

Synopsis TER

Modélisation mathématique et informatique de neurones : création d'une bibliothèque Python/Tensorflow (BRNN - Biologically realistic neural network) pour la simulation de réseaux de spiking-neurons et applications

Granier Arno

encadré par C.Shlick et B.Ainseba

1^{er} mars 2018

Table des matières

1	Introduction	3
2	Modélisation d'un neurone	3
2.1	Rappel sur les aspects neurobiologiques	3
2.2	Limites des neurones formels	3
2.3	Integrate and fire - Leaky Integrate and fire	3
2.4	Modèle de Hodgkin-Huxley	3
2.5	Modèle de FitzHugh-Nagumo	3
2.6	Modèle de Izhikevich	4
2.7	Modèle par processus de Poisson	4
2.8	Implémentations supplémentaires conditionnelles au fait que le reste soit fait	4
3	Modélisation de réseaux de neurones	4
3.1	Modèle de synapses entre deux neurones pour un spike	4
3.2	Generalisation à plusieurs spikes	4
3.3	Generalisation à plus de deux neurones	4
3.4	Modèles et architectures de réseaux	4
3.5	Apprentissage	4
3.6	Implémentations supplémentaires conditionnelles au fait que le reste soit fait	5
4	En ce qui concerne la partie informatique	5

5 Applications concrètes et applications futures envisageables	5
6 Notes	5
Références	6

1 Introduction

Présentation du sujet, approches prises et définitions. Intérêts scientifiques, notamment comparaison sous plusieurs angles de vues des réseaux de neurones formels et des réseaux proposés ici. Aspects philosophiques.

2 Modélisation d'un neurone

2.1 Rappel sur les aspects neurobiologiques

Neurones, synapses. PPSE, PPSI, potentiel d'action, dynamique des potentiels d'actions -mécanismes chimiques et dynamiques temporelles.

2.2 Limites des neurones formels

Présentation des neurones formels et justification de ce modèle. Limites de ce modèle, notamment l'absence d'une composante temporelle dans les comportements de ce modèle. Ces limites justifient que l'on n'utilise pas les neurones formels dans le but de modéliser des neurones biologiques. Besoin de la théorie des systèmes dynamiques.

2.3 Integrate and fire - Leaky Integrate and fire

Présentation des modèles. Passage de Integrate and fire à Leaky Integrate and fire avec période réfractaire à Leaky Integrate and fire. Comparaison avec données électrophysiologiques. Mise en évidence des limites et du non réalisme de ces modèles. Présentation et commentaire d'une implémentation en Python.

2.4 Modèle de Hodgkin-Huxley

Présentation complète du modèle - modèle en circuit RC mis en relation avec les mécanismes neuro-électriques et neurochimique, explication des dynamiques d'inactivation et d'activation des canaux ioniques, explication du comportement du modèle, notamment explication du rôle des constantes temporelles (m, h, n). Réalisme biologique mais problème en termes de complexité de computation. Présentation et commentaire d'une implémentation en Python. Discussion autour des deux approches possibles pour travailler à partir de Hodgkin-Huxley : complexifier ou simplifier. Résumé rapide des approches où on complexifie - Cable Theory, modèles multi-compartimentaux. Dans la suite on s'intéressera aux simplifications de Hodgkin-Huxley, dans des soucis d'implémentation de réseaux de grandes tailles.

2.5 Modèle de FitzHugh-Nagumo

Explication de la réduction à partir de Hodgkin-Huxley. Etude à partir de la théorie des systèmes dynamiques : points fixes, isoclines, plan de phase, dynamiques, diagrammes

de bifurcation.

2.6 Modèle de Izhikevich

Explication de la réduction à partir de Hodgkin-Huxley. Etude à partir de la théorie des systèmes dynamiques : points fixe, isoclines, plan de phase, dynamiques, diagrammes de bifurcation.

2.7 Modèle par processus de Poisson

2.8 Implémentations supplémentaires conditionnelles au fait que le reste soit fait

Adaptation des modèles simples à des données électrophysiologiques : introduction des méthodes de Spike-triggered average avec bruit blanc gaussien, analyse en composante principales, divergence de Kullback-Leibler, adaptation aux données d'un neurones modélisés par un processus de Poisson.

3 Modélisation de réseaux de neurones

3.1 Modèle de synapses entre deux neurones pour un spike

A partir du modèle en circuit RC, à partir de Hodgkin-Huxley. Différents types de synapses : fonctions alpha, délais. Présentation d'une implémentation en Python.

3.2 Generalisation à plusieurs spikes

Filtre linéaire pour intégrer plusieurs spikes

3.3 Generalisation à plus de deux neurones

3.4 Modèles et architectures de réseaux

Feedforward, recurrent. Présentations et commentaires de plusieurs implémentation en Python-Tensorflow.

3.5 Apprentissage

Règle de Hebb, règle de Oja. Spike-timing dependent synaptic plasticity. Présentation d'une implémentation en Python.

3.6 Implémentations supplémentaires conditionnelles au fait que le reste soit fait

Apprentissage non-supervisé - apprentissage par compétition, Algorithme EM. Apprentissage supervisé. Apprentissage par renforcement. Reste à voir si la réalité biologique permet d'implémenter ce genre d'algorithme, puisque le but de ce TER est de rester proche de la réalité biologique. On pourra étudier les modèles des ganglions de la base en relation avec l'apprentissage par renforcement.

4 En ce qui concerne la partie informatique

Tout ce qui a été présenté dans 2 et 3 sera implémenté dans une bibliothèque Python 3+ / Tensorflow. Pour ce qui est de la simulation d'un seul neurone, tensorflow ne devrait pas être nécessaire et on s'intéressera plutôt aux différents ODE solver disponible dans Python (scipy et autres). Pour ce qui est de l'implémentation des réseaux, on essaiera d'utiliser au maximum les outils de tensorflow pour obtenir une implémentation rapide. En plus de l'implémentation des modèles, on aura également une grosse partie affichage et visualisation de données (euler plot, isoclines, plans de phase, raster plot, affichage du comportement d'un ou plusieurs neurones dans un réseau, etc ..), certainement basée sur matplotlib. Si l'usage de Tensorflow se révèle excessif, on ne l'utilisera pas. Tout sera accompagné d'une documentation "technique", bien sur.

5 Applications concrètes et applications futures envisageables

Pour des applications plus concrètes, et en utilisant l'outil qu'on aura mis en place, on pourra présenter les propriétés générales des réseaux qu'on retrouve dans un réseau du type présenté dans (Izhikevich, 2003) ou (Brette et al., 2007). On pourra reproduire le modèle des ganglions de la base de (Hérick et al., 2016). Dans un registre plus clinique on pourra s'intéresser à un modèle de la maladie de Parkinson (proposé par M.Ainseba, pour l'instant moi je sais pas précisément comment on ferai ça). Enfin pour conclure on pourra parler des applications futures possibles de ce type de réseaux de neurones, notamment au niveau de l'implémentation possible de la simulation d'un cerveau entier en temps réel (ou proche de réel), et des implications -cartographie du cerveau du vers C.Elegans, fonctionnalisme. Notamment, est-ce qu'on verrait l'apparition d'un phénomène similaire à la conscience ? (c'est une ouverture interrogative, je compte pas répondre à la question [enfin si, plus tard]).

6 Notes

La bibliographie est temporaire et incomplète, elle sera étoffée au fur et à mesure du TER.

Références

- Brette, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower, J. M., ... others (2007). Simulation of networks of spiking neurons : a review of tools and strategies. *Journal of computational neuroscience*, 23(3), 349–398.
- Computational neuroscience course by university of washington.* (s. d.). Consulté sur <https://www.coursera.org/learn/computational-neuroscience/> (Accessed : 2018-01-17)
- FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophysical journal*, 1(6), 445–466.
- Fountas, Z., & Shanahan, M. (2017). The role of cortical oscillations in a spiking neural network model of the basal ganglia. *PloS one*, 12(12), e0189109.
- Goodman, D., & Brette, R. (2008). Brian : a simulator for spiking neural networks in python. *Frontiers in neuroinformatics*, 2.
- Héricé, C., Khalil, R., Moftah, M., Boraud, T., Guthrie, M., & Garenne, A. (2016). Decision making under uncertainty in a spiking neural network model of the basal ganglia. *Journal of integrative neuroscience*, 15(04), 515–538.
- Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4), 500–544.
- Izhikevich, E. M. (2003, Nov). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6), 1569–1572. doi: 10.1109/TNN.2003.820440
- Izhikevich, E. M. (2007). *Dynamical systems in neuroscience*. MIT press.
- Izhikevich, E. M., & Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proceedings of the national academy of sciences*, 105(9), 3593–3598.
- MacLeod, M., & Nersessian, N. J. (2018). Modeling complexity : cognitive constraints and computational model-building in integrative systems biology. *History and philosophy of the life sciences*, 40(1), 17.
- McCulloch, W. S., & Pitts, W. (1943, 01 Dec). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133. Consulté sur <https://doi.org/10.1007/BF02478259> doi: 10.1007/BF02478259
- Neuron documentation.* (s. d.). Consulté sur <https://www.neuron.yale.edu/neuron/docs> (Accessed : 2018-01-17)
- Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, 275(5306), 1593–1599.
- Sen, S., Miller, K. D., & Abbott, L. F. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, 3(9), 919.
- Vreeken, J. (2003). *Spiking neural networks, an introduction*. Utrecht University : Information and Computing Sciences.