

Rapport de TIPE

Comment créer une intelligence artificielle grâce à un algorithme génétique ?

Préambule

Les algorithmes génétiques apportent une réponse innovante aux problèmes d'intelligence artificielle (problèmes consistant à imiter un comportement humain, souvent complexe). En effet, la manière classique de construire une intelligence artificielle consiste à résoudre le problème sous une forme mathématique, puis à construire un algorithme décrivant le comportement optimal à adopter. Ainsi, les algorithmes jouant aux échecs ont battu les grands champions dès 1980 en utilisant des recherches en arbres complétées de l'expérience tactique des experts. Mais de nombreux problèmes d'intelligence artificielle ont de trop nombreux paramètres et sont trop complexes pour pouvoir établir une solution algorithmique. Les algorithmes génétiques s'inspirent de la théorie de l'évolution pour s'affranchir de ces contraintes et établissent une méthode pour réaliser des intelligences qui apprennent par elles-mêmes.

Mon objectif a été d'implémenter un algorithme génétique dans le but de créer une intelligence artificielle capable de jouer au jeu Mario Bros.

Introduction

J'ai réalisé mon travail en deux étapes : la première a consisté en la réalisation du programme implémentant l'algorithme génétique, la seconde en son exécution et les transformations nécessaires à l'obtention de résultats significatifs.

Démarche expérimentale

J'ai tout d'abord choisi une version du jeu Mario écrite en Python, que j'ai dû adapter à mes besoins. Cette adaptation du jeu fut un véritable défi pour moi. En effet, la version du jeu que j'ai choisie utilise la librairie graphique Pygame que je n'avais jamais utilisée, et il fallait améliorer de façon drastique les performances du jeu pour espérer pouvoir évaluer un grand nombre d'intelligences artificielles. Il m'a donc fallu réussir à comprendre le code source du jeu et à le modifier. Parmi les nombreuses modifications effectuées, j'ai notamment optimisé la vitesse d'exécution du jeu, résolu des bugs faisant planter le jeu épisodiquement, ce qui aurait interrompu mon programme, et modifié la façon de recevoir les entrées (mon programme n'utilisant pas des boutons physiques mais des événements). Il m'a fallu également fabriquer un pont logiciel prenant en charge la communication entre le jeu et mon programme.

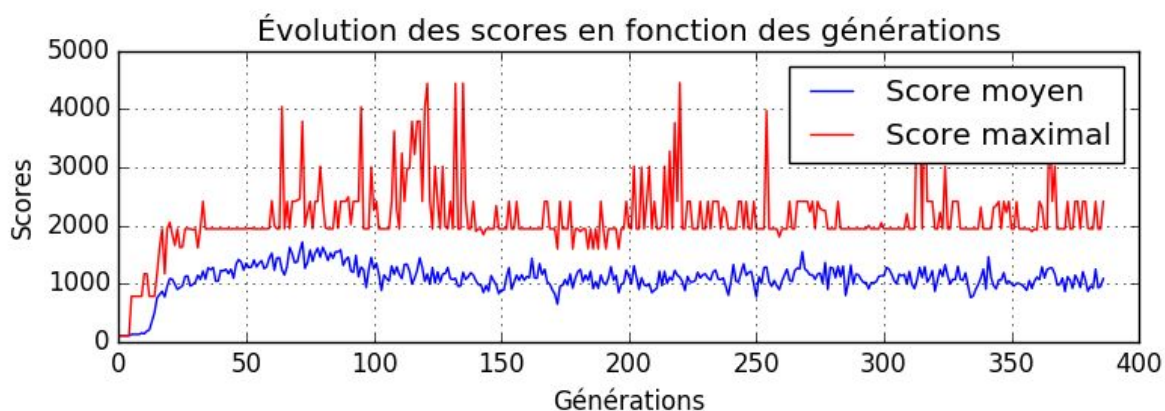
J'ai ensuite construit les réseaux de neurones artificiels modélisant les intelligences artificielles, ainsi que des composants pour les manipuler, puis j'ai programmé l'algorithme génétique. J'ai élaboré un composant ayant pour rôle de faire jouer les intelligences artificielles et de leur attribuer un score.

Afin de pouvoir récupérer et analyser les résultats, j'ai fait en sorte que toutes les données décrivant les intelligences artificielles créées par le programme soient enregistrées sur mon ordinateur au fur et à mesure de l'exécution. J'ai enfin réalisé une application utilisable en ligne de commande permettant de lancer le processus, de l'arrêter, de le reprendre, de faire jouer séparément une intelligence artificielle spécifiée et enfin d'imprimer un tableau des résultats.

Une fois l'application prête, le moment décisif est arrivé : vais-je avoir assez de puissance de calcul pour faire tourner l'algorithme ? Vais-je obtenir des résultats concluants ? J'ai alors fait tourner le programme pendant quelques jours. Malgré des débuts très prometteurs, les résultats se sont mis à stagner, les intelligences ne parvenant pas à franchir certains obstacles. Il a alors fallu que je m'inspire de ce que j'avais appris dans les livres pour améliorer mon procédé. Cela m'a permis de faire un nouvel essai avec un algorithme plus performant, dont les résultats se sont révélés bien meilleurs.

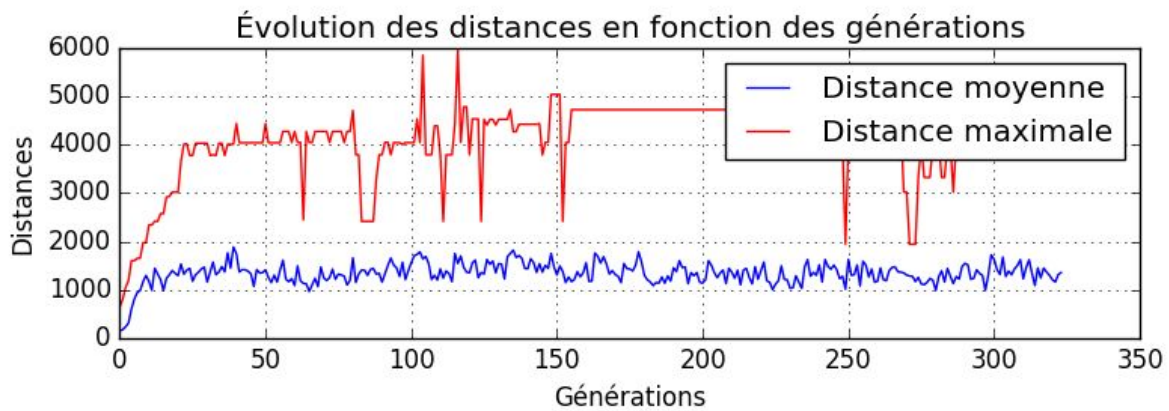
Résultats

Voici les résultats obtenues suite à la première exécution :



On observe sur cette courbe deux tendances. Lors des 30 premières générations, le score obtenu par les intelligences artificielles (la distance parcourue) augmente progressivement mais de manière considérable (de 0 à 2000). Cela correspond à la période où les intelligences artificielles apprennent les mouvements de base du jeu et montre qu'elles réussissent petit à petit à progresser sur le terrain. Cependant, à partir de $x = 2000$, un obstacle trop difficile pour elles survient et les nouvelles mutations aléatoires ne parviennent pas à les aider à franchir l'obstacle : le score maximal stagne alors aux alentours de 2000 sur les 300 prochaines générations.

Mes différentes lectures sur le sujet m'ont cependant beaucoup appris, et j'ai pu trouver de nombreuses pistes d'amélioration. J'ai alors apporté trois modifications majeures à mon algorithme : le chaînage des neurones et l'utilisation d'un opérateur de crossing-over à la place de la reproduction, la régularisation de la fonction d'évaluation et l'introduction d'un principe de survie. Grâce à ces apports, j'ai pu lancer une seconde exécution dont voici les résultats :



Ces nouveaux résultats sont très satisfaisants : la meilleure intelligence a en effet réussi à parcourir 69% du niveau, contre seulement 23% la première fois ! Les intelligences ont réussi à trouver des innovations génétiques leur permettant de passer l'obstacle sur lequel elles bloquaient auparavant, et on observe des comportements complexes leur permettant de continuer à avancer. Aucune n'a cependant réussi à passer au delà de $x = 6000$. À cette abscisse se situe en effet à une série d'épreuves particulièrement techniques, difficiles même pour un joueur humain : il est donc normal de les voir s'arrêter là.

Conclusion

Ainsi, l'utilisation d'un algorithme génétique pour la création d'une intelligence artificielle jouant à Mario Bros c'est révélée être une méthode fructueuse apportant des résultats significatifs. Si de nombreuses améliorations pourraient encore être apportées à l'algorithme afin d'obtenir de meilleurs résultats, ceux-ci sont suffisants pour mettre en lumière l'intérêt des algorithmes génétiques. Dans l'industrie, que ce soient pour les jeux-vidéos, pour la réalisation de robots humanoïdes ou pour d'autres domaines de l'intelligences artificielles comme la reconnaissance d'images, les algorithmes génétiques sont un procédé simple et peu coûteux à mettre en oeuvre en comparaison de l'expertise nécessaire à la mise en oeuvre d'une solution plus classique (quand c'est seulement possible). Si le fonctionnement précis des intelligences obtenus par un algorithme génétique peut être difficile à interpréter, la rétro-ingénierie sur celles-ci peut permettre de découvrir des innovations; c'est ainsi que AlphaGo a pu bouleverser 2 000 ans de théorie en introduisant une quarantaine de coups nouveaux.

L'ensemble du projet est disponible à cette adresse : <https://bitbucket.org/Zzortell/tipe/src>