recherche agronomique et Irstea, Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture. Il est sous la tutelle conjointe des ministères en charge de la Recherche, l'Enseignement supérieur, la Recherche et l'Innovation (MESRI) et celui en charge de l'Agriculture et de l'Alimentation (MAA). Il développe des collaborations importantes avec le ministère en charge de la Transition écologique et solidaire (MTES) selon des conventions cadre dédiées. », Site internet de l'INRAE



La problématique agricole

Depuis une cinquantaine d'année, l'évolution de la démographie, de la consommation et de la technologie mondiale a amené divers changements dans le domaine de l'agriculture. En effet, la production se doit de répondre à une demande croissante en termes de quantité tout en respectant les enjeux écologiques au moyen et long terme. Elle se doit de répondre à une surface, à des ressources et des moyens d'actions limités. Le paradigme actuel est une production avec, comme contrainte, de limiter l'impact environnemental, mais avec pour priorité une production en grandes quantité à faible coûts. On fait intervenir alors, quand c'est possible, des OGM et des pesticides dispersés à grande échelle, peu régulièrement ce qui implique un traitement intensif, ou bien quand ces techniques sont impossibles, par la nature de la plante ou bien la législation, une délocalisation à la recherche d'une production à faible coût de main d'œuvre, ce qui implique l'exploitation humaine et écologique de pays lointains, avec le coût écologique supplémentaire du transport de ces marchandises.

La mission et les moyens

Mission :

Face à ces problématiques écologiques, techniques, sociales et politiques, l'INRAE a pour mission de développer de nouveaux paradigmes de production agricole. Cette volonté se traduit par la recherche sur de nombreux domaines, bien qu'elle se concentre principalement autour de la biologie, de la chimie et de géologie, la recherche reste large et ouverte à de nombreuses solutions possibles comme la robotique, l'électronique ou encore la mécanique. Toutes ces voies visent à converger vers l'agroécologie de demain, en traitant de problèmes concrets et précis. Comme indiqué sur leur site :

« Nos recherches visent à :

- *Préserver les milieux cultivés*
- *Réduire l'utilisation d'intrants de synthèse (pesticides, antibiotiques, engrais) grâce aux régulations biologiques*
- *Diversifier les productions agricoles à toutes les échelles, du champ à l'assiette.*

Avec également comme autres missions soulever :

- *Connaitre, préserver, restaurer la biodiversité*
- *S'adapter au changement climatique et gérer les risques*
- *La bioéconomie : production et une utilisation durable et circulaire des ressources biologiques*
- *L'alimentation au cœur de notre santé*
- *Favoriser les systèmes agricoles et alimentaires durables »*

Moyens :

Par la problématique à traiter et la mission confiée à l'INRAE, l'organisme possède diverses ressources pour pouvoir accomplir sa tâche.
En quelques chiffres, l'INRAE c'est :

- 18 centres de recherches
- 14 départements scientifiques
- 11523 salariés dont 8413 titulaires
- 268 unités de recherche, de service et expérimentales
- 1 milliard d'euros de budget annuel
- 10 000 ha d'expérimentation
- plus de 50 % des publications en collaboration avec un autre pays »

En effet, l'INRAE dispose de nombreuses ressources humaines et financières pour mener à bien sa mission, au demeurant, la problématique et l'enjeu à traiter restent difficiles et larges. En effet, la France est le premier pays producteur agricole d'Europe avec un chiffres d'affaires d'environ 80 milliards pour environ 3 % du PIB. Face à un si gros secteur, les chercheurs redoublent de créativité et de rigueur pour parvenir à développer et déployer de nouvelles solutions dans le domaine.

Organisation et méthode de travail

Les unités de recherche sont organisé autour d'un chef d'unité, dont le rôle est de coordonner le travail de chaque personne afin de réaliser en temps et en qualité les missions attribuées à l'unité, il a aussi un rôle déterminant dans la recherche de mission et de financement, bien que l'unité possède un budget et du personnel « automatique », la recherche de missions et de financements permets de recruter des contractuels (doctorant, post-doctorants, ingénieur de recherche, ingénieur d'études…) pour permettre à l'unité d'avancer sur des problématique d'envergure.

Méthode de travail :

Par les sujets larges traités, chaque personne collaborant sur un sujet doit pouvoir avoir un maximum de liberté. Le travail s'organise en déléguant beaucoup au sein des équipes. Le choix de l'organisation, les chargées de recherches et les directeurs d'unité vont être les moteurs premiers de cette organisation. Une fois un projet mis en place, le travail va se répartir assez naturellement entre les différents corps de métier, avec une forte collaboration pour pouvoir mener à bien les différentes expérimentations. Spécifiquement au sein de mon équipe de robotique, les chercheurs ont la chance de pouvoir travailler individuellement sur leurs parties et de pouvoir combiner leurs travaux pour arriver à un résultat final, ce qui laisse la possibilité d'une organisation asynchrone.

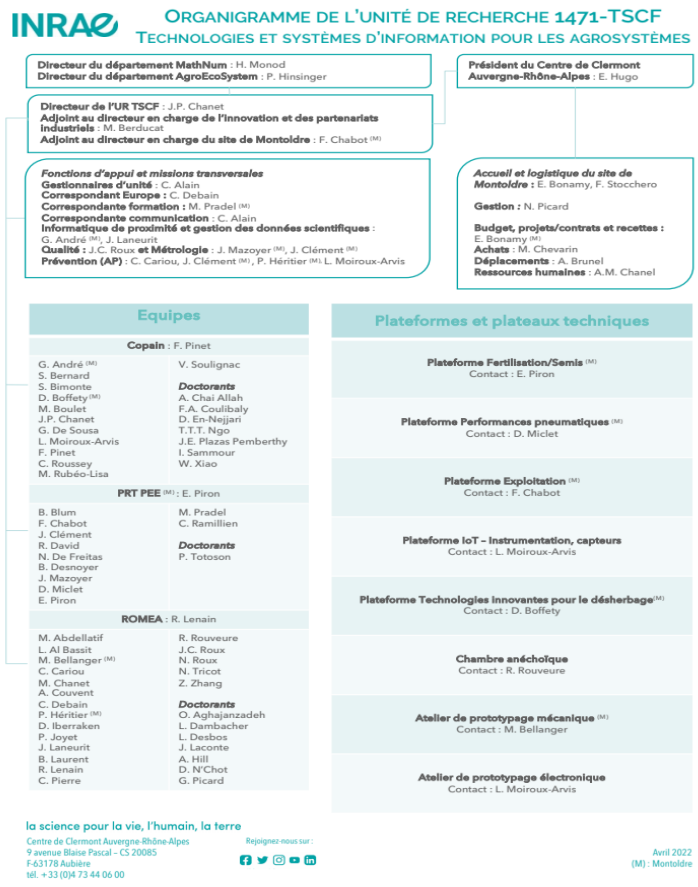
Ambiance de travail :

J'ai eu la chance d'être accueilli dans une ambiance motivante et chaleureuse, les échanges entre les différents corps de métiers se sont révélés être naturels, agréables et intéressants. C'est aussi le reflet des méthodes, de l'organisation et de la mission générale de l'INRAE. En effet, l'organisme à une approche avant tout humaine du travail, les encadrants sont compréhensifs des difficultés de chacun, ce qui pourrait être moins le cas dans le cas où l'entreprise est confrontée à des enjeux économiques et des délais stricts de rendu de travail.

Ces défis ouvrent la possibilité à plusieurs axes de recherche, dont celui de l'unité de recherche TSCF (Technologies et systèmes d'information pour les agro-systèmes), se concentrant sur l'aspect robotique afin de développer des technologies permettant un traitement autonome, ciblé et régulier des plantes, permettant ainsi une méthode de production moins polluante. L'idée derrière ce domaine de recherche est de favoriser des techniques écologiques à déployer à grande échelle, afin de faire concurrence aux techniques de traitement peu régulières, à grandes échelles et par conséquent nocifs pour l'écosystème agricole (remplacer les gros traitements chimiques peu régulier par des traitements naturels plus spécifique et régulier).

Présentation de l'équipe ROMÉA

L'équipe ROMÉA (RObotique et Mobilité pour l'Environnement et l'Agriculture), encadrée par M LENAIN Roland, se concentre sur la création et le développement de technologies robotiques pour le déplacement, la gestion et l'autonomie d'engins agricoles. Cette équipe regroupe plusieurs chercheurs, ingénieurs de recherche, ingénieurs d'études, doctorants et post-doctorants, pour un effectif total d'une vingtaine de personnes. Les axes techniques de recherche sont : le contrôle en automatisme, le traitement de lidar, le contrôle de bras robot en milieu agricole et le développement de la géolocalisation des différents équipements. L'équipe se concentre sur la robotique pure, alors que les autres appartenant à TSCF vont traiter d'autres problématiques par l'informatique (PRT PEE, Plateau de Recherche Technologique Pôle Epandage Environnement) ou l'étude et la création de systèmes embarqués (COPAIN, Systèmes d'information communicants et agri-environnementaux).



J’ai été accueilli sous l’encadrement de Monsieur ZHANG Zhongkai, chargé de recherche, dont le travail se concentre sur le domaine de l’intelligence artificielle afin de développer de nouveaux outils permettant une plus grande autonomie et adaptabilité des engins agricoles.

L’équipe Roméa, par la nature de son travail, ne va pas beaucoup collaborer avec des équipes hors TSCF, mais va, par exemple, collaborer avec l’équipe Copain pour le développement et la mise en place de capteurs plus spécifiques. En revanche, la collaboration reste très présente entre les chercheurs et avec le monde industriel, cela permet d’accélérer la recherche par un financement, une problématique réelle, une application directe des procédés et technologies développées au sein de l’INRAE.

Par exemple, il y a le projet TIARA en collaboration avec l’entreprise privée SABI AGRI (spécialisée dans la création d’engins agricoles électriques). Le but est de développer un véhicule électrique agricole avec plusieurs niveaux d’autonomie, la recherche se concentre actuellement sur le suivi de trajectoire multi-capteurs (GPS, satellite, balise GPS de proximité, lidar), couplé avec des recherches en parallèle sur la segmentation des données lidar pour mieux appréhender son environnement, ainsi que le franchissement des obstacles.

Figure 1: Projet TIARA, véhicule agricole autonome

Poste de travail :

Mon poste de travail était celui de stagiaire chercheur. Ma mission première était d'amener dans le domaine de l'apprentissage par renforcement, la technologie des réseaux auto-générés (AutoML) et d'y voir l'impact positif de leurs utilisations. En effet, cette nouvelle technologie permet, face à une problématique nécessitant une résolution par intelligence artificielle, notamment dans le domaine de la reconnaissance d'image, de trouver la meilleure configuration de réseau de neurones possible. Cette technique a fait ses preuves en surpassant des réseaux de neurones conçus par des humains dans le domaine de la classification d'images.
Concrètement, cela signifie qu'une IA, face à une tâche précise, est meilleure qu'un humain pour choisir la disposition des couches, le nombre de neurones, les fonctions d'activation entre autres. Mais dans le cadre d'une tâche supervisée et au prix d'une multitude d'essais de réseaux neuronaux.
En effet, l'entraînement en AutoML se fait en deux boucles imbriquées, la plus petite est le réseau qui s'entraîne, et la deuxième est celle qui, en fonction du résultat de l'entraînement de la première, va modifier les réseaux neuronaux pour améliorer la performance de ceux-ci.

Pour mener à bien mes recherches et mes expérimentations, j'avais accès à un bureau et un ordinateur. En effet, par le but qui m'a été confié, j'ai dû faire beaucoup de recherches et de lectures d'articles scientifiques pour mieux comprendre l'utilisation et les différents types de réseaux de neurones, ainsi que les problématiques principales dans l'intelligence artificielle pour le contrôle direct de robots. J'ai également expérimenté sur l'utilisation des solutions techniques trouvés afin de constituer une solution finale. Ces expérimentations n'ont malheureusement pas amené d'idées innovatrices par rapport aux solutions déjà existantes, mais m'ont appris beaucoup sur les bases techniques nécessaire pour mon étude finale comme l'utilisation plus bas niveau des librairies pytorch et tensorflow, ainsi que la manipulation de matrices et de couches neuronales.

Concernant l'environnement humain, par la nature de mon travail de recherche, il poussait à beaucoup d'autonomie, mais si jamais j'avais besoin de réponses, de conseils ou d'explications, mon maitre de stage, tout comme les autres chercheurs étaient prêts à m'aider et à me guider. Je n'ai pas eu beaucoup de comptes à rendre ou de travaux spécifiques à fournir, et en plus de cela, à la suite d'un évènement personnel de mon maitre de stage, j'ai été laissé en totale autonomie pendant environ deux mois.

Ce poste, par son environnement et les responsabilités qui en découlent, m'a parfaitement convenu. Il était assez motivant de travailler en autonomie sur des sujets complexes et exigeants, avec pourtant, la liberté de pouvoir comprendre et expérimenter de la manière qui me paraissait être la plus efficace pour atteindre l'objectif fixé par le stage.

I : Problématique principale

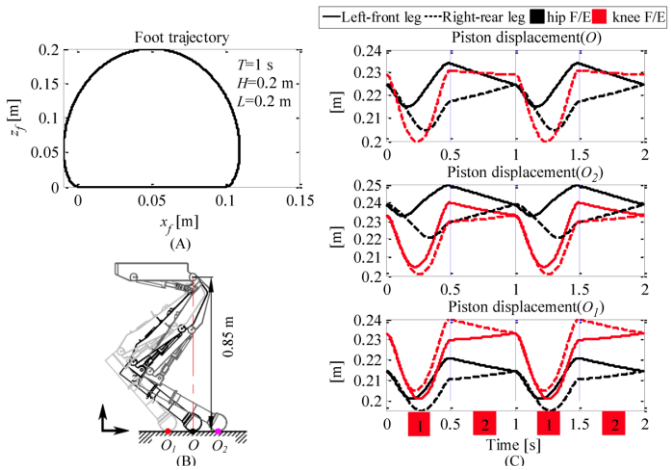
Utilisation de l'automatisme dans la robotique

Depuis des dizaines d'années, la recherche autour de l'automatisme nous a permis d'apporter énormément de solutions technologiques dans le domaine de la robotique pour répondre à des besoins industriels divers et variés. Les lois de commande automatique permettent d'exécuter une tâche de manière contrôlée et fiable. Par les solutions potentielles que cela apporte, l'équipe de robotique Romea au sein de l'INRAE utilise et développe l'automatisme afin de pouvoir contrôler correctement des véhicules dans un contexte agricole, ce qui implique des problématiques nouvelles comme l'instabilité du terrain, l'adhérence du véhicule, la sécurité des agriculteurs et du matériels, des problèmes de franchissement d'obstacle entre autres.
L'automatisme possède de nombreuses applications dans le domaine de l'autonomie des véhicules comme le suivi de trajectoires GPS, l'appréhension d'un chemin de terre via des nappes lidar (« appréhension » car c'est un traitement statistique, le véhicule détecte l'obstacle ou la rangé de plantes de manière empirique, elle peut se tromper, elle ne comprend pas réellement où se situe le chemin).

La recherche autour de l'automatisme nous apporte déjà beaucoup de solutions, notamment sur des lois de commandes avec une part d'inconnus pour le contrôle de jambes robotiques. On peut modéliser des problèmes complexes à plusieurs degré de liberté et les résoudre de manière optimale. Ces techniques permettent, dans l'industrie, d'avoir des tâches qui demandent à la machine de moduler son comportement selon la situation à laquelle elle est confrontée. Ces techniques, bien que plus avancées, ne permettent pas à la machine de prendre des décisions, ou d'analyser une situation de manière autonome. Ces algorithmes pourraient être directement implémenté dans le monde agricole, notamment pour les suivis de trajectoires et le contrôle des jambes robotiques, cela fonctionnerait à 99 % mais l'adaptation à un terrain aléatoire en serait compromise. Ces techniques de contrôle sont incapables de s'adapter efficacement aux situations de la vie réelle.

Ces techniques sont très efficaces pour appréhender un monde physique connu. Cependant, depuis quelques années, la recherche autour de l'intelligence artificielle appliqué au domaine de la robotique se développe, car elle peut permettre des solutions techniques pour autonomiser des outils, plus particulièrement dans le domaine agricole, pour permettre à un robot d'accomplir sa tâche dans un environnement aléatoire (autant par les déplacements sur un terrain naturel que le travail effectué sur des plantes aux formes et à la rigidité inconnues).

La réaction d'un corps non-rigide face à une action mécanique rend la tâche de contrôle beaucoup plus compliquée, on ne peut pas effectuer une commande robuste si on ne connaît pas les réactions du système face à l'instrument. En vulgarisant, la cueillette est une tâche encore plus dure que le déplacement de robot, les techniques d'automatisme trouvent leurs limites dans ce domaine, par le déplacement du robot selon le terrain et la réaction de la plante par rapport à une manipulation. Agir sur un environnement qui évolue de manière constante et aléatoire rends impossible l'utilisation de l'automatisme pour mener à bien ces tâches.



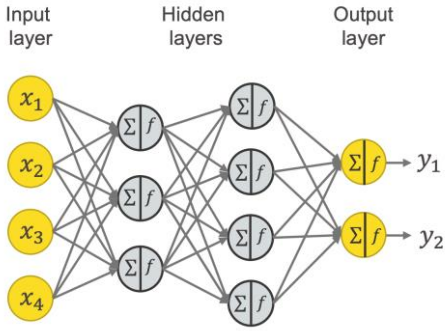
Lois de commande automatique pour un robot biped

Bras robotique dans un contexte agricole (les plantes sont spécifiquement positionnées pour que la tâche puisse être accompli)

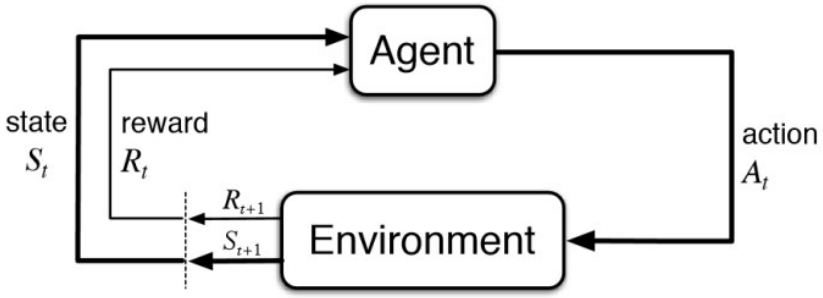
L'IA pour aller plus loin que l'automatisation

Le monde réel peut être difficilement appréhendé par une machine purement automatique. L'idée d'y implémenter des algorithmes adaptatifs selon les différents travaux requis est une bonne solution pour accomplir avec succès les différentes tâches du monde agricole. En effet, la technologie du réseau de neurones permet un traitement unique et adaptatif de chaque situation, c'est grossièrement une loi automatique de commande, des dizaines d'entrées, des dizaines de sorties et des centaines de paramètres. Ce paradigme de résolution permet, en théorie, quand l'algorithme est confronté à des situations qu'il n'a jamais traitées auparavant, de malgré tout trouver la meilleure solution pour accomplir sa tâche. La limite de ce paradigme est dans la manière de trouver les paramètres, c'est là que se trouve « l'intelligence » de l'algorithme.
Le point central de l'utilisation d'un réseau de neurones est l'entraînement face à une tâche donné, on doit mettre en forme au mieux :
- l'entrée des capteurs (passage en gris pour une image, filtrage du bruit et discrétisation pour un signal analogique)
- la sortie (contrôle direct des actionneurs, prédictions comparées avec des données réelles)
- la correction des paramètres (quand et comment punir ou encourager le réseau par une descente de gradient)

Et plus particulièrement dans le cadre d'un apprentissage par renforcement :
- Le comportement à adopter malgré les prédictions du réseau (exploration/exploitation, minimiser l'entropie, maximiser la curiosité). C'est pour cela qu'on distingue explicitement l'agent de prise de décision du réseau de neurones, l'agent prendra sa décision en fonction de plusieurs facteurs et potentiellement en fonction de plusieurs réseaux de neurones, selon la politique de résolution de la tâche.



Réseau de neurones à 4 entrées, deux sorties et deux couches cachées



Détermination d'un comportement par apprentissage par renforcement

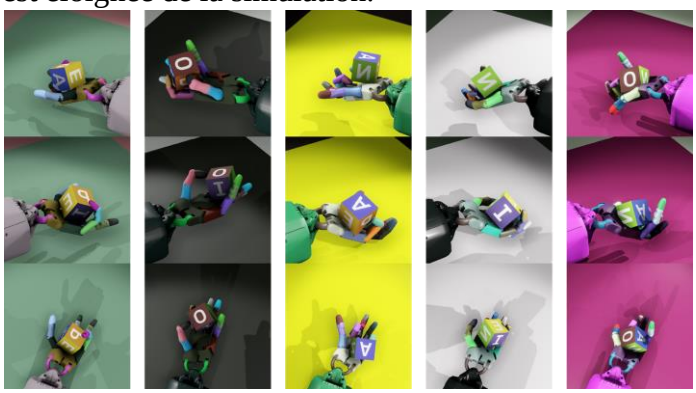
Par la nature même de l'IA, plusieurs points sont importants à prendre en compte :

La fiabilité du résultat :

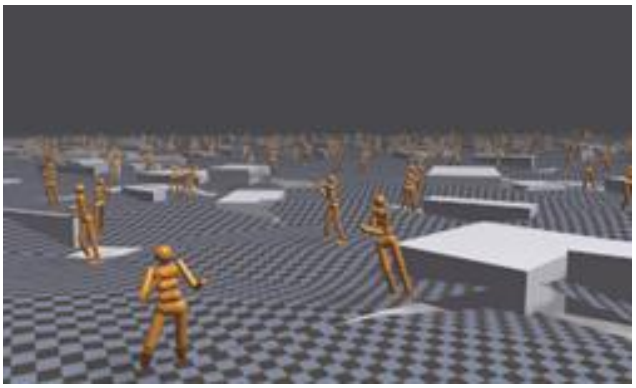
Le modèle de prédiction ou bien de contrôle doit permettre, via l'implémentation, une fiabilité dans le résultat, ainsi qu'une sécurité dans le contrôle direct. Malgré l'aspect « boite noire » du réseau de neurones, on doit pouvoir appliquer des règles spécifiques au contrôle de robot comme la limitation de l'accélération des moteurs, un signal peu bruité entre autres facteurs.

Généralisation :

Le modèle doit également permettre une bonne généralisation, c'est à dire que malgré l'entraînement spécifique qu'il a reçu, il doit pouvoir s'adapter à de nouvelles situations et fournir des performances stables. C'est est un point critique de l'application des modèles neuronaux à des données réelles, en effet, la prise de données réelles est coûteuse, on doit manuellement définir l'information que l'on voudrait prédire, et ainsi créer en base d'entraînement un large panel de situations différentes. La simulation nous permet facilement d'obtenir des informations venant de toutes les situations, sans extraction manuelle de l'information importante, pourtant la simulation permet difficilement de représenter la réalité. La méthode la plus utilisée actuellement est d'entraîner une simulation avec un maximum de paramètres différents, par exemple, les frictions, la couleur ou encore la gravité pour obtenir un modèle qui peut s'adapter à n'importe quelle situation réelle même si elle est éloignée de la simulation.



Multitude d'expérimentations de manipulation d'un cube par main robotique, Open AI



Multitude d'expérimentations de simulation d'un mannequin, Open AI

L’efficacité d’échantillon :

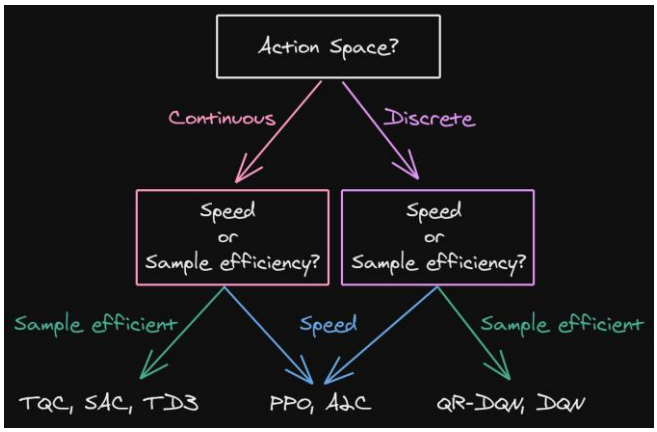
L’efficacité d’échantillon désigne, pour une quantité de données limitée, la capacité d’un modèle à obtenir une bonne performance. C’est un enjeu important, en améliorant ce point-donné, on réduit les coûts d’un modèle, ou on augmente sa performance. On peut obtenir une meilleure performance d’échantillon en enregistrant chaque étape et en réentraînant le modèle sur l’expérience passée. Cependant, cette technique n’est pas forcément permise pour tous les problèmes rencontrés. L’entraînement peut nécessiter une expérience « fraiche », c’est à dire, que la décision prise impacte l’entraînement, et qu’elle doit être proche du modèle actuel, l’IA ne peut parfois pas s’entrainer sur des données « aléatoires ». C’est la différence entre une politique d’entraînement dite « on-policy » et « off-policy » (respectivement, nécessite des données « fraîches », peut réutiliser l’expérience passé).

Exécution temps réel :

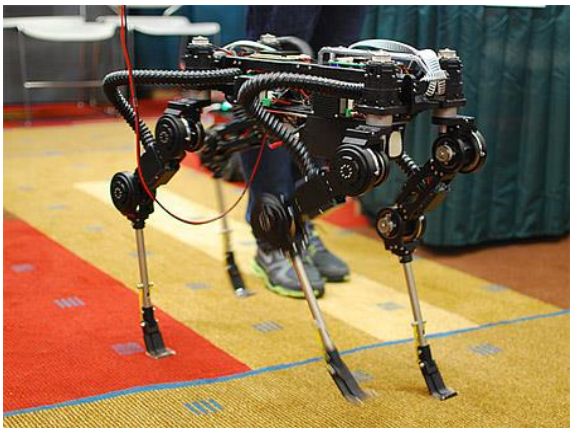
La robotique étant de plus en plus répandue avec des tâches de plus en plus complexes, l’étude devra également prendre en compte l’exécution en temps réel, en effet, dans le cadre d’application réelle, on embarque le modèle neuronal dans un microcontrôleur, avec ainsi, une puissance et un stockage limité. L’IA embarquée est un axe important de recherche, la possibilité que le robot exécute directement sur place l’algorithme de traitement est un point important, surtout s'il y a une multitude de réseaux de neurones exécuté en même temps. Dans le cadre de mon stage, l’exécution supplémentaire restera très correcte, mais on doit tout de même estimer la complexité des solutions techniques proposées, notamment pour le traitement d’images qui nécessite une grande puissance de calcul. La solution la plus courante pour faire du traitement d’images en temps réel est de prendre une fréquence assez basse (de 1 à 20 fps selon la nature de la tâche), pour ensuite interpoler la loi de commande finale entre chaque frame. De plus, l’étude portée sur les Fourier Features vise à être un maximum large pour permettre à d’autres chercheurs ou ingénieurs d’implémenter cette solution, et ce, pas forcément dans le cadre prévu initialement.

Cahier des charges de l’étude :

Dans le but de proposer une solution technique, on va devoir prendre en compte des contraintes supplémentaires, l’étude se devra de pouvoir traiter un maximum de problématiques différentes. En effet, par la nature du problème à traiter, les choix techniques et les contraintes des équipes de développement, les chercheurs seront contraints d’utiliser une politique d’entraînement spécifique. La proposition d’amélioration se doit alors de couvrir un maximum de problèmes et de cas différents, tout en ayant une vision des avantages et des inconvénients explicites, on ne peut pas se permettre de résoudre une tâche partiellement ou localement, l’utilisation de cette technique, en théorie adaptée, se verrait alors infructueuse une fois l’implémentation réalisée.



Classement des politiques d'entrainements en fonction de l'espace d'action et de ce qu'on recherche le plus

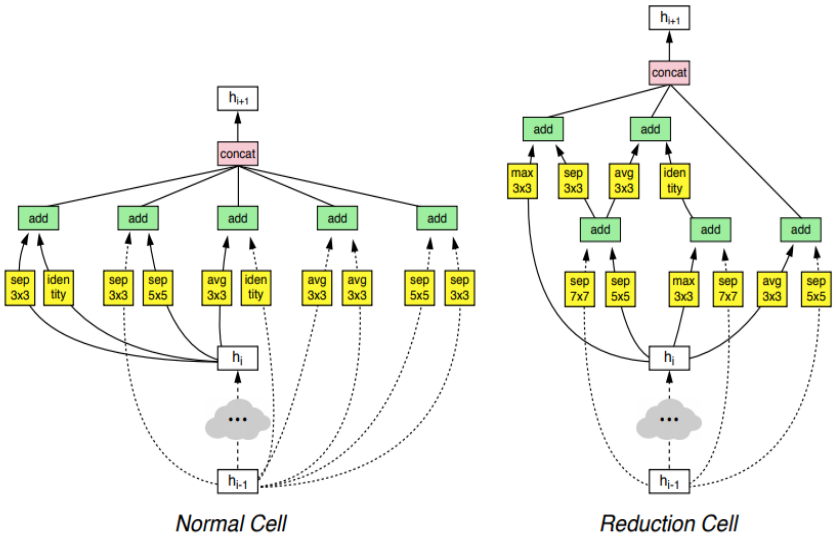


2Robot quadrupède contrôler par un apprentissage par renforcement, ANYmal

II : Etat de l'art des réseaux auto générés (AutoML) dans le cadre de la robotique

Présentation de l'AutoML:

L' AutoML (Automated machine learning) est un axe de recherche qui visent à concevoir des algorithmes d'IA plus précis et généraliste par la modification automatique du réseau de neurones. Ce domaine est vaste car la modification du réseau de neurones peut intervenir de plusieurs manières différentes, avec plusieurs méthodes de modification.



Réseau de neurones auto générés

Supervisé / Non-supervisé :

Dans le cadre de l'entraînement d'un réseau de neurones, on dit qu'une tâche est supervisée quand on donne directement ce que l'on doit nous obtenir comme résultat (classification d'images, détection de maladie, estimation du prix d'une maison par exemple). Alors qu'une tâche est non-supervisé quand on ne donne pas à l'algorithme la solution, on lui laisse prendre des décisions et on le récompense s'il atteint l'objectif donné initialement (investissement financier, jouer à un jeu vidéo ou contrôler un bras robotique par exemple).

Le domaine du non-supervisé est très propice à une « nouvelle robotique », en effet, en recherche, on fait l'hypothèse que là où le domaine de l'automatisme est limité par la nature de la tâche (prise de décision, environnement aléatoire, manipulation d'objet flexible) le domaine de l'intelligence artificielle peut amener une solution qui permettrait d'avoir un comportement plus adapté et robuste.

L'attrait des réseaux auto-générés est la possibilité de combiner plusieurs boucles d'apprentissages face à un problème, le modèle s'entrainant à trouver sa bonne architecture pour ensuite s'entrainer à accomplir cette tâche.

Gain en généralisation :

L'avantage sur ce type d'apprentissage est que l'on peut utiliser le résultat de l'algorithme sur l'échantillon de test afin de trouver la meilleure architecture. On entrainera le réseau non pas à obtenir le meilleur taux d'apprentissage sur sa base d'entraînement, mais on l'entrainera à obtenir une bonne généralisation sur sa base de test, ce qui incite le modèle à trouver une solution générale et optimale (qui résout au mieux tous les problèmes, du meilleur des manières).

Cahier des charges rempli :

En partant du principe que l'on a réussi à trouver un réseau de neurones optimisé pour une tâche d'apprentissage par renforcement, on remplit parfaitement le cahier des charges imposé par la robotique en milieu incertain contrôlé par intelligence artificielle. En effet, les réseaux de neuronaux autogénérés possède les même contraintes qu'un réseau de neurones classique, mais avec l'avantage d'être calibré spécifiquement pour la tâche que l'on souhaite traiter.

Fiabilité du résultat :

Concernant la fiabilité du résultat, on a la même problématique que pour un réseau classique, mais la fiabilité sera augmentée par la spécialisation du réseau dans une tâche donné. Cependant, il faut faire attention au surapprentissage, un réseau de neurones entrainé spécifiquement pour être performant dans un domaine risque d'avoir une moins bonne généralisation face à une situations inhabituelle. Comme dans beaucoup de sous-domaines de l'intelligence artificielle, un entraînement riche et varié est la meilleure de solution pour obtenir un modèle neuronal efficace et généraliste.

Généralisation

Similaire au point précédent, la généralisation désigne la capacité du modèl de contrôle à s'adapter à de nouvelles situations, un réseau de neurones ayant une architecture soumise à une fonction de récompense sera optimisé pour répondre à celle-ci, selon l'entraînement effectué, les réseaux les plus spécialisés et les plus sur-appris peuvent être malheureusement favorisés. L'entraînement d'un tel réseau est un point critique, les avantages et inconvénients d'un réseau classique sont amplifiés selon le type d'entraînement.
On se doit donc de veiller à avoir une base de données robuste, ainsi qu'une fonction de récompense bien réglée pour les tâches d'apprentissages par renforcement.

L'efficacité d'échantillon

Par la sélection du réseau qui aura le plus de récompenses, on peut émettre la conjoncture qu'on sélectionnera le réseau qui apprend le plus vite et le plus efficacement, et, par conséquent, le réseau autogénéré améliore l'efficacité d'échantillon. Ceci est valable pour une tâche supervisée, où les données traitées sont fixes. En revanche, dans le cadre d'un entraînement non-supervisé, l'entrée du réseau évolue en fonction des décisions prises par celui-ci, par conséquent, trouver la bonne architecture nécessitera un plus grand nombre de prises de données, ce qui peut annuler le gain d'efficacité d'échantillon.
Généralement, on emploie un réseau auto-généré pour obtenir une meilleur performance et fiabilité du modèle final (notamment dans des taches de classification d'images).

Exécution temps réel :

L'exécution en temps réel des réseaux de neurones auto-générés est un compromis, ils ont été de base conçue pour obtenir les plus grandes performances possibles dans la classification d'images. C'est un compromis dans le sens où l'on peut guider l'entraînement de recherche d'architecture vers des petits réseaux, aux prix d'obtenir un réseau moins performant mais bien plus rapide que le réseau de neurones idéale mais potentiellement énorme.

III : Problématique de l'entraînement non-supervisé face à l'AutoML

Difficulté principale de l'apprentissage supervisé :

Dans le cas où l'on entraine un réseau de neurones à prédire une réalité à partir d'une donnée brut (classification d'image, prédiction météo…), la difficulté pour un réseau dans une tâche supervisé est de déceler les éléments important de son entrée pour pouvoir prédire correctement la sortie.
L'utilisation de l'AutoML permet d'améliorer la performance du réseau face à une tâche donné.
Dans le cadre de la classification d'images, l'AutoML à permis d'augmenter les performances des réseaux désigner manuellement, cela s'explique par un plus grand est plus spécifique traitement de l'image par un réseau de neurones désigner spécialement pour cette tâche (un réseau entrainé à différencier des chats et des chiens sera probablement totalement différent d'un réseau différenciant les visages hommes et les visages femmes).

Difficulté principale de l’apprentissage non-supervisé :

Pour le cas où l’on entraine un réseau de neurones à sélectionner le bon comportement face à une tâche donnée, la difficulté pour celui-ci est de déterminer les bonnes prédictions à effectuer. On ne peut pas l’entrainer à nous donner la bonne sortie car on ne vise pas un comportement défini. C’est ce qu’on appelle un entraînement non-supervisé, contrairement à un entraînement classique, on ne va pas dire au réseau quel sortie prédire selon son entrée, mais seulement s'il a globalement correctement fait ses prédictions à la fin d’une séquence d'actions.
L’utilisation de réseau de l’AutoML est plus délicate que pour un apprentissage supervisé. On ne peut pas déterminer précisément si un réseau apprend mieux qu’un autre.

Un réseau neurones auto-générés sera bon pour mieux traiter les entrées par rapport à une tâche spécifique, mais pas forcément pour apporter une meilleure convergence du réseau global, plus une entrée est traitée, plus déterminer le bon comportement est compliqué. Il y a également le problème de la fonction de récompense, le modèle peut “tricher” et satisfaire la génération mais pas forcément la performance finale, par exemple, un réseau simpliste peut surpasser les plus spécialisés selon l’entraînement et l’adaptabilité demandés.
Cela vient du fait que la récompense obtenue n’est qu’un indicateur partiel de la performance de l’agent dans la résolution d’une tâche, on ne sait pas réellement s'il obtient un bon score car il a bien appris à s’adapter à son environnement, ou bien s'il a trouvé le meilleur moyen d’obtenir de la récompense.
Ce problème ne pose généralement pas trop de difficultés dans le cas d’un entraînement supervisé classique car, si la fonction de récompense est bien implémentée, l’agent, même s'il “triche” au début, devra quand même trouver le bon comportement pour obtenir la meilleure récompense.
Mais dans le cadre d’un réseau auto-généré, on ne peut pas effectuer l’entraînement en entier, on doit se contenter de la première portion, où l’agent le plus performant n’est pas forcément le plus adapté à la tâche.

L’apprentissage par transfert :

Une méthode possible (et utilisée par mon binôme), pour permettre la création d’une architecture optimisée dans le cadre d’un apprentissage par renforcement, est de d’abord l’entrainer sur une tâche supervisée d’une des solutions possibles jusqu’aux comportements finaux attendus. Une fois l’architecture déterminée, on l’applique à l’apprentissage non supervisé, pour lui permettre, en théorie, d’être mieux adapté à la tâche donnée. Avec un entraînement varié, on obtient un réseau de neurones capable de plutôt bien apprendre, mais il sera totalement inefficace pour la découverte et l’exploration qui seraient en dehors de son entraînement initial, car dans l’entraînement supervisé, l’agent ne sera confronté qu’à des situations prédéfinies, ses actions n’auront aucun impact sur la conduite du robot. En somme, il sera bon pour recopier mais mauvais pour apprendre, surtout dans des situations nouvelles. Le modèle reste cloisonné à son entraînement supervisé, le création manuelle du réseau par l’homme reste alors la meilleure des solutions pour obtenir un bon compromis entre sur-apprentissage et généralisation.
Cette difficulté ne m’a pas paru être dans un premier un temps un frein à ma recherche, l’utilisation direct de l’AutoML était compromise, mais il y a tellement de possibilité possible dans l’amélioration d’un modèle d’intelligence artificielle qu’il y a forcément une solution permettant d’améliorer celui-ci.

Ces difficultés ne restent tout de même pas trop problématiques, l'utilisation de cette technique est souvent couplée à un entraînement guidé par mimétisme. On va d'abord entrainer le réseau à recopier un comportement pré établi, pour ensuite le laisser en autonomie pour apprendre à maximiser sa récompense, cette méthode est efficace si l'on sait déjà quel est le meilleur moyen de contrôler les actionneurs, et que l'on veut y ajouter une couche d'autonomie.

IV Solution envisagées :

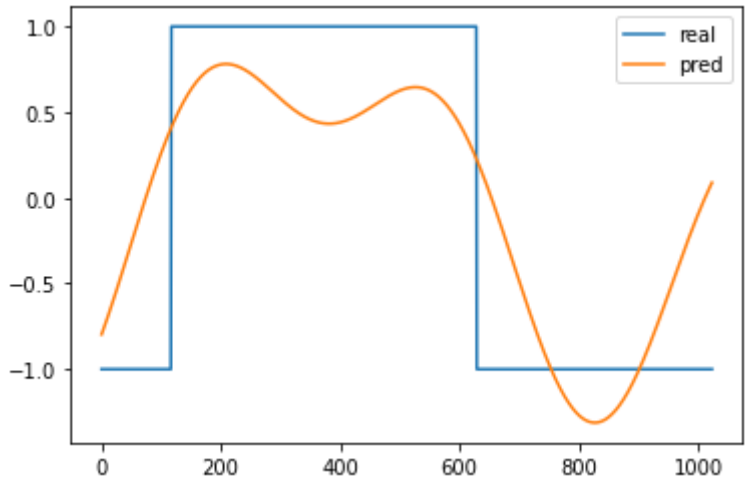
Face à cette problématique, j'ai essayé d'amener la technologie des réseaux auto-générés dans des points plus spécifiques, si le réseau de neurones entier peut être difficilement amélioré, une solution possible est de formuler d'une autre manière le contrôle des actionneurs pour arriver à notre but, et dans cette manière d'être contrôlé, y implémenter une recherche automatique de réseau.

Hypothèse et problématique générale :

Composer des mouvements robotiques à partir d'un réseau de neurones classique comporte de nombreux inconvénients et n'est pas adaptés à la tâche donnée, il faudrait trouver une solution qui permet de composer et de corriger un mouvement de la manière la plus simple possible. En effet, la composition de sigmoid ou de fonction relue nous amène à une bonne description de l'environnement, mais dans le cas du *reinforcement learning*, cette méthode est peu efficace car l'erreur se répercute sur toutes les fonctions qui ont créé le mouvement, et par conséquent le modèle de contrôle peut avoir du mal à déterminer et composer un mouvement qui lui permettrait d'atteindre son objectif.
L'hypothèse est double, le réseau de neurones classiques n'est pas capable d'interpréter une entrée ou de composer une sortie réelle. On part de l'hypothèse que le réseau de neurones est un outil statistique très puissant pour prédire une sortie, mais peu adapté pour déterminer un comportement robotique.

Composition de la sortie via un Transformée de Fourier Inverse :

Dans le cadre d'un contrôle robotique, la première idée de recherche possible est la création d'un signal via la recherche de coefficients, pour ensuite composer un signal via une transformation de Fourrier. En théorie, cela peut permettre de réaliser n'importe quelle commande, et d'ajuster les paramètres en temps réel selon la tâche, et ce, avec un réseau qui a moins de choses à trouver mais qui, avec une plus grande intelligence se prêterait bien à l'auto génération de réseaux de neurones. Cette idée n'a pas pu être bénéfique car l'erreur répercuté sur le réseau ne lui donne pas le bon comportement à adopter, en punissant le signal on a du mal à impacter correctement les coefficients. La limite de cette idée est très vite visible, le réseau de neurones n'a aucune idée de l'impact du choix des coefficients, et surtout dans le cadre de l'apprentissage par renforcement, la recherche du comportement par recherche de bon coefficient se fera par tâtonnement et non par compréhension, la composition d'un signal par transformée de Fourier inverse est donc délicate et inadaptée dans le cadre de l'apprentissage par renforcement.



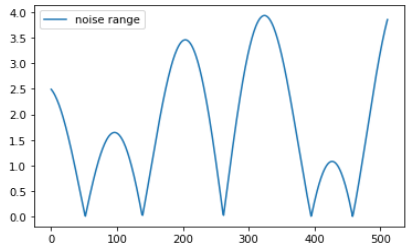
Approche d'une fonction carrée par composition de sinusoïde par un réseau de neurones

Décomposition du signal via un PID inversé

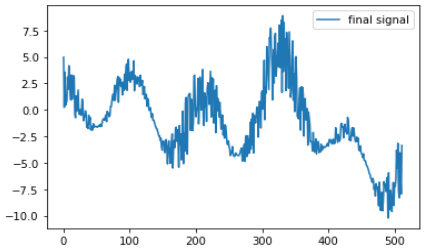
Une deuxième idée pour que le réseau de neurones comprenne mieux son environnement, notamment pour traiter des données venant d'une nappes lidar (capteurs beaucoup utilisé dans le monde agricole par la robustesse d'utilisation), est de décomposé le signal en sa proportionnelle (lui-même), son intégrale (à quel point le neurone a été stimulé auparavant), et sa dérivée (l'évolution du signal général). Malheureusement, cette idée était impossible à mettre en place dans le cadre d'un réseau de neurones :

- La proportionnelle est déjà présente.
- L'intégrale nécessiterait un réglage manuel d'un décalage, elle ne doit surtout pas se cumuler à l'infini si jamais le signal n'est pas cloisonné autour de zéro.
- Le dérivé est fortement soumis au bruit, et elle ne donne pas forcément une information temporelle exploitable et intéressante.

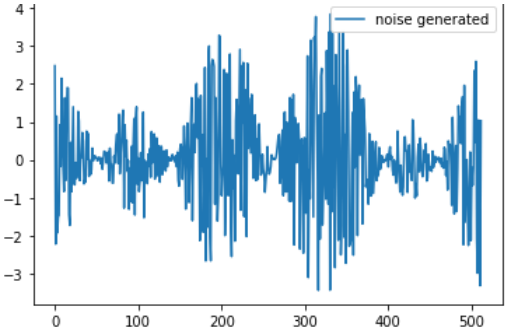
Pour mettre en place cette solution, il faudrait :
Un système d'oubli ou bien une régulation constante pour l'intégrale (ax+b)
Un lissage du signal pour la dérivée, mais on perd de son intérêt, une intervention manuelle et fine est nécessaire.
Ces solutions spécifiques à la robotique sont possibles mais on peut d'intérêt, le modèle serait amené à trouver une solution local défini par l'ingénieur qui a mis en forme ces données plutôt qu'une solution générale optimisé pour répondre à la problématique de la tâche. Il vaudrait mieux désigner une solution déterministe en automatisme qu'un réseau de neurones « boite noires ».
La recherche autour de la robotique met en avant une technologie à la base créer pour le traitement de texte et la composition de musique : une couche de neurones récurrents, et plus particulièrement le LSTM qui a à la fois une mémoire court terme et long terme, mais aussi un mécanisme d'oubli.



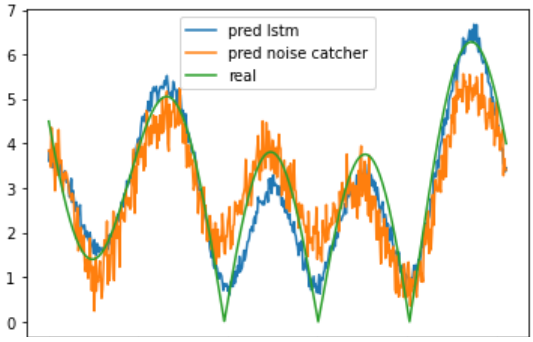
Amplitude du bruit



Signal bruité généré

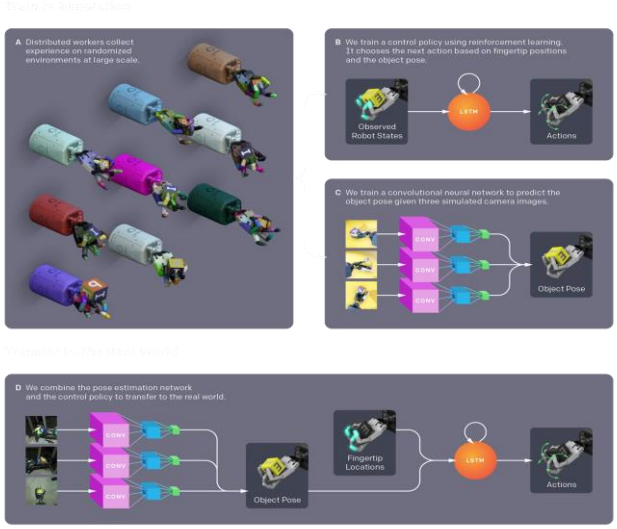


Bruit généré



Amplitude du bruit prédit à partir du signal, la méthode créée spécialement pour cette tâche n'est que légèrement meilleur qu'un LSTM de base

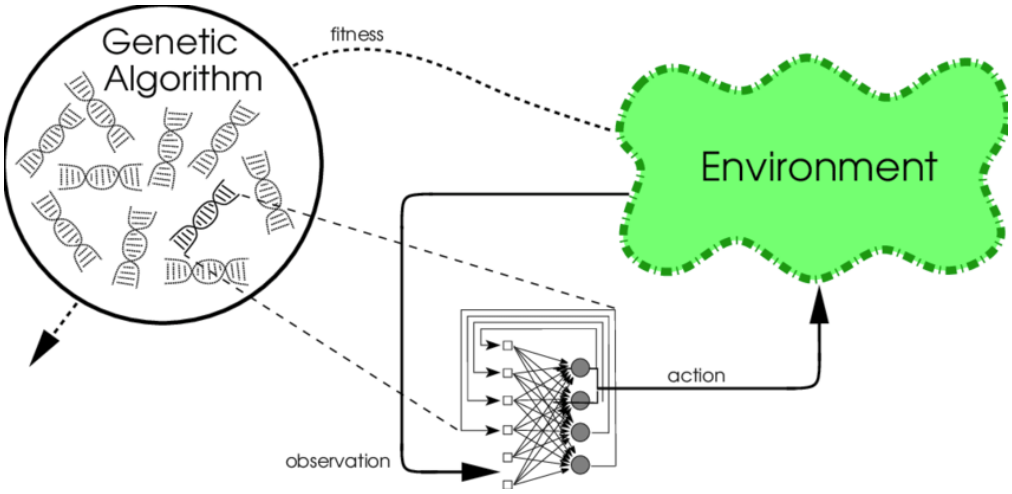
Mes recherches documentaires et mes expérimentations m'ont amené à découvrir les qualités du LSTM dans le contrôle robotique, mais c'est en réalité déjà connu et utilisé, son couplage à un réseau de neurones classique permet notamment de mettre en place une fonction de coût particulière sur la dérivée de l'action, pour pouvoir limiter l'accélération et par conséquent la force appliquée sur le moteur (pour le cas d'un contrôle direct, une solution plus simple qui combine intelligence artificielle et automatisme, est de mettre un PI qui lisse la commande des actionneurs).



Utilisation de LSTM dans le cadre de la manipulation d'un cube via une main robotique à l'aide de caméras

Evolution génétique :

Dans le cadre d'une recherche d'un réseau de neurones pour résoudre une tâche par apprentissage par renforcement, la résolution par évolution génétique est une méthode efficace pour traiter efficacement des problèmes simples, elle peut se traduire de plusieurs manières mais la base est de générer une population avec des comportements différents, la tester face au problème et sélectionner les meilleurs individus afin de générer une population dérivant de celle-ci.



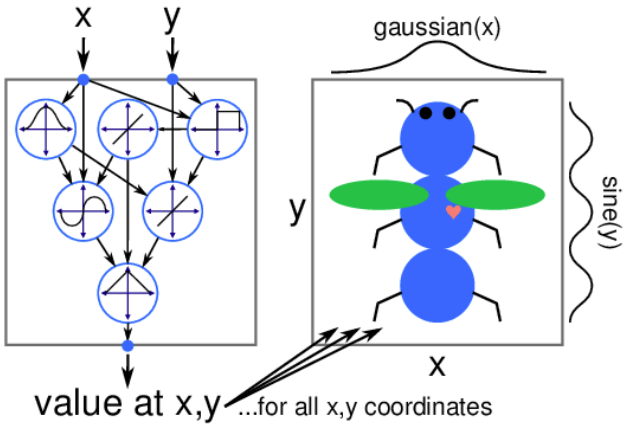
Cette technique peut amener une solution si le problème est simple, ou bien qu'on ai déjà une idée de comment résoudre le problème car la solution trouvée ne sera que la meilleure configuration du réseau de base, et par la méthode d'évolution, on ne pourra pas apprendre spécifiquement un comportement comme pour les méthodes plus traditionnelle, la découverte d'un comportement qui apporterait une meilleure récompense, ou bien une analyse plus poussée de l'environnement est difficile à voir émerger, le modèle n'apprend pas réellement, il garde juste les meilleurs.
Cette méthode est extrêmement efficace pour une tâche bien précise avec peu d'entrée et peu de sortie, mais trouve vite ses limites pour trouver un comportement adapté à une situation qui nécessite une meilleure analyse.

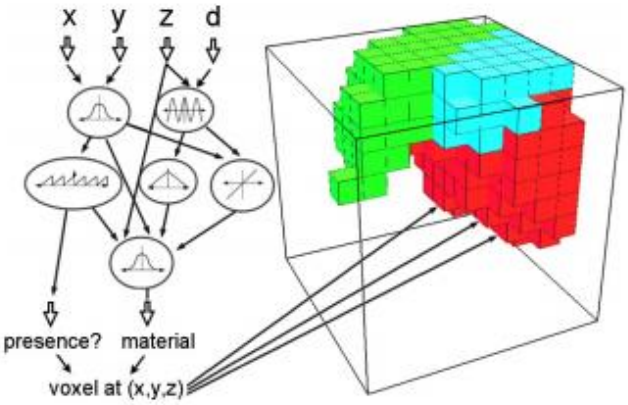
Compositional pattern-producing network (CPPN) :

La troisième idée revient aux réseaux de neurones autogénérés par la création de réseaux de neurones à fonctions mathématiques, au lieu de composer une solution par modification des poids, on va composer une solution par arrangement de fonction mathématique, ce qui permet, en théorie de traiter n'importe quel problème robotique.

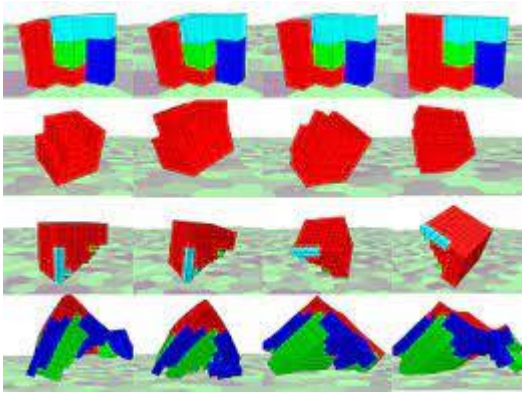
Une solution apportée par la neuro-évolution et les réseaux autogénérés est le réseau de neurones à génération de paterne, c'est un réseau de neurones particulier, contrairement à un réseau classique :

- Il est composé de fonction mathématique complexe
- Son architecture est « chaotique », les neurones sont connectés en un graphe maillé orienté acyclique (cad des entrées vers les sorties, chaque connexion a un sens unique, sans convention global, et ça ne fait pas de boucle)
- Son entraînement modifie son architecture et ses poids, en même temps
- Permet la redondance de paterne dans la sortie

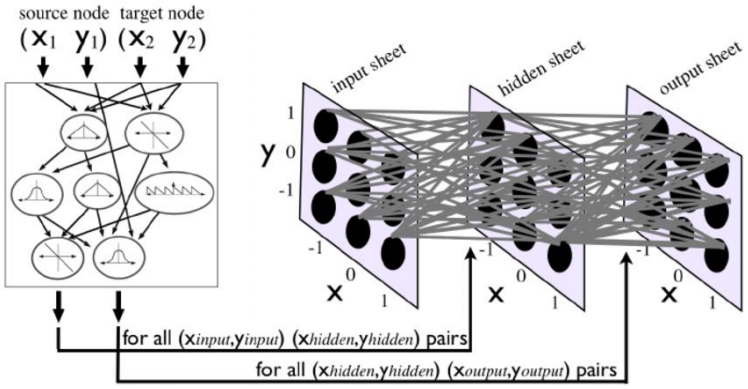




Ce réseau a beaucoup d'applications différentes, il peut générer dans une dimension simple, une solution à un problème par renforcement, il peut par exemple, essayer de ressembler à un modèle préexistant (à gauche), ou bien créer des solutions à des problèmes, ici la création d'une créature de partie flexible, solide et musclé (se contracte et décontracte), dans le but de faire avancer le plus loin possible celle-ci (à gauche et ci-dessous à gauche). Ou bien encore définir les poids d'un réseau, l'entrée du réseau va être le neurone et la connexion, et la sortie va être son poids.



Génération de "créatures" par CPPN, Jeff Clune



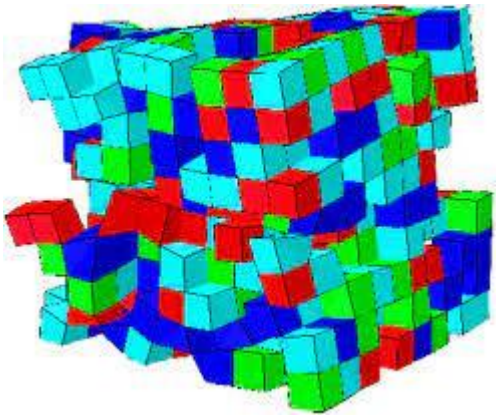
Détermination des poids d'un réseau de neurones par CPPN

Avantages des CPPN :

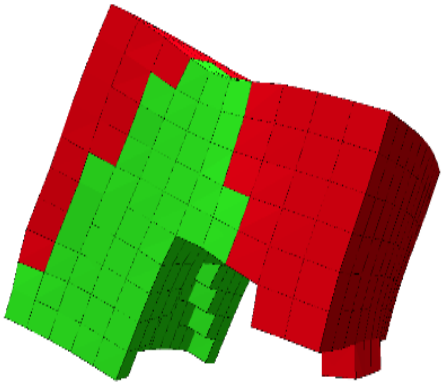
- Permet une forte exploration par son système multi-agent
- Permet une approche robotique, l'environnement ne peut être vu que par des fonctions mathématiques, c'est similaire à de la recherche de PID de manière automatique (empirique, pas de réelle compréhension de son environnement).
- Permet une solution homogène (les actions résultant de ce model seront peu bruités car composé de moins de fonctions, mais plus précise, contrairement à un réseau classique qui crée sa sortie à partir d'une composition d'une multitude de fonctions d'activation

Inconvénient :

- L'évolution se fait par génétique, par conséquent, le réseau de neurones n'apprend pas de nouveau comportement, ils les possèdent déjà et les meilleurs sont sélectionnés, cela implique une exploration par le test de toutes les solutions possibles. Par conséquent, la dimension d'expression du CPPN est limité, on peut l'entrainer à reconstituer une image, un modèle 3D, mais pour prendre en compte une dizaine de dimensions différentes comme c'est souvent le cas en robotique générale, son implémentation est impossible.



Entrainement par encodage géométrique direct



Encodage par encodage indirect

Comparaison entre une évolution par codage direct (à gauche) et codage indirect (à droite), on y voit explicitement à la fois l'avantage du CPPN et son défaut, il peut simplement approximer une solution mais serait incapable d'appréhender des subtilités locales, pour la résolution agricole et pour la robotique en général, un entraînement non-supervisé n'amène pas de solution générale et exploitable.

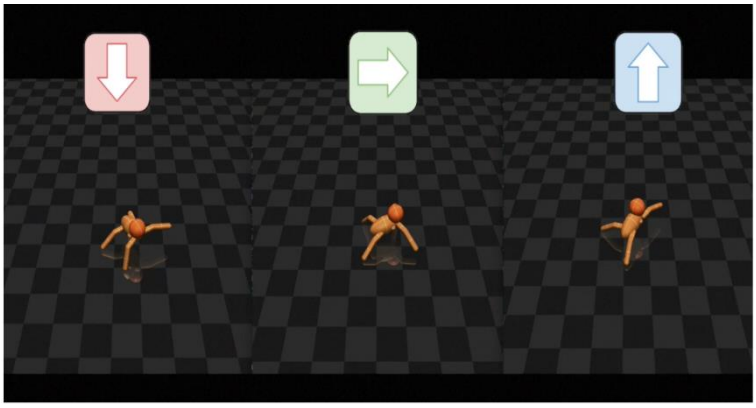
Hypothèse et utilisation d'un CPPN dans le contrôle robotique :

Dans le but de contrôle de robot, le paradigme de l'apprentissage par renforcement actuel n'est pas adapté au contrôle réel, sur chaque timestep, le réseau de neurones peut prendre une action différente de ses actions précédentes, il n'y a pas de prise de décision global. Dans le but d'améliorer la performance générale du model de contrôle direct, j'ai orienté mes recherches dans la séparation de la prise de décision avec l'exécution, en effet, en robotique, surtout quand la dimension d'actions est élevée, les actions comme les décisions peuvent être difficiles à mettre en place, la séparation des deux permettrait théoriquement une meilleure convergence. L'idée de recherche était d'avoir un réseau classique dédié à la prise de décision (ce qui annulerai le défaut de l'inadaptation d'un CPPN à prendre en compte une dimension élevée), et un réseau CPPN auto généré pour exécuter ces décisions.
Dans le fonctionnement optimal, le réseau classique à l'aide de son observation prendra la bonne décision, et le CPPN par cette décision appliquerai les bonnes fonctions mathématiques de commande.
Dans l'entraînement idéal, une fois la bonne décision prise face à une situation, le réseau s'entraînerait uniquement à effectuer la bonne action. Si jamais ces deux réseaux étaient correctement assemblés et entrainé, on pourrait avoir une méthode qui peut appréhender des situations complexes, avec une multitude de comportement possible, le tout avec un entraînement rapide et précis.

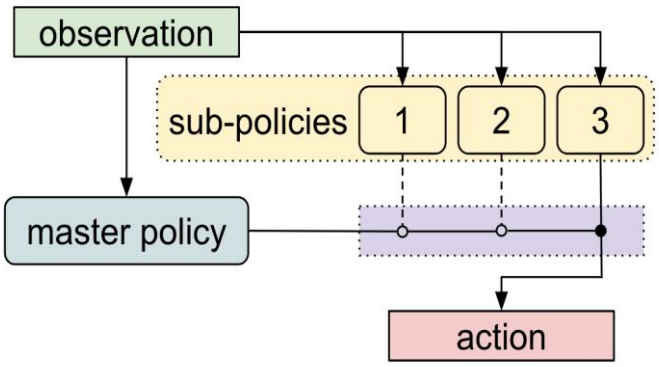
Cet axe est théoriquement un bon axe de recherche pour l'utilisation de réseau auto-générés dans le contrôle de robot par intelligence artificielle, mais cette voie n'est pas implémentable. Il est impossible pour le modèle de savoir si c'est sa décision ou son action qui était bonne, lors de son entraînement, par la nature du réseau de neurones et de son entraînement non-supervisé, la prise de décision est « aléatoire » et bruité, par conséquent, composé et évaluer à partir de ce réseau est impossible. C'est un problème irrésolvable, il y a deux équations à résoudre avec deux variables à trouver, et pour trouver une variable, il faut déjà avoir trouvé la deuxième.

Il y a néanmoins deux moyens de résoudre cette situation, mais malheureusement non adapté dans notre étude :

La première est d'entrainer manuellement le CPPN à effectuer des actions bien précises, le MLP sera quant à lui entrainer uniquement à sélectionner le meilleur comportement.
Le seul moyen actuel d'arriver à ce genre de résultat est d'entrainer manuellement les différentes actions, et d'entrainer un deuxième réseau de neurone à prendre les bonnes décisions.



Contrôle d'une araignée à sortir d'un labyrinthe, Open Ai



Prise de décision par bloques indépendamment entrainer

La deuxième méthode serait d'extraire les décisions de manière contourner via un auto-encoder, le but serait de convertir les entrées en une prédiction (ou bien l'entrée de nouveau), en passant par une matrice plus petite ayant extrait les données les plus importante, par exemple, pour le contrôle d'un véhicule, on pourrait entrainer un premier réseau à prédire les forces et moments appliqués sur chaque roue, à partir de la séquence d'angle de braquage, d'accélération, de vitesse du véhicule… Le tout passant par la plus petite couche neuronale possible. Après ce premier entraînement, on pourrait prendre la matrice intermédiaire en entrée du CPPN, pour qu'à partir des données les plus importante, on définisse une fonction mathématique permet le contrôle souhaité du véhicule. Cette solution est impossible à mettre en place car il n'y a pas de compression possible des données, chaque articulation est indépendante, et de plus, il n'y a également pas de données prévisibles permettant l'extraction des commandes.

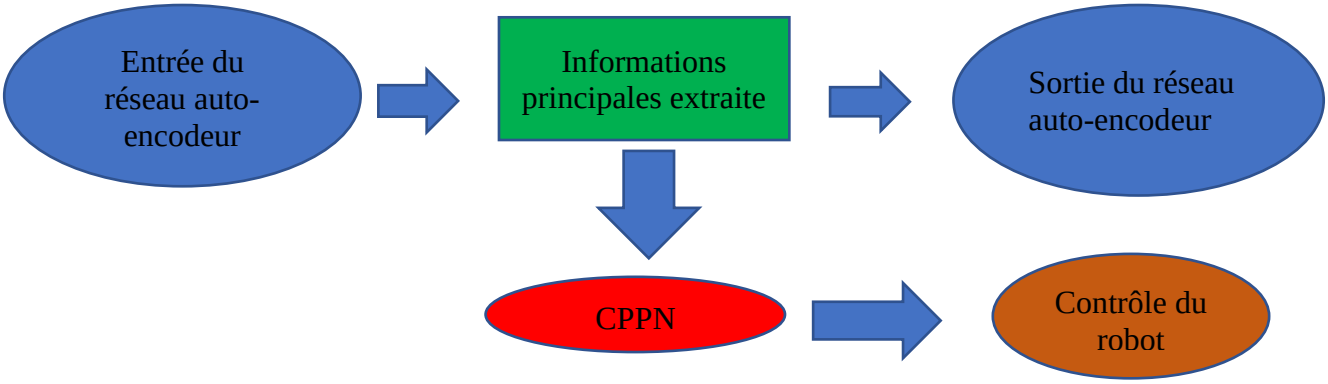


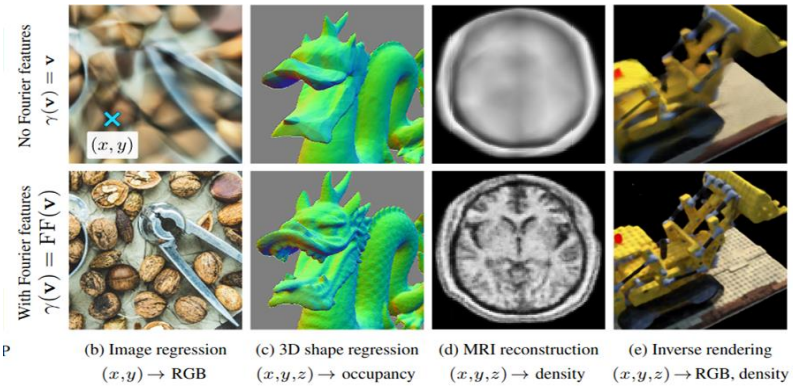
Schéma du fonctionnement possible de la création d'un mouvement par CPPN

V Etude :

Analyse de l'existant avant le démarrage de l'étude :

Méthode d'interprétation de l'entrée :

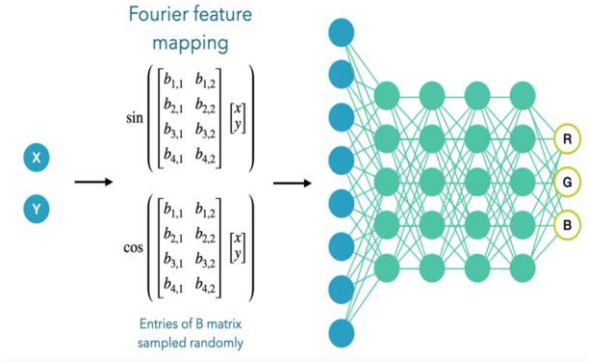
À la suite de la lecture d'un article scientifique (Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains, Tancik et al.), le but étant, similaire à la reconstitution d'une image par CPPN, était de reconstituer une image avec un réseau de neurones qui prenait en entrée les coordonnées d'un pixel, et en sortie la couleur de ce pixel. La technique consiste à décomposer l'entrée pour pouvoir en tirer plus d'information et que le réseau de neurones puisse mieux composer son résultat.

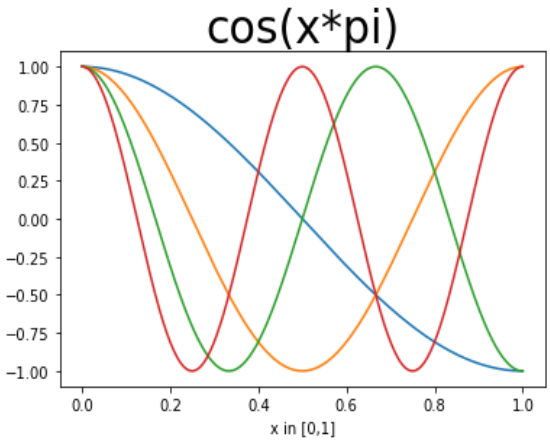


Comparaison entre l'interprétation linéaire de la position et l'interprétation fréquentielle, Tancik et al

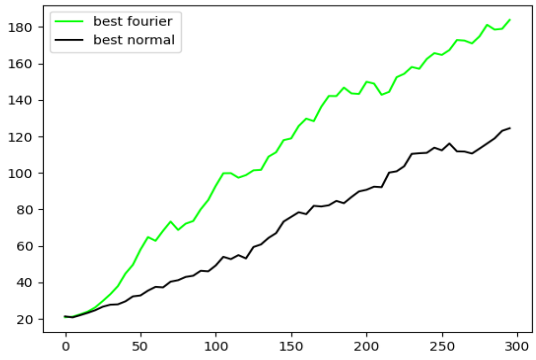
Fonctionnement :

L'entrée va être périodisée et exprimée selon plusieurs fréquences, ce qui permet au réseau de neurones une compréhension local et globale de son espace, permettant notamment, dans la reconstitution d'une image une meilleure différentiation de deux pixel côtes à côtes.

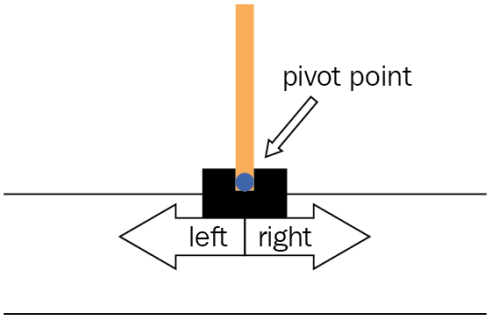




Suite à cette découverte, j'ai eu l'idée d'appliquer cette technique au domaine de l'apprentissage par renforcement, mais suite à mes recherches, quelqu'un l'avait déjà fait, et on avait montré que le traitement par Fourier pouvait permettre une meilleure convergence d'un réseau de neurones face à une tâche d'apprentissage par renforcement.



Expérimentation personnelle de l'interprétation de l'entrée sur le problème du CartPole



Problème du CartPole, le but est de faire tenir en équilibre le bâton en bougeant de droite à gauche son chariot

Critique de l'existant :

Malgré le brillant article scientifique (Fourier Features in Reinforcement Learning with Neural Networks, Brellmann et al. , 2020) théorisant l'impact des « Fourier Features » dans le l'apprentissage par renforcement, beaucoup de problématiques restaient présente :

- La consommation en plus de cpu
- L'adaptabilité de la solution à plusieurs politiques/problèmes
- L'évaluation de certaines variantes théorisées dans d'autres domaines

Au global, on a appris que les Fourier Features pouvaient être, dans certains cas, sous certaines configurations, une bonne solution, mais sa mise en place, surtout dans le domaine de la robotique, reste encore incertaine.
La critique a apporter à cette étude est qu'il n'y a pas assez de problèmes et de politiques étudiées, mais cette critique est valable que si on regarde l'étude dans le but d'amener la technique présentée dans une application robotique en temps réel. L'étude portée n'avait pas pour but de tester globalement la méthode, mais plutôt de poser la première pierre, de prouver que le pré-traitement par une Fourier Feature peut permettre une meilleure convergence, prouver l'hypothèse pour la transformer en une réalité scientifique.

Proposition de solutions

J'ai donc décidé de mener une étude ayant pour but de :

- De tester un maximum de problèmes différents
- De tester les « meilleurs » politiques d'entraînements (les plus utilisés/connus)
- De tester un maximum de nouvelles méthodes d'extraction de données
- D'évaluer les avantages et les inconvénients des méthodes d'extraction
- De faciliter la mise en place de modèle d'apprentissage par renforcement dans le monde de la robotique par une meilleure convergence face à une quantité de données limité

Il y a également une technique tirée par hasard d'un article scientifique qui compare des fonctions d'activation dans le but de re créer un environnement 3D, et elle a montré que la fonction sinus était une bonne solution. (Implicit Neural Representations
with Periodic Activation Functions, Sitzmann et al., 2020)

Grace à cette idée, ça nous permet, en théorie, d'obtenir une variante de Fourier Feature, où la périodisation serait apprise par le réseau lui-même, cette technique, apporte, en théorie, énormément d'avantages, elle peut s'utiliser comme une couche normale mais avec simplement une fonction périodique en fonction d'activation.

Etude :

Objectif et cahier des charges :

Cette étude a été motivée par la volonté de répondre aux problématiques de la robotique agricole, et plus largement pour la robotique réelle :
- Tourne en temps réel
- Efficace dans la compréhension des données réels
- Une solution qui généralise bien
- Utilisation d'une mémoire temporelle

Hypothèse de l'algorithme :
- Efficace sur toutes les politiques, pour tous les problèmes
- Surcout de calcul ≤ Gain de convergence
- Peu de sur-apprentissage par la conversion fréquentielle

Avantage déjà présent :
- Pas d'impact sur l'utilisation de cellule récurrente

Notre but idéal est de pouvoir répondre positivement à ces problématiques, tout en gardant en vision critique et global des solutions proposées. Dans le cadre de l'intelligence artificielle, un algorithme peut être théoriquement meilleur qu'un autre, mais en regardant de plus près, il peut « tricher » pour arriver à ses fins. La vision du chercheur et le fonctionnement du modèle neuronal peuvent fortement biaiser une étude, on se doit de rester détacher du cahier des charges avant de mener à bien l'étude, et puis seulement d'évaluer les solutions techniques proposées.

Mise en place des problèmes :

Dans le cadre d'une étude d'évaluation de différent modèles de contrôle par apprentissage par renforcement, mon étude se devait d'avoir une multitude de problèmes à résoudre, et plus particulièrement des problèmes à la fois bien représentatif de problèmes généralement traité, et à la fois spécifique à la vision et a l'application de l'étude. En effet, beaucoup d'étude se concentre sur la résolution de jeux vidéo, ce qui est très pertinent dans la recherche dans le domaine des intelligences artificielles autonome, mais dans notre cas, on se concentre sur la résolution de problèmes physiques.

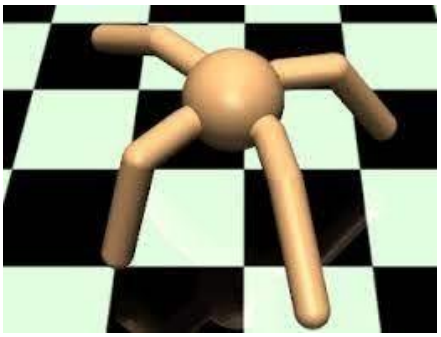
Les problèmes se devaient également d'avoir une caractéristique importante dans le domaine de l'apprentissage par renforcement, c'est d'avoir un espace d'action à la fois discret et continue :

Discret :

La prise de décision du réseau de neurones se porte sur plusieurs actions différentes, comme pour un jeu de Snake ou un jeu d'échecs

Continue :

La prise de décision vise à déterminer une ou plusieurs valeurs de manière linéaire, généralement pour le contrôle d'élément physique (il arrive quelques fois qu'on ait recours à la discrétisation des valeurs physiques pour permettre une prise de décision discrète pour la résolution d'un problème continue). Par exemple, contrôler une araigné en déterminant les forces et moment appliqué sur ses articulations.



Araignée articulée "Ant", mujoco, Open AI

Dans le cadre de la recherche en intelligence artificielle, pour que mon étude soit valide, l'étude impose aux problèmes à traiter de :

- Pouvoir être reproduit
- Pouvoir être modifié
- Pouvoir facilement être implémenté
- Pouvoir simuler un robot de manière customisable et correct physiquement
- Pouvoir être facilement exécutable

Par conséquent, ils devaient avoir comme caractéristiques :

- Être open source
- Être connu et validé dans la communauté scientifique

Le choix s'est donc porté sur les environnements de gym, une librairie développée par OpenAI.

Mise en place des politiques d'entraînement

La politique d'entraînement est à la fois, comme le model via un modèle neuronal va prendre une décision, mais aussi comment il va être entrainé. J'ai au début de mon stage étudié quelques uns de leurs fonctionnements, mais leurs implémentations de manière stable et adapté à une multitude de problème est délicate.

Les politiques d'entraînement doivent avoir les mêmes contraintes et caractéristique que les environnements de l'étude
J'ai choisi d'utiliser les politiques d'entraînement de stable-baseline3, une librairie également développée par OpenAI et étant parfaitement compatible avec les environnements de gym.

Présentation et critique de Open AI :

Durant mon stage, j'ai utilisé beaucoup d'outil provenant d'OpenAI, on se doit de rester toujours méfiant et éclairé sur les outils et les publications scientifiques qui nous sont proposées
Open AI est une entreprise américaine à but lucratif plafonné, cherchant à développer et commercialiser l'intelligence artificielle.
Actuellement, dans le cadre du domaine de l'apprentissage par renforcement, elle permet de financer des recherches et des outils open source pour tous. C'est avantageux mais il faut avoir conscience des défauts de ce système, la recherche se concentre autour d'intérêts privés d'une entreprise pas forcément éthique et bienveillante, mais par l'aspect open source et communautaire, on évite les dérives. Je ne suis pas très informé de ces sujets là mais il y a aussi la question de l'aspect propriété intellectuel sur les créations autour de leurs outils, ou les recherches publiées par des chercheurs financé par OpenAI. Dans le cadre de mon stage, menant une recherche open source sans brevêt industrielle ou découverte scientifique majeur en jeu, l'utilisation d'outil externe et privée n'entre pas en conflit avec mon stage.
Par conséquent, dans le cadre de mon stage, l'utilisation des outils proposés par Open AI m'a été bénéfique.

Mise en place des transformations :

La première étape a été de bien comprendre les subtilités de la mise en place de l'extraction de données, pour ensuite les implémenter via pytorch (que j'ai heureusement, à cette étape de mon stage, bien appris à maitriser grâce à toutes les expérimentations précédentes).

Pour résumer les différentes transformées de Fourier leurs variantes (l'explication explicite et complète se trouve en annexe), il y a méthodes principales d'extraction de données :

La Fourier Feature :
Pour la première couche, elle combine sur chaque neurone, la somme pondérée de chaque entrée du réseau, et selon l'ordre, chaque combinaison possible est présente, cette méthode est avantageuse pour des problèmes avec un petit nombre d'entrée, mais devient vite inutilisable par sa théorie.

Par exemple, pour deux entrées (x,y) et une FF d'ordre 2, on aura sur la première couche :

```
cos(x * 0 + y * 0)
cos(x * 0 + y * 1)
cos(x * 0 + y * 2)
cos(x * 1 + y * 0)
cos(x * 1 + y * 1)
cos(x * 1 + y * 2)
cos(x * deux + y * 0)
cos(x * deux + y * 1)
cos(x * deux + y * 2)
```

nb neurones = (order + 1)^nb_inputs

On comprend vite à la fois la puissance et l'inadaptation de cette technique pour traiter des problèmes de robotique où l'on a facilement deux ou 3 informations par articulation.

La Fourier Light Features :

Similaire à la FF, mais avec pour avantage de ne pas combiner les entrées, chaque entrée va être projetée sur plusieurs neurones, chaque neurone représentera alors une fréquence différente

Par exemple, pour deux entrée (x,y) et une FF d'ordre 2, on aura sur la première couche :

cos(x * 1)
cos(x * 2)
cos(y * 1)
cos(y * 2)

nb neurones = ordre*nb_input

Cette technique est plus simple mais possède l'énorme avantage d'avoir un nombre de neurones en sorti strictement proportionnelle au nombre d'entrées et à l'ordre.

Avec ces deux méthodes (FF et FLF), il y a aussi 3 paradigmes différents de mise en place :

- Déterministe (employé dans l'explication précédente) :

Chaque coefficient est déterminé par une équation, c'est la plus simple et la plus proche d'une transformée de fourier classique

On l'exprimera sous forme DFF / DFLF (Determinist Fourier Feature, Determinist Fourier Light Featured)

- Aléatoire :

Chaque coefficient est déterminé de manière aléatoire, cela permet d'avoir un ordre de décomposition plus grande en limitant les hautes fréquences.

On l'exprimera sous forme RFF / RFLF (Randoms Fourier Features, Random Fourier Light Featured)

Appris :

C'est la variante « découverte » sur les réseaux de neurones à fonction d'activation périodique

On l'exprimera sous forme LFF / LFLF (Learned Fourier Features, Learned Fourier Light Features)

Mise en place du système de recueille de résultat :

Pour pouvoir récolter autant de résultat, de manière asynchrone (exécution paralléliser) on a mis en place un système de log par écriture ligne à ligne dans des fichiers csv, chaque fichier représentant via son nom et son emplacement :

- L'environnement
- La politique
- L'extraction
- La variante d'extraction
- L'index (pour faire tourner plusieurs fois une même expérience)

Chaque fichier de résultat contient, pour chaque bloque de 1000 étapes, un couple de valeur :
(le timestep, la vitesse d'exécution, la récompense)

Ce système est loin d'être parfait, notamment pour le classement et la manipulation de cette multitude de données, mais il est efficace pour l'usage que j'en ai fait.

Nombre d'expériences à exécuter :

Cette étude, par les variations internes possible à chaque expérience, a nécessiter beaucoup de calculs :

- 5~10 expériences : avoir des expériences redondantes pour faire une moyenne générale et annuler l 'aspect aléatoire de chaque entraînement
- 17 environnements d'exécutions : les problèmes à traiter, un nombre conséquent pour couvrir un maximum de possibilité et ne pas trop favoriser une méthode d'extraction qui serai adaptée localement à un problème spécifique et pas globalement.
- 3~5 politiques d'entraînements : la manière de prendre une décision et d'entrainer le model (ça varie car toutes les politiques ne sont pas adaptés à tous les espaces d'actions (continue/discret))
- 10 méthodes d'extractions : manière de prétraiter l'entrée du réseau de neurones, là est le point central de notre étude
- 3 variations : modification de l'ordre de Fourier pour les méthodes d'extractions

Ce qui nous donne au total environ 15 000 expériences de 1.000.000 de steps chacune (c'est beaucoup)

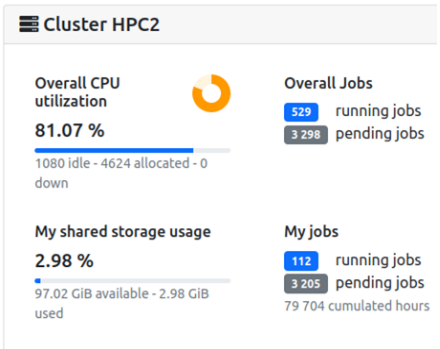


Tableau de bord de la ferme de calcul lors du lancement d'une grandes quantité de calculs

Architecture de programmation :

Par les contraintes de ce projet, l'architecture de code doit pouvoir permettre des modifications simples et un réassemblage aisé, en effet, c'est du développement et du test en continu, une des plus grandes difficultés de ce projet a été de tester et d'implémenter les différentes solutions techniques. Par l'aspect intelligence artificielle du problème, il est plus dur de savoir si le programme « marche » comme il faudrait, le réseau peut s'exécuter normalement, et même donner un résultat convenable alors que les transformations de matrices sont potentiellement mal agencées.

J'ai opté pour une structure de code un maximum atomique, séparé un maximum chaque fonctionnalité, chaque paramètre pour pouvoir corriger facilement les erreurs découvertes que bien plus tard, je n'avais jamais mené ce genre d'étude auparavant, j'ai donc pris en compte les « erreurs de débutant » que j'allais commettre.

L'optimisation des calculs a également eu une place importante dans ma conception algorithmique, de nombreux choix et changements ont été faits :

- Suppression du stockage des modèles neuronaux et des mémoires d'actions (trop de place, pas d'intérêt particulier)
- Passage d'un encodage des logs du json (encodage explicit) à du csv (encodage implicite, mais plus léger)
- Arrondi des valeurs pour conserver que l'information importante

Utilisation d'une ferme de calcul :

Malgré une bonne architecture et une bonne optimisation, exécutés tous ces calculs sur un simple ordinateur personnelle aurait été impossible, j'ai donc eu la chance de pouvoir avoir accès à un super ordinateur de calculs parallélisé (le mésocentre). Notamment que l'exécution finale n'est pas simple, il faut déjà faire tourner beaucoup de calculs afin de déterminer les variantes viables de chaque transformation.

Présentation technique du mésocentre :

- Connexion ssh via tunel/vpn
- Upload de code via git
- Fonctionne par une multitude de tâches en liste d'attente, environ 200 jobs en parallèle possible, soit 20 à 50 fois la puissance de mon ordinateur de stage (un bon ordinateur de base)

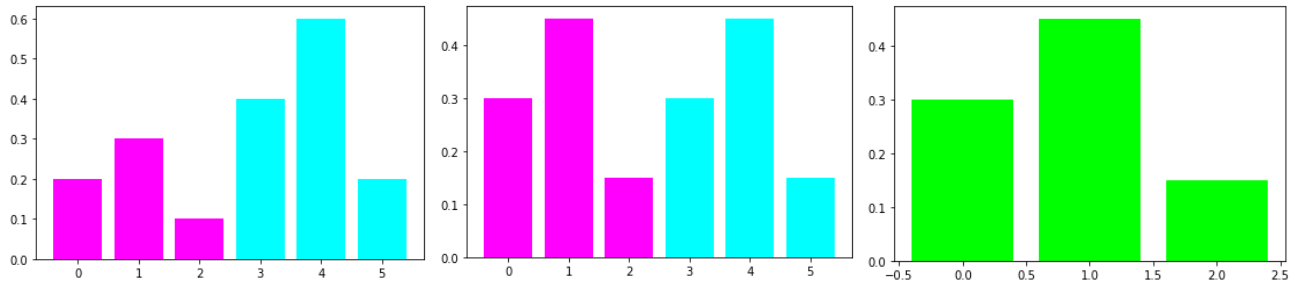
Mise en place et suivi des solutions

Une fois l'étude menée, il faut traiter, analyser et mettre en forme le résultat final. Par chance, malgré la grande quantité d'expériences

Une fois que tous les calculs exécutés, il faut mettre en forme le résultat pour pouvoir en tirer une conclusion :

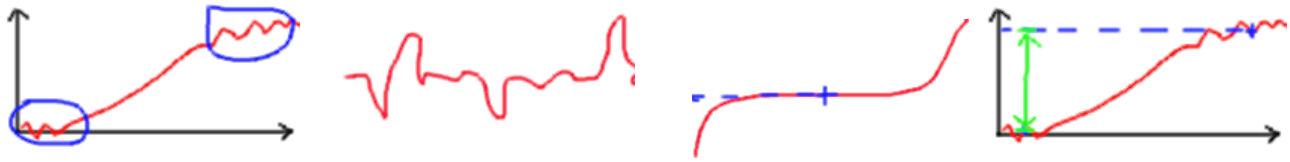
<u>Mixer les différentes expériences pour faire une moyenne :</u>

Pour tirer une conclusion de la consommation, garder problèmes par problèmes, politique d'entraînement par politique d'entraînement n'a pas beaucoup d'importance, mais un mix de tout en même temps est impossible, il y a des écarts entre les problèmes traiter et les politiques. Il nous faut donc normaliser en fonction de la moyenne de toutes les méthodes d'extractions pour obtenir un résultat général pour une politique en particulier, puis recommencer l'opération pour obtenir globalement, la différence de consommation de chaque extraction.
Par la manière que j'ai eu de programmé le tout, le matériel spécifique, les mises à jour spécifique, autant pour la pertinence que pour la précision, donné une valeur d'exécution n'a pas de sens, il vaut mieux parler d'écart relatif.



Données brutes, violet et bleu sont 2 situations différentes → *Données normalisés* → *Moyenne des données*

L'étude de gain de récompense est quant à elle plus délicate, pour définir le gain relatif de récompense de chaque méthode d'extraction, on doit définir une borne minimum global et une borne maximum pour chaque expérience, tout en éliminant les perturbations liées à la simulation. On va sélectionner la zone à étudier, on trie les nombres de manière croissante, et on prend la médiane.

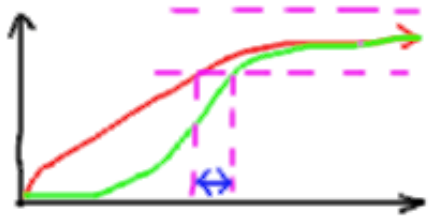


Selection des zones → *Zone brute* → *Trie et sélection de la médiane pour éliminer les "fausses" données* → *Ecart relatif entre le min et le max*

Pour la borne minimum, on va faire une moyenne de tous les problèmes, toutes les politiques.

L'étude de convergence est plus délicate, on doit comparer la rapidité que prend un algorithme pour atteindre les récompenses d'un autre, à quel point il apprend plus vite et a besoin de moins de données qu'une autre.

Pour ce faire, on définit une fenêtre de « résolution » du problème, défini à + ou – 20 % de la récompense maximale, et comparé le nombre de step pour que l'algorithme rentre dans cette zone :



Estimation du gain de convergence

Ces méthodes sont sensibles et empiriques par rapport à la courbe d'entrée, mais pour l'étude menée de comparaison relative, le résultat obtenu est suffisant et qualitatif.
Une autre méthode plus précise aurait été, par régression linéaire, d'approximer la courbe par une fonction prédéfini (ax+b pour le plus simpliste, dans notre cas d'étude de convergence, a*e^b*x +b aurait été plus adapté), puis comparer directement la dérivée de ces formules pour à la fois, comparé la vitesse de convergence, et la zone.

VI Conclusion de l 'étude

(Résultats finaux encore inconnus, mais déjà des résultats intéressants)

L'étude à confirmer les travaux précédents, une méthode extrayant mieux l'information permet de mieux comprendre l'espace et de mieux converger vers une solution globale.
L'apport de l'étude montre également, à travers toutes ces expériences, que cette méthode d'extraction est efficace mais soumise à une forte part de variations selon l'extraction, le problème, et la politique d'entraînement employée. **(En théorie)**

Cette étude permet également, d'apporter un premier pilier regroupant et théorisant les différentes techniques d'extractions, et par la suite, par un travail personnel ou globale, amener d'autres méthodes d'extractions, et converger d'ici quelques années à une méthode meilleur que les autres, de la même manière que les politiques d'entraînement se sont théorisée, développer, tester et déployer, c'est un travail collaboratif multi-domaine qui nécessite une collaboration global et organiser.

Pour pouvoir répondre à cette ambition, il faut :

- Finir le compte rendu scientifique, corriger l'anglais et le soumettre aux pairs pour le valider et le rependre aux plus grands nombres
- Mettre correctement en forme le travail github pour améliorer la reproductibilité et les forks de l'étude
- Implémenter et soumettre ces couches de transformations à l'intérieur des librairies pytorch et tensorflow, appuyer de la théorie et des conclusions du compte rendu l'étude

Les points améliorables de cette étude :

Durant mon travail et ma découverte de toutes les problématiques liées à l'étude, j'ai pris connaissance de certaines problématiques que j'aurai pu prendre en compte plus tôt et améliorer ma rigueur et mon efficacité :

- Base de données SQL

Pour l'enregistrement et la manipulation de cette énorme quantité de données provenant d'endroits différents, je privilégierais l'implémentation d'une base de données SQL, un peu plus dure à mettre en place mais plus efficace pour l'interprétation des résultats finaux.

- Utilisation de git

J'ai utilisé git en tant qu'un simple dépôt de code, mais l'utilisation multibranches aurait pu être intéressante pour visualiser clairement les modifications et l'historique de changement. Cette contrainte d'organisation permettrait, paradoxalement, une plus grande liberté de changements global du code, par sa conception, j'ai dû prendre en compte et me souvenir de toutes les modifications effectuées, ce qui était peu ergonomique, surtout quand il faut prendre en compte plusieurs exécutions sur plusieurs projets différents (déterminer la puissance consommée ne pouvait se faire que en local par exemple).

Point améliorable pour le travail du stage :

Collaboration à plusieurs :

Malgré qu'impossible dans ce projet, la collaboration aurait pu être intéressante pour plus facilement trouver une structure de code pertinente et normé, ainsi que détecter les erreurs de code bas niveau. Je n'avais jamais travaillé sur un projet informatique d'une telle rigueur scientifique, et cette expérience m'a appris que travailler seul ne permet pas forcément d'aller plus vite car la liberté amène plus de rapidité, mais aussi plus d'erreurs (« tout seul on va plus vite, à plusieurs on va plus loin »).

Bibliographie :

Une bonne gestion de bibliographie est aussi importante, j'aurai du prendre en compte plus rigoureusement cela. Une simple liste n'est pas suffisante pour un travail efficace et ordonné, j'aurai

Conclusion de stage

Techniquement, ce stage m'a permis d'en apprendre énormément sur le développement et l'utilisation d'intelligence artificielle, et plus loin que ça, leurs rôles et leurs placements dans le domaine de la robotique. Par mes recherches et la volonté de répondre à des problématiques bien précises, j'ai également compris, bien qu'indirectement le rôle et l'importance du domaine de l'automatisme dans le cadre d'un asservissement d'actionneurs. Et enfin, j'ai découvert plusieurs compétences autres que purement théorique, qui, pendant ce stage, m'ont permis de mener à bien mon étude et mon travail, comme par exemple, la compréhension et le classement de bibliographie, la présentation d'un travail technique aux autres (comment les chercheurs font pour partager le plus efficacement leurs travaux). Le point le plus important de ce stage fut, pour moi, l'organisation et la rigueur, le travail en autonomie et la liberté m'ont finalement appris la difficulté de devoir cadrer et faire avancer son travail tout seul.

Ma formation m'a également beaucoup apporté, la licence d'informatique m'a permise d'être à l'aise avec le développement général en python de mes expérimentations, ce qui m'a permis de concentrer mes efforts et mes difficultés sur les points les plus importants. Mon master automatique robotique, m'a permis d'avoir une base théorique solide en automatisme et en intelligence artificielle, pour pouvoir appréhender beaucoup plus facilement les nouvelles notions présentées. Un autre point important que m'a appris le master est l'organisation du travail et des connaissances, pouvoir, à partir d'une information théorique, pouvoir élaborer une réflexion large et adaptée aux problèmes traiter, plus loin que simplement comprendre la théorie, savoir comment l'appliquer à un problème précis.

Et enfin, personnellement, ce stage m'a permis de m'immerger dans un environnement professionnel correspondant, pour la première fois à mon domaine d'étude et à mes aspirations professionnelles. Bien que le monde de la recherche m'ait énormément plu, j'attends de voir et de connaître le travail en entreprise pour participer à des projets d'équipe plus organisés avec des contraintes techniques, humaines et temporelles bien plus présentes. L'aspect recherche m'a beaucoup plu par la liberté et la recherche documentaire du travail d'autres chercheurs, mais a aussi le point négatif d'être peu concret et d'avoir peu de contrainte, le travail fourni n'est pas évalué par un cahier des charges précis mais par un apport scientifique. C'est un tout autre paradigme, le but n'est pas de répondre au mieux à une problématique, mais plutôt de rechercher et d'évaluer les bonnes approches et de fournir un travail dans cette direction, le résultat et le fonctionnement importe finalement peu, c'est la collaboration globale avec le monde scientifique qui juge la valeur du travail accompli.

BIBLIOGRAPHIE :

Concernant la recherche autour de l'AutoML :

AutoML: A Survey of the State-of-the-Art, 2021
Xin He, Kaiyong Zhao, Xiaowen Chu,

DARTS: Differentiable architecture search, 2019.
H. Liu, et al.

Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network, 2021
Alex Sherstinsky,

Automated Reinforcement Learning (AutoRL): A Survey and Open Problems,2022
Jack Parker-Holder, Raghu Rajan, Xingyou Song et al.

Concernant la robotique par intelligence artificielle :

Learning Quadrupedal Locomotion over Challenging Terrain, 2020
Joonho Lee, Jemin Hwangbo , Lorenz Wellhausen, et al.,

Towards Sample Efficient Reinforcement Learning2018
Yang Yu ,

Learning Dexterous In-Hand Manipulation, 2019
Marcin Andrychowicz, Bowen Baker, Maciek Chociej, et al.

Concernant les CPPN (Compositional pattern-producing network) :

Automatic Content Generation in the Galactic Arms Race Video Game, 2009
Erin J. Hastings, Ratan K. Guha, and Kenneth O. Stanley,

Evolving Neural Networks That Are Both Modular and Regular: HyperNeat Plus the Connection Cost Technique 2014
Joost Huizinga, Jean-Baptiste Mouret, Jeff Clune,

Concernant les Fourier Features :

Fourier features let networks learn high frequency functionsin low dimensional domains, 2020.
Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, et al.

Fourier features in reinforcement learning with neural networks, 2022.
David Brellmann, Goran Frehse, and David Filliat.

Implicit neural representations with periodic activation functions. In Proc. NeurIPS, 2020.
Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, et al.

Concernant les outils utilisés pour l'études :

Openai gym. 2016
Greg Brockman, Vicki Cheung, Ludwig Pettersson, et al.

Stable-baselines3 , 2021
Antonin Raffin, Ashley Hill, Adam Gleave, et al.

Rl baselines3 zoo, 2020.
Antonin Raffin

ANNEXE :

Projet :
https://github.com/clement-chupin/BenchNeuralNework

Les fichiers les plus intéressant sont dans ./utils_lib/

Compte rendu (joint également au pdf) :

Il manque les résultats et la conclusion, mais la partie théorique et explication des choix techniques est presque complète (j'attends les résultats et les relectures pour les finir totalement)