CentraleSupélec

Deep learning with coherent nanophotonic

Article Review, Photonic Computing

DOI: 10.1038/NPHOTON.2017.93

Antonin Grateau

Février 2022

Table des matières

1 Contexte

Les "réseaux de neurones" sont un formalisme de Machine Learning directement inspiré du cerveau humain. Ils consistent en l'interconnection d'un grands nombres de neurones qui interagissent entre eux pour traiter une information. Leur récent développement dans les dernières décennies a fait d'eux des outils indispensables. Pour cause, ils dépassent aujourd'hui l'intelligence humaine dans de nombreux domaines (comme la reconnaissance d'image, le diagnostique médical, ou même le jeu d'échecs). Ils sont aujourd'hui implémentés sur des systèmes hardware électroniques où le paradigme de calcul est celui de Von Neumann. Ces architectures sont relativement inadaptées pour les réseaux de neurones. Pour cause, un réseau de neurones consistent en la parallélisation d'un grand nombres d'opérations alors que le calcul de Von Neumann est intrinsèquement séquentiel. Il en résulte que la consommation énergétique est énorme, et que la vitesse de calcul atteint ses limites. Par exemple, le célèbre AlphaGo (algorithme de Google Deepmind qui a battu le champion du monde de Go) consomme en moyenne 1 MW, soit presque 10 000 fois plus que ses adversaires humains. L'amélioration des systèmes physiques support de réseaux de neurones constitue donc un enjeu énergétique énorme.
L'article étudié [1] propose une nouvelle approche photonique pour le hardware des réseaux de neurones. Le système est complètement optique et atteint des performances qui dépasseraient celles des systèmes électroniques à l'état de l'art.

2 Quelques exemples de systèmes optiques neuro-inspirés

Bien que la recherche sur les réseaux de neurones optiques soit très récente, certains systèmes ont déjà fait leurs preuves. Le but de cette partie et de mentionner les idées les plus intéressantes et les plus prometteuses.
Cette partie est largement inspirée de l'article de revue [2].

2.1 Fonctionnement d'un neurone

Pour comprendre les ingrédients nécessaires pour construire de tels systèmes, il convient de rappeler préalablement le fonctionnement d'un neurone.

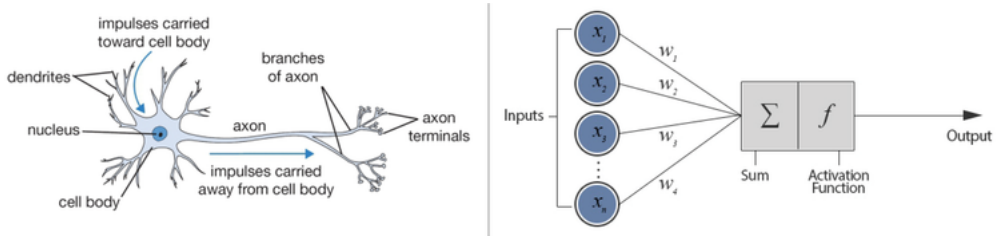

FIGURE 1 – Neurone biologique VS Neurone artificiel

1. Les noeuds d'entrée sont associés à des valeurs x_i réelles.
2. Les connections entre les noeuds d'entrée et le neurone sont associées à des poids w_i qui sont aussi des valeurs réelles.
3. Les entrées sont rassemblées sous la forme d'une somme pondérée à laquelle on rajoute généralement un biais : $b + \sum x_i w_i$
4. La sortie est alors donnée par le produit de cette somme par une fonction non-linéaire (qui, généralement, ne s'active que lorsqu'un certain seuil est dépassé) : $y = f(b + \sum x_i w_i)$

On comprend ainsi que les deux éléments essentiels d'un neurone sont :

— **une opération linéaire** : $b + \sum x_i w_i$

— **une opération non-linéaire** : la fonction d'activation f.

2.2 Réalisation optique d'une opération linéaire

2.2.1 Implémentation via la transformée de Fourier

L'idée est de faire la multiplication $x_i w_i$ en réalisant un produit de convolution dans l'espace fréquentiel ($F(\nu_x, \nu_y) * G(\nu_x, \nu_y) \Leftrightarrow f(x, y)g(x, y)$). Pour se faire le faisceau d'entrée contenant toute l'information des x_i va traverser un modulateur spatiale de lumière (SLM) placé dans le plan focal d'une lentille. Cette dernière va, par nature, réaliser la transformée de Fourier inverse mais également l'opération de sommation (grâce à la focalisation). On récupère ainsi une intensité $\sum x_i w_i$ dans le plan de Fourier [3].

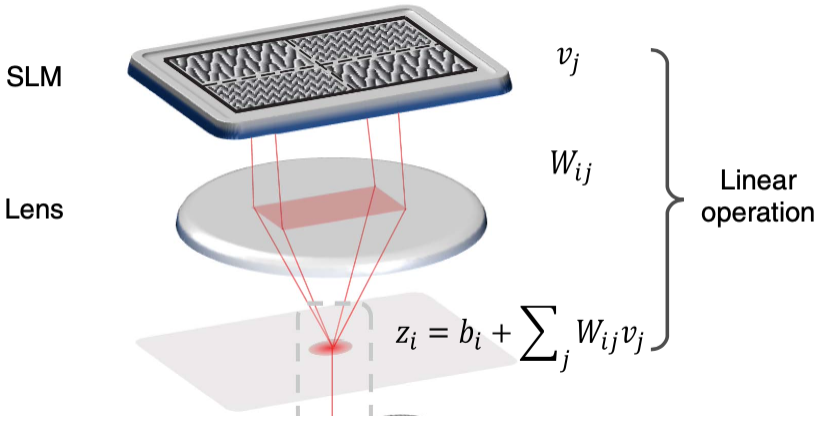

FIGURE 2 – Opération linéaire grâce à un système SLM-Lentille [3].

2.2.2 Implémentation via la diffusion

La diffusion va, par nature, mélanger la lumière incidente dans l'espace. Le principe de la méthode est qu'en contrôlant la position et la concentration des agents diffusant, on est capable en quelques sortes de contrôler les poids w_i. Le résultat est donc similaire à l'opération linéaire matricielle du réseau de neurone classique.
Les milieux capables de réaliser une telle tâche sont appelés "nano-photonic neural medium" (NNM).

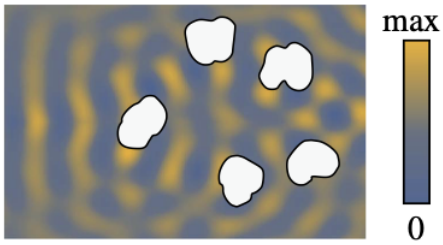

FIGURE 3 – Lumière spatialement redistribuée par des nanostructures [4]

2.2.3 Implémentation via le multiplexage en longueur d'onde

Le multiplexage en longueur d'onde permet de faire passer simultanément plusieurs informations (encodées sur plusieurs longueurs d'ondes différentes) dans une même fibre optique. L'idée est alors de réaliser l'opération linéaire en affectant des poids différents aux différentes gammes de longueur d'onde. Pour se faire, il s'agit dans un premier temps de séparer ces longueurs d'onde dans le temps grâce à une fibre dispersive, puis de les moduler grâce à un modulateur d'intensité. Enfin, on recombine le signal temporellement élargi en une impulsion grâce à une autre fibre dispersive. Ainsi, on a bien réalisé une opération de la forme $\sum x_i w_i$ [5].

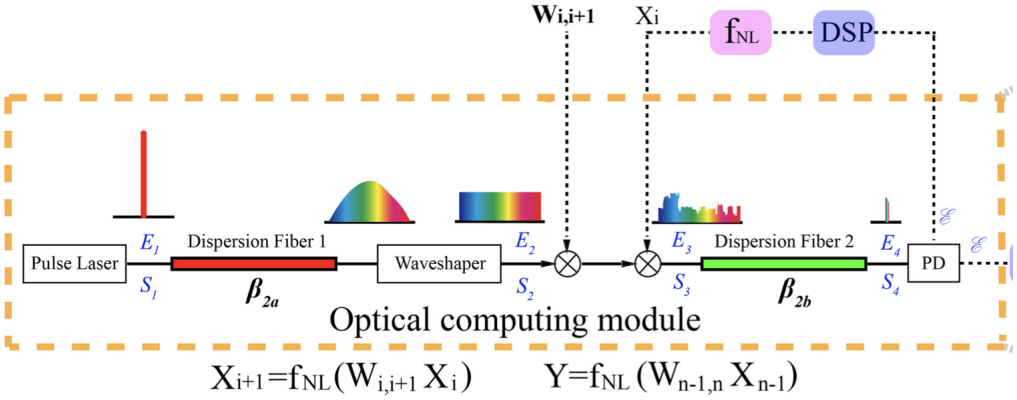

FIGURE 4 – Schéma de la réalisation d'une opération linéaire à l'aide du multiplexage [5].

2.3 Réalisation optique d'une opération non linéaire

Il est relativement difficile de réaliser une opération non linéaire optique. C'est un casse-tête qui limite le développement des réseaux de neurones photoniques (la plupart du temps, les fonctions non linéaires sont implémentées électroniquement). Néanmoins, certains effets optiques non linéaires pourraient permettre de réaliser optiquement ce genre d'opérations.

2.3.1 Implémentation via l'absorption saturable

Certains milieux vont voir leur propriétés physiques changer sous l'effet de la lumière. En particulier, ils existent des cristaux photoniques qui absorbent différemment en fonction de l'intensité du champ électrique. Leur coefficient d'absorption suit une loi de la forme : $\alpha(I) = \alpha_0/(1 + \frac{I}{I_0})$. Un rapide coup d'oeil sur cette formule nous fait comprendre la dynamique de ce genre de cristaux : le coefficient d'absorption est une fonction décroissante de l'intensité d'entrée.

Remarque : Ceci est vrai pour tous les matériaux, cependant cette variation est palpable pour des intensités faibles (de l'ordre de I_c) dans le cas des absorbants saturables.

2.3.2 Implémentation via un phénomène de transparence induite par électromagnétisme

Des atome ^{85}Rb (rubidium) sont confinés dans un piège magnétique. Un laser (ω_c) dit "sonde" traverse le piège. L'intensité de sortie de ce dernier est alors contrôlée par l'intensité d'un autre laser dit "couple" (ω_p) (figure 5, (a)). Le laser couple est en résonance avec les transitions $|1\rangle \longrightarrow |3\rangle$ des atomes alors que le laser couple est en résonance avec les transitions $|2\rangle \longrightarrow |3\rangle$ (figure 5, (b)). Sans présence du laser

couple, les atomes absorbent naturellement les photons du laser sonde. En présence du laser couple, le niveau d'énergie 3 est repeuplé et le milieu devient plus transparent pour la lumière du laser sonde [3].

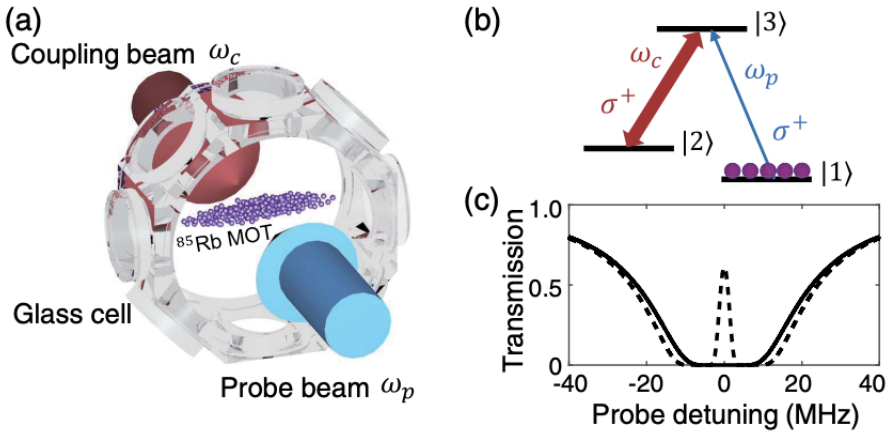

FIGURE 5 – (a), représentation du piège magnétique. (b), transitions énergétiques des atomes ^{85}Rb. (c), transmission vue par le laser sonde avec et sans présence du laser couple.

2.3.3 Effet Kerr et bistabilité optique

Dans certains cristaux, une biréfringence induite est observée lorsque le milieu est soumis à un fort champ électrique : c'est l'effet Kerr (effet non linéaire d'ordre 3). La différence d'indice entre les deux directions de polarisation est alors proportionnelle à l'intensité du champ électrique incident :
$\Delta n = n_{\parallel} - n_{\perp} = \alpha E_0^2$.
Cet effet peut être utilisé pour créer des systèmes bistables. La réponse optique à une entrée optique de ce genre de système ressemble alors très fortement à une sigmoid (voir figure 6).

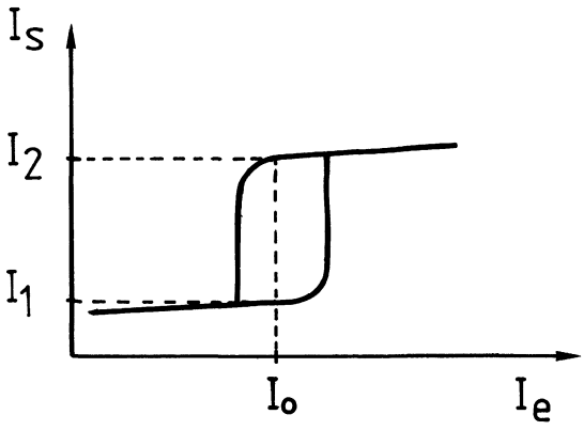

FIGURE 6 – Bistabilité optique : intensité de sortie en fonction de l'intensité d'entrée.

2.4 Le Reservoir Computing

Le Réservoir Computing est une catégorie très particulière des réseaux de neurones. L'idée est de concilier une architecture complexe et proche de celles des réseaux de neurones récurrents avec une phase d'entraînement simple. Pour se faire, les poids des connexions neuronales sont choisis aléatoirement et fixés. Lors de la phase d'entraînement, seuls les poids de la couche de sortie sont entraînés (il s'agit donc d'une simple régression).

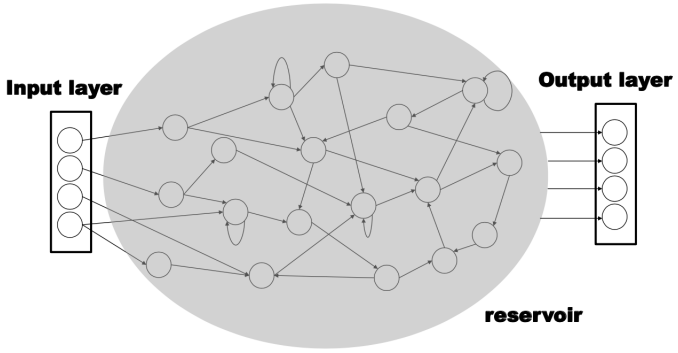

FIGURE 7 – Représentation d'un réseau "Reservoir Computing". Le coeur du système est une boîte noire qui ne sera pas modifiée à l'entraînement. Source : [6].

Le Reservoir Computing permet d'envoyer les données dans un espace de plus grande dimension (grâce à ses propriétés non linéaires). Un vecteur d'entrée de taille p est alors envoyé dans un espace de taille $n \geq p$.

Ceci peut faciliter la séparation des données par des hyper plans (voir figure 8).

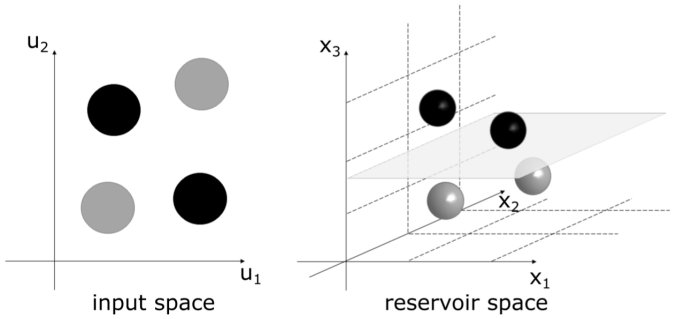

FIGURE 8 – Le "mapping" des données d'entrée vers l'espace Reservoir de plus grande dimension rend possible la séparation par hyper plan entre différents cluster.

Remarque : Le terme "Reservoir" provient du premier système physique utilisé pour créer ce genre de réseaux. Il s'agissait d'un sceau d'eau et de 8 actionneurs mécaniques qui produisaient une surface d'onde complexe (utilisée comme espace de plus grande dimension).

2.4.1 Le Reservoir Computing à retard

Principe

Le Reservoir Computing à retard est un réseau d'un seul noeud physique soumis à sa propre réponse avec un retard τ. Cette architecture simplifie considérablement la difficulté à implémenter physiquement

le réseau. Plutôt que de considérer plusieurs noeuds physiques répartis dans l'espace, on considère un seul noeud physique qui génère des noeuds virtuels répartis uniformément dans le temps le long de la boucle à retard (voir figure 9).

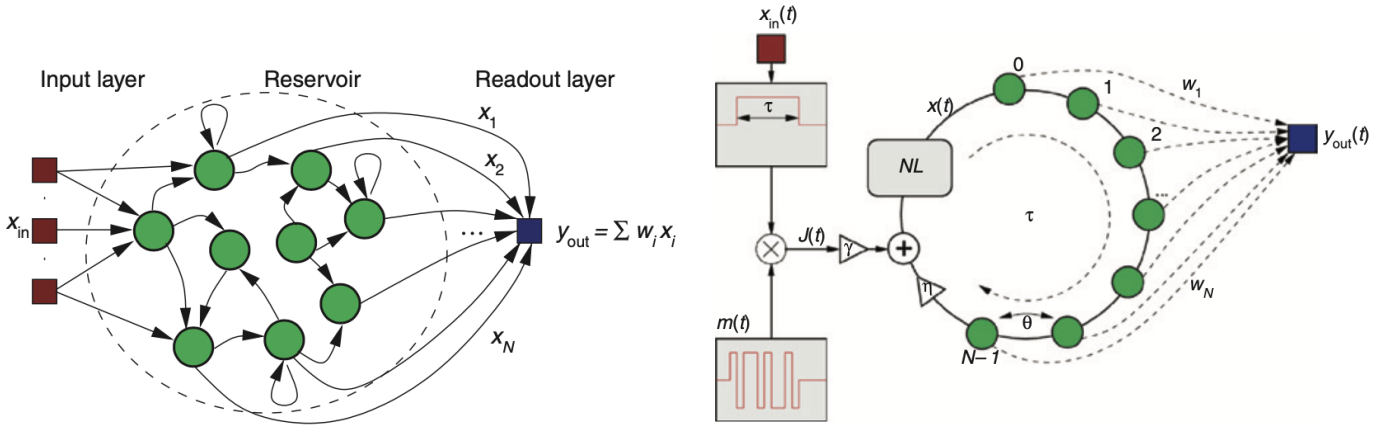

FIGURE 9 – À gauche : un réseau Reservoir classique. À droite : un réseau Reservoir à retard. Source : [7]

La dynamique de ce genre de réseau est donnée par une équation différentielle à retard de la forme :

$$\frac{dx(t)}{dt} = F(t, x(t), x(t-\tau))$$

Le nombre de neurones du réseau est facilement modifiable puisqu'il s'agit seulement d'échantillonner le signal plus ou moins rapidement le long de la boucle à retard. En notant θ le temps le temps inter-neurone, le nombre de neurones du réseau est donné par $N = \tau/\theta$.
L'injection se fait de la manière suivante : une valeur d'entrée u est maintenue pendant une durée τ (pour que chaque neurone de la boucle puissent ressentir les effets de cette entrée). En d'autres termes, une puissance optique proportionnelle à u est injectée sur un intervalle $[t, t+\tau]$.

Procédure de masquage

Un réseau Reservoir Computing à retard nécessite une procédure de pré-traitement des données appelée "masquage". L'idée est de convoluer le signal d'entrée avec un signal masque (voir figure 11), de sorte à ce que le système n'est jamais le temps de retourner à sa valeur stationnaire. En effet, en notant T le temps de réponse du système, si $\theta \geq T$ alors tous les neurones vont répondre similairement à l'entrée. Le rôle du masque est donc de pas laisser le temps au système de se stabiliser.

Généralement, les masques sont à valeurs dans [-1,1] ou dans [0,1]. Le temps entre deux valeurs différentes de masquage est θ (le même que le temps inter-neurone). Chaque plateau de symbole de durée τ du signal d'entrée est multiplié par un même masque de durée τ. Il est important que le masque soit le même pour tous les plateaux de symboles (pour ne pas trop dénaturer le signal d'entrée).

Le masquage permet de renforcer l'interaction entre les neurones. En effet, sans le masque (ou avec un masque mal dimensionné), le système atteint son régime stationnaire et la réponse du système ne dépend plus que de la valeur d'entrée à l'instant considéré. Il n'y a plus d'effet mémoire et l'intéraction entre les neurones virtuels est faible voir nulle. Au contraire, avec un masque bien choisi ($\theta \approx T/5$), le système n'a pas le temps d'atteindre son régime stationnaire. En conséquence, la valeur de sortie du système à un instant donné dépend des valeurs du passé : l'interaction entre les neurones est fortes.

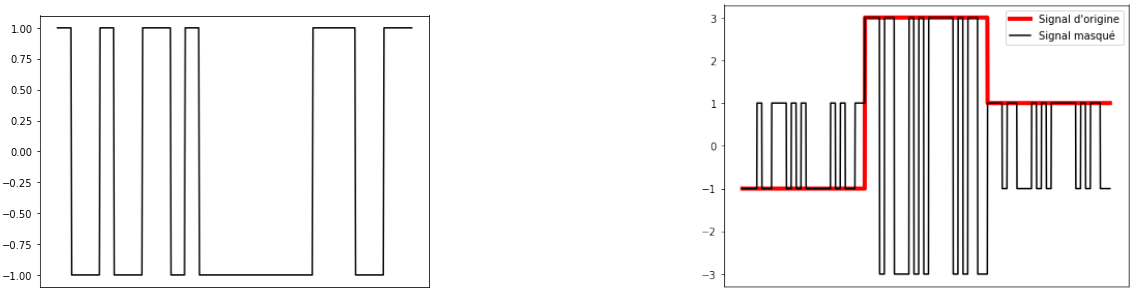

FIGURE 10 – À gauche : un masque à valeurs dans [-1,1]. À droite, la différence entre le signal d'entrée avant et après masquage.

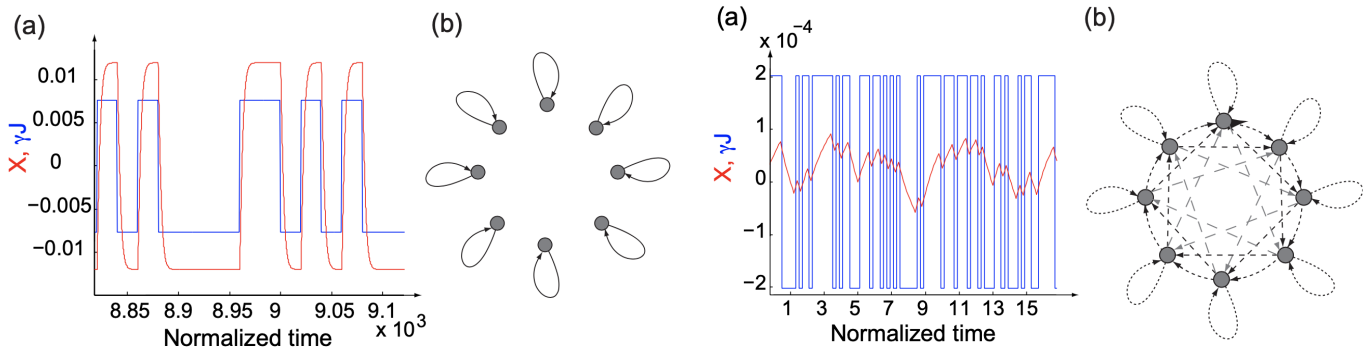

FIGURE 11 – À gauche : un masque mal dimensionné ($T > \theta$). Le système se stabilise **(a)** et l'interaction entre neurones est nulle **(b)**. À droite : un masque bien dimensionné ($\theta \approx T/5$). Le système ne se stabilise pas **(a)** et l'interaction entre neurones est forte **(b)**. Source : [8]

Entraînement

L'entraînement du réseau est supervisé. Pour une tâche donnée, il s'agit de nourrir le réseau avec un jeu de données de la forme [entrées, cibles attendues] et d'optimiser les poids de sortie pour que la réponse du système soit, en moyenne, la plus proche possible de la cible.
Mathématiquement, on associe à chaque entrée $u(k)$ injectée entre $[k(\tau-1), k\tau]$ un vecteur d'état $\mathbf{x}(k)$ qui contient la réponse de tous les neurones.

$$\mathbf{x}(k) = [s(k\tau), s(k\tau-\theta), s(k\tau-2\theta), ..., s(k\tau-(N-1)\theta)]^T$$

Avec s la réponse du système et N le nombre total de neurones virtuels.

À partir de ce vecteur d'état $\mathbf{x}(k)$, on calcule la sortie $\hat{y}(k)$ en faisant le produit scalaire avec les poids de sortie :

$$\hat{y}(k) = \mathbf{W}^{out}\mathbf{x}(k) = \sum_{i=0}^{N-1} w_i s(k\tau - i\theta)$$

En notant $\mathbf{y}$ le vecteur cible et $\mathbf{X}$ la matrice de tous les vecteurs d'états, l'entraînement du réseau consiste à minimiser la quantité $\|\mathbf{y} - \mathbf{W}^{out}\mathbf{X}\|^2$.

Grâce à l'inversion de Moore-Penrose, on possède une expression analytique de cette solution optimale :

$$\mathbf{W}^{out,*} = \mathbf{y}(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}$$

Caractéristiques d'un réseau Réservoir Computing performant

Il existe plusieurs moyens de mesurer les performances d'un bon réservoir. Bien sûr, on peut regarder sa capacité à résoudre une tâche donnée. Mais il existe également des métriques universelles qui permettent d'évaluer les performances d'un réseau indépendamment de la tâche.

Pour faire court, un bon réseau Reservoir doit être :

- consistant : deux entrées proches doivent données des réponses proches (au contraire, deux entrée distantes doivent données des réponses significativement différentes).
- avoir une mémoire éphémère.

D'après [7], pour remplir ses caractéristiques, il faut se placer à un endroit où le système est stationnaire mais proche d'une bifurcation (en jouant sur les paramètres physiques).

2.4.2 Un exemple avec une architecture photonique

L'avantage du Reservoir Computing à retard est qu'il ne nécessite que d'un seul noeud physique soumis à sa propre réponse avec un retard τ.
Un exemple d'architecture photonique : le noeud utilisé est un laser (dit "esclave"). Une fibre optique permet de ré-injectée partiellement la lumière émise par ce laser dans le passé (avec un retard τ qui dépend de la longueur de la fibre). Un atténuateur est placé en fin de boucle afin d'empêcher une dynamique trop chaotique. Les différentes valeurs des neurones virtuels au cours du temps sont mesurées grâce à une photodiode reliée à un oscilloscope. L'injection optique est réalisée grâce à un autre laser (dit "maître) (voir figure 12).

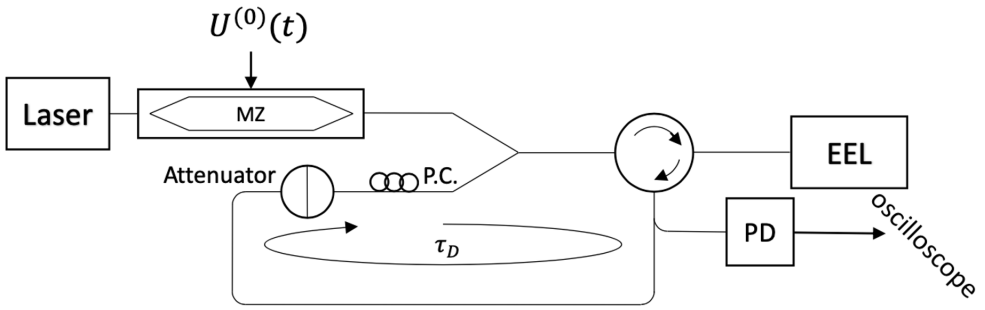

FIGURE 12 – Architecture photonique du réseau Reservoir Computing à retard.

3 Le système de l'article

3.1 Architecture théorique

3.1.1 Structure générale

Le réseau de neurones optique présenté dans l'article est constitué d'une succession de couches reliées les unes à la suite des autres par des guides d'ondes. Chaque couche est constituée d'une unité d'opération linéaire et d'une unité d'opération non linéaire (conformément au fonctionnement d'un neurone, voir 2.1). Des pulses de lumière sont injectées en entrée et se propagent dans le système jusqu'à la couche de sortie.

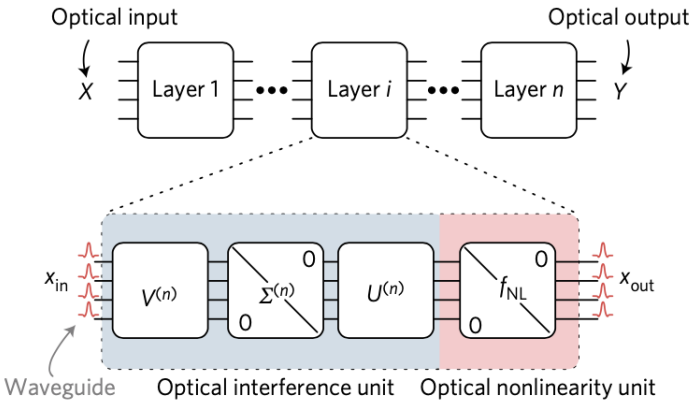

FIGURE 13 – Représentation schématique de l'architecture théorique du réseau de neurone de l'article étudié [1].

3.1.2 Focus sur l'unité d'opération linéaire

L'unité d'opération linéaire d'une couche doit être capable de transformer les entrées x_i en différentes sommes pondérées $b_j + \sum x_i w_{j,i}$ (où j correspond à l'indiçage des neurones de la couche). Ces sommes sont ensuite transmises à l'unité d'opération non linéaire qui les transforment à son tour pour donner les sorties de la couche.
Il est évident que cette transformation linéaire est équivalente à une multiplication de la forme $WX + B$ où X serait le vecteur d'entrée, B le vecteur des biais, W la matrice des poids. La difficulté est donc de réaliser optiquement n'importe quelle multiplication matricielle de la forme WX. Pour faire cela, l'article propose de décomposer W en valeurs singulières : $W = U\Sigma V^\dagger$ où U est une matrice unitaire, Σ diagonale, et $V^\dagger$ la matrice adjointe d'une matrice unitaire V.
Toute la subtilité du système réside dans le fait que la lumière utilisée est cohérente. Il est donc possible de faire interférer différents faisceaux entre eux. La littérature montre alors qu'il est possible d'implémenter optiquement les opérations $U, V^\dagger$ à l'aide de séparateurs et de déphaseurs. L'opération Σ est beaucoup plus simple et est réalisée grâce à des amplificateurs ou des atténuateurs (en fonction de si les valeurs diagonales sont plus ou moins grandes que 1).

3.1.3 Focus sur l'unité d'opération non linéaire

L'article propose de réaliser l'unité d'opération non linéaire à l'aide de couches de graphène. Ce matériau est à absorption saturable (voir 2.3.1). Sa transmittance T_m est fonction de l'intensité incidente I_0 selon

la loi suivante :

$$\sigma\tau_s I_0 = \frac{1}{2}\frac{\ln\left(T_m/T_0\right)}{1-T_m}$$

Avec σ, τ_s, T_0 des paramètres physiques caractéristiques du matériau.
À partir de cette formule, on trace l'intensité de sortie en fonction de l'intensité d'entrée et observée la réponse non linéaire :

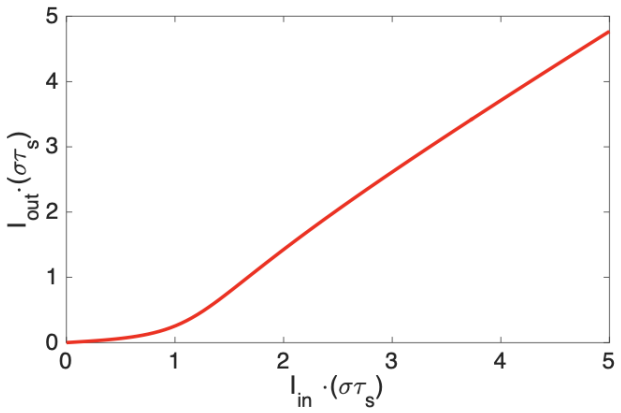

FIGURE 14 – Réponse non linéaire de la couche de graphène

Remarque : L'effet seuil est relativement léger et la réponse du graphène est très linéaire à partir de la valeur critique.

3.2 L'expérience

Pour tester les performances pratiques de leur système théorique, les chercheurs ont construit 2 couches (voir figure 15) selon l'architecture décrite dans la partie 3.1.1. Ils ont ensuite entraîné le système à la reconnaissance de voyelles et mesuré les performances.

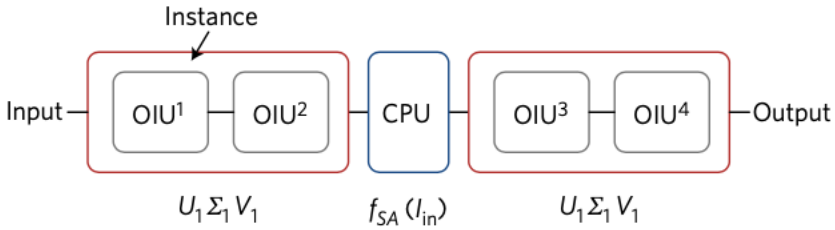

FIGURE 15 – Architecture du réseau de neurones expérimentale, il n'est constitué que de deux couches

3.2.1 Le système hardware

Les chercheurs ont frabiqué un circuit photonique sur silicium en partenariat avec l'entreprise OPSIS. Ce dernier comportait pas moins de 56 interféromètres de Mach-Zender (chacun étant appareillé avec deux déphaseurs thermo-optiques modulabes, judicieusement placés et programmés).
Comme expliqué dans la partie 3.1.3, le but est de réaliser une multiplication par une matrice unitaire $U(N)$ où N est le rang (qui correspond au nombre de neurones par couches). Le subterfuge mathématiques utilisé est le suivant :

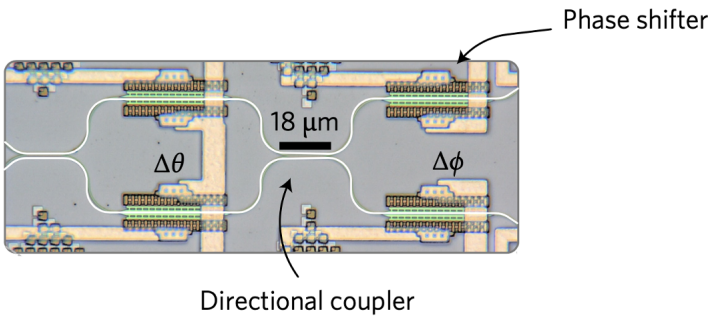

FIGURE 16 – Représentation schématique de la configuration utilisée : un interféromètre Mach-Zender et deux déphaseurs pris en "sandwich".

Toute matrice unitaire $U(N)$ se décompose en un produit de matrices de rotation R, elles-mêmes construites par bloc à partir de matrice de rotation du plan $SU(2)$.

Implémentation d'une matrice SU(2) quelconque :

Un séparateur de faisceaux à transmission/réfléxion variable auquel on rajoute un déphaseur sur une des deux sorties réalise une opération de la forme :

$$\begin{pmatrix} e^{i\phi}\sin\theta & e^{i\phi}\cos\theta \\ \cos\theta & -\sin\theta \end{pmatrix}$$

Avec $\sqrt{R} = \cos\theta$ (resp. $\sqrt{T} = \sin\theta$) le coefficient de réfléxion (resp. transmission) du séparateur, et ϕ le déphasage induit par le déphaseur.

Plutôt que d'utiliser cette configuration, les chercheurs de l'article utilisent une configuration équivalente à l'aide d'un Mach-Zender et de deux déphaseurs :
La matrice unitaire associée est la suivante :

$$\frac{1}{2}\begin{pmatrix} e^{i\phi}(e^{i\theta}-1) & ie^{i\phi}(1+e^{i\theta}) \\ i(e^{i\theta}+1) & 1-e^{i\theta} \end{pmatrix}$$

Bilan :

L'objectif était de réaliser la multiplication de X par n'importe quelle matrice $W = U\Sigma V^\dagger$. Grâce à une cascade de blocs "Mach-Zender + 2 déphaseurs" bien organisée, le circuit (une fois bien paramétré) réalise successivement les multiplications par $V^\dagger$, puis par Σ, puis par U.
La partie "rouge" de la figure 4 est responsable des matrices unitaires U,V et la partie "cyan" de la matrice diagonale Σ.

Quatre passages dans le circuit sont nécessaires pour réaliser les multiplications par :

1. $U^{(1)}\Sigma^{(1)}$
2. $V^{(1)}$
3. $U^{(2)}\Sigma^{(2)}$
4. $V^{(2)}$

Ces notations sont celles de la figure 15.

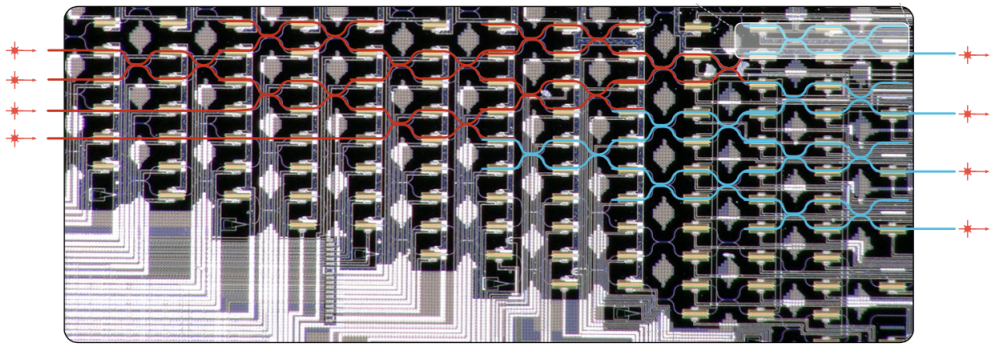

FIGURE 17 – Le circuit photonique sur silicium utilisé pour l'expérience

3.2.2 Les limites du "tout optique"

Avant de discuter des performances de leur système, il est important de comprendre les limites de leur démarche.

1. **Le vecteur d'entrée :**
   Comme expliqué précédemment, les entrées du système sont des pulses optiques (le pulse i aura alors une intensité proportionnelle à x_i). Une étape de pré-traitement est nécessaire pour modéliser la voyelle d'entrée en un vecteur de grande dimension X. Ce pré-traitement est réalisé à l'aide d'un système électronique qui analyse la réponse fréquentielle dans plusieurs bandes de fréquences. Néanmoins, cette étape est énergiquement peu coûteuse devant l'inférence.
2. **L'opération non linéaire :**
   L'opération non linéaire n'est pas proprement réalisée à l'aide d'un absorbant saturable type graphène. Elle est implémentée sur un ordinateur qui simule le comportement de la figure 14.
3. **L'entraînement via à un clone électronique du réseau**
   Pour calculer les poids optimaux qui maximisent les performances du système, le réseau est entraîné sur un ordinateur classique grâce à l'algorithme standard de "backpropagation" sur une réplique du réseau. Cette opération est peu coûteuse en énergie pour 2 couches, mais on ne pourrait pas en dire autant si le réseau en possédait nettement plus.

3.2.3 Les performances

Le Dataset utilisé pour l'expérience était constitué de 4 types de voyelles prononcées par 90 personnes. Soit un total de $4 \times 90 = 360$ données. La moitié à été utilisé pour entraîner le réseau de neurone (sur un ordinateur), et l'autre moitié pour tester les performances du réseau de neurone optique.
Voici les résultats obtenus (mis en parallèle avec les résultats de la classification sur un ordinateur 64-bit classique) :
On remarque que les performances sont presque au niveau de celles obtenues par l'électronique. Les deux réseaux ont les mêmes difficultés à bien classifier les voyelles C et D.

Facteurs limitants :

Plusieurs paramètres peuvent être à l'origine d'incertitudes et d'erreurs de classification. En voici une liste non-exhaustive :

- Diaphonie liée aux échanges thermiques entre les différents déphaseurs thermo-optiques
- Dérive sur les couplages optiques
- Précision limitée sur la phase des déphaseurs (16 bits dans le cas de l'article)
- Bruit de détection sur les photodiodes

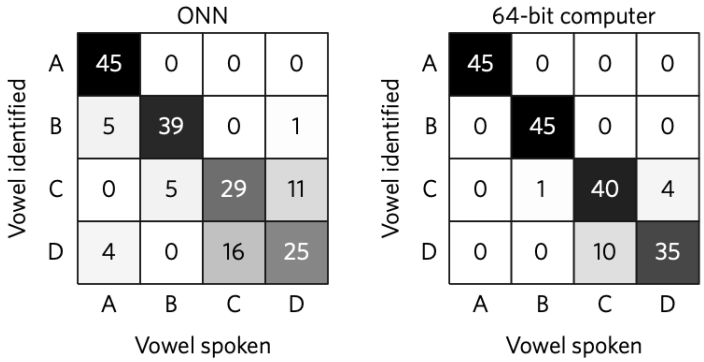

FIGURE 18 – Comparaison des résultats de classification entre le réseau de neurones optique et celui implémenté sur un ordinateur 64-bit.

Considérations énergétiques :

Dans un premier temps, le système a besoin d'une d'alimentation de 10 mW par modulateur. Cet aspect pourrait être amélioré en utilisant des matériaux "à changement de phase non volatils" qui fonctionnent sans apport d'énergie.
Cependant, l'aspect le plus énergivore du système est lié à la puissance minimale nécessaire pour déclencher la fonction non linéaire et à la nécessité d'avoir un bon rapport signal sur bruit (SNR) au niveau des détecteurs. Les calculs de l'article montrent que pour ce réseau de neurones de petite taille, l'efficacité énérgétique est déjà 5 fois meilleure que pour un réseau de neurone équivalent qui serait implémenté sur un processeur graphique classique. Théoriquement, cet avantage énergétique des réseaux de neurones optiques par rapport aux réseaux de neurones électroniques est sensé être de plus en plus significatif au fur et à mesure que la taille du système augmente. En effet, une multiplication matricielle électronique $N \times N$ demande une énergie en $O(N^2)$ (alors qu'elle ne consomme, a priori, pas d'énergie si elle est réalisée optiquement).

Vitesse de calcul :

Pour ce qui est des considérations de vitesse, le réseau optique est principalement limité par les photodiodes (100 GHz). Un réseau de neurone électronique classique ne pouvant excéder le gigahertz, l'optique est 100 fois plus rapide.
De plus, la latence (temps entre l'injection de l'information et l'inférence) est plus faible.

4 Conclusion

Cet essai démontre le potentiel des réseaux de neurones "tout optique". Il en montre l'intérêt en terme de performances de calcul et de consommation d'énergie. Cependant, le système est loin d'être parfait et montre de grosses limites (entraînement sur un clone électronique, difficulté à "scaler", opération non linéaire réalisée électroniquement, etc).
Pour autant, plusieurs perspectives de progression sont mentionnées : entraînement "on-chip" et "forward" (pas de backpropagation), possibilité de rajouter une dimension verticale pour augmenter facilement le nombre de neurones.
Tout ceci présage un avenir optimiste pour les réseaux de neurones "tout optique" qui finiront probablement par dépasser leurs jumeaux électroniques.

Références

[1] Yichen Shen et al. "Deep learning with coherent nanophotonic circuits". In : *Nature Photonics* 11.7 (2017), p. 441-446.

[2] Jia Liu et al. "Research progress in optical neural networks : theory, applications and developments". In : *PhotoniX* 2.1 (2021), p. 1-39.

[3] Ying Zuo et al. "All-optical neural network with nonlinear activation functions". In : *Optica* 6.9 (2019), p. 1132-1137.

[4] Erfan Khoram et al. "Nanophotonic media for artificial neural inference". In : *Photonics Research* 7.8 (2019), p. 823-827.

[5] Yubin Zang et al. "Electro-Optical Neural Networks Based on Time-Stretch Method". In : *IEEE Journal of Selected Topics in Quantum Electronics* 26.1 (2020), p. 1-10.

[6] Jérémy Vatin. "Neuro-inspired photonics for telecommunication applications". In : (2020).

[7] Lennert Appeltant et al. "Information processing using a single dynamical node as complex system". In : *Nature communications* 2.1 (2011), p. 1-6.

[8] Lennert Appeltant. "Reservoir computing based on delay-dynamical systems". In : (2012).