

A. Présentation du programme :

Notre programme permet de simuler l'évolution d'un réseau de 12'500 neurones selon le modèle de Brunel (2000).

Il comporte deux classes principales.

1. La première classe est la classe « Neuron ».

Il y a deux types de neurones, (leur rapport numérique est de 0.25) :

- Les neurones excitateurs : ils envoient un courant excitateur aux neurones auxquels ils sont connectés. Il y en a 10'000 sur 12'500.
- Les neurones inhibiteurs : ces derniers envoient un courant inhibiteurs (valeur négative). Il y en a 2'500 sur 12'500

(La seule chose les différenciant l'un de l'autre sont le courant qu'ils envoient aux autres neurones, tâche gérée dans la classe Network, c'est pour cela qu'une seule classe est nécessaire pour représenter les deux types.)

Chaque neurone a un certain potentiel membranaire qui évolue en fonction du temps et du courant qu'on lui envoie.

Si ce potentiel membranaire dépasse le seuil fixé, on enregistre que le neurone a fait un pic, un « spike », c'est lorsque qu'un neurone fait un pic qu'il va transmettre une impulsion à ses neurones cible.

Etant donné une probabilité de connexions de 0.1 (Selon les données de Brunel), pour un système de 12'500 neurones, chaque neurone est lié aléatoirement à 1000 autres neurones excitateurs et 2500 inhibiteurs. C'est de ces liaisons que s'occupe la classe Network.

2. La deuxième classe « Network » représente donc le réseau de neurones et gère leurs liaisons. Pour se faire elle crée un tableau qui recense les connexions aléatoires de chaque neurone.

Un « Background noise » (bruit de fond) est aussi pris en compte dans notre simulation, il représente l'activité neuronale du reste du cerveau qui impact l'amplitude du potentiel membranaire des neurones en suivant la distribution aléatoire de Poisson.

Dans notre simulation, le courant qui doit être choisi est 0.0 V, les divers spikes dépendent donc uniquement du background noise et des impulsions reçues des autres neurones.

Notre programme vise à reproduire la figure 8 du papier Brunel (p.197). Ces différentes images représentent le nombre de neurones dont le potentiel membranaire. Il vous sera expliqué dans la compilation et utilisation du programme comment les reproduire.

B. Compilation du programme :

Le programme fonctionne avec CMake.

Ouvrez le terminal, tapez le chemin nécessaire pour vous trouver dans le fichier « build ».

Lancer le programme :

```
cmake ..          « enter »  
make              « enter »  
./Neuron          « enter »
```

Lancer les Google tests :

```
cmake ..          « enter »  
make              « enter »  
./neuron_unittest « enter »
```

Générer la documentation Doxygène :

Videz au préalable votre dossier « build » de préférence

```
cmake ..          « enter »  
make              « enter »  
make doc          « enter »
```

C. Utilisation du programme :

Graphiques : Le graphique suggéré par le professeur via un code Python sur Jupyter donne des figures similaires à celles du papier Brunel, cependant, n'ayant aucune connaissance en code Python je n'ai pas compris pourquoi l'histogramme (bas de la figure) ne correspondait pas exactement. J'ai alors créé un fichier sur Numbers qui génère un histogramme du nombre de spike par pas de temps, qui lui, correspond aux proportions et unités demandées.

Pour générer le graph sur python :

- Ouvrez Jupyter (sur internet)
- uploadez sur le fichier texte plotX.txt
- cliquez sur New/ Python 3
- Copiez le code python correspondant au fichier texte (PythonGraphX.txt) et cliquez sur « run »

Lorsque vous suivez les instructions « Lancer le programme » ci-dessus. Il vous est demandé d'entrer certaines informations nécessaires à la simulation :

Si vous voulez reproduire les figure de Brunel, il est important d'entrer le nombre de pas correspondant au graphie (graph A : 6000 pas / graphs B,C et D : 12'000 pas)
Ainsi qu'un courant externe nul (0.0), entez ensuite le chiffre correspondant au graphique désiré.

Voici un exemple pour générer le fichiers texte de données du graphique B de la figure 8 du papier de Brunel.

```
Enter the number of steps you want for the simulation: (1step represents 0.1ms)
( To generate the Brunel's graphs you need :
A: 6000 steps
B,C and D: 12'000 steps)
12000
Enter a External Current:
0.0
Wich brunel graph's numbers list do you want to generate ? 0 = None , 1 = graph A, 2 =
graph B, 3 = graph C, 4 = graph D
3
The list of numbers will be generate, to see the plot you have to upload it on Jupyter
and use the python code provided on my GitHub
```

Le résultat de cette simulation sont deux fichiers :

- Le premier « PlotX.txt » (X= A,B,C ou D) est le fichier à upload sur jupyter en exécutant le code python présent sur le GitHub afin d'obtenir le graphique correspondant.
La colonne de gauche représente le **pas de simulation courant**, celle de droite **l'index du neurone qui a eu un spike**.
- Le deuxième « PlotNumbersX.txt » X=A,B,C ou D est le fichier nécessaire à la création des histogramme sur l'application Numbers (type excel)
Les colonnes représente cette **fois le pas courant** et le **nombre de spikes qui ont eu lieu lors de ce pas de temps**.

Concernant les **Google test**, un des tests dans le fichier « neuron_unittest » est commenté, il prend beaucoup de temps à s'exécuter donc j'ai choisi d'en créer un autre moins complet mais plus court, si vous désirez quand même l'exécuter, il vous suffit de le décommenter. Il est indiqué par les commentaires du fichier.