

Machine Gaming

Thomas Ibanez

22 mai 2018

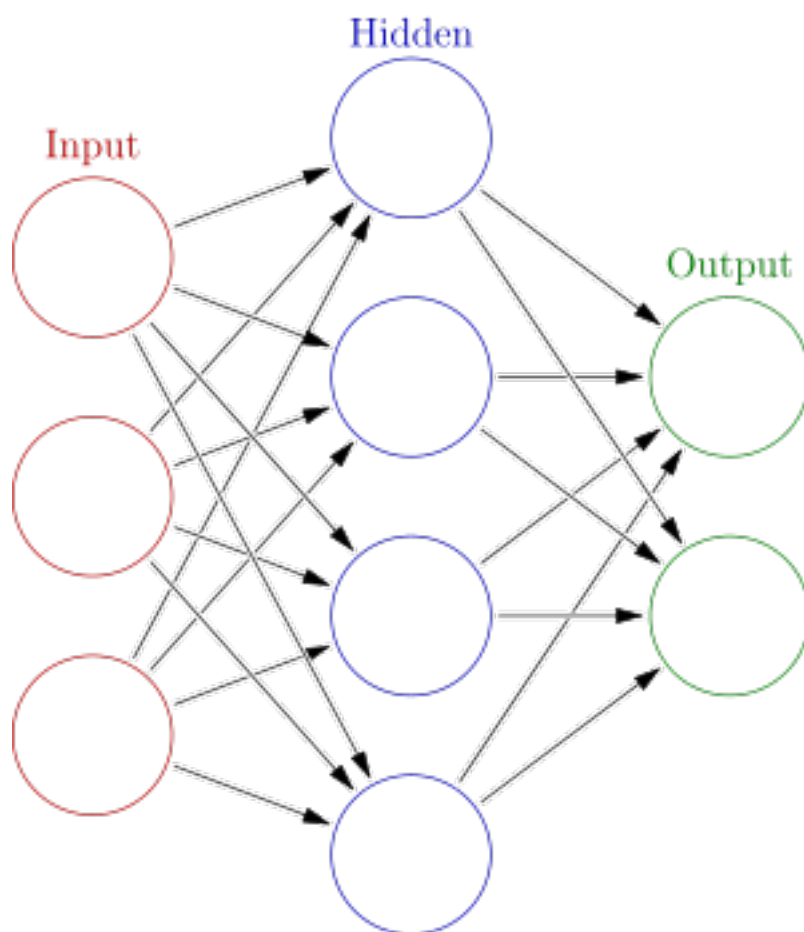


Table des matières

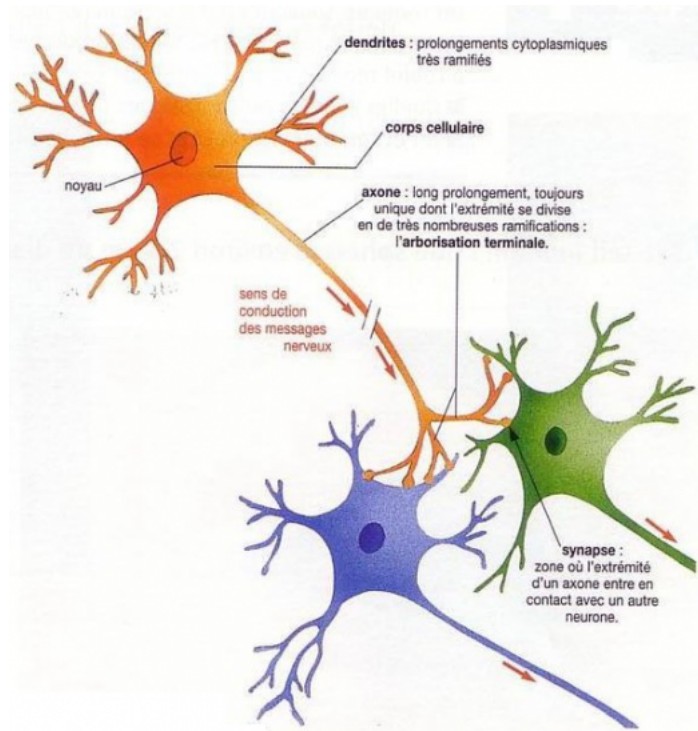
1	Introduction	3
2	Architecture Réseau	4
2.1	Protocole	5
2.2	Serveur	5
2.3	Client	5
3	Algorithme Génétique	5
3.1	Population initiale	5
3.2	Elitisme	5
3.3	Crossover	5
3.4	Mutation	5
3.5	Génome	5
3.6	Phénome	5
4	Réseaux de Neurones	5
4.1	Perceptron Multicouche	5
4.2	NEAT	5
5	Jeux	5
5.1	Asteroid	5
5.1.1	Principe	5
5.1.2	Entrées	5
5.1.3	Sorties	5
5.1.4	Résultats	5

1 Introduction

Le but de ce travail de bachelor est d'étudier et de comparer le comportement de différentes techniques de Machine Learning basés sur la neuroévolution en analysant la faculté des algorithmes à apprendre à jouer à des jeux-vidéos.

La neuroévolution est une technique qui consiste à faire évoluer des réseaux de neurones artificiels afin qu'ils arrivent à effectuer une tâche.

Un réseau de neurones est un modèle, inspiré du fonctionnement du cerveau humain, qui va consister en un ensemble de neurones artificiels (aussi appelés perceptrons), disposés en couches qui vont communiquer en propageant une information. En effet les neurones de notre cerveau vont collecter les signaux en provenance de leurs dendrites, puis si les signaux sont assez forts envoyer une impulsion le long de leur axone vers les neurones suivants qui vont faire de même. A noter que les connexions entre deux neurones peuvent être plus ou moins fortes (le signal va donc se propager avec une intensité variable).



Les perceptrons quant à eux vont imiter (de manière simplifiée) ce comportement, chaque perceptron va prendre le signal envoyée par chacun de ses voisins de la couche précédente, puis multiplier cette valeur par le poids de la connexion et finalement faire la somme de toutes les valeurs pondérées et passer cette somme dans une fonction (dites fonction d'activation) qui va placer cette somme dans un interval (entre 0 et 1 par exemple) et renvoyer le résultat de cette fonction à tous ses voisins de la couche suivante qui vont faire de même.

D'un point de vue mathématique le comportement d'un perceptron peut être écrit comme :

$$o = f\left(\sum_{i=0}^n S_i * W_i\right) \quad (1)$$

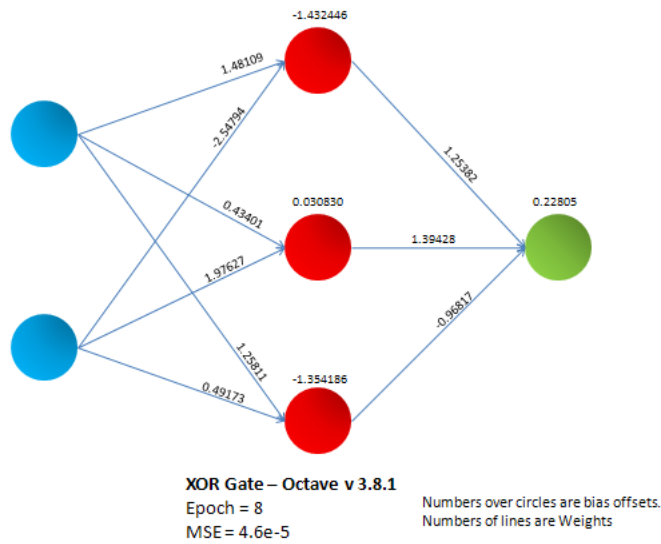
Où

o est le signal qui va sortir du perceptron

n est le nombre de voisins de la couche précédente

S l'ensemble des signaux des voisins la couche précédente

W l'ensemble des poids entre le perceptrons et ses voisins de la couche précédente



La méthode classique pour entrainer des réseaux de neurones est la rétropropagation du gradient (backpropagation). Cette technique consiste à comparer la sorties du réseaux à la sortie attendu pour une entrée donnée. L'avantage de cette technique est qu'elle va converger très rapidement vers le résultat attendu (à condition que les hyperparamètres du réseau soit adaptés au problème). Le désavantage est qu'il faut avoir à sa disposition un grand nombres d'entrées dont on connais la nature afin de pouvoir les comparer aux résultats du réseau. Par exemple il existe une base de données de chiffres écrit à la main avec leur valeur réelle sur <http://yann.lecun.com/exdb/mnist/> (70'000 entrées).

Cependant générer une telle base de données est un travail énorme, ainsi ces dernières années nous avons assisté à l'émergence de nouvelles technique ne nécessitant pas d'exemples

2 Architecture Réseau

Afin d'optimiser la vitesse de calcul, un système distribué à été mis en place. Celui-ci repose sur une architecture client-serveur classique où le client effectue les simulations et le serveur gère les génomes et s'occupe de distrubuer de manière équitable le travail entre tous les clients.

2.1 Protocole

2.2 Serveur

2.3 Client

3 Algorithme Génétique

3.1 Population initiale

3.2 Elitisme

3.3 Crossover

3.4 Mutation

3.5 Génome

3.6 Phénomène

4 Réseaux de Neurones

4.1 Perceptron Multicouche

4.2 NEAT

5 Jeux

5.1 Asteroid

5.1.1 Principe

5.1.2 Entrées

5.1.3 Sorties

5.1.4 Résultats