Apprentissage par Renforcement en temps continu

Proposé par : Alain Dutech (`Alain.Dutech@loria.fr, members.loria.fr/ADutech, 03.83.59.20.95`)

Problématique

Une des thématiques de recherche de l'équipe BISCUIT s'intéresse au cadre de l'Apprentissage par Renforcement [3] pour guider l'adaptation de systèmes composés d'un grand nombre d'unités de calcul simples, asynchrones, homogènes, décentralisées, distribuées et favorisant des communication locales (par exemple, certains types de réseaux de neurones). L'apprentissage supervisé n'est pas une option viable pour apprendre à ces systèmes à s'adapter et c'est pourquoi nous voulons y utiliser des méthodes issues de l'apprentissage par renforcement.

L'une des difficultés soulevées concerne le fait que les algorithmes d'apprentissage par renforcement "canoniques" s'appuient sur une notion de temps centralisée et synchronisée. Les décises à un instant commun t et l'environnement évolue, selon des transitions probabilistes markoviennes, pour arriver dans un nouvel état au temps $t + 1$. Cette modélisation du temps n'est pas compatible avec des systèmes que nous voulons asynchrones et décentralisés. C'est pourquoi nous voulons explorer et analyser des mécanismes d'apprentissage en **temps continus**.

Sujet

Ce travail s'appuiera principalement sur deux publications : [1] et [2]. Dans un premier temps, nous voudrions étudier les algorithmes proposés par Kenji Doya sur les environnements de tests proposé par Nicolas Frémaux. Puis, nous chercherons à comprendre si une implémentation des algorithmes avec des architectures neuronales à *spike*, à l'image de ce qu'a fait Nicolas Frémaux, nécessite de modifier ou d'adapter les concepts proposés par Doya. Enfin, si le temps le permet, nous envisagerons la portabilité des algorithmes aux modèles étudiés par l'équipe BISCUIT.

- Etudier les algorithmes de [1]
- Les mettre en œuvre sur les environnements de test de [2] : acrobot, navigation d'un robot dans un labyrinthe.
- Comprendre en quoi les architectures à spike utilisées par Frémaux nécessitent, ou non, des modifications *ad hoc*
- Etudier l'utilisation de ces algorithmes dans le cadre des systèmes étudiés dans l'équipe BISCUIT.

Compétences

C/C++ (ou python), Linux, Optimisation, quelques notions de mathématiques (probabilités, équations différentielles).

Bibliographie

[1] K. Doya *Reinforcement learning in continuous time and space* Neural Computation, Vol. 12, 2000.

[2] Frémaux, Nicolas and Sprekeler, Henning and Gerstner, Wulfram *Reinforcement Learning Using a Continuous Time Actor-Critic Framework with Spiking Neurons* PLOS Computational Biology, Vol. 9(4), 2013.

[3] R. Sutton and G. Barto, *Reinforcement Learning*, Bradford Book, MIT Press, Cambridge, MA, 1998.