

Création d'un robot de combat basé sur un algorithme génétique et un réseau de neurones

DOCUMENTATION UTILISATEUR

Alexis BUSSENEAU, Guillaume VAILLAND,
Romain BOIZUMAULT, Vianney MATHIEU
Luc GEFFRAULT

Encadrant : Christian RAYMOND

Résumé

La documentation technique explique l'ensemble des manipulations à faire pour pouvoir utiliser correctement Robocode ainsi que la configuration pour lancer l'apprentissage supervisé. En dernière partie de ce document, vous trouverez des explications sur le lancement de l'algorithme génétique.

1 Installation de Robocode

Télécharger le **.jar** de la version la plus récente à l'adresse suivante : .

Exécuter le fichier téléchargé afin de lancer l'installation (lors de l'exécution, vous allez pouvoir choisir le dossier d'installation).

N'hésitez-pas à aller consulter la documentation en ligne de Robocode : afin de vous familiariser avec les règles et le fonctionnement du jeu.

2 Installation de notre projet

Notre projet est disponible sur **GitHub** (<https://github.com/Naeoth/Darwini>). Vous pouvez le cloner facilement en vous déplaçant en ligne de commande dans un répertoire d'installation de votre choix puis en exécutant `git clone https://github.com/Naeoth/Darwini.git`.

Un répertoire "Darwini" devrait se créer avec plusieurs dossiers notamment les données pour exécuter l'algorithme génétique et les sources de notre projet.

Il ne reste plus qu'à ouvrir le projet depuis votre IDE de votre choix (**IntelliJ** ou **Eclipse** par exemple) à partir du dossier "**src**". N'oubliez pas d'intégrer le dossier "**data**" à votre projet, afin que les ressources soient disponibles.

Le projet utilisant des classes de **Robocode**, il faut importer depuis votre projet de votre IDE, la librairie "**robocode.jar**". Cette librairie est disponible facilement dans notre projet, car nous avons fait une copie des librairies de Robocode dans le dossier "**libs**". Cependant elle contient la version **1.9.2.5** du jeu. Si une version plus récente est disponible, vous pouvez mettre à jour ces librairies en copiant l'ensemble des **.jar** du dossier "**libs**" située dans le repertoire d'installation de Robocode dans le dossier "**libs**" de votre projet.

La dernière étape est l'ajout des robots au sein du jeu Robocode. Lancer le jeu en exécutant "**./robocode.sh**", contenu dans le dossier d'installation

de Robocode. Une fois le jeu lancé, allez dans **Options -> Preferences -> Development Options**, puis ajouter le chemin du dossier contenant les **.class** du projet ("**out/production/Darwini**" pour **IntelliJ** et **bin/Darwini** pour **Eclipse**). Vous devriez voir les robots du projet apparaître lorsque vous souhaitez lancer une partie.

3 Apprentissage supervisé avec Iriselm

Avant de démarrer, il faut s'assurer que la **limite d'écriture** que Robocode autorise soit assez grande pour récolter toutes les données lors de l'acquisition. Pour cela, il faut aller dans le répertoire **"/config"** dans le dossier d'installation du jeu. Vous devriez trouver un fichier **robocode.properties**. Modifier le et ajouter à partir de la **troisième ligne** en partant de la fin la ligne suivante : **robocode.robot.filesystem.quota=999999999**. Si besoin, vous pourrez augmenter ce quota.

Lancer Robocode avec **"./robocode.sh"**, puis démarrer une partie avec un robot de votre choix et contre **AcquisitionBot** qui est notre robot qui va récolter des informations.

A la fin de la partie, un fichier au format **ssvm** du nom de **"data.ssvm"** sera créé dans le dossier des **.class** de votre projet. Robocode crée automatiquement un dossier **AcquisitionBot.data** contenant les fichiers de sortie du Robot **AcquisitionBot**. (Par exemple pour **IntelliJ** le chemin complet sera : **/out/production/Darwini/controller/AcquisitionBot.data**

Une fois le fichier **ssvm** créé, vous pouvez l'utiliser avec **Iriselm** qui est une librairie développée par Christian RAYMOND qui va pouvoir apprendre à partir des données récoltées et retourner un fichier au format **XML** contenant les poids d'un réseau de neurones. Renseignez-vous au près de M. RAYMOND afin d'avoir davantage d'informations sur les commandes à exécuter pour obtenir ce que vous voulez exactement.

Le robot **Darwini** est configuré par défaut pour lire un fichier **XML** produit par **Iriselm** afin d'initialiser son réseau de neurones. A cause de la sécurité de Robocode, ce fichier doit être obligatoirement mis dans le dossier **"Darwini.data"** (sous-répertoire du dossier des **.class** du projet). Le nom du fichier doit également s'appeler **"Perceptron.xml"**. Ainsi pour lire depuis **IntelliJ**, le chemin sera : **"/out/production/Darwini/controller/Darwini.data/Perceptron.xml"**

Vous pouvez ainsi voir si le Robot s'est amélioré ou non en comparant les résultats d'un combat avant/après contre le même robot.

4 Apprentissage non-supervisé avec l'algorithme génétique

Le processus de l'algorithme génétique a été entièrement automatisé. Il suffit simplement de le lancer et d'attendre la fin de l'algorithme pour obtenir le meilleur robot. Pour ce faire, il faut exécuter la classe **NaturalSelection.java** depuis un IDE. A la fin, le meilleur robot sera affiché sur la sortie standard.

Si vous voulez le tester, il suffit de copier le fichier **XML** du meilleur robot situé dans **"data/population"** dans le sous-répertoire **"Darwini.data"** en le renommant **"Perceptron.xml"**. (Cf : apprentissage supervisé pour en savoir davantage sur la procédure).

Vous pouvez bien évidemment modifier les constantes des classes `NaturalSelection.java` et `GeneticAlgorithm.java` pour changer les paramètres de l'algorithme génétique (nombre de génération, taille de la population, etc.).

5 Contact

Si vous avez un problème n'hésitez-pas à contacter Alexis BUSSENEAU à l'adresse alexis.busseneau@insa-rennes.fr pour plus d'explications ou d'éclaircissements.