

Bataille navale : De l'optimisation d'une stratégie gagnante par programmation classique et intelligence artificielle

L'informatique, et en particulier l'intelligence artificielle, reste la seule méthode pour traiter certains problèmes efficacement. Les TIPE me permettent alors une approche de ce domaine et la réalisation d'un projet qui me tient à coeur : la détermination d'une stratégie gagnante à la bataille navale.

La détermination et l'optimisation de différents programmes codant une stratégie gagnante à la bataille navale s'inscrivent pleinement dans le thème OCEAN, d'autant plus que les programmes implémentés ici peuvent être adaptés à des problèmes complexes plus proches de la réalité militaire des conflits en mer.

Ce TIPE fait l'objet d'un travail de groupe.

Liste des membres du groupe :

- GONZALEZ--THAUVIN Milan
- VESCHAMBRE Julie

Positionnement thématique (ETAPE 1)

INFORMATIQUE (Informatique pratique), INFORMATIQUE (Informatique Théorique), MATHEMATIQUES (Autres).

Mots-clés (ETAPE 1)

Mots-Clés (en français)	Mots-Clés (en anglais)
<i>Bataille navale</i>	<i>Battleship</i>
<i>Intelligence artificielle</i>	<i>Artificial Intelligence</i>
<i>Apprentissage supervisé</i>	<i>Supervised learning</i>
<i>Probabilités</i>	<i>Probability</i>
<i>Apprentissage par renforcement</i>	<i>Reinforcement learning</i>

Bibliographie commentée

La bataille navale, petit jeu de société qui oppose deux joueurs tentant de saborder la flotte adverse, dont la position est inconnue, semble ne relever que du pur hasard, ou tout du moins ne pas employer de stratégies aussi complexes que les échecs et le jeu de go [1]. Et pourtant, il existe bel et bien des stratégies pouvant drastiquement augmenter les chances de défaire l'adversaire. Celles-ci ont notamment été développées par Nick Berry [2] et Alex Alemi [3], et l'une d'entre elle semble l'emporter sur toutes les autres : l'approche probabiliste.

La question qui se pose alors est celle-ci : peut-on faire encore mieux ? Aucune stratégie employant des algorithmes classiques ne peut aujourd'hui prétendre obtenir de meilleurs résultats que

l'approche probabiliste. C'est pour cela que notre regard s'est tourné vers les intelligences artificielles et plus précisément l'apprentissage machine.

Une intelligence artificielle peut être définie ici comme un algorithme, dont l'architecture est souvent inspirée de systèmes naturels (comme les réseaux de neurones qui s'inspirent du cerveau humain), et dont le but est d'approcher une fonction notamment grâce à une phase dite d'apprentissage [4].

Plusieurs types d'intelligences artificielles existent, certaines plus simples à comprendre que d'autres. Le premier type d'intelligence artificielle auquel nous nous sommes intéressés est l'apprentissage supervisé, utilisant des réseaux de neurones. Un réseau de neurones peut être apparenté à une matrice de transition entre un vecteur d'entrée (une grille de jeu par exemple) et un vecteur de sortie (une cible à choisir par exemple). Ce réseau est initialement généré aléatoirement, puis entraîné sur un jeu de couples (entrée, sortie) correspondant à la fonction que l'on souhaite approcher (ici, la fonction probabiliste). La programmation de réseaux de neurones étant plutôt simple, nous avons décidé d'en coder nous-mêmes grâce au site de Michael Nielsen [5].

D'autres intelligences artificielles sont bien plus dures à coder et à comprendre. Nous nous sommes particulièrement intéressés à des algorithmes d'apprentissage non supervisé. Ceux-ci fonctionnent grossièrement comme les algorithmes d'apprentissage supervisé, à la différence que l'on ne connaît pas a priori la fonction vers laquelle doit converger l'algorithme. Il doit pouvoir, pour un certain jeu de données en entrée, les analyser et déterminer comment les classer, c'est-à-dire quelle sortie renvoyer pour une certaine entrée [6]. Les algorithmes d'apprentissage non supervisé étant plus complexes et l'année scolaire particulièrement courte, nous nous appuierons grandement sur le travail de Jonathan Landy [7].

Bien heureusement, il existe des bibliothèques donnant des fonctions de base pour coder des intelligences artificielles : par exemple, une fonction permettant de créer un réseau de neurones. Nous avons décidé d'utiliser la bibliothèque TensorFlow [8], une des bibliothèques les plus fournies pour le langage Python. Ainsi, nous aurons trois types d'intelligence artificielles différentes : des algorithmes d'apprentissage supervisé codés par nos soins, des algorithmes d'apprentissage supervisé codés à l'aide de TensorFlow, et des algorithmes d'apprentissage non supervisé codés à l'aide de TensorFlow selon le modèle de Landy. Le deuxième type d'intelligence artificielle n'est pas en soi très différent du premier, mais nous aidera dans l'utilisation de la bibliothèque.

Notre travail s'inscrit pleinement dans un travail de recherche, en cela qu'il fait la synthèse du travail déjà effectué par d'autres chercheurs sur le sujet, en y ajoutant notre propre réalisation : la comparaison des performances de tous ces algorithmes. Bien évidemment, la bataille navale reste un jeu très simpliste qui peut paraître très éloigné de la réalité des combats en mer. Cependant, nos algorithmes sont transposables si l'on change les règles. Par exemple, on pourrait choisir de donner la possibilité aux bateaux de se déplacer, de voir les cases dans un certain périmètre autour d'eux, de tirer plusieurs missiles (salve) en même temps. En complexifiant légèrement les règles, on ne

change pas la manière de le résoudre et on crée par la même occasion des algorithmes qui pourraient utiliser des bâtiments militaires.

Problématique retenue

Il s'agit de concevoir, implémenter et optimiser différents programmes pouvant jouer à la bataille navale. On s'intéressera à des méthodes de programmation classiques et à des intelligences artificielles, et on évaluera leurs performances au problème considéré.

Objectifs du TIPE

Nous nous proposons :

- d'implémenter une interface permettant de jouer à la bataille navale ;
- d'implémenter différents programmes jouant à la bataille navale n'utilisant pas l'intelligence artificielle et d'évaluer leurs performances ;
- de déterminer les méthodes utilisant l'intelligence artificielle adaptées au problème ;
- d'implémenter différents programmes jouant à la bataille navale utilisant l'intelligence artificielle, d'évaluer leurs performances et de les comparer aux programmes précédents.

Références bibliographiques (ETAPE 1)

- [1] Bataille navale (jeu) : https://fr.wikipedia.org/wiki/Bataille_navale_%28jeu%29, consulté le 4 novembre 2019
- [2] N. BERRY : Battleship : <http://www.datagenetics.com/blog/december32011/index.html>, consulté le 4 novembre 2019
- [3] A. ALEMI : The linear theory of battleship : <https://thephysicsvirtuosi.com/the-linear-theory-of-battleship/>, consulté le 17 novembre 2019
- [4] C.-A. AZENCOTT : Introduction au Machine Learning : Dunod, 2019
- [5] M. NIELSEN : Neural Networks and Deep Learning : <http://neuralnetworksanddeeplearning.com/>, consulté le 12 décembre 2019
- [6] C. NICHOLSON : A Beginner's Guide to Deep Reinforcement Learning : <https://pathmind.com/wiki/deep-reinforcement-learning>, consulté le 20 décembre 2019
- [7] J. LANDI : Deep reinforcement learning, battleship : <http://efavdb.com/battleship/>, consulté le 4 janvier 2020
- [8] B. RAMSUNDAR & R. BOSAGH ZADEH : Tensorflow for Deep Learning : O'Reilly, 2018