Résumé du Projet : LIM / Wang

Dans le cadre de ce projet, nous devions réimplémenter une partie du contenu de l'article « *Self-Improving Reactive Agents Based On Reinforcement Learning, Planning and Teaching* ». Ce dernier a pour objet l'évaluation des performances d'algorithmes d'apprentissage par renforcement sur des tâches complexes. Plusieurs algorithmes sont testés mais dans notre contexte, nous devions implémenter l'algorithmes QCON (Q-Learning + réseau de neurones), ainsi que sa variante experience replay, afin d'obtenir des résultats similaires à ceux présentés à la fin de l'article. Cependant, ayant eu des difficultés lors de l'implémentation du réseau de neurones, nous n'avons pas eu le temps d'implémenter la variante experience replay.

Nous avons dans un premier temps codé l'environnement de simulation, qui est représenté par une grille de taille 25 x 25, ainsi que les différents éléments existants : un agent, quatre ennemies, de la nourriture et des obstacles. L'agent choisi sa prochaine action en fonction de l'output du réseau de neurones, mais afin de tester notre code, l'agent se déplaçait, dans un premier temps, à l'aide des touches du clavier. Lorsqu'il consomme de la nourriture, il récupère de l'énergie. Chaque déplacement d'un ennemi a une probabilité qui évolue de sorte que plus l'ennemi est proche de l'agent, plus la probabilité de se déplacer vers ce dernier est grande. Nous avons ensuite implémenté les différents capteurs que possède l'agent, notamment leur placement et leur mise à jour.

Nous avons ensuite implémenté l'algorithme Q-CON, qui est présenté dans l'article comme étant un réseau de neurones composé de trois couches avec : 145 entrées dans la couche input, 30 neurones dans la couche cachée, et 1 sortie dans la couche output. Nous avons initialisé les poids des connexions avec les valeurs données dans l'articles, et avons utilisés la fonction d'activation présenté dans l'article : f(x) = (1/(1+e^(-x))) – 0.5.

En ce qui concerne la partie rétropropagation, nous nous sommes aidés d'un cours en pdf trouvé sur internet (https://helios2.mi.parisdescartes.fr/~bouzy/Doc/AA1/ReseauxDeNeurones1.pdf) qui explique les formules que nous avons utilisés.

La courbe de nourritures collectées en moyenne obtenue par notre algorithmes QCON a une tendance similaire à celle présentée dans l'article. Cependant, vers la fin de l'expérience, cette valeur arrive aux alentours de 12 dans l'article, alors que nous avons une valeur entre 8 et 10.

Parmis les articles mis en références à la fin du papier, nous avons consulté l'article « *Programming robots using reinforcement learning and teaching* » écrit par le même auteur en 1991 et qui a pour objet l'utilisation de l'apprentissage par renforcement dans la programmation des robots. Dans cet article, nous avons remarqué qu'ils utilisent une reward négative de -0.5 lorsque le robot entre en collision lorsqu'il se déplace. Nous avons donc rajouté cette reward négatif pour les collisions avec les obstacles, ainsi qu'une reward négative de -0.5 lorsque le robot effectue le mouvement contraire à son dernier déplacement, et revient donc vers la case qu'il occupait à l'itération précédente. Cela afin d'éviter que l'agent boucle sur deux cases. A l'aide de ces nouvelles reward, nous avons maintenant une courbe similaire à celle de l'article, qui atteint 12 nourritures collectées en moyennes vers la fin de l'expérience.