

AI/RL - Assignment 2

GILLE Cyprien

LARROUBINE Tristan

I. Introduction

L'objectif de ce projet est d'entraîner un ANN à jouer au jeu Freeway d'Atari (i.e. à effectuer une Value Function Approximation), et d'étudier l'influence de la topologie du réseau sur les différents aspects du processus (performance au jeu, accuracy...).



Screenshot du jeu Freeway

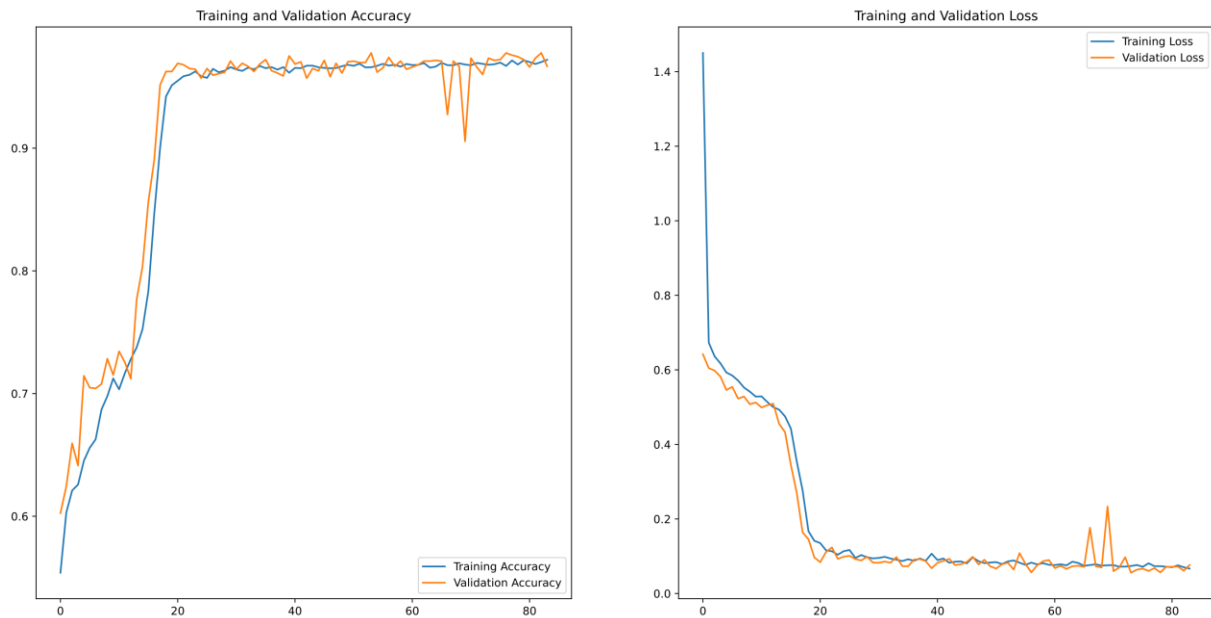
Le jeu Freeway est simple : le joueur contrôle le personnage en jaune en bas de l'écran, et doit l'amener de l'autre côté de la route, tout en évitant les voitures. L'objectif est de faire un high score, le temps de jeu étant limité (se faire toucher par une voiture fait reculer de quelques pas, et donc perdre du temps). Le joueur ne peut se déplacer que verticalement (l'action space est donc constitué de "stay", "up" et "down" uniquement).

Il est à noter que nous stockerons comme données non des screenshots du jeu, mais l'état de sa RAM à un instant donné. Cette dernière est composée de 128 bytes, ce qui se traduit en pratique par une liste de 128 nombres compris entre 0 et 255.

II. Implémentation

L'implémentation est assez classique. Tout d'abord, on joue quelques parties afin de créer notre database. On entraîne ensuite un réseau de neurones dense sur ces données, pour finalement faire jouer cet agent à Freeway. Relevons cependant quelques points notables avant de passer à l'étude de la topologie de l'ann:

- Pour inclure dans les données des informations temporelles, on stocke en fait la différence entre deux états consécutifs de la RAM.
- La phase de training est arrêtée par la fonction d'early stopping de keras, qui interrompt l'entraînement si une quantité ne s'améliore plus depuis un certain nombre d'époques. Dans notre cas, la quantité est la validation accuracy.



III. Influence de la topologie du réseau de neurones

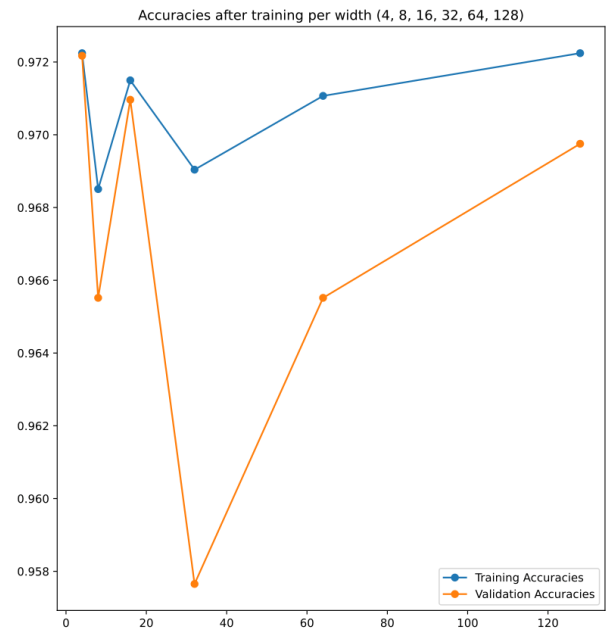
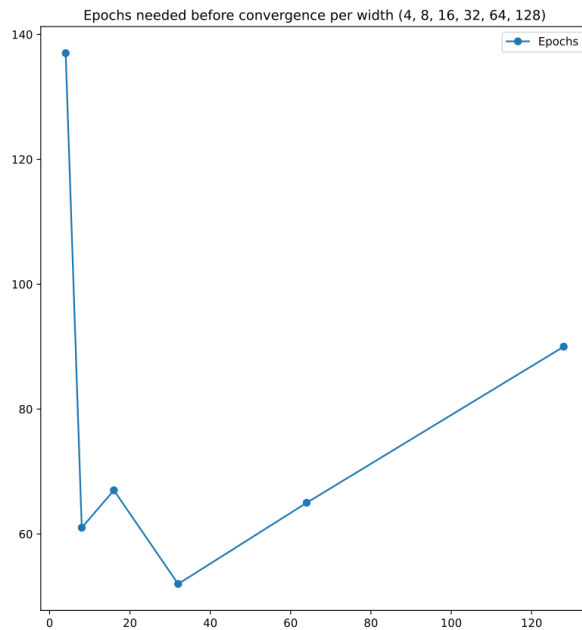
Avant toute chose, pour avoir un point de comparaison, un agent qui ne fait que courir vers le haut à chaque frame obtient un score de 21. Le plus haut score que l'on ait obtenu avec un agent entraîné est de 22, mais le score moyen des agents entraînés est de 12. Il semble qu'un ANN, même avec une haute accuracy de validation, ne soit pas très bon au jeu. Voyons l'influence des paramètres du réseau sur ces résultats.

1. Architecture

Le choix entre des layers denses ou de convolution est assez instinctif : une des forces des CNNs dans le traitement d'images réside dans leur capacité à prendre en compte la répartition spatiale des données, et les similarités locales que l'on trouve naturellement dans la plupart des images. Cependant, nos données ne sont pas des images, et deux bytes de RAM voisins n'ont pas de lien particulier. Cela rend l'usage d'un réseau de neurones convolutif bien moins adapté à notre cas.

Cette intuition se vérifie dans la pratique, où l'usage d'un CNN amène à des validation accuracy qui plafonnent entre 60% et 70%, contre plus de 95% pour un réseau dense.

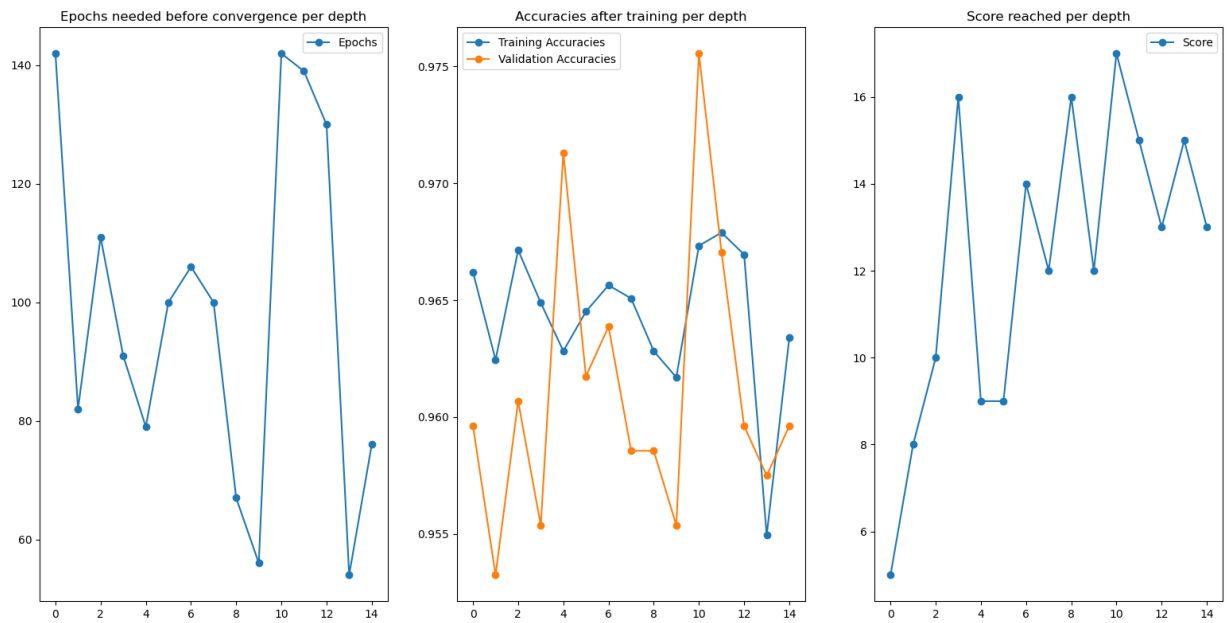
2. Largeur



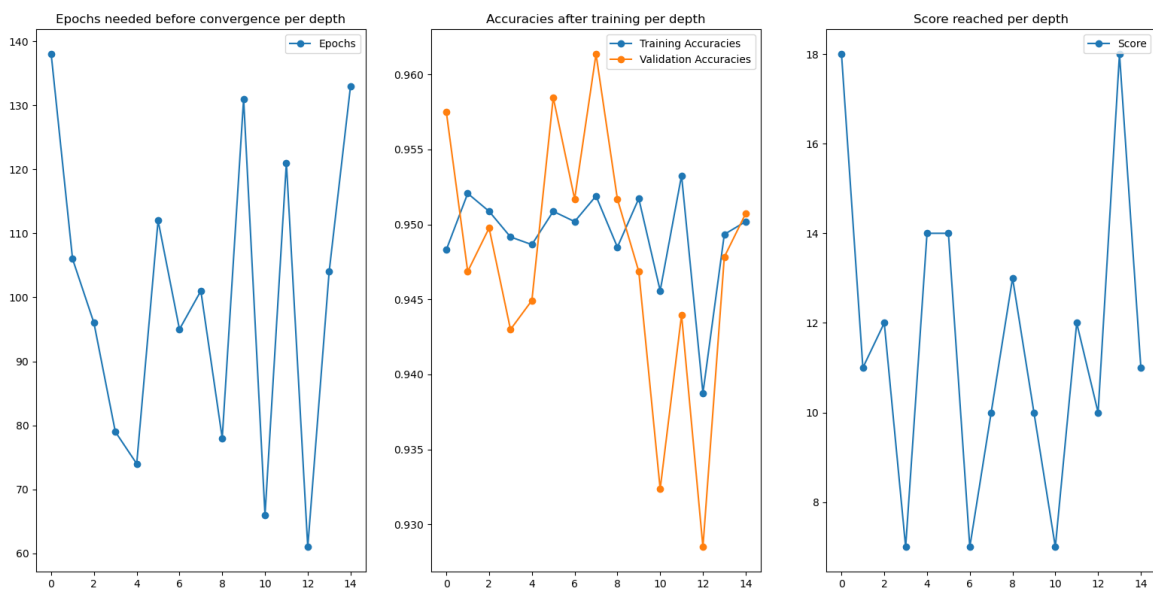
Ici, on a fait varier le nombre de neurones dans les hidden layers (avec toujours 5 hidden layers). On constate que la largeur du réseau n'a pas une grande influence sur sa précision : un réseau de profondeur 5 sait atteindre des précisions entre 95% et 97% quelle que soit sa largeur. Pour ce qui est du temps de convergence, en gardant à l'esprit la variance due à l'initialisation aléatoire, il semble que tant que le réseau n'est pas étroit à l'extrême (cas avec 4 neurones par hidden layer), le nombre d'epochs avant convergence augmente régulièrement avec la largeur.

Il semble donc que la profondeur soit le paramètre à modifier pour avoir un impact sur le réseau. (Note: depuis les travaux de LeCun et Al., on sait en effet qu'il est bien plus efficace de faire un réseau plus profond qu'un réseau plus large)

3. Profondeur



Il s'avère que la profondeur n'a pas véritablement d'influence sur la précision ni sur la vitesse de convergence, ni même sur la performance : si l'on entraîne le même modèle sur un autre jeu de données créé exactement de la même façon que le précédent mais juste un peu plus grand, on obtient les résultats suivants :



On constate alors que les éventuels résultats sur la profondeur que l'on aurait pu conjecturer avec les premiers résultats ne se vérifient plus : il n'y a pas de corrélation entre vitesse de convergence, accuracy, ou performance et profondeur du réseau.

IV. Conclusion

L'utilisation d'un ANN pour créer une IA jouant à Freeway en lisant la RAM n'est pas une bonne solution : un agent qui ne fait qu'aller vers le haut à chaque instant obtiendra dans la majorité des cas un meilleur score. En effet, l'agent de l'ANN est très hésitant : puisque l'on passe du temps dans le jeu à ne pas bouger pour attendre qu'un chemin soit libre, l'agent en fait de même - mais il base sa décision sur la RAM, qui n'est pas une structure de données "parlante". En effet, la RAM n'est pas pensée pour être exploitée ainsi : les bytes ne sont pas spécialement ordonnés, et leurs valeurs ne sont pas pensées pour être indicatives de l'état du jeu. De plus, l'information donnée par la RAM est hautement bruitée: il est souvent inutile de connaître la position de plus de la moitié des voitures.

En conclusion, la topologie de l'ANN n'a pas une grande importance, car les données brutes (RAM) se prêtent mal à une résolution par réseau de neurones. Ici, il sera probablement préférable d'utiliser des images du jeu, en les rognant pour donner à l'agent une vision "locale", moins bruitée, et bien plus structurée spatialement.