

# Compréhension de la STDP et classification

0 1 2 3 4

5 6 7 8 9

Nous attestons que nous sommes les auteurs du présent travail et que tout ce qui a été emprunté est attribué à sa source et proprement référencé

Auteurs: Soline Bernard, Antoine Marion, Victor Yon

Groupe : P2

Temps de travail: 68h/personne

Date: 19/11/2020



github.com/victor-yon/gei723

## Objectif

Comprendre le fonctionnement de la STDP ainsi que ses limites (forces et faiblesses)

Etre capable de modifier la STDP et de l'adapter suivant le type de connexion synaptique (inhibitrice, excitatrice)

Comprendre l'établissement de courbes d'accord et de champs récepteurs par l'apprentissage

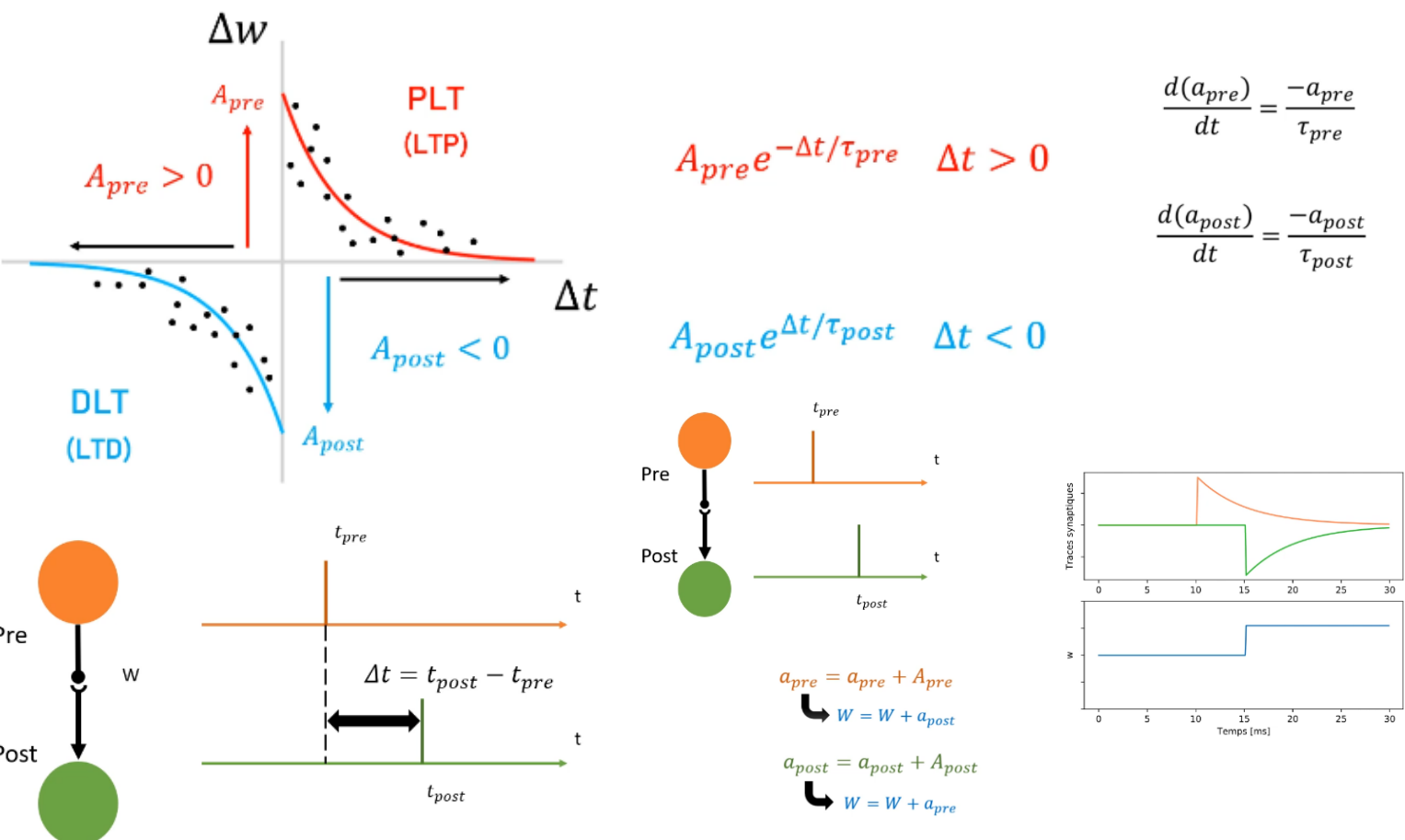
Comprendre l'impact des diverses configurations, des paramètres et des différents facteurs affectant les performances de réseaux de neurones avec STDP dans un contexte de classification.

Comprendre l'équivalence entre les champs récepteurs des neurones et une série de filtres représentant les éléments de base des signaux d'entrée (ici, des images).

## Partie I : Etude des différentes formes de STDP

La STDP (Spike Timming Dependent Plasticity) est une règle de plasticité.

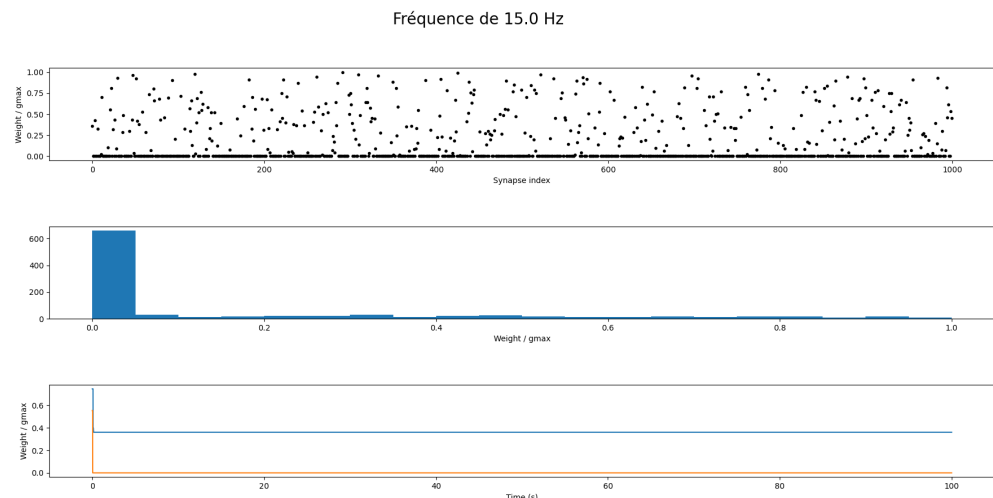
La STDP a pour rôle de modifier les poids synaptiques en fonction du temps de décharge entre les neurones pré et post synaptique. Nous en expliquons le principe avec la forme la plus classique. On associe aux neurones pré et post synaptique des variables *apre* et *apost* qui sont les traces de l'activité synaptique. Ces traces sont régies, dans le cadre de la forme classique de STDP, par des équations différentielles du premier ordre. Si le neurone post synaptique décharge après le présynaptique dans un temps suffisamment court, alors son potentiel est augmenté d'un poids positif. Si le *post* décharge avant le *pre* dans un temps suffisamment court, le potentiel du neurone post synaptique est diminué de la valeur du poids.



On peut rendre ce modèle asymétrique en modifiant les valeurs de *Apre* et *Apost* ou les constantes de temps des traces pré et post synaptiques .

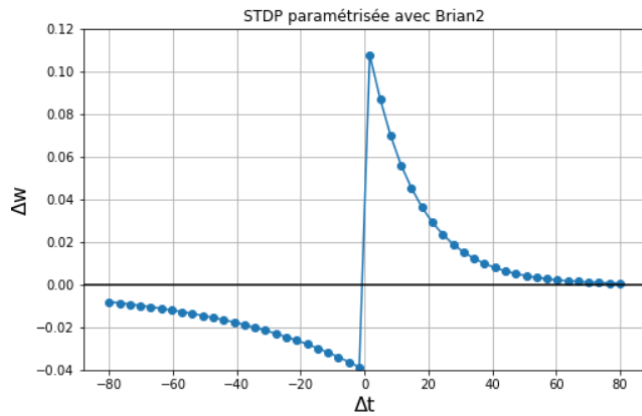
### Avec la forme de STDP classique symétrique de Excitateur à Excitateur

Nombre de neurone N	1000
Constante de temps trace présynaptique	200 ms
Constante de temps trace postsynaptique <i>taupre</i>	200 ms
Constante Pré synaptique <i>Apre</i>	0.01*gmax
Constante Post Synaptique <i>Apost</i>	-0.01*gmax
Poids maximum <i>gmax</i>	0.01
Fréquence de décharge F	15 Hz



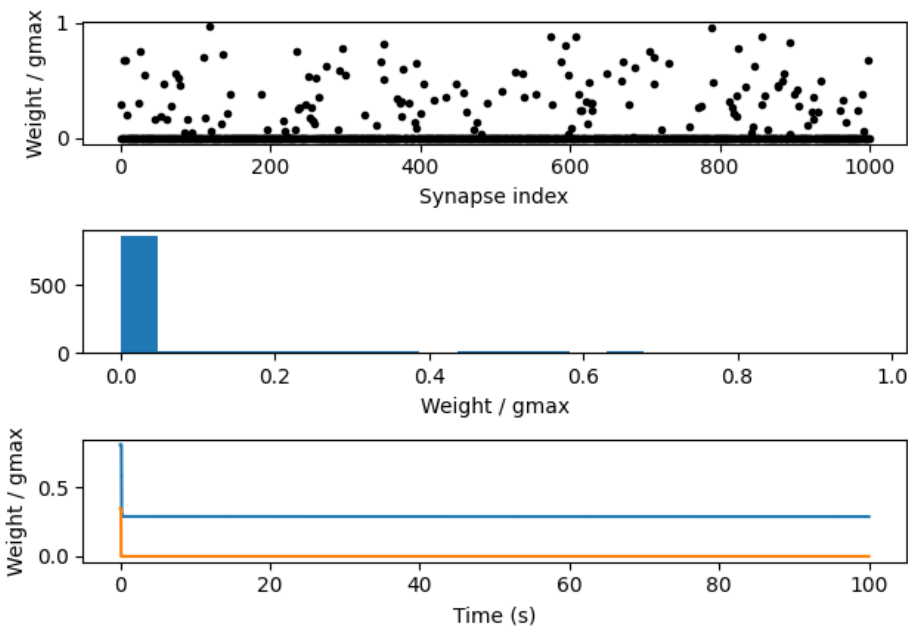
## Dynamique d'évolution des poids en fonction de la forme de STDP

### Avec la forme I de Excitateur à Excitateur



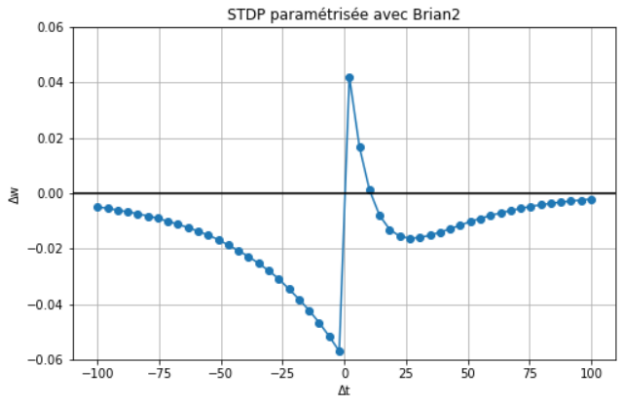
Pour obtenir cette forme, nous prenons une forme classique de STDP que nous rendons asymétrique. Cette forme atténue fortement les poids pour des décharges inférieures à 22 Hz (ANNEXE 1) . En effet, 22Hz correspond à une moyenne de 1 décharge sur 45 ms, et graphiquement, on voit que cette forme de STDP inhibera plus qu'elle n'excitera pour des temps entre décharges supérieur à 45 ms.

STDP forme I, excitateur à excitateur fréquence de 15.0 Hz



<i>taupre</i>	15ms
<i>taupost</i>	50 ms
<i>Apre</i>	0.12
<i>Apost</i>	-0.04

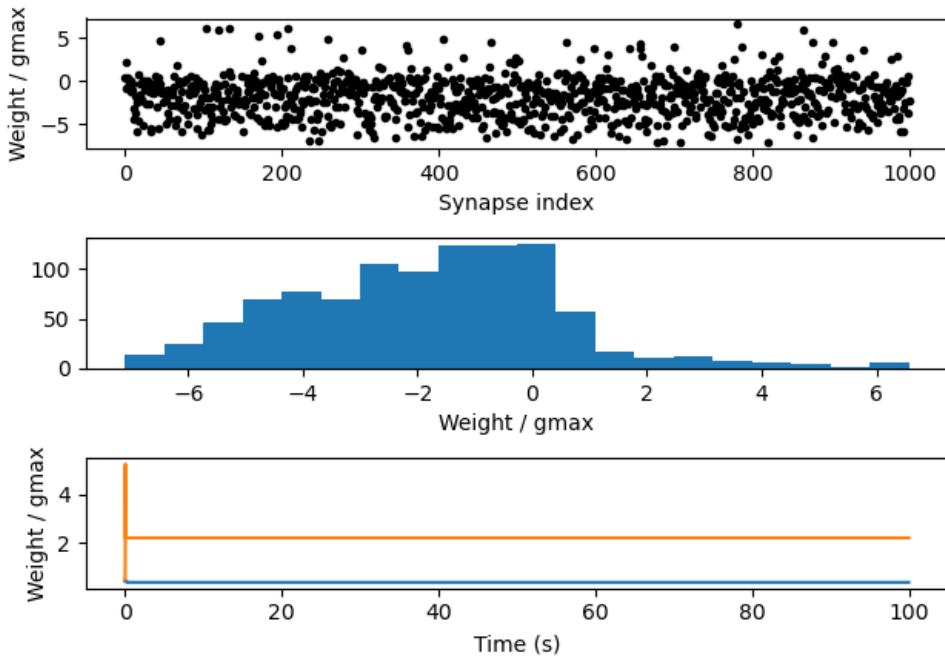
### Avec la forme II de Excitateur à Excitateur



Cette forme est obtenue avec une forme classique de STDP pour la trace post-synaptique, et une trace en double exponentielle pour la trace pré-synaptique. Cette forme donne une répartition des poids à peu près homogène entre 0 et 1 jusqu'à 12 Hz car on est sur des différences de temps > à 80 ms, et la STDP n'a alors presque pas d'effet. Pour des fréquences supérieures à 12Hz, les poids deviennent progressivement négatifs, car, peu importe l'ordre d'arrivée des décharges, le poids sera incrémenté d'une valeur négative. (ANNEXE 1). Ici la visualisation est effectuée sans borner le poids.

$$\text{Pour } \Delta t > 0 : W = A_{pre1} e^{-\frac{t}{\tau_{pre1}}} + A_{pre2} e^{-\frac{t}{\tau_{pre2}}}$$

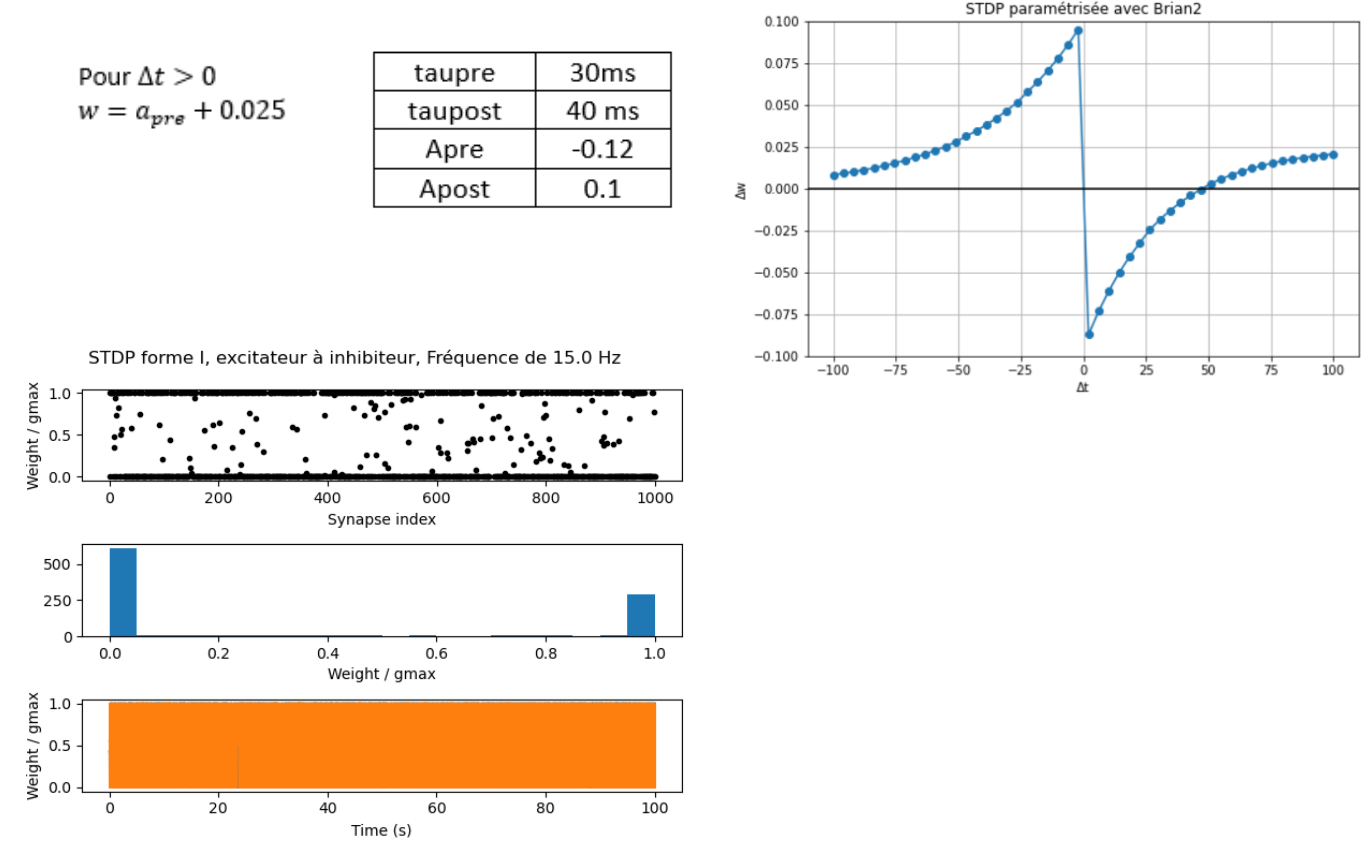
STDP forme II, excitateur à excitateur Fréquence de 15.0 Hz



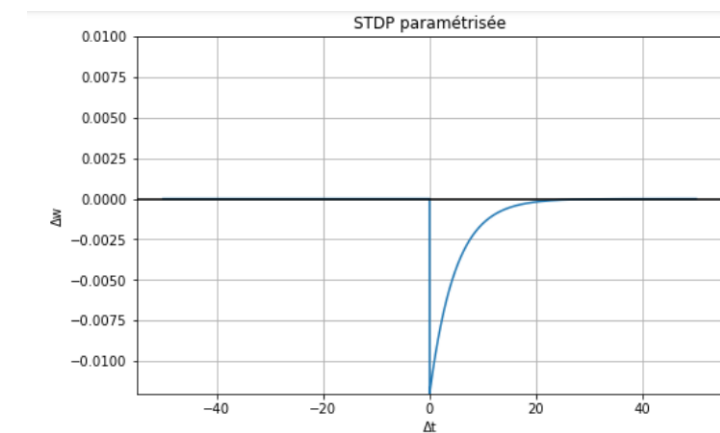
<i>taupre1</i>	10ms
<i>taupre2</i>	30ms
<i>taupost</i>	40 ms
<i>Apre1</i>	0.12
<i>Apre2</i>	-0.06
<i>Apost</i>	-0.06

### Avec la forme I de Excitateur à Inhibiteur

Il s'agit d'une forme classique de STDP avec des valeurs positives pour la trace post-synaptique, et négative pour la trace pré-synaptique. Un *offset* à été ajouté à la trace pré-synaptique afin que les poids soient incrémenté d'une valeur positive si le temps de décharge est supérieur à 50 ms. Pour des fréquences inférieures à 11Hz, les poids sont répartis de façon homogène entre 0 et 1. Pour des fréquences supérieures à 12Hz, les poids sont légèrement plus réduits plutôt qu'augmenté pour un *delta\_t* de temps équivalent (car *tau\_post* > *tau\_pre*) . On observe alors une répartition des poids en 2 colonnes.

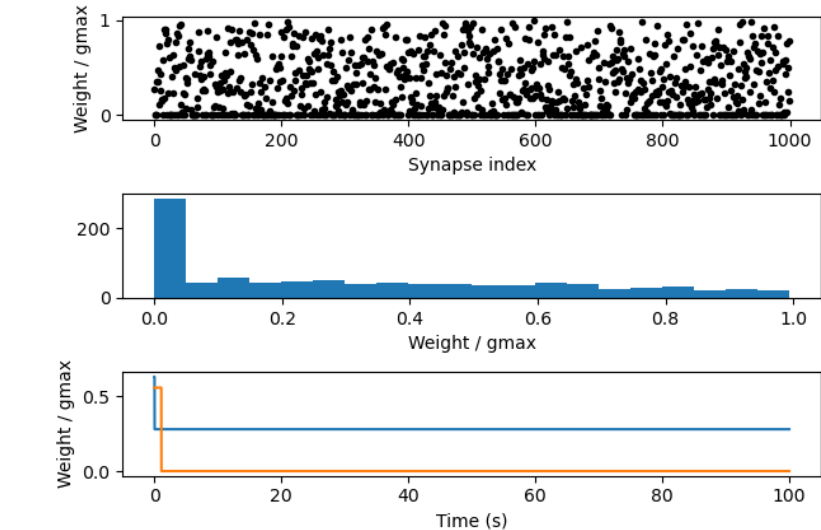


### Avec la forme II de Excitateur à Inhibiteur



Cette forme est obtenue avec uniquement la trace de la décharge présynaptique, qui est négative. Cette forme inhibe fortement les hautes fréquences en ajoutant des poids négatifs au moment des décharges, alors qu'elle va laisser inchanger les fréquences faibles.

Fréquence de 15.0 Hz



<i>taupre</i>	4ms
<i>taupost</i>	20 ms
<i>Apre</i>	-0.01
<i>Apost</i>	0

**Conclusion** Pour que la STDP soit peu sensible aux changements de fréquence, il faut que le poids associé au hautes fréquences (des temps courts) soit faible, et que celui associé au basses fréquences soit plus élevé (temps longs). On peut implémenter cela par une fonction linéaire (affine passant par 0) . Les résultats sont donnés en ANNEXE 1. Une autre solution est d'implémenter l'homéostasie. Chaque fois qu'un neurone décharge, la valeur seuil de décharge à atteindre est augmentée d'un terme *theta* qui décroît au court du temps. Ainsi, pour des fréquences élevées, le changement des poids est diminué car la valeur à atteindre pour provoquer une décharge est plus élevé.



# Réseau de neurones pour classification MNIST



## Bibliographie

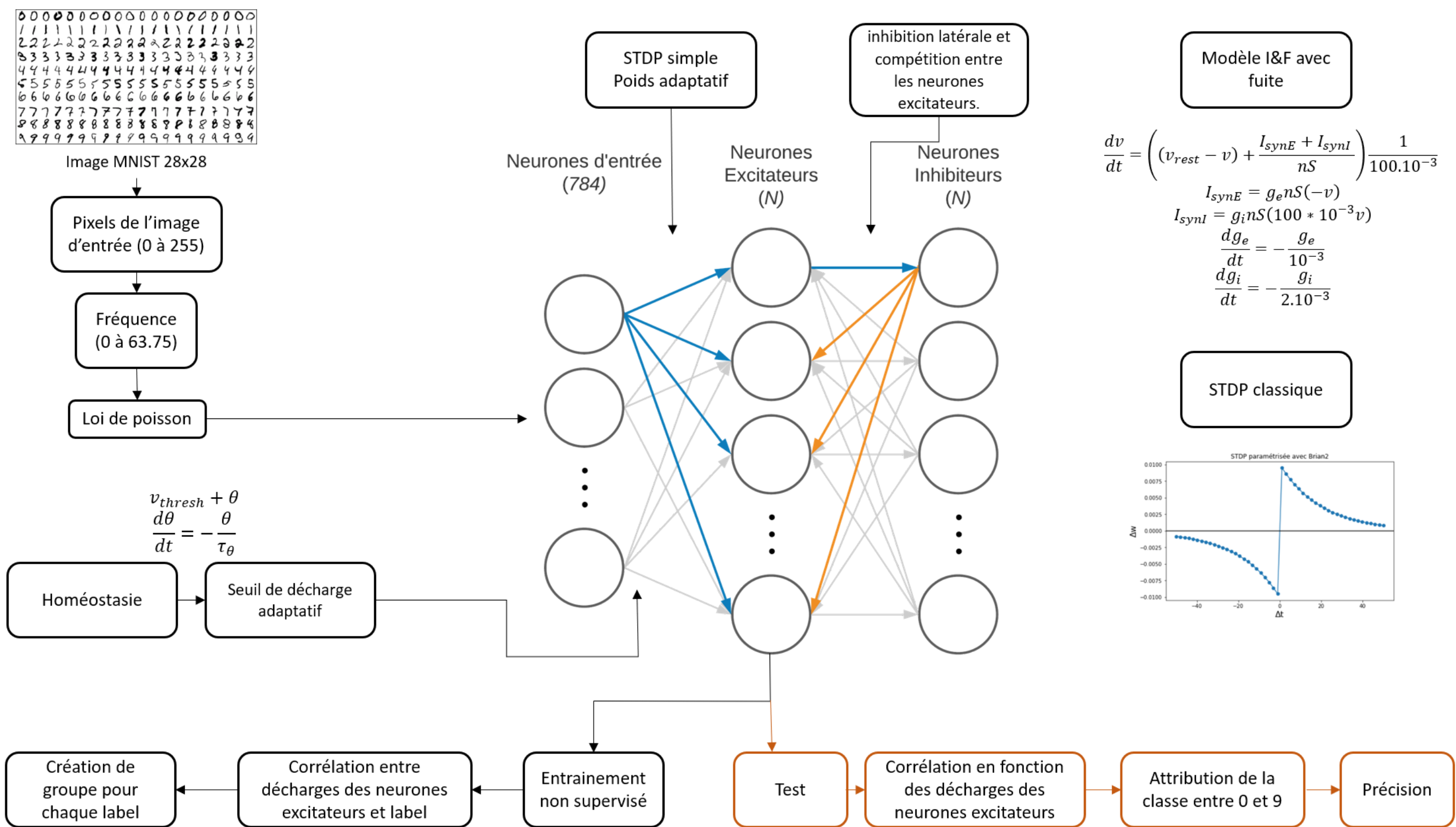
[1] Diehl, Peter U., and Matthew Cook. “Unsupervised Learning of Digit Recognition Using Spike-Timing-Dependent Plasticity.” *Frontiers in Computational Neuroscience* 9 (August 3, 2015). <https://doi.org/10.3389/fncom.2015.00099>.

[2] Goodhill, Geoffrey J., and Harry G. Barrow. “The Role of Weight Normalization in Competitive Learning.” *Neural Computation* 6, no. 2 (March 1994): 255–69. <https://doi.org/10.1162/neco.1994.6.2.255>.

[3] Song, Sen, Kenneth D. Miller, and L. F. Abbott. “Competitive Hebbian Learning through Spike-Timing-Dependent Synaptic Plasticity.” *Nature Neuroscience* 3, no. 9 (September 2000): 919–26. <https://doi.org/10.1038/78829>.

## Remerciements

Brian2, GitHub, LucidChart et StackOverflow  
Base de données MNIST : <http://yann.lecun.com/exdb/mnist>  
stdp-mnist : <https://github.com/peter-u-diehl/stdp-mnist>  
Brian2STDPMNIST : <https://github.com/zxzhijia/Brian2STDPMNIST>



## Effets des paramètre du modèle

Les effets des différentes variables du modèle sont difficiles à étudier car l'espace des paramètres est très grand, et chaque simulation est longue. Cependant nous avons remarqué que certains paramètres peuvent être optimisés afin d'améliorer la performance du réseau. (Figures en Annexe 2)

- Le nombre d'échantillons présentés au réseau lors de l'entrainement augmente généralement la qualité de l'apprentissage.
  - Des taux d'apprentissage élevés permettent d'adapter les poids synaptiques plus rapidement. Nous avons remarqué qu'une valeur trop forte mène à une grande diminution des poids, ce qui, à terme, réduit le nombre de décharge des neurones excitateurs, jusqu'à bloquer l'apprentissage.
  - La normalisation des poids est un autre facteur important car cela permet d'empêcher un synapse prendre une valeur trop extreme et ainsi influencer à lui seul la décharge d'un neurone. Quelques façons de procéder on été évaluées, mais la forme retenue est conforme à celle de [2]. Cependant, nous avons observé un effet négatif de normalisation sur l'apprentissage de notre réseau (jusqu'à -10% de précision). Ces résultats sont contradictoire avec ceux de la littérature mais nous supposons que cela s'explique par le nombre restreint d'exemples d'apprentissage que nous avons utilisés lors de nos entraînements.
  - La connectivité déterministe entre les couches excitatrices et inhibitrices à donné les meilleurs résultats. Les connecter de façon aléatoire et parcimonieuse a fait perdre plus de 20% de précision au réseau.
- La méthode de sélection des neurones de *labels* a été un critère déterminant pour améliorer la précision de notre réseau. Notre première approche consistait à sélectionner un seul neurone par classe, mais cette méthode ne s'est pas révélée assez fiable. Une seconde approche plus robuste, associe un groupe de neurones à chaque classe. Le simple passage de la première à la deuxième méthode a permis de gagner plus de 25% de précision.
  - La variation du nombre de neurones de la couche excitatrice et inhibitrice n'a étonnement pas montré un grand impact sur la performance du réseau. Par exemple, le changement de 400 à 100 neurones dans chacune de ces deux couches n'a pas été corrélé avec une baisse de précision lors de nos expériences. Nous supposons cependant qu'un nombre très faible de neurones réduira les capacités de classification du réseau. Et un nombre très grand nécessiterait un long apprentissage avant de montrer un différence de performance.
  - Le fait de permettre la variation du seuil de décharge, c'est à dire d'appliquer l'homéostasie, a permis de bonifier la performance de la classification. Ne pas l'appliquer fait perdre jusqu'à 20% de précision de classification après l'entrainement.

## Discussion des résultats, conclusion

Il n'a pas été aisé de trouver les bons paramètres permettant au réseau de neurones de converger vers une solution satisfaisante pour résoudre le problème de classification MNIST. En effet, comme le taux d'apprentissage est faible, les poids ne change pas rapidement avec les itérations et ainsi plusieurs jours de calculs serait nécessaire pour exploiter pleinement un tel modèle. De plus, la carte d'activation montre qu'un faible de nombre de décharge suffit à générer de l'apprentissage. Cependant une exploration de ces paramètres sur un petit sous ensemble du jeu de donne nous a permis au final d'entraîner un réseau ayant une précision de 63%.

L'article [1] montre que plusieurs règles de STDP peuvent être envisagées et qu'elles ne produisent pas les mêmes résultats. La STDP vu en classe a été évaluée dans un premier temps, puis nous avons essayé une équation permettant de reproduire la forme II de excitateur à excitateur présentée précédemment. Il est difficile de démontrer la supériorité d'une forme par rapport à une autre en s'appuyant uniquement sur les courtes simulations que nous avons pu réaliser. Étendre ces dernières pourrait faire partie d'un projet ultérieur. De plus, l'application des règles de la STDP pourrait aussi se faire dans les groupes de synapses entre la couche excitatrice et la couche inhibitrice alors que dans les articles et nos travaux, les poids de ces synapses étaient fixes.



Annexe 1 – Variation de la fréquence

Forme I de Excitateur à Excitateur

Forme I de Excitateur à Inhibiteur

Forme II de Excitateur à Excitateur

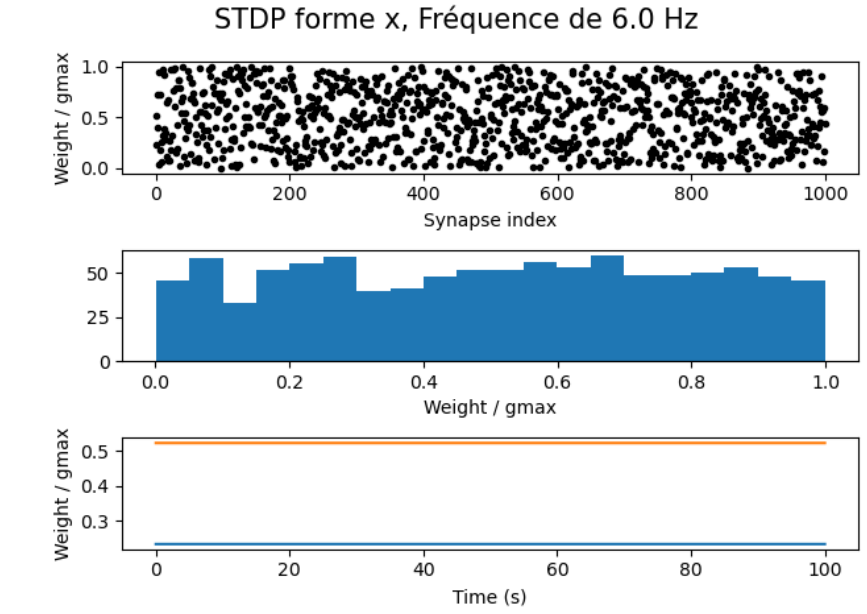
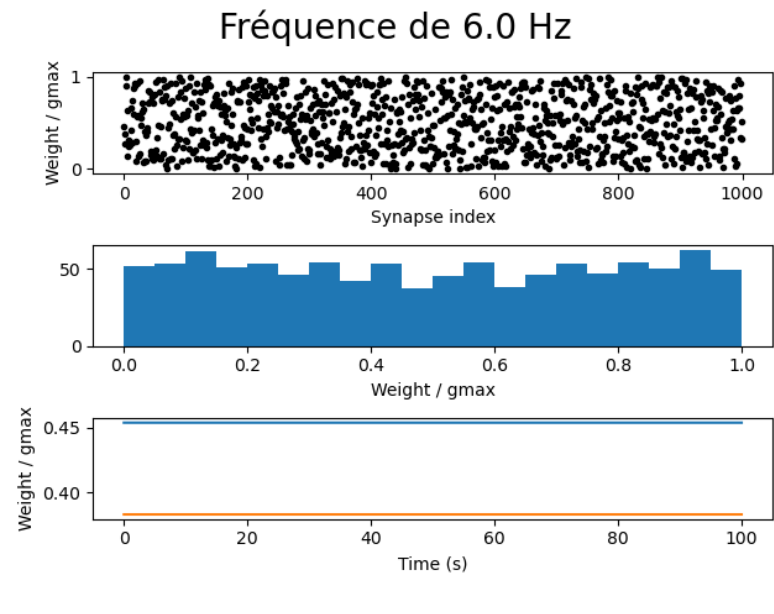
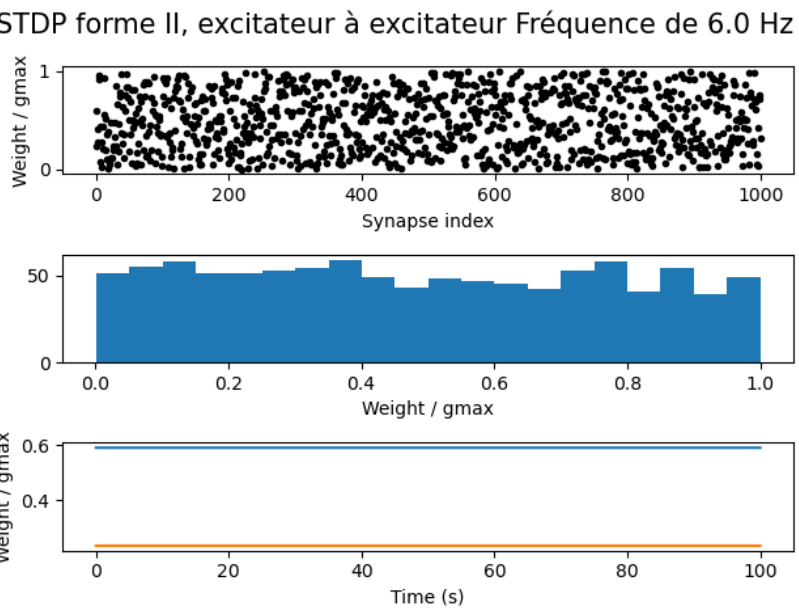
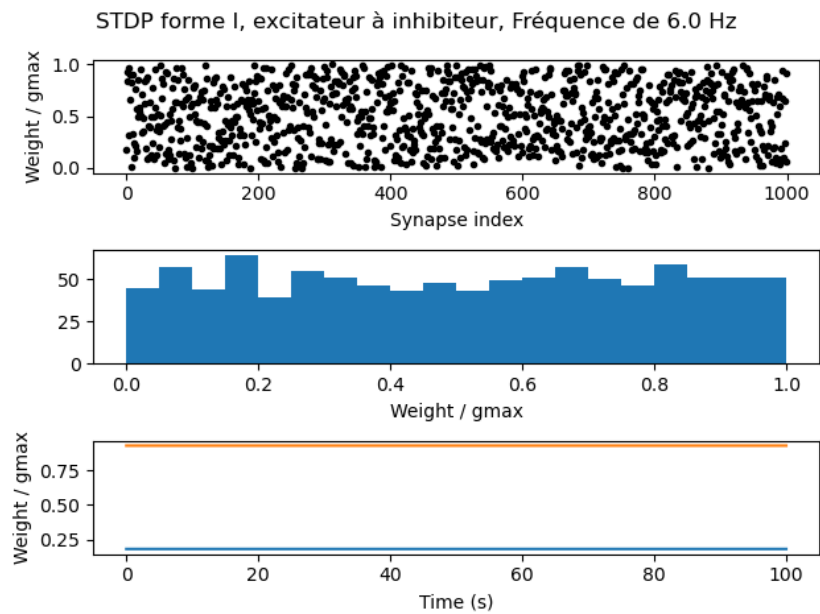
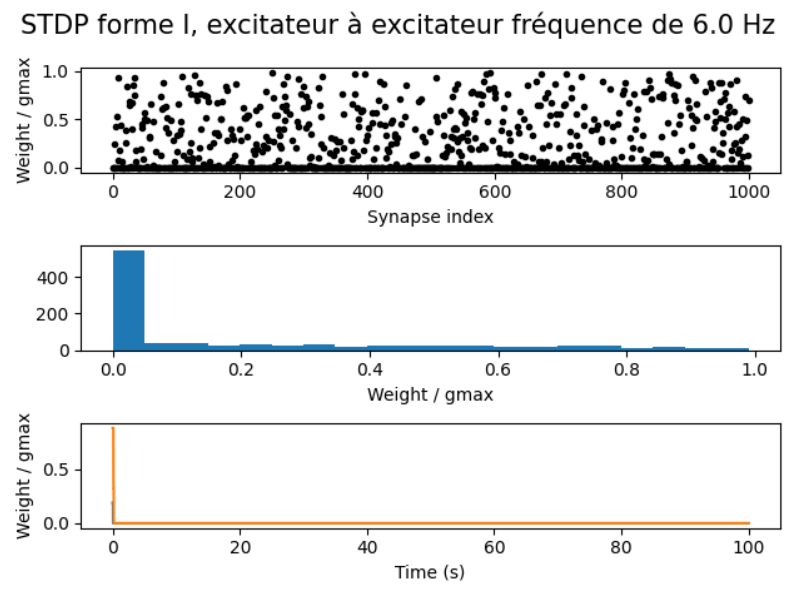
Forme II de Excitateur à Inhibiteur

Forme linéaire

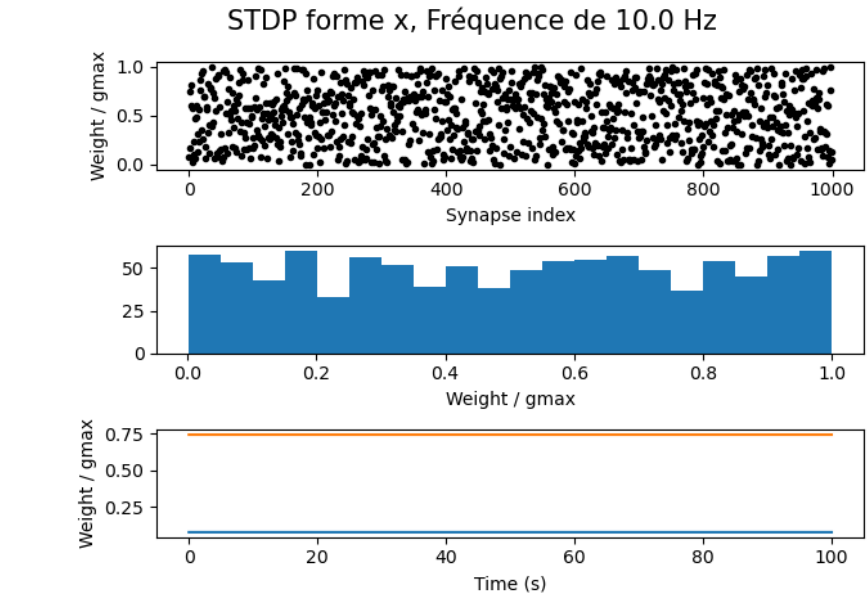
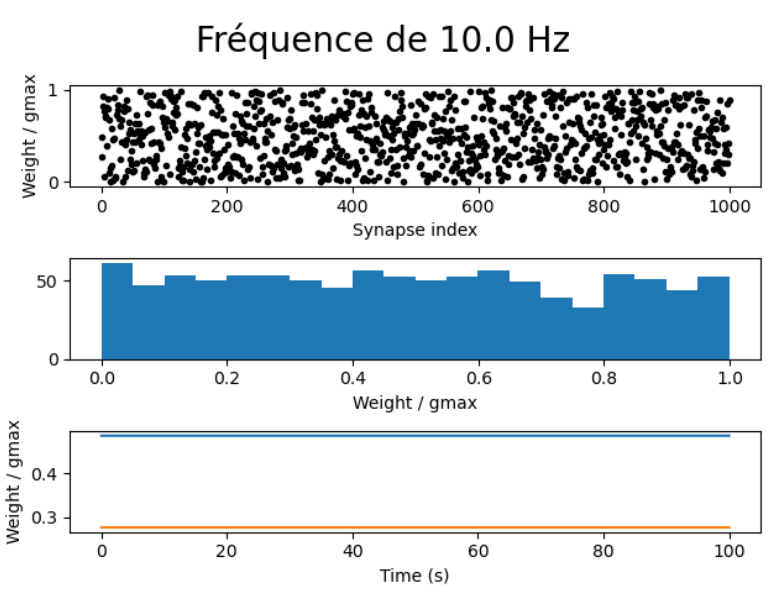
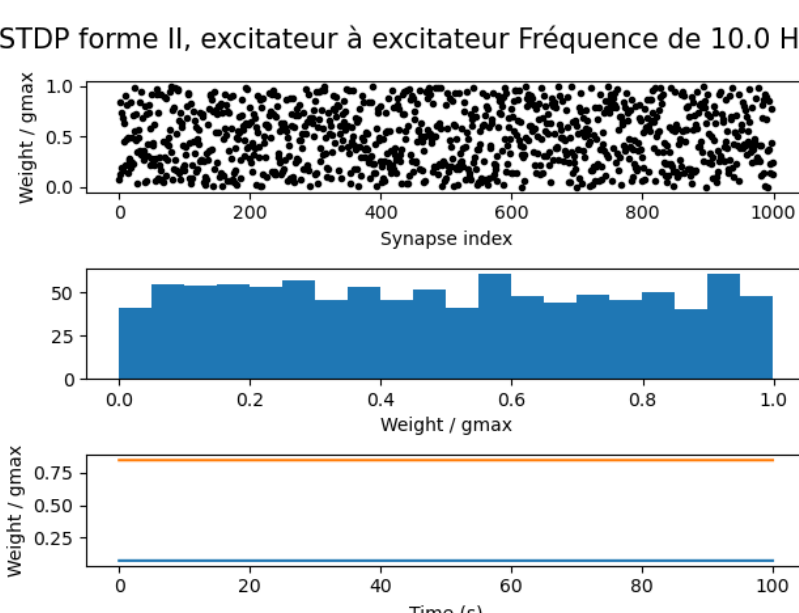
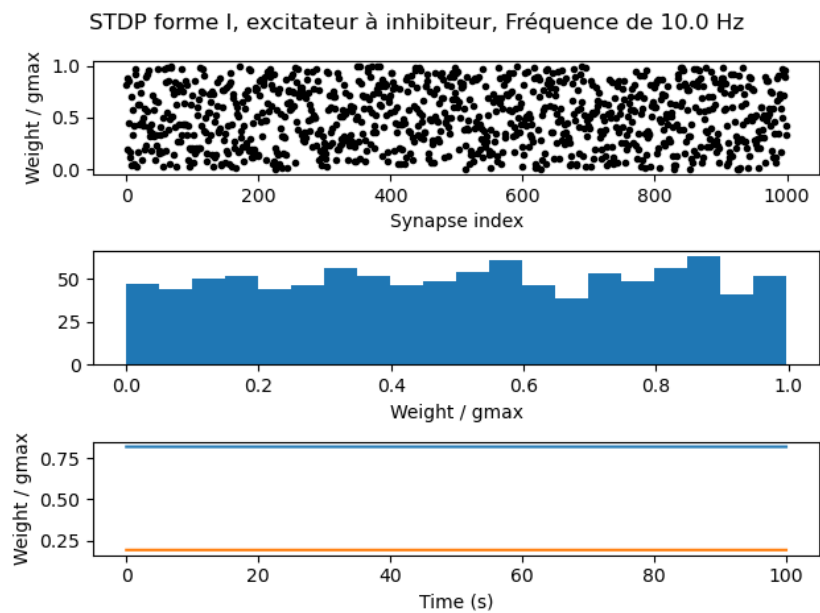
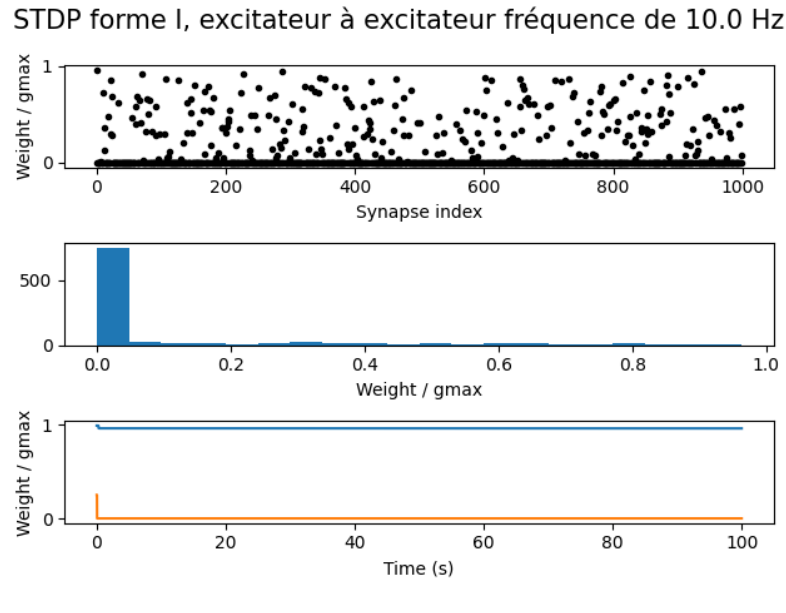
$$w = w + \frac{a_{post}}{\tau_{post}} \Delta t = w + \frac{a_{pre}}{\tau_{pre}} \Delta t = w + 5|\Delta t|$$



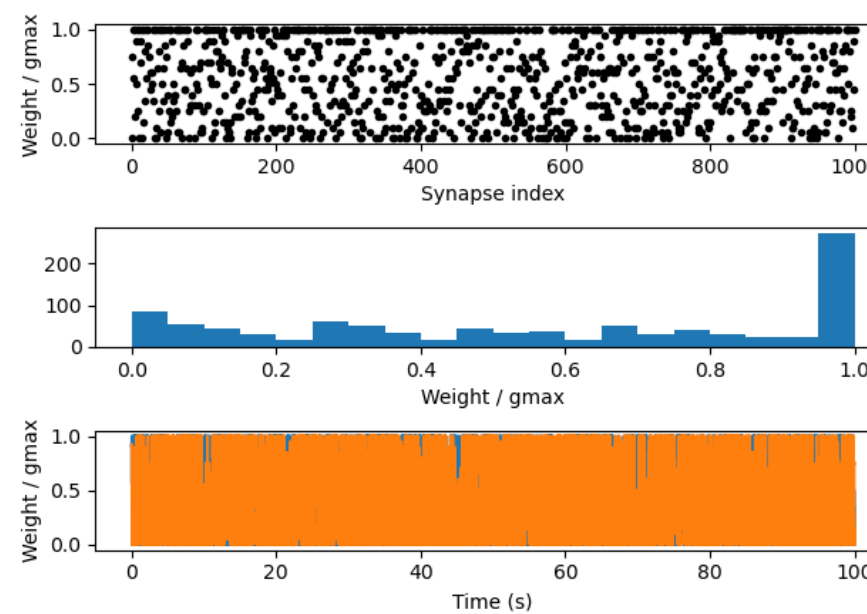
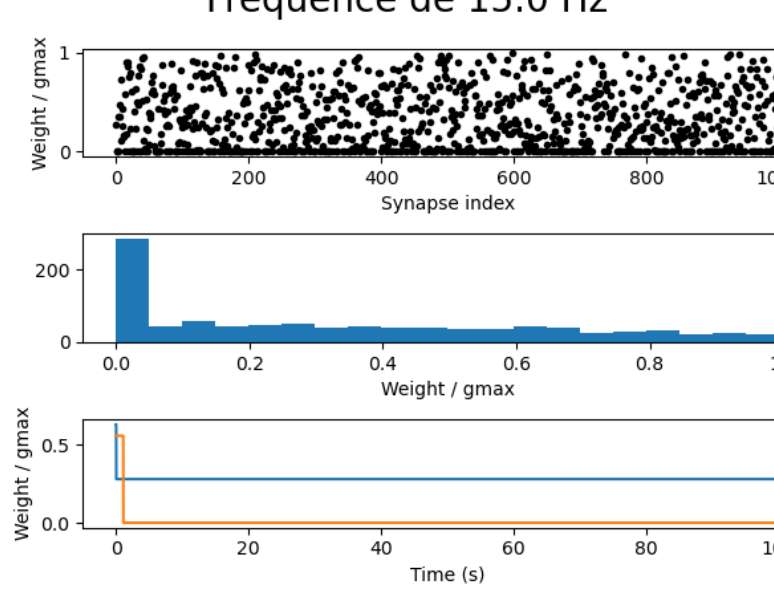
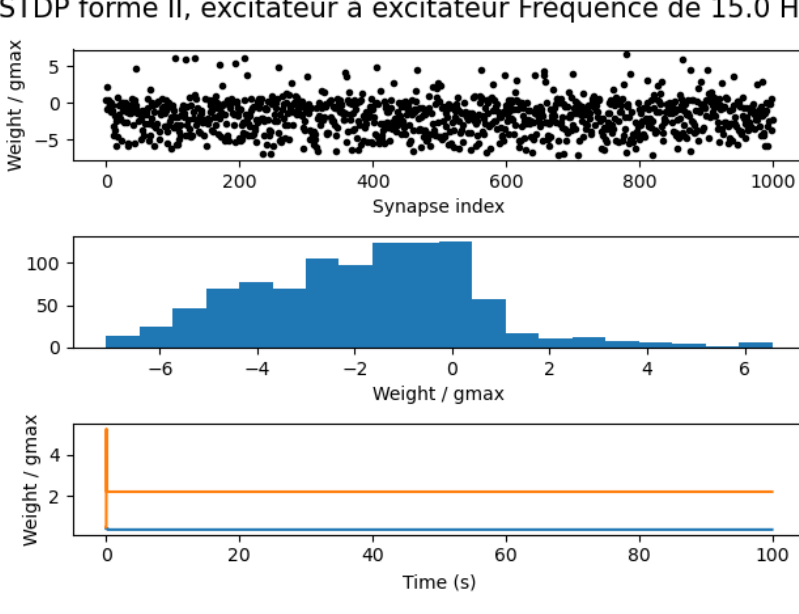
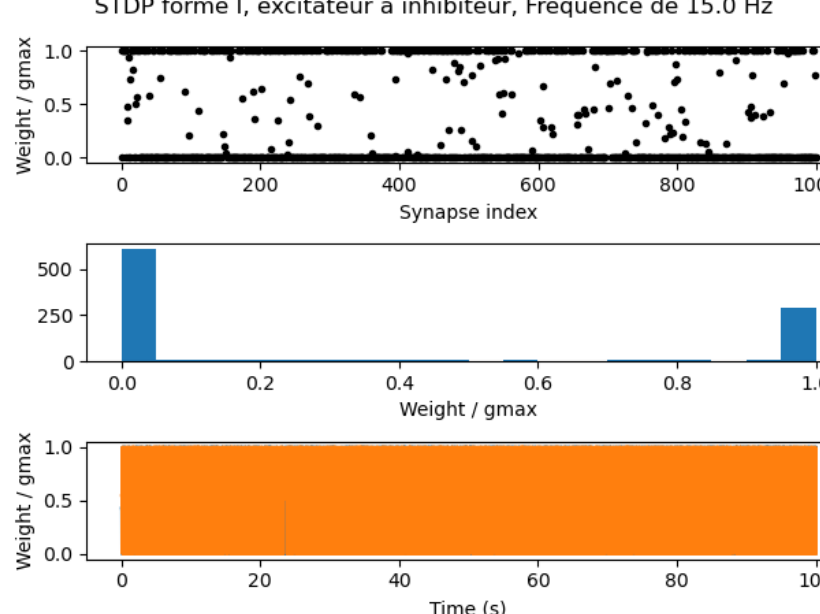
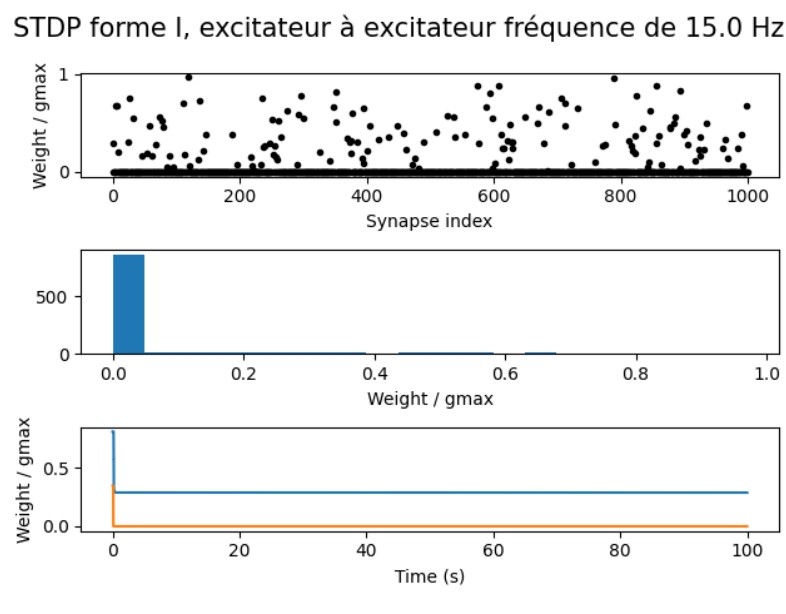
6Hz



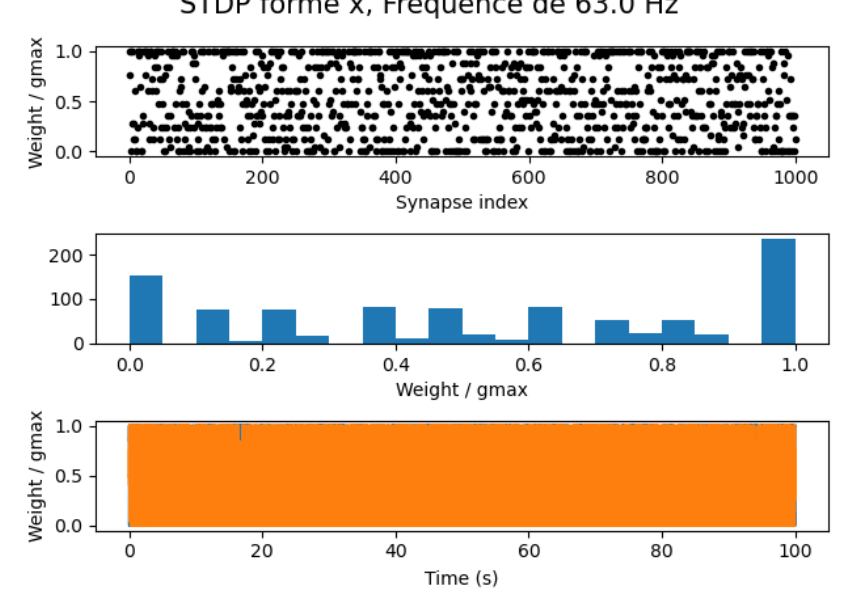
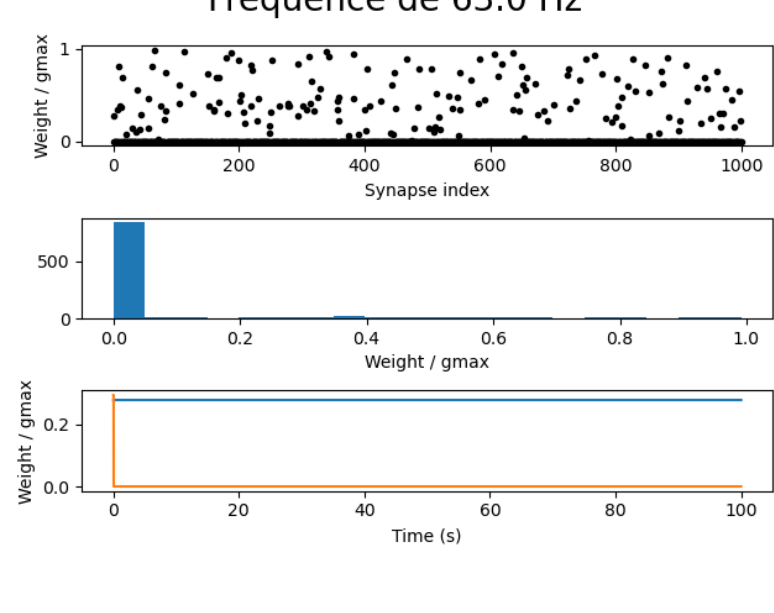
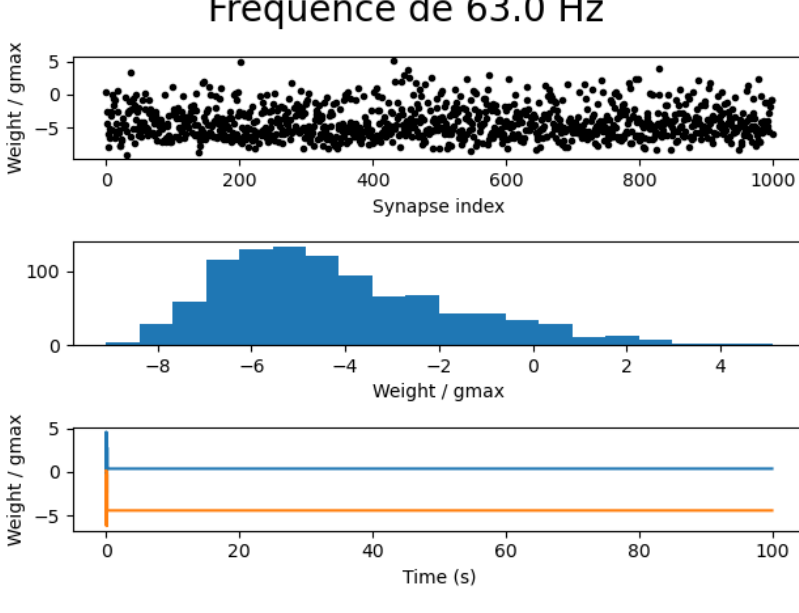
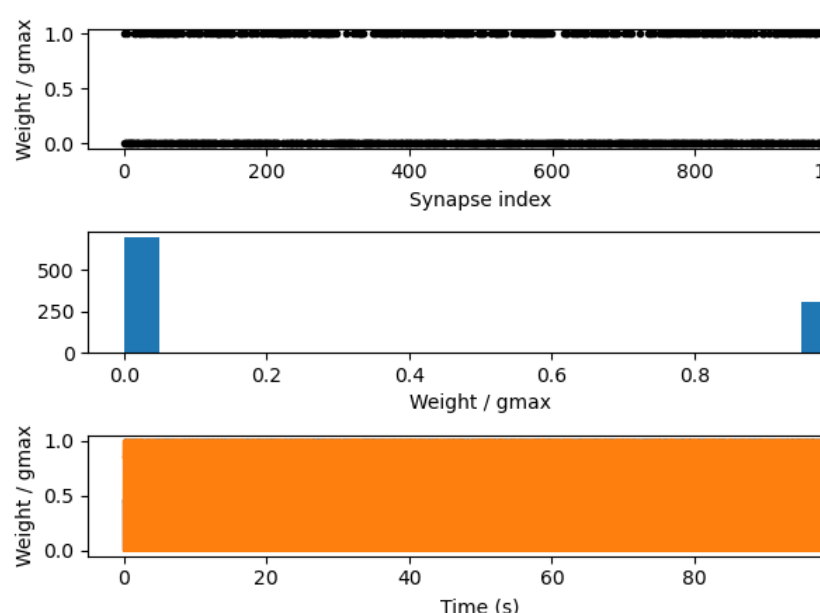
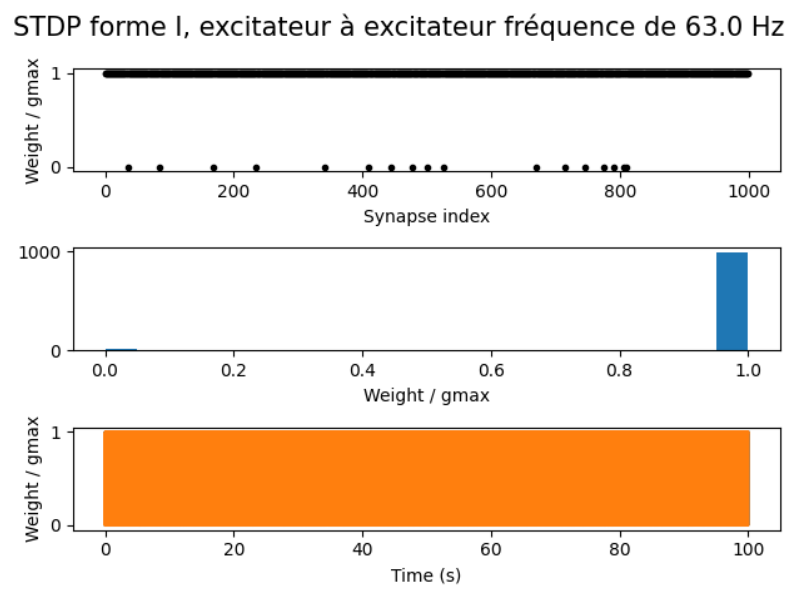
10Hz



15Hz



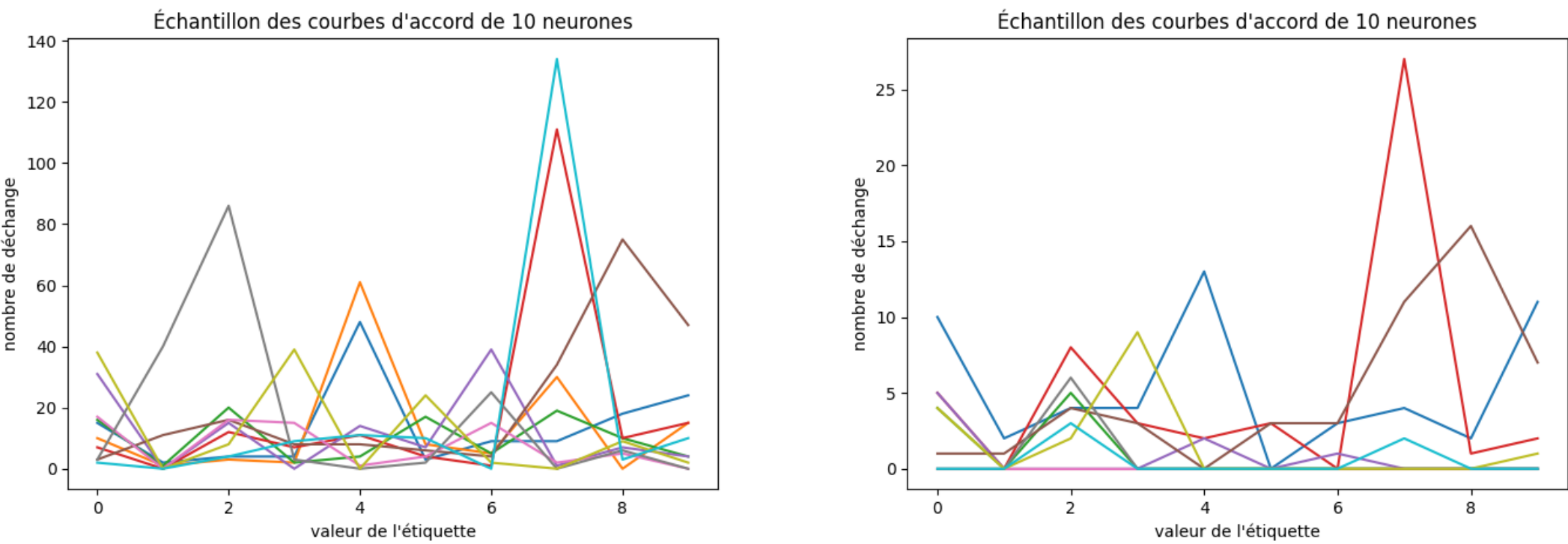
63Hz



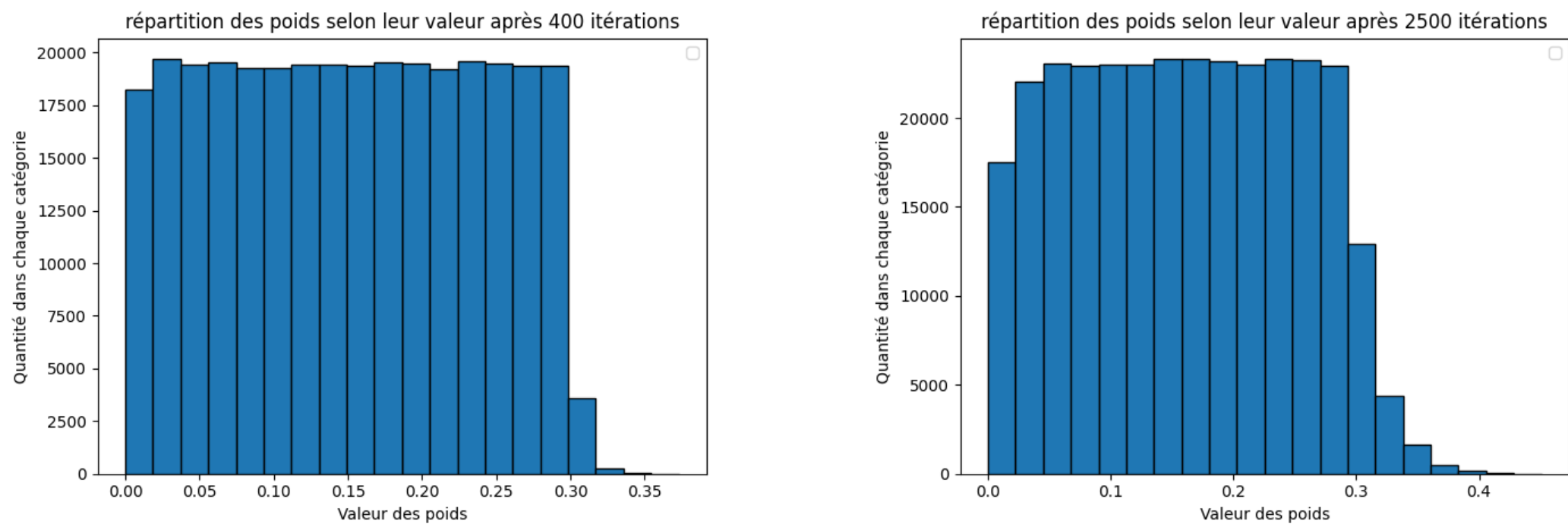


Annexe 2 – Apprentissage MNIST

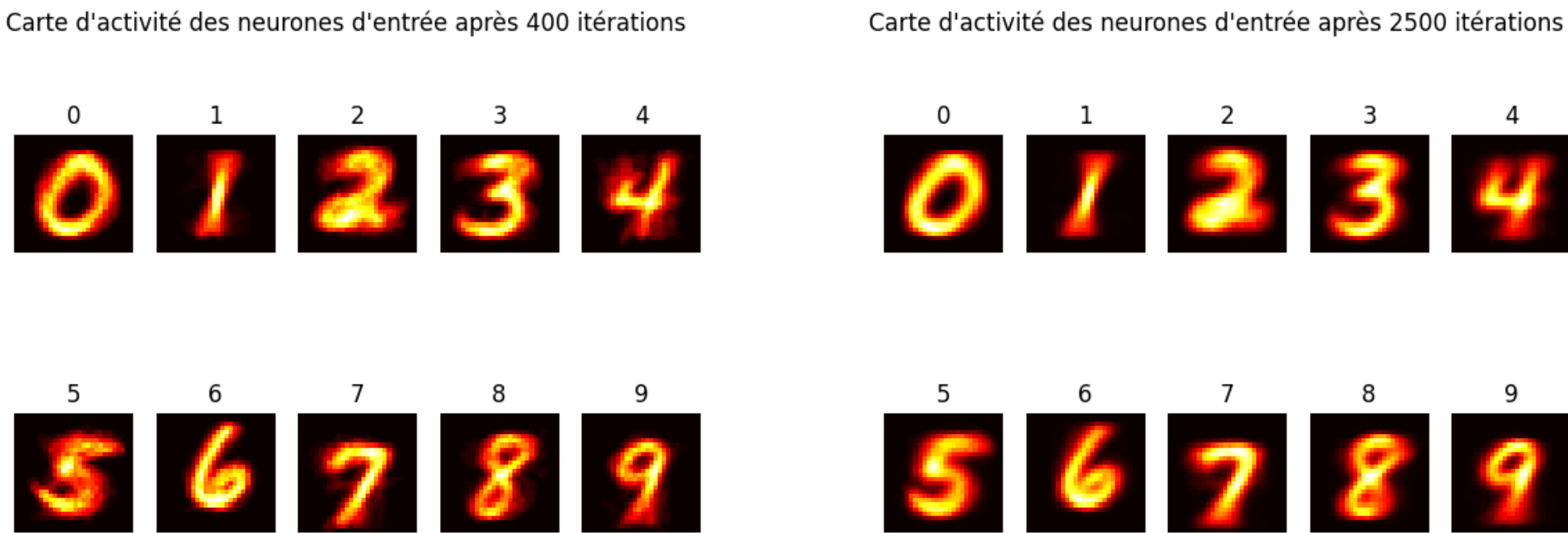
Précision de classification respective pour cette colonne: 48% et 63%



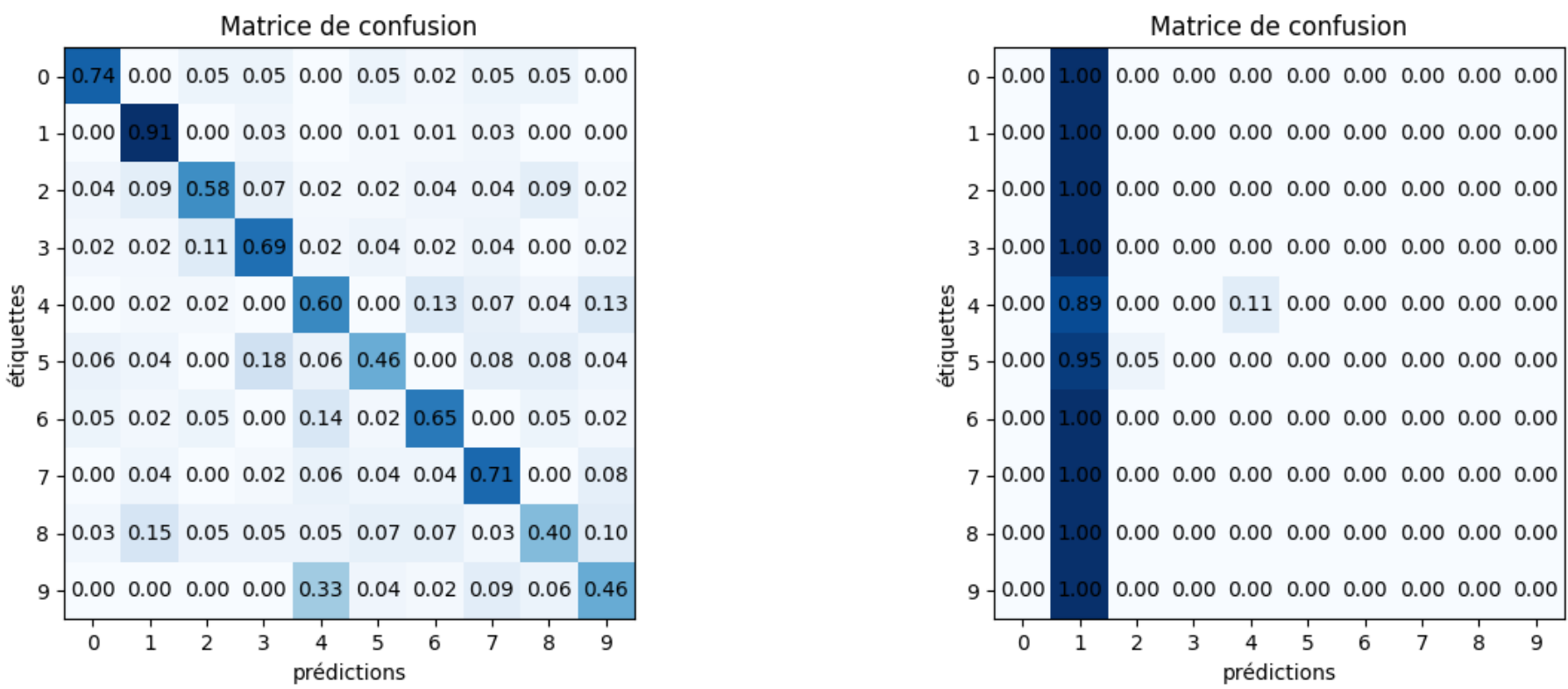
Comparaison des courbes d'accord après 400 images (gauche) et 2500 (droite).



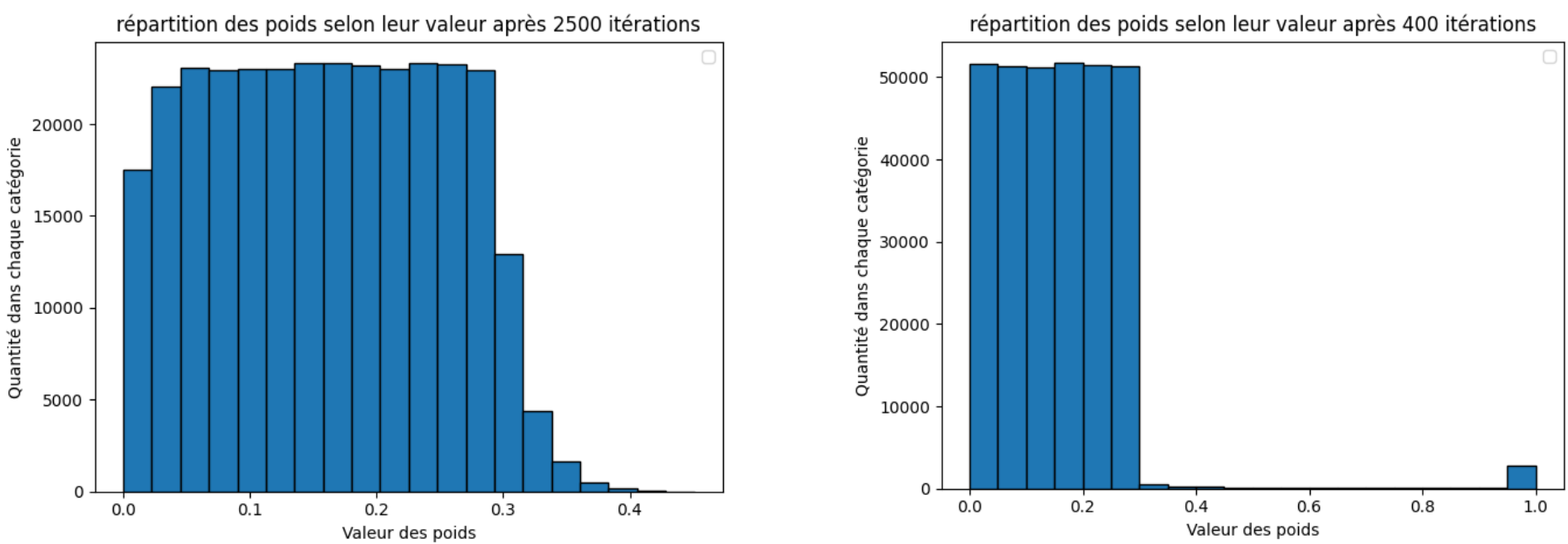
Comparaison des histogramme des poids des synapses après 400 images (gauche) et 2500 (droite)



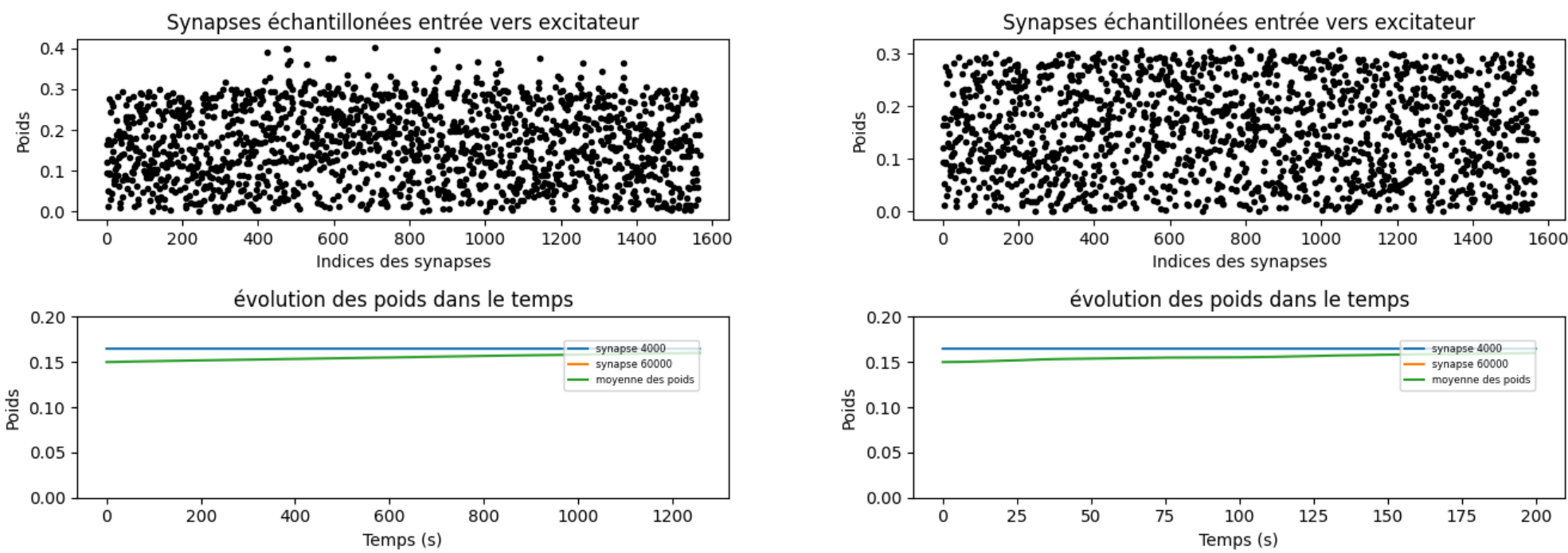
Comparaison des cartes d'activité des neurones de la couche d'entrée après 400 images (gauche) et 2500 (droite)



Comparaison de matrices de confusion suite à un apprentissage réussi ayant 63% de précision (gauche) et un qui a échoué avec 15% de précision (droite)



Comparaison de la répartition des poids suite à un apprentissage réussi ayant 63% de précision (gauche) et un qui a échoué avec 15% de précision (droite)



Comparaison de l'évolution des poids suite à un apprentissage réussi ayant 63% de précision (gauche) et un qui a échoué avec 15% de précision (droite)



Annexe 3 – Résultats en fonction des paramètres de simulation

Variable	Valeurs	Valeurs proposées pour nos tests	Nombre de possibilité
Nombre d'image d'entrainement	70000	200, 800, 6000	3
Nombre d'image de test	20% de Xtrain	20% de Xtrain	1
Variable d'Homoéostasie	Activé/Désactiver	Activé/Désactiver	2
Taux d'apprentissage pré synaptique	de 1e-6 à 1	par puissance de 10	7
Taux d'apprentissage post synaptique	de 1e-6 à 1	par puissance de 10	7
Poids Maximal	De 10 à 0.01	10;1;0.1;0.01	4
Nombre de neurone exciteateur	100, 400, 1600	100, 400, 1600	3
Nombre de fois que l'on présente le jeu de donné	1,3,6	1,3,6	3
Connectivité entre Entrée et exciteateur	0 à 100 %	Pas de variation	1
Connectivité entre couche excitatrice et inhibitrice	0 à 100 %	0.5; 0.25; 0.9, 1; i = j	5
Connectivité entre couche inhibitrice et excitatrice	0 à 100 %	0.5; 0.25; 0.9, 1; i != j	5
Normalisation des poids	True or False	True or False	2
Type de classification	Single or Group	Single or Group	2

- **Nombre d'images d'entraînement** Nombre d'images que l'on présente au réseau pendant la phase d'entraînement.
- **Nombre d'images de test** Nombre d'images que l'on présente au réseau pendant la phase de test.
- **Homéostasie** Nous activons ou désactivons l'homéostasie pour chaque test.
- **Taux d'apprentissage** Ce qui permet de réguler l'apprentissage en augmentant le poids de façon partielle pendant le processus de STDP. Cela permet d'éviter d'atteindre des valeurs de poids trop élevées trop rapidement.
- **Nombre de neurones excitateurs** Nombre de neurones que l'on définit pour la couche excitatrice ET inhibitrice.
- **Nombre de fois où l'on présente le jeu de données (époques)** Permet d'améliorer l'apprentissage sur certains types d'images.
- **Connectivité entre les couches** Permet de faire varier l'influence des neurones inhibiteurs sur le réseau et d'obtenir plus ou moins de compétition entre les réseaux.
- **Normalisation des poids** Normalisation des poids par colonnes d'après [2] avant chaque itération d'entraînement.
- **Type de classification** Deux types de classification, *single* et *group*. La classification *single*, prend le neurone qui a davantage déchargé pendant l'entraînement pour un *label* donné, et la classification pendant le test sera effectuée en fonction des 10 neurones *labels* sélectionnés. Pour la classification en groupe de neurone, chaque neurone de la couche excitatrice est associé au label pour lequel il a le plus déchargé pendant l'entrainement. Pendant la phase de test, le groupe de neurone correspondant à un *label* dont la moyenne de décharge est la plus grande, donnera le *label* de l'image testée.

Variables :	nb_train_samples	nb_test_samples	θ et τ_θ	nu_pre	nu_post	wmax	nb_excitator_neurons	nb_epoch	connectivity[0]	connectivity[1]	connectivity[2]	normalization	labeled_neurone	final accuracy
Valeur défaut :	60000	10000	ACTIVE	0,0001	0,01	1	400	1	1	i = j	i != j	TRUE	1 par classe	10,0%
Simulations	200	20	default	0,01	default	default	100	default	0,5	0,25	0,9	default	default	15,0%
	200	20	DESACTIVE	0,01	default	default	100	default	0,5	0,25	0,9	default	default	5,0%
	200	20	default	default	0,1	default	100	default	0,5	0,25	0,9	default	default	20,0%
	200	20	default	default	0,0001	default	100	default	0,5	0,25	0,9	default	default	15,0%
	200	20	default	0,000001	0,1	default	100	default	0,5	0,25	0,9	default	default	30,0%
	200	20	default	0,000001	0,1	0,1	100	default	0,5	0,25	0,9	default	default	10,0%
	200	20	default	0,000001	0,1	10	100	default	0,5	0,25	0,9	default	default	30,0%
	200	100	default	0,000001	0,1	10	100	default	0,5	0,25	0,9	default	default	22,0%
	800	100	default	0,000001	0,1	10	100	default	0,5	0,25	0,9	default	default	14,0%
	800	100	default	0,000001	0,1	10	100	default	0,5	0,9	0,9	default	default	10,0%
	800	100	default	0,000001	0,1	10	100	default	0,5	0,05	0,9	default	default	20,0%
	200	100	default	default	0,1	10	default	default	0,8	default	default	default	default	26,7%
	400	200	default	0,000001	0,1	10	100	default	default	default	default	default	default	18,0%
	400	200	DESACTIVE	0,000001	0,1	10	100	default	default	default	default	default	default	16,0%
	400	200	DESACTIVE	0,001	0,1	10	100	default	default	default	default	default	default	15,0%
	400	200	default	0,001	0,1	10	100	default	default	default	default	default	default	13,0%
	400	200	default	default	0,1	default	100	default	default	default	default	default	default	30,0%
	400	200	default	default	0,1	10	100	default	default	default	default	default	Groupe	33,0%
	400	200	default	default	0,1	default	100	default	default	default	default	default	Groupe	32,0%
	800	200	default	0,00001	default	default	1000	default	default	default	default	default	Groupe	49,0%
	800	200	DESACTIVE	0,00001	default	default	1000	default	default	default	default	default	Groupe	11,0%
	800	200	default	0,00001	0,001	default	1000	default	default	default	default	default	Groupe	33,0%
	6000	200	default	0,00001	0,001	default	1000	default	default	default	default	default	default	42,0%
	800	200	default	0,000001	0,001	default	1000	default	default	default	default	default	Groupe	50,0%
	800	200	DESACTIVE	0,00001	0,001	default	1000	default	default	default	default	default	default	8,5%
	400	200	default	0,000001	0,001	default	100	default	default	default	default	default	Groupe	41,0%
	400	200	default	0,000001	0,001	default	100	default	default	default	0,5	default	Groupe	21,0%
	400	200	default	0,000001	0,001	default	100	default	default	0,5	default	default	Groupe	21,0%
	2500	750	default	0,001	0,05	default	default	default	default	default	default	default	default	16,0%
	400	200	default	default	default	default	default	default	default	default	default	default	default	18,5%
	400	200	default	default	default	default	default	default	default	default	default	FALSE	default	23,0%
	800	200	default	default	default	default	default	default	default	default	default	FALSE	default	16,0%
	400	200	default	0,05	0,5	default	default	default	default	default	default	default	default	Pas terminé
	400	200	default	default	default	default	default	default	default	default	default	default	Groupe	39,5%
	400	200	default	default	default	default	100	default	default	default	default	default	Groupe	38,0%
	400	200	default	default	default	default	default	default	default	default	default	FALSE	Groupe	48,0%
	400	200	default	0,0005	0,05	default	default	default	default	default	default	FALSE	Groupe	15,0%
	400	200	default	0,005	0,05	default	default	default	default	default	default	FALSE	Groupe	27,5%
	400	200	default	default	default	default	default	default	default	0,0025	0,9	FALSE	Groupe	18,5%
	400	200	default	default	default	default	20	default	default	default	default	FALSE	Groupe	39,0%
	2500	500	default	default	default	default	default	default	default	default	default	FALSE	Groupe	63,0%
	400	100	DESACTIVE	0,005	0,1	10	default	default	0,8	default	default	FALSE	Groupe	Problème mem
	350	100	DESACTIVE	0,005	0,1	10	default	default	0,8	default	default	FALSE	Groupe	30,0%

Nous avons 13 paramètres que nous pouvons faire varier indépendamment. Pour analyser leur impact sur l'apprentissage, nous avons défini des plages de variations. Si nous devons effectuer toutes les combinaisons, il faudrait donc effectuer 1.058.400 simulations. Nous choisissons d'étudier l'impact de certains paramètres de façon progressive, de manière itérative afin d'obtenir le meilleur apprentissage.

Répartition des tâches

Fonctionnement STDP, adaptation et influence de la fréquence : Soline Bernard  
Code informatique, fonctionnement, choix des types de STDP : Antoine Marion, Soline Bernard, Victor Yon  
Normalisation des poids : Antoine Marion  
Type de classification : Bernard Soline  
Implémentation, choix et affichage des graphiques : Antoine Marion  
Organisation du code informatique : Victor Yon  
Simulations et tests : Antoine Marion, Soline Bernard, Victor Yon