

# Flappy Bird - Discussion

## 1 Méthode utilisée

La méthode utilisée dans ce projet de renforcement est le "Deep Q Learning" en utilisant un réseau de neurones qui apprend directement sur le vecteur d'état "state" qui contient 8 variables décrivant l'état instantané du jeu. Dans chaque itération, le réseau de neurone apprend sur un échantillon d'expériences choisies aléatoirement d'une liste stockée en mémoire. Chaque expérience est composée de l'état  $s$ , l'action  $a$ , le nouveau état, la récompense  $r$ , et une variable binaire indiquant s'il s'agit d'un état terminal.

## 2 Architecture du réseau de neurones

Durant la période allouée au projet, différentes architectures et différent choix de couches du réseau ont été testés.

La meilleure performance est obtenue à notre étonnement avec un réseau de neurone très simple composé d'une seule couche cachée qui contient 500 nœuds. On a pas utilisé de "dropout" avec cette architecture.

## 3 Remarques

- Au cours des premières tentatives, on a essayé des réseaux de neurones avec plusieurs couches cachées couplées avec des couches de type "dropout". On a aussi essayé le cas sans et avec "experience replay" mais les résultats n'étaient pas satisfaisants et on n'arrivait pas à converger à un minimum local.
- On a remarqué aussi que le modèle était très sensible au choix du taux d'apprentissage "learning rate", le fait de diminuer le taux d'apprentissage à une valeur de  $5.e^{-5}$  rendait l'algorithme beaucoup plus performant.
- L'apprentissage est lancé sur 20 000 parties du jeu, le meilleur modèle n'était pas obtenu aux dernières itérations mais bien avant. Dans notre cas, le meilleur modèle est obtenu après 18 000 jeux.
- La modification des récompenses données par le jeu était très déterminante. dans le code fourni, on a multiplié les récompenses par un entier positif pour que la différence entre l'échec et la réussite soit très grande. dans le cas où on prend des récompenses par défaut (1 en cas de réussite et -5 en cas d'échec) les poids prédits par le réseau de neurones seront très proches ce qui nuit à la performance de l'algorithme.