



HAL
open science

Application de méthodes d'apprentissage profond à l'analyse et modélisation de la dynamique non-linéaire dans les fibres optiques

Mehdi Mabed

► **To cite this version:**

Mehdi Mabed. Application de méthodes d'apprentissage profond à l'analyse et modélisation de la dynamique non-linéaire dans les fibres optiques. Optique / photonique. Université Bourgogne Franche-Comté, 2024. Français. NNT : 2024UBFCD021 . tel-04798633

HAL Id: tel-04798633

<https://theses.hal.science/tel-04798633v1>

Submitted on 22 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UBFC

UNIVERSITÉ
BOURGOGNE FRANCHE-COMTÉ



**UNIVERSITÉ DE
FRANCHE-COMTÉ**

THÈSE DE DOCTORAT DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ

PRÉPARÉE A L'UNIVERSITÉ DE FRANCHE-COMTÉ

École doctorale n° 37

Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Optique et photonique

par

M. MABED Mehdi

**Application de méthodes d'apprentissage profond à l'analyse et modélisation
de la dynamique non-linéaire dans les fibres optiques**

Application of deep learning methods to the analysis and modeling of nonlinear dynamics in optical fibers

Thèse présentée et soutenue à Besançon, le 11 Juillet 2024

Composition du Jury :

M. LEVENSON Ariel	Directeur de Recherche CNRS au Centre de Nanosciences et de Nanotechnologies	Rapporteur
M. MUSSOT Arnaud	Professeur à l'université de Lille	Rapporteur
M. KIBLER Bertrand	Directeur de Recherche CNRS au Laboratoire Interdisciplinaire Carnot de Bourgogne	Examineur et président
M. WETZEL Benjamin	Chargé de Recherche CNRS à l'Institut de Recherche XLIM	Examineur
M. DUDLEY John	Professeur à l'université de Franche-Comté	Directeur de thèse

Remerciements

Table des matières

Introduction	1
1 Propagation non-linéaire dans les fibres optiques	5
1.1 L'équation de Schrödinger non-linéaire	5
1.1.1 Équations de Maxwell	6
1.1.2 Équation de Schrödinger non-linéaire généralisée	6
1.1.2.1 Équation de propagation	6
1.1.2.2 Effet Raman	12
1.1.3 Forme normalisée	16
1.1.4 Intégration numérique	17
1.2 Présentation des phénomènes physiques	19
1.2.1 Élargissement induit par la dispersion	19
1.2.2 Auto-modulation de phase	21
1.2.3 Instabilité de modulation	22
1.2.4 Solitons	24
1.2.5 Génération de supercontinuum	28
1.2.6 Vagues scélérates optiques	30
1.3 Problématique	30
2 Apprentissage machine	31
2.1 Introduction	31
2.2 Le perceptron	33
2.3 Réseau de neurones artificiels à propagation avant	36
2.4 Descente stochastique du gradient	41
2.4.1 Implémentation	48
2.4.1.1 Propagation et rétro-propagation	48
2.4.1.2 <i>Mini-batch</i>	49
2.4.1.3 Mise en forme des données	50
2.4.1.4 Toolbox utilisées	51
2.4.2 Variantes de l'algorithme	53
2.4.2.1 Variantes du modèle	53
2.4.2.2 Variantes de l'algorithme d'optimisation	53
3 État de l'art et vérification	55
3.1 Motivation	55
3.1.1 Mesures expérimentales	55
3.1.2 Simulation numérique	56
3.2 Instabilité de modulation et supercontinuum	57

3.2.1	Instabilité de modulation	57
3.2.2	Supercontinuum	61
4	Résultats	67
4.1	Instabilité de modulation : onde continue	67
4.1.1	Propagation	67
4.1.2	Analyse par réseau de neurones	71
4.2	Laser à soliton dissipatif	76
4.2.1	Propagation	76
4.2.2	Analyse par réseau de neurones	80
4.3	Extraction de l'information	86
4.4	Intégration de l'ESNLG par réseau de neurones	90
	Conclusion	97
	Annexes	119
A	Code MATLAB de réseau de neurones	121
B	Bestiaire des architectures de réseaux de neurones	125
C	Publications et conférences	127

Introduction

"Je crois que dans une cinquantaine d'années il sera possible de programmer des ordinateurs, avec une capacité de mémoire d'à peu près 10^9 , pour les faire si bien jouer au jeu de l'imitation qu'un interrogateur moyen n'aura pas plus de 70 pour cent de chances de procéder à l'identification exacte après cinq minutes d'interrogation."

– Alan TURING, *Computing machinery and intelligence*, 1950

L'observation de phénomènes physiques extrêmes a toujours été source d'interrogation pour les scientifiques de toute époque, ces phénomènes étant, par définition, compliqués à observer, donc rares, et donc aux frontières de la physique contemporaine à ces observations. Ainsi, leur étude est souvent féconde en ce qu'elle oblige à enrichir les modèles déjà existants afin de rendre compte de ces phénomènes, voire à réinterpréter les modèles eux-mêmes. Parmi ces événements extrêmes, celui des vagues scélérates en est un particulièrement intéressant, puisque son étude a mené au développement de nouvelles équations dont la résolution a fourni un socle interprétatif riche pour l'analyse de ces événements extrêmes dans de nombreux domaines de la physique.

Les vagues scélérates sont des vagues océaniques très hautes (plusieurs mètres) et soudaines, apparaissant de manière imprévisible. Il est compliqué de savoir à quand remonte les premières observations de telles vagues, ces dernières ayant longtemps été considérées comme des légendes de marins. Il est par exemple rapporté que l'explorateur et scientifique français Jules Dumont d'Urville a observé en 1826 une vague haute de plus de 30 mètres au large de l'Océan Indien, son témoignage ayant été décrédibilisé par le scientifique François Arago. Les premières études scientifiques de vagues au comportement "anormal" eurent lieu une vingtaine d'années plus tard, avec les travaux de John Scott Russell en 1844 [1]. Ce dernier avait alors observé la formation d'une vague générée par l'arrêt brusque d'une barge, et dont la propagation semblait inaltérée sur une très grande distance ; il suivit cette vague à cheval pendant plus d'un kilomètre. Bien que n'ayant pas tout de suite été mises en lien avec les observations de "vagues monstrueuses" décrites par les marins, ces mesures ont été soutenues par certains physiciens de l'époque tels que Joseph Valentin Boussinesq [2], ou encore Lord Rayleigh [3] dans les années 1870. Les études analytiques menées à cette époque sur les équations développées par ces scientifiques ont conduit à la découverte de l'existence de solution d'un genre nouveau : les "ondes solitaires" (*solitary waves*), qui présentaient bien un comportement semblable à celui décrit par Russell. C'est encore une autre vingtaine d'années plus tard (en 1895) que l'équation de Korteweg-de Vries sera développée pour décrire ces phénomènes [4], équation qui sera d'une importance capitale dans les années à venir.

Durant de nombreuses années, l'étude de ces modèles théoriques a été assez laborieuse, les outils mathématiques pour analyser ces équations non-linéaires étant très complexes et limités, et les protocoles expérimentaux étant, par définition de ce qu'est un phénomène extrême, très

compliqués à mettre en place. Cependant, la méthode de la transformée de Fourier non-linéaire (*inverse scattering*) près de soixante-dix ans après la mise en équation (en 1967) a permis de fournir une solution analytique exacte de l'équation de Korteweg-de Vries [5], et ainsi mieux comprendre la nature de ces "ondes solitaires", désormais baptisées **solitons**. De plus, l'essor des méthodes numériques dans la seconde moitié du XX^e siècle a pu mettre en lumière toute la richesse des modèles développés pour décrire cette nouvelle physique des ondes solitoniques [6]. Parmi ces modèles, l'**équation de Schrödinger non-linéaire** (ESNL) est d'une particulière richesse, et a été l'objet de nombreuses études théoriques, expérimentales et numériques depuis son développement dans les années 1960-70 ; cette équation doit son nom à sa similitude avec l'équation de Schrödinger utilisée en mécanique quantique, mais ne décrit cependant pas du tout le même phénomène physique.

En effet, l'ESNL a été développée pour la première fois dans les années 1960 afin de décrire la propagation de la lumière dans des matériaux non-linéaires [7]. Cet intérêt pour la propagation lumineuse dans de tels matériaux est dû à l'essor de la technologie laser dans cette même décennie. Quelques années à peine après le développement du premier laser par Theodore H. Maiman [8], les expérimentations démontrant l'existence de phénomènes non-linéaires optiques virent le jour et permirent alors d'observer des effets tels que la diffusion Raman stimulée en 1962 [9], la dépendance en intensité de l'indice de réfraction en 1964 [10], ou encore le mélange à quatre ondes en 1966 [11]. De plus, le développement des technologies de fibres optiques durant ces mêmes années a poussé la communauté des chercheurs du domaine à étudier ces nouveaux systèmes, jusqu'à la découverte des solitons optiques par Akira Hasegawa en 1973 [12] ; il paraissait désormais clair que l'équation de Schrödinger non-linéaire allait jouer un rôle important dans l'étude des phénomènes optiques extrêmes, tels que l'émergence de solitons. C'est ainsi que la physique des fibres optiques non-linéaires est née et que l'étude des ondes extrêmes dans ces fibres a commencé. Parmi ces phénomènes, la génération de **supercontinuum** qui a été observée pour la première fois en 1970 [13] est particulièrement remarquable et complexe, et bien que son observation ait précédé de quelques années celle du soliton optique, c'est ce dernier qui a permis de comprendre les mécanismes en jeu dans le supercontinuum. Ces observations empiriques ont rapidement poussés les outils de mesures dans leurs limites, si bien qu'à mesure que de nouveaux phénomènes toujours plus complexes étaient observés, de nouveaux protocoles expérimentaux devaient être développés afin de pouvoir saisir au mieux la richesse de ces phénomènes. Cependant, certaines méthodes expérimentales restent, encore aujourd'hui, très délicates à mettre en œuvre, si bien que de nouvelles approches ont été tentées, utilisant les outils numériques actuels d'apprentissage machine.

Ces outils algorithmiques d'intelligence artificielle (IA) existent dans une très grande variété, permettant d'effectuer de nombreuses tâches pour lesquelles les algorithmes traditionnels échouent. Les fondations de l'intelligence artificielle remontent aux travaux d'Alan Turing en 1950 [14], et des progrès assez rapides ont été faits durant les années qui ont suivi. Ces progrès ont souvent été guidés par des analogies avec la biologie et les avancées dans ce domaine, notamment en génétique et en neurosciences. Ainsi, dès la fin des années 1950, le premier **réseau de neurones artificiels** vit le jour ; le "Mark I perceptron" de Rosenblatt [15] était capable de reconnaître un homme d'une femme (avec un taux d'erreur convenable) lorsqu'une photo de visage lui était présentée, après avoir été formé sur un grand ensemble de photos. Cette machine imposante ne constituait que les prémices des outils d'apprentissage machine, puisque la recherche en intelligence artificielle a été extrêmement prolifique durant les années qui ont suivi, avec l'avènement des ordinateurs et l'explosion de leur puissance de calcul, telle que prédite par la loi de Moore [16]. Une consécration symbolique des outils d'apprentissage machine eue lieu lorsque le super-ordinateur Deep Blue développé par IBM a vaincu en 1997 le cham-

pion du monde d'échecs de l'époque Garry Kasparov après avoir "appris" à jouer aux échecs ; dès lors, le développement de tels algorithmes capables de s'adapter aux données qui leur sont fournies fût au centre de l'attention de la communauté scientifique, et leur utilisation dans de nombreux domaines a alors vu le jour.

Les exemples récents sont particulièrement troublants, puisque des algorithmes tels que MuZero ou Agent57 sont capables de fournir de meilleures performances que l'humain moyen à la fois aux échecs, au jeu de Go, au shogi, et à une sélection de 57 jeux d'Atari, le tout sans avoir à être formés sur des données pré-existantes (des parties déjà jouées par exemple), mais en apprenant par eux-mêmes les règles de chacun de ces jeux et en développant des "représentations internes" de ces jeux [17, 18]. Dans un autre domaine, l'algorithme ChatGPT (dans sa quatrième version aujourd'hui) est l'un des exemples les plus marquants de ce dont est capable aujourd'hui une machine au "jeu de l'imitation" introduit par Alan Turing, puisque ce *large language model* (LLM) est désormais capable de tenir des conversations tout à fait crédibles sur un grand nombre de sujets [19].

C'est ainsi que la communauté des physiciens s'est saisie de ce nouvel outil comme allié dans sa quête de compréhension de la nature. En particulier, les chercheurs en optique ont utilisé les algorithmes d'apprentissage machine pour contrôler et étudier la génération et la propagation d'impulsions lumineuses, ainsi que l'interaction de ces dernières avec la matière [20]. L'équipe du département d'optique de l'institut Femto-ST dans laquelle je me suis intégré a déjà entrepris depuis quelques années d'implémenter les outils d'apprentissage machine à la photonique, par exemple en développant des dispositifs physiques capables d'effectuer des calculs de réseaux de neurones en exploitant les fonctions toute optique, ou encore en utilisant des algorithmes génétiques pour contrôler et optimiser les sources laser. Cette thèse s'inscrit dans cette démarche et a donc eu pour but d'utiliser les outils d'intelligence artificielle, et plus particulièrement les réseaux de neurones, afin d'étudier la propagation d'impulsions lumineuses ultra-courtes dans des fibres optiques non-linéaires, ces impulsions exhibant des comportements chaotiques et extrêmes, hors de portée des outils d'analyse standards. En particulier, l'objectif a été de prédire des données temporelles sur la seule base de données spectrales, sans aucune information sur la phase.

Chapitre 1

Propagation non-linéaire dans les fibres optiques

"La science donne à celui qui y consacre son travail et ses recherches beaucoup de satisfaction, à celui qui en apprend les résultats, fort peu."

– Friedrich NIETZSCHE, *Humain, trop humain*, 1878

Comme indiqué dans l'introduction, l'objectif de cette thèse est d'examiner comment les méthodes d'apprentissage machine, et notamment les réseaux de neurones, peuvent analyser les phénomènes d'instabilité qui se développent lors de la propagation non-linéaire d'un champ électro-magnétique dans les fibres optiques. Pour cela, il convient tout d'abord de présenter la physique de ces instabilités, en décrivant le modèle de propagation sous-jacent à Schrödinger non-linéaire (ESNL) ainsi que sa version généralisée. Ce sera alors l'occasion de décrire les phénomènes clés que sont l'instabilité de modulation et la génération de supercontinuum.

1.1 L'équation de Schrödinger non-linéaire

L'objet d'étude de cette thèse est la propagation lumineuse dans les fibres optiques. Ces dernières sont constituées d'un long fil de diélectrique, appelé le cœur, dans lequel se propage la lumière, et qui est recouvert d'une gaine ayant des propriétés optiques différentes. Dans une modélisation géométrique de la propagation lumineuse, leur différence d'indice optique permet à la lumière de se réfléchir à l'interface cœur-gaine et ainsi rester piégée dans le cœur ; la lumière peut alors se propager sur de longues distances de manière guidée et avec une atténuation raisonnable. La description détaillée de la propagation d'une onde lumineuse fait appel aux équations de Maxwell.

1.1.1 Équations de Maxwell

Lorsque l'intensité lumineuse est suffisamment faible, le comportement de l'onde électromagnétique est très bien décrit par des équations ne faisant intervenir que des termes linéaires, c'est-à-dire proportionnels au champ. Cependant, l'utilisation de sources lasers avec d'importantes puissances crêtes ou encore de fibres spéciales à modes confinés (fibre à cristaux photoniques, fibre hautement non-linéaire, ...) impose de tenir compte des effets non-linéaires pour correctement décrire les observations expérimentales.

De manière générale, le champ électromagnétique est décrit de la façon suivante par les équations de Maxwell :

$$\left\{ \begin{array}{l} \nabla \cdot (\varepsilon_0 \vec{E} + \vec{P}) = \rho_f \\ \nabla \cdot (\mu_0 \vec{H} + \vec{M}) = 0 \\ \nabla \times \vec{E} = -\frac{\partial (\mu_0 \vec{H} + \vec{M})}{\partial t} \\ \nabla \times \vec{H} = \vec{J} + \frac{\partial (\varepsilon_0 \vec{E} + \vec{P})}{\partial t} \end{array} \right. \quad \begin{array}{l} (1.1a) \\ (1.1b) \\ (1.1c) \\ (1.1d) \end{array}$$

où \vec{E} est le champ électrique, \vec{H} est le champ magnétique, \vec{P} la polarisation électrique induite par le matériau, \vec{M} la polarisation magnétique induite par le matériau, ρ_f la densité de charges, \vec{J} la densité de courant, et c_0 la vitesse de la lumière dans le vide ; de manière générale, les champs vectoriels présentés dépendent de l'espace et du temps :

$$\vec{V} = \vec{V}(x, y, z, t)$$

1.1.2 Équation de Schrödinger non-linéaire généralisée

1.1.2.1 Équation de propagation

Le système d'équations de Maxwell est alors étudié dans le cadre des fibres optiques non-linéaires afin d'aboutir à l'équation de Schrödinger (et sa version généralisée). Les premières dérivations de cette équation datent des années 1960, mais la démarche présentée ici suit l'approche plus moderne développée durant les années 1980, et plus particulièrement celle de Govind Agrawal [21].

On suppose tout d'abord que le milieu n'est pas magnétique, et exempt de charges libres :

$$\left\{ \begin{array}{l} \rho_f = 0 \\ \vec{M} = \vec{0} \\ \vec{J} = \vec{0} \end{array} \right. \quad \begin{array}{l} (1.2a) \\ (1.2b) \\ (1.2c) \end{array}$$

Ainsi, en prenant le rotationnel de l'équation 1.1c, et en faisant l'approximation temporaire $\nabla \cdot \vec{E} = 0$, on obtient l'équation d'onde

$$\Delta \vec{E} - \frac{1}{c_0^2} \frac{\partial^2 \vec{E}}{\partial t^2} = \mu_0 \frac{\partial^2 \vec{P}}{\partial t^2} \quad (1.3)$$

L'expression rigoureuse de \vec{P} nécessite de faire appel à un développement plus complet que celui présenté ici [22]. Cependant, une première approche phénoménologique permet d'écrire

$$\vec{P} = \varepsilon_0 \left(\vec{P}^{(1)} + \vec{P}^{(2)} + \vec{P}^{(3)} + \dots \right)$$

où les termes $\vec{P}^{(k)}$ s'expriment

$$\vec{P}^{(k)} = \int_{-\infty}^t \dots \int_{-\infty}^t \chi^{(k)}(t - t_1, \dots, t - t_k) : \vec{E}(x, y, z, t_1) \dots \vec{E}(x, y, z, t_k) dt_1 \dots dt_k \quad (1.4)$$

où $\chi^{(k)}$ est la susceptibilité électrique d'ordre k du matériau; cette équation traduit la non-instantanéité de la réponse du matériau. L'approche standard [23] consiste premièrement à supposer que la réponse du matériau est instantanée, ce qui permet d'écrire

$$\vec{P} \simeq \varepsilon_0 \left(\chi^{(1)} \cdot \vec{E} + \chi^{(2)} : \vec{E} \vec{E} + \chi^{(3)} : \vec{E} \vec{E} \vec{E} + \dots \right) \quad (1.5)$$

L'équation de Schrödinger non-linéaire est une équation qui a été développée afin de modéliser le comportement d'ondes (électro-magnétiques, fluides, ...) se déplaçant dans un milieu non-linéaire, présentant les propriétés nécessaires à ce que seuls les effets non-linéaires d'ordre trois soient excités (les effets linéaires sont évidemment pris en compte). Dans le cas des fibres optiques, ceci est assuré par la centrosymétrie de ces dernières qui annule les effets d'ordre pair, et en supposant que les effets d'ordre impair supérieurs à trois sont assez faibles pour être négligés; on obtient alors

$$\vec{P} \simeq \varepsilon_0 \left(\chi^{(1)} \cdot \vec{E} + \chi^{(3)} : \vec{E} \vec{E} \vec{E} \right) \quad (1.6)$$

Bien que les hypothèses sur lesquelles reposent l'ESNL puissent paraître restrictives, elles sont en réalité largement valables dans une grande partie des protocoles expérimentaux modernes. Parmi ces hypothèses, les premières concernant la réponse du matériau ont été énoncées plus haut. On suppose de plus que la partie non-linéaire restante de la polarisation ($\chi^{(3)}$) est suffisamment faible pour être traitée comme une perturbation de la polarisation totale, et également que les effets de polarisation sont négligeables, ce qui permet de traiter le tenseur $\chi^{(3)}$ comme un scalaire, et donc de supposer que les champs \vec{E} et \vec{P} qui sont polarisés selon une certaine direction (\vec{x} par exemple) le resteront tout au long de la propagation dans la fibre. De plus, dans l'hypothèse où l'enveloppe du champ électrique est lentement variable, il est utile de séparer la porteuse oscillant à ω_0 de cette enveloppe, d'où

$$\vec{E}(x, y, z, t) = \frac{E(x, y, z, t)e^{-i\omega_0 t} + \bar{E}(x, y, z, t)e^{i\omega_0 t}}{2} \vec{x} \quad (1.7)$$

où E est l'enveloppe lentement variable et \bar{E} son complexe conjugué. En injectant cette expression de \vec{E} dans l'équation 1.6, on obtient un terme oscillant à une pulsation de $3\omega_0$; on

suppose alors que ce terme est négligeable, les conditions d'accord de phase nécessaires à cette génération de troisième harmonique n'étant généralement pas respectées. On obtient donc

$$\vec{P}(x, y, z, t) = \varepsilon_0 \left(\chi^{(1)} + \frac{3}{4} \chi^{(3)} |E|^2 \right) \vec{E} \quad (1.8)$$

En injectant cette expression de \vec{P} dans l'équation d'onde 1.3, en projetant sur \vec{x} , et en posant

$$\varepsilon(x, y, z, t) = 1 + \chi^{(1)} + \frac{3}{4} \chi^{(3)} |E(x, y, z, t)|^2 \quad (1.9)$$

on obtient

$$\Delta E - \frac{1}{c_0^2} \frac{\partial^2 (\varepsilon E)}{\partial t^2} = 0 \quad (1.10)$$

La dépendance temporelle de ε rend l'équation 1.10 très compliquée à résoudre en général. Or, la nature perturbative de $\chi^{(3)}$ et le fait que E soit lentement variable permet de supposer momentanément ε comme étant indépendant du temps, d'où l'équation

$$\Delta E - \frac{\varepsilon}{c_0^2} \frac{\partial^2 E}{\partial t^2} = 0 \quad (1.11)$$

En prenant la transformée de Fourier de cette équation, l'opérateur $\partial^2/\partial t^2$ devient $-\omega^2$, et la transformée de Fourier \tilde{E} de E satisfait alors l'équation d'Helmholtz

$$\Delta \tilde{E} + \frac{\tilde{\varepsilon} \omega^2}{c_0^2} \tilde{E} = 0 \quad (1.12)$$

où $\tilde{\varepsilon}$ est la transformée de Fourier de ε :

$$\tilde{\varepsilon}(x, y, z, t, \omega) = 1 + \tilde{\chi}^{(1)}(\omega) + \frac{3}{4} \chi^{(3)} |E(x, y, z, t)|^2 \quad (1.13)$$

La dépendance temporelle de $\tilde{\varepsilon}$ peut sembler étrange mais est due au fait que cette dépendance a été momentanément négligée, par hypothèse d'enveloppe lentement variable. Afin de résoudre l'équation 1.12, une hypothèse de découplage entre les effets longitudinaux (le long de l'axe de propagation, \vec{z} dans ce manuscrit) et transverses (perpendiculaire à cet axe de propagation) est faite ; les effets longitudinaux sont donc séparés entre la porteuse oscillant à ω_0 et une enveloppe $A(z, t)$

$$\tilde{E}(x, y, z, \omega) = F(x, y) \tilde{A}(z, \omega) e^{i\beta_0 z} \quad (1.14)$$

où β_0 est le vecteur d'onde à la pulsation ω_0 ($\beta_0 = \beta(\omega_0)$). $F(x, y)$ est le mode transverse de l'onde, obtenu en résolvant la partie spatiale des équations de Maxwell ; le détail de cette résolution ne sera pas traité ici, mais l'expression de F sera importante dans la détermination de l'aire effective du mode, elle-même intervenant dans le coefficient non-linéaire γ .

En injectant l'expression de \tilde{E} dans l'équation d'Helmholtz 1.12, on obtient

$$\left(\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2}\right) \tilde{A} + F \left(\frac{\partial^2 \tilde{A}}{\partial z^2} + 2i\beta_0 \frac{\partial \tilde{A}}{\partial z} - \beta_0^2 \tilde{A}\right) + \tilde{\varepsilon} k_0^2 F \tilde{A} + \tilde{\beta}^2 F \tilde{A} - \tilde{\beta}^2 F \tilde{A} = 0 \quad (1.15)$$

$$\implies \left(\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2} + (\tilde{\varepsilon} k_0^2 - \tilde{\beta}^2) F\right) \tilde{A} + \left(\frac{\partial^2 \tilde{A}}{\partial z^2} + 2i\beta_0 \frac{\partial \tilde{A}}{\partial z} + (\tilde{\beta}^2 - \beta_0^2) \tilde{A}\right) F = 0 \quad (1.16)$$

où k_0 est le vecteur d'onde dans le vide ($k_0 = \omega/c_0$) et $\tilde{\beta}$ le vecteur d'onde dans le matériau. $A(z, t)$ étant considérée comme lentement variable, sa dérivée seconde est négligeable ($\partial \tilde{A}^2 / \partial z^2 \ll \partial \tilde{A} / \partial z$), et donc la distribution modale $F(x, y)$ et l'enveloppe lentement variable $A(z, t)$ doivent satisfaire le système

$$\begin{cases} \frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2} + (\tilde{\varepsilon}(\omega) k_0^2 - \tilde{\beta}(\omega)^2) F = 0 & (1.17a) \end{cases}$$

$$\begin{cases} 2i\beta_0 \frac{\partial \tilde{A}}{\partial z} + (\tilde{\beta}(\omega)^2 - \beta_0^2) \tilde{A} = 0 & (1.17b) \end{cases}$$

De plus, la constante diélectrique ε est reliée à l'indice optique complexe n du matériau par

$$\varepsilon = n^2 = \left(\tilde{n} + i \frac{\tilde{\alpha}}{2k_0}\right)^2 \quad (1.18)$$

Or ε dépend de l'intensité $|E|^2$, donc les quantités \tilde{n} et $\tilde{\alpha}$ également, et il est alors utile de faire intervenir cette dépendance comme suit :

$$\begin{cases} \tilde{n} = n + \bar{n}_2 |E|^2 & (1.19a) \end{cases}$$

$$\begin{cases} \tilde{\alpha} = \alpha + \alpha_2 |E|^2 & (1.19b) \end{cases}$$

où n est l'indice de réfraction linéaire et \bar{n}_2 est une mesure de la non-linéarité de la fibre ; ce dernier est lié au coefficient de réfraction non-linéaire. La dépendance en longueur d'onde de l'indice de réfraction de la silice est visible dans la figure 1.1. En supposant \bar{n}_2 et $\tilde{\alpha}$ suffisamment petits pour pouvoir considérer que n est légèrement perturbé par une petite quantité Δn telle que $(\Delta n)^2 \ll 1$, une approche perturbative est alors possible, et on a

$$\varepsilon = (n + \Delta n)^2 \quad (1.20)$$

$$\simeq n^2 + 2n\Delta n \quad (1.21)$$

avec

$$\Delta n = \bar{n}_2(\omega) |E|^2 + \frac{i\tilde{\alpha}}{2k_0} \quad (1.22)$$

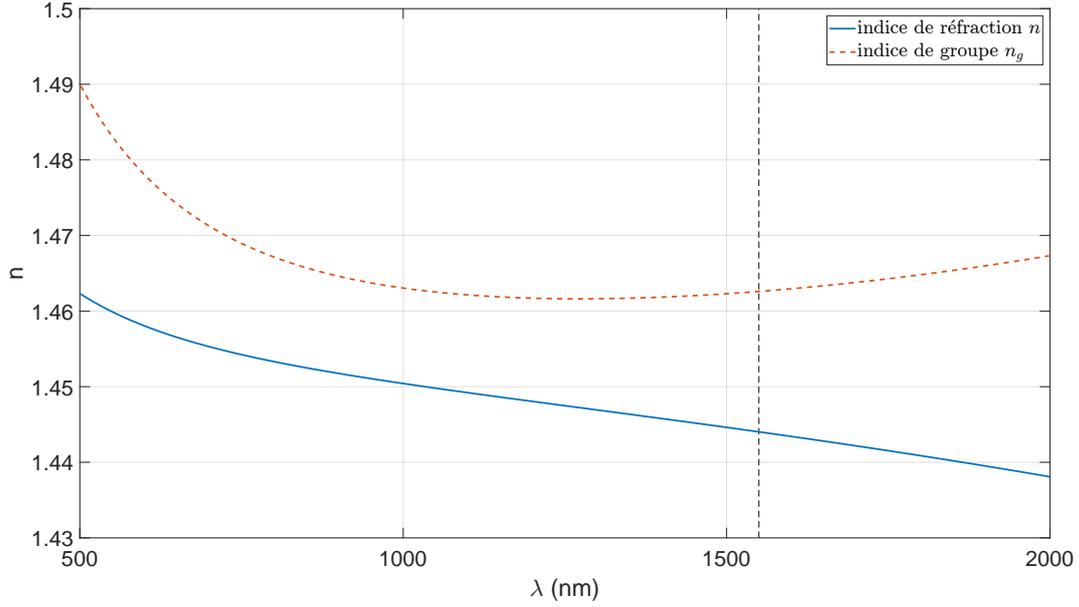


FIGURE 1.1 – Évolution de l'indice de réfraction n et de l'indice de groupe n_g de la silice fondue en fonction de la longueur d'onde λ de l'impulsion en entrée de fibre. La ligne en pointillés noirs correspond à la longueur d'onde de 1550 nm qui est celle qui a été utilisée pour la plupart des études effectuées lors de cette thèse.

L'équation 1.17a est tout d'abord résolue en négligeant la perturbation de la constante diélectrique ($\varepsilon \simeq n^2$), ce qui permet de déterminer le profil transverse de l'onde $F(x, y)$. La perturbation $\Delta n(\omega)$ est ensuite réintroduite, et il est possible de montrer qu'elle a pour seul effet de perturber le vecteur d'onde $\tilde{\beta}(\omega)$ qui devient $\tilde{\beta}(\omega) + \Delta\beta(\omega)$ (la distribution modale $F(x, y)$ n'est pas affectée), avec

$$\Delta\beta(\omega) = \frac{n(\omega)\omega^2 \iint_{-\infty}^{+\infty} \Delta n(\omega) |F(x, y)|^2 dx dy}{c_0^2 \tilde{\beta}(\omega) \iint_{-\infty}^{+\infty} |F(x, y)|^2 dx dy} \quad (1.23)$$

En utilisant le fait que

$$\tilde{\beta}(\omega) \simeq \frac{n(\omega)\omega}{c_0} = n(\omega)k_0 \quad (1.24)$$

$$\Rightarrow \frac{n(\omega)\omega^2}{c_0^2 \tilde{\beta}(\omega)} \simeq k_0 \quad (1.25)$$

et en supposant que \bar{n}_2 et $\tilde{\alpha}$ ont la même valeur dans le cœur et dans la gaine de la fibre (i.e. qu'ils sont indépendants de x et y), on obtient

$$\Delta\beta(\omega) \simeq k_0 \bar{n}_2(\omega) |A(z, t)|^2 \frac{\iint_{-\infty}^{+\infty} |F(x, y)|^4 dx dy}{\iint_{-\infty}^{+\infty} |F(x, y)|^2 dx dy} + \frac{i\tilde{\alpha}(\omega)}{2} \quad (1.26)$$

De plus, $\tilde{\beta}(\omega)$ étant suffisamment proche de β_0 et la perturbation $\Delta\beta(\omega)$ suffisamment petite, on a

$$\left(\tilde{\beta}(\omega) + \Delta\beta(\omega)\right)^2 - \beta_0^2 \simeq 2\beta_0 \left(\tilde{\beta}(\omega) + \Delta\beta(\omega) - \beta_0\right) \quad (1.27)$$

et donc l'équation 1.17b peut être réécrite

$$\frac{\partial \tilde{A}}{\partial z} = i \left(\tilde{\beta}(\omega) + \Delta\beta(\omega) - \beta_0\right) \tilde{A} \quad (1.28)$$

Cette équation montre que chaque composante spectrale se voit déphasée au cours de la propagation dans la fibre, et que ce déphasage est proportionnel à l'amplitude de la composante spectrale considérée.

En considérant que la largeur spectrale $\Delta\omega$ de l'impulsion est petite devant la fréquence ω_0 de la porteuse ($\Delta\omega \ll \omega_0 \implies \omega \simeq \omega_0$), on peut alors approximer $\tilde{\beta}(\omega)$ par sa série de Taylor à un ordre quelconque, et $\Delta\beta(\omega)$ par sa série de Taylor à l'ordre un, au voisinage de ω_0

$$\begin{cases} \tilde{\beta}(\omega) \simeq \beta_0 + \beta_1 (\omega - \omega_0) + \beta_2 \frac{(\omega - \omega_0)^2}{2!} + \dots \\ \Delta\beta(\omega) \simeq \Delta\beta(\omega_0) \end{cases} \quad (1.29)$$

où

$$\beta_m = \left(\frac{\partial^m \beta}{\partial \omega^m} \right)_{\omega=\omega_0} \quad (1.30)$$

En faisant la transformée de Fourier inverse de l'équation 1.28, on obtient

$$\frac{\partial A}{\partial z} \simeq i \left(i\beta_1 \frac{\partial}{\partial t} + i^2 \frac{\beta_2}{2!} \frac{\partial^2}{\partial t^2} + i^3 \frac{\beta_3}{3!} \frac{\partial^3}{\partial t^3} + \dots + \Delta\beta_0 \right) A \quad (1.31)$$

$$\simeq i \left(\sum_{k=1}^{+\infty} \frac{i^k \beta_k}{k!} \frac{\partial^k A}{\partial t^k} + \gamma |A|^2 A \right) - \frac{\alpha}{2} A \quad (1.32)$$

où

$$\gamma(\omega_0) = k_0 \bar{n}_2(\omega_0) \frac{\iint_{-\infty}^{+\infty} |F(x, y)|^4 dx dy}{\iint_{-\infty}^{+\infty} |F(x, y)|^2 dx dy} \quad (1.33)$$

Il est alors utile de normaliser l'amplitude A de manière à ce que son module au carré $|A|^2$ représente la puissance optique. Pour cela, on définit l'amplitude A' telle que

$$|A'|^2 = \frac{1}{2} \varepsilon_0 c_0 n \mathcal{A}_m |A|^2 \quad (1.34)$$

$$= I_0 |A|^2 \quad (1.35)$$

où

$$\mathcal{A}_m = \iint_{-\infty}^{+\infty} |F(x, y)|^2 dx dy \quad (1.36)$$

est l'aire du mode spatial. On montre alors que A' vérifie également l'équation 1.32 mais où le coefficient non-linéaire γ est remplacé par γ'

$$\gamma' = \frac{\gamma}{I_0} \quad (1.37)$$

En définissant l'aire effective du mode \mathcal{A}_{eff} comme étant

$$\mathcal{A}_{\text{eff}} = \frac{\left(\iint_{-\infty}^{+\infty} |F(x, y)|^2 dx dy \right)^2}{\iint_{-\infty}^{+\infty} |F(x, y)|^4 dx dy} = \frac{\mathcal{A}_m^2}{\iint_{-\infty}^{+\infty} |F(x, y)|^4 dx dy} \quad (1.38)$$

on obtient

$$\gamma' = \frac{k_0 n_2}{\mathcal{A}_{\text{eff}}} \quad (1.39)$$

avec

$$n_2 = \frac{2\bar{n}_2}{\varepsilon_0 c_0 n} \quad (1.40)$$

n_2 est communément appelé indice de réfraction non-linéaire. Dès lors, et afin de ne pas surcharger les notations, l'amplitude normalisée A' et le coefficient non-linéaire normalisé γ' seront noté A et γ .

1.1.2.2 Effet Raman

La démarche mise en œuvre jusqu'ici s'est basée sur de nombreuses hypothèses. Bien que celles-ci soient valides dans les régimes où les impulsions considérées sont de l'ordre de quelques pico-secondes, les impulsions plus courtes (de l'ordre de quelques femto-secondes) et/ou plus intenses ne satisfont plus ces hypothèses et sont donc mal décrites par les équations précédentes. Il est néanmoins possible d'aboutir à une équation plus générale qui permet de décrire les impulsions aussi courtes que quelques cycles optiques.

L'effet majeur qui a été négligé jusqu'à maintenant est l'effet Raman. Ce dernier est l'effet induit par les vibrations moléculaires du matériau; la lumière excite les molécules qui, en se désexcitant, émettent une nouvelle onde. L'onde émise n'a alors pas la même fréquence; la diffusion principale ayant lieu dans les fibres optiques étant la diffusion Stokes, où l'onde créée a une plus petite fréquence que l'onde incidente (décalage vers le rouge), l'onde anti-Stokes (décalée vers le bleu) nécessitant que la vibration moléculaire soit parfaitement ajustée en fréquence et quantité de mouvement, ce qui est donc plus rare. De plus, l'émission de cette onde se fait avec un certain retard dû au temps de réponse du matériau [24].

Pour tenir compte de cet effet, il ne faut alors plus supposer que les effets non-linéaires sont instantanés, ce qui se traduit par le fait qu'il ne faille pas utiliser la formule 1.6 mais plutôt 1.4 pour modéliser la polarisation induite \vec{P} . De plus, la prise en compte des effets retardés modifie également l'expression du tenseur $\chi^{(3)}$, si bien que la polarisation non-linéaire induite

$P^{(3)}$ devient [22]

$$P^{(3)} \simeq \frac{3}{4} \chi^{(3)} E(x, y, z, t) \int_{-\infty}^t R(t - t_1) |E(x, y, z, t)|^2 dt_1 \quad (1.41)$$

où $R(t)$ est la réponse non-linéaire totale du matériau. Cette fonction inclue l'effet Kerr et l'effet Raman, et vaut

$$R(t) = (1 - f_R) \delta(t) + f_R h_R(t) \quad (1.42)$$

où $h_R(t)$ modélise la réponse retardée du matériau (effet Raman), et f_R représente la contribution de cet effet. Cette expression fait donc apparaître la réponse instantanée des électrons (effet Kerr) ainsi que la réponse retardée des noyaux, et f_R est ajusté empiriquement pour contrôler la contribution des effets retardés ; en pratique, $f_R = 0,18$.

Le fait de tenir compte de la réponse Raman dans $R(t)$ a pour effet de modifier la valeur de la perturbation de l'indice de réfraction Δn (1.22)

$$\Delta n(\omega) = n_2(\omega) \int_0^{+\infty} R(t') |E(x, y, z, t - t')|^2 dt' + \frac{i\tilde{\alpha}(\omega)}{2k_0} \quad (1.43)$$

ce qui a pour conséquence de modifier l'expression de la perturbation $\Delta\beta$ du mode fondamental. De plus, afin de tenir compte de la dépendance en fréquence de n_2 (à cause de \bar{n}_2) et d' $\tilde{\alpha}$, on utilise leur décomposition en série de Taylor

$$\left\{ \begin{array}{l} \gamma(\omega) \simeq \gamma_0 + \gamma_1 (\omega - \omega_0) + \gamma_2 \frac{(\omega - \omega_0)^2}{2!} + \dots \end{array} \right. \quad (1.44a)$$

$$\left\{ \begin{array}{l} \alpha(\omega) \simeq \alpha_0 + \alpha_1 (\omega - \omega_0) + \alpha_2 \frac{(\omega - \omega_0)^2}{2!} + \dots \end{array} \right. \quad (1.44b)$$

où

$$\left\{ \begin{array}{l} \gamma_m = \left(\frac{\partial^m \gamma}{\partial \omega^m} \right)_{\omega=\omega_0} \end{array} \right. \quad (1.45a)$$

$$\left\{ \begin{array}{l} \alpha_m = \left(\frac{\partial^m \alpha}{\partial \omega^m} \right)_{\omega=\omega_0} \end{array} \right. \quad (1.45b)$$

il est également possible de montrer que

$$\frac{\gamma_1}{\gamma_0} = \frac{1}{\omega_0} + \frac{1}{n_2} \left(\frac{\partial n_2}{\partial \omega} \right)_{\omega=\omega_0} - \frac{1}{\mathcal{A}_{\text{eff}}} \left(\frac{\partial \mathcal{A}_{\text{eff}}}{\partial \omega} \right)_{\omega=\omega_0} \quad (1.46)$$

Les effets induits par la dépendance fréquentielle de l'aire effective \mathcal{A}_{eff} dans la génération de supercontinuum ont été étudiés en détail dans la littérature [25]. Ainsi, lorsqu'on considère des impulsions longues de plusieurs cycles optiques (une largeur supérieure à 100 fs), seul le premier terme de l'équation 1.46 contribue significativement (donc $\gamma_1 \simeq \gamma_0/\omega_0$, $\alpha_{m \geq 1} \simeq 0$, et

$|A(z, t - t')|^2$ est approximé par sa série de Taylor à l'ordre un

$$|A(z, t - t')|^2 \simeq |A(z, t)|^2 - t' \frac{\partial |A(z, t)|^2}{\partial t} \quad (1.47)$$

L'équation 1.32 devient alors, en tronquant γ à l'ordre un et α à l'ordre zéro [26]

$$\frac{\partial A}{\partial z} = i \left(\sum_{k=1}^{+\infty} \frac{i^k \beta_k}{k!} \frac{\partial^k A}{\partial t^k} + \gamma_0 \left(|A|^2 A + \frac{i}{\omega_0} \frac{\partial |A|^2 A}{\partial t} - T_R \frac{\partial |A|^2}{\partial t} A \right) \right) - \frac{\alpha_0}{2} A \quad (1.48)$$

où

$$T_R = \int_0^{+\infty} t' R(t') dt' \quad (1.49)$$

Le terme proportionnel à i/ω_0 résulte de la dépendance fréquentielle de $\Delta\beta$ (1.23), et le terme proportionnel à T_R provient de la réponse Raman retardée. Une bonne approximation de la réponse du matériau $h_R(t)$ est donnée par [27]

$$h_R(t) = \left(\frac{1}{\tau_1^2} + \frac{1}{\tau_2^2} \right) \tau_1 e^{-t/\tau_2} \sin\left(\frac{t}{\tau_1}\right) \quad (1.50)$$

et donc au final

$$R(t) = (1 - f_R) \delta(t) + f_R \left(\frac{1}{\tau_1^2} + \frac{1}{\tau_2^2} \right) \tau_1 e^{-t/\tau_2} \sin\left(\frac{t}{\tau_1}\right) \quad (1.51)$$

Cette équation plus générale modélise la non-conservation de l'énergie induite par la diffusion Raman, et permet, lorsque suffisamment de termes dispersifs sont considérés, de traiter de la génération de supercontinuum. Le terme $R(t)$ (et plus précisément $h_R(t)$) reste néanmoins compliqué à estimer. Il est bon de noter que dans le cas où l'impulsion considérée est plus longue que le temps de réponse Raman, la fonction $h_R(t)$ est quasi-nulle, ainsi que la contribution Raman f_R (et donc $R(t) \simeq \delta(t)$), et le terme γ_1 est négligeable ; ainsi, l'équation 1.48 se réduit à 1.32.

La vitesse de groupe d'une onde étant définie comme

$$\frac{1}{v_g} = \frac{\partial \beta}{\partial \omega} = \beta_1 \quad (1.52)$$

β_2 , qui est la dérivée de β_1 par rapport à ω , est alors la mesure de l'influence de la pulsation de l'onde (et donc sa fréquence) sur sa vitesse de groupe, d'où son nom : le paramètre de dispersion de vitesse de groupe

$$\beta_2 = \frac{\partial \left(\frac{1}{v_g} \right)}{\partial \omega} = -\frac{1}{v_g^2} \frac{\partial v_g}{\partial \omega} \quad (1.53)$$

La dépendance en fréquence (et donc en longueur d'onde) de la vitesse de groupe, et donc de β_2 sont visibles dans la figure 1.2 ; il y figure également l'évolution du paramètre de dispersion de vitesse de groupe D

$$D = \frac{d\beta_1}{d\lambda} = -\frac{2\pi c}{\lambda^2} \beta_2 \quad (1.54)$$

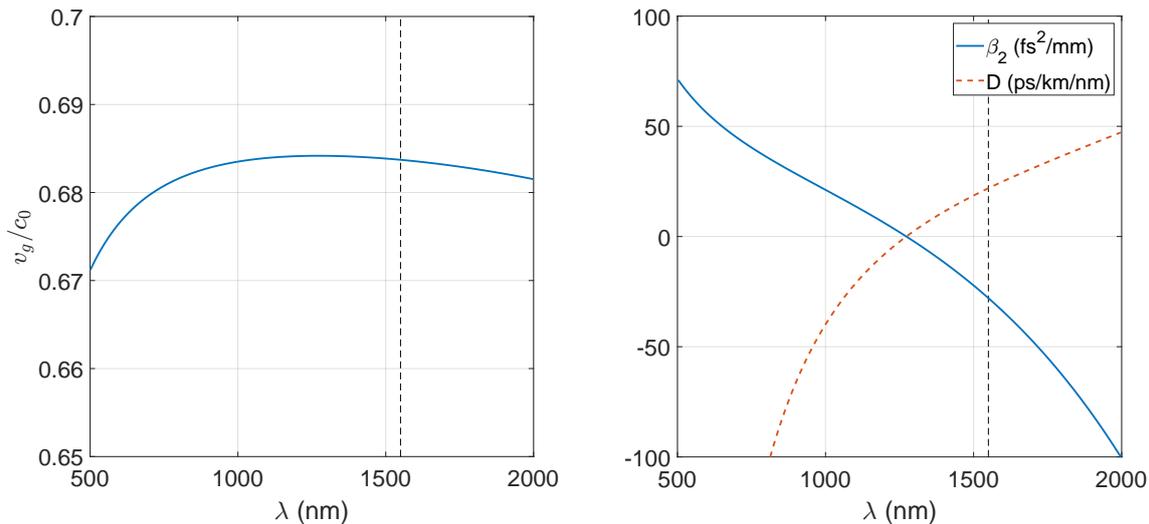


FIGURE 1.2 – Évolution de la vitesse de groupe v_g (figure de gauche), du paramètre de dispersion D et de β_2 (figure de droite) de la silice fondue en fonction de la longueur d'onde λ de l'impulsion en entrée d'une fibre en silice fondue. La ligne en pointillés noirs correspond à la longueur d'onde de 1550 nm qui est celle qui a été utilisée pour la plupart des études effectuées lors de cette thèse.

Malgré son appellation, β_2 n'est pas le seul paramètre à gouverner les effets dispersifs lors de la propagation ; en effet, tous les effets dispersifs d'ordre supérieurs à deux y contribuent, mais ces derniers sont souvent négligés. Ainsi, en présence de dispersion de la vitesse de groupe, chaque composante spectrale évolue à une vitesse différente, ce qui a pour effet d'altérer l'impulsion initiale ; ce point sera discuté plus en détail dans la section concernant la présentation des phénomènes physiques.

En se plaçant dans le référentiel comobile à l'onde électro-magnétique, i.e. celui se déplaçant à la vitesse de groupe v_g de l'onde, quelques simplifications s'opèrent. Pour cela, on effectue le changement de variable

$$T = t - \frac{z}{v_g} = t - \beta_1 z \quad (1.55)$$

et l'équation 1.48 devient alors

$$\frac{\partial A}{\partial z} = i \left(\sum_{k=2}^{+\infty} \frac{i^k \beta_k}{k!} \frac{\partial^k A}{\partial T^k} + \gamma_0 \left(|A|^2 A + \frac{i}{\omega_0} \frac{\partial |A|^2 A}{\partial T} - T_R \frac{\partial |A|^2}{\partial T} A \right) \right) - \frac{\alpha_0}{2} A \quad (1.56)$$

Dans le cas où l'impulsion est plus large que 5 ps, la dépendance fréquentielle de $\Delta\beta$ (1.23) et la réponse Raman retardée (T_R) sont négligeables, et l'équation 1.56 devient

$$\frac{\partial A}{\partial z} = i \left(\sum_{k=2}^{+\infty} \frac{i^k \beta_k}{k!} \frac{\partial^k A}{\partial T^k} + \gamma_0 |A|^2 A \right) - \frac{\alpha_0}{2} A \quad (1.57)$$

qui a la même forme que l'équation 1.32, et lorsque seul l'effet dispersif d'ordre deux est retenu, cette équation se réduit à son tour en l'équation de Schrödinger non-linéaire (ESNL)

$$\frac{\partial A}{\partial z} = i \left(-\frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + \gamma_0 |A|^2 A \right) - \frac{\alpha_0}{2} A \quad (1.58)$$

Cette équation est l'une des plus prolifiques de la physique moderne. En effet, elle trouve de fortes analogies en océanographie [28] dans l'étude des vagues scélérates, en physique des plasmas [29], ou encore dans la modélisation des condensats de Bose-Einstein. Sa force est d'être concise tout en permettant de modéliser des dynamiques complexes, notamment lors de l'utilisation de sa forme généralisée. Dans la suite, afin de ne pas alourdir les notations, α_0 sera remplacé par α , et γ_0 par γ .

1.1.3 Forme normalisée

Il est utile de normaliser l'équation et d'introduire certaines grandeurs caractéristiques du système afin d'analyser l'équation au mieux. On commence en normalisant les durées par la durée de l'impulsion T_0 .

$$\tau = \frac{T}{T_0} \implies \frac{\partial^k}{\partial T^k} = \frac{1}{T_0^k} \frac{\partial^k}{\partial \tau^k} \quad (1.59)$$

Pour une impulsion gaussienne, T_0 correspond à sa largeur à $1/e$, et est proportionnelle à la largeur à mi-hauteur (FWHM) par la relation

$$A(0, T) = \sqrt{P_0} e^{-T^2/(2T_0^2)} \implies \text{FWHM}_{\text{gauss}} = 2\sqrt{\ln(2)}T_0 \quad (1.60)$$

et pour une impulsion en sécante hyperbolique (sech), une telle relation existe également

$$A(0, T) = \sqrt{P_0} \operatorname{sech}\left(\frac{T}{T_0}\right) \implies \text{FWHM}_{\text{sech}} = 2 \ln\left(1 + \sqrt{2}\right) T_0 \quad (1.61)$$

En introduisant l'amplitude normalisée $U(z, \tau)$ telle que

$$A(z, \tau) = \sqrt{P_0} e^{-\alpha z/2} U(z, \tau) \quad (1.62)$$

où P_0 est la puissance crête de l'impulsion initiale ($P_0 = \max(A(0, \tau)^2)$), l'équation 1.57 devient

$$\frac{\partial U}{\partial z} = i \left(\sum_{k=2}^{+\infty} \frac{i^k \beta_k}{k! T_0^k} \frac{\partial^k U}{\partial \tau^k} + \gamma P_0 e^{-\alpha z} |U|^2 U \right) \quad (1.63)$$

En posant

$$\begin{cases} L_D = \frac{T_0^2}{|\beta_2|} \\ L_{NL} = \frac{1}{\gamma P_0} \end{cases} \quad (1.64)$$

l'ESNL (1.58) devient

$$\frac{\partial U}{\partial z} = i \left(\frac{-\operatorname{sgn}(\beta_2)}{2L_D} \frac{\partial^2 U}{\partial \tau^2} + \frac{e^{-\alpha z}}{L_{NL}} |U|^2 U \right) \quad (1.65)$$

Les grandeurs L_D et L_{NL} sont appelées respectivement **longueur de dispersion** et **longueur non-linéaire**, et permettent de quantifier l'ordre de grandeur des distances sur lesquelles les effets dispersifs et/ou non-linéaires sont dominants ou non ; dans le cadre de l'ESNL (1.58),

on peut définir la quantité

$$N^2 = \frac{L_D}{L_{NL}} = \frac{\gamma P_0 T_0^2}{|\beta_2|} \quad (1.66)$$

et il est alors possible de montrer que

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} z \ll L_D \\ z \ll L_{NL} \end{array} \right. \implies \text{les effets dispersifs et non-linéaires sont négligeables} \\ z \approx L_D \ll L_{NL} \implies N^2 \ll 1 : \text{les effets non-linéaires sont négligeables} \\ z \approx L_{NL} \ll L_D \implies N^2 \gg 1 : \text{les effets dispersifs sont négligeables} \\ z \approx L_D \approx L_{NL} \implies N^2 \approx 1 : \text{les effets dispersifs et non-linéaires sont non-négligeables} \end{array} \right.$$

L'interprétation plus précise du paramètre N sera discuté plus bas dans la section dédiée aux solitons.

1.1.4 Intégration numérique

L'ESNL est en général trop compliquée pour être intégrée analytiquement, si bien qu'une approche numérique est nécessaire. Parmi les différentes méthodes algorithmiques, les deux plus utilisées sont la méthode pseudo-spectrale de *split-step Fourier* et celle des différences finies. Dans les deux cas, l'espace et le temps doivent être discrétisés en un nombre de points suffisamment grand pour que les diverses approximations numériques (dérivée, transformée de Fourier, ...) restent valides ; les pas respectifs de ces discrétisations seront notés δ_z et δ_t .

La méthode des différences finies intègre l'équation dans le domaine temporel en approximant les dérivées grâce à des schémas tels que celui de Runge-Kutta. Ce dernier est largement répandu, sa variation la plus utilisée étant la méthode d'ordre quatre (RK4) qui s'est avérée être très appropriée à l'intégration numérique de dynamique supercontinuum [30]. Dans le schéma d'intégration RK4 standard (c'est lui qui est implémenté dans la toolbox MATLAB d'optimisation via la fonction ODE45 par exemple, bien que cette version de l'algorithme soit une méthode à pas adaptatif), un problème général de type

$$\frac{\partial A(z, T)}{\partial z} = f(z, A(z, T)) \quad (1.67)$$

est résolu (approximativement) de manière itérative par

$$\left\{ \begin{array}{l} A_{n+1} = A_n + \frac{\delta_z}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ z_{n+1} = z_n + \delta_z \end{array} \right. \quad (1.68)$$

avec

$$\left\{ \begin{array}{l} k_1 = f(z_n, A_n) \\ k_2 = f\left(z_n + \frac{\delta_z}{2}, A_n + \delta z \frac{k_1}{2}\right) \\ k_3 = f\left(z_n + \frac{\delta_z}{2}, A_n + \delta z \frac{k_2}{2}\right) \\ k_4 = f(z_n + \delta_z, A_n + \delta z k_3) \end{array} \right. \quad (1.69)$$

et où z_0 et $A_0 = A(z_0, T)$ sont les conditions initiales connues.

La méthode *split-step Fourier*, elle, fait l'hypothèse selon laquelle les effets dispersifs et non-linéaires sont découplés localement et agissent de manière isolée; cette méthode est très largement utilisée dans le domaine de l'optique fibrée [31, 32]. En réécrivant l'équation 1.56 sous la forme

$$\frac{\partial A(z, T)}{\partial z} = (\hat{D} + \hat{N}) A(z, T) \quad (1.70)$$

avec

$$\left\{ \begin{array}{l} \hat{D} = i \sum_{k=2}^{+\infty} \frac{i^k \beta_k}{k!} \frac{\partial^k}{\partial T^k} - \frac{\alpha}{2} \\ \hat{N} = i\gamma \left(|A|^2 + \frac{i}{\omega_0 A} \frac{\partial |A|^2 A}{\partial T} - T_R \frac{\partial |A|^2}{\partial T} \right) \end{array} \right. \quad (1.71)$$

l'opérateur de dispersion \hat{D} et l'opérateur non-linéaire \hat{N} apparaissent explicitement. Le schéma d'intégration consiste alors à propager l'équation dans le domaine temporel sur une distance $\delta_z/2$ en négligeant les effets dispersifs ($\hat{D} = 0$), puis de nouveau sur une distance $\delta_z/2$, mais cette fois dans le domaine spectral (d'où l'appellation de méthode pseudo-spectrale) et en négligeant les effets non-linéaires ($\hat{N} = 0$). Le passage dans le domaine spectral se fait grâce à l'algorithme de FFT qui est très rapide. La solution exacte lorsque \hat{N} est supposé indépendant de z est alors

$$\left\{ \begin{array}{l} A_{n+1} = e^{\delta_z(\hat{D} + \hat{N})} A_n \\ z_{n+1} = z_n + \delta_z \end{array} \right. \quad (1.72)$$

Le calcul exact de l'exponentielle étant très compliqué, des approximations doivent être faites, et la solution communément retenue est

$$A_{n+1} \simeq e^{\delta_z \hat{D}} e^{\delta_z \hat{N}} A_n \quad (1.73)$$

Le choix de la méthode d'intégration numérique n'a en pratique que très peu d'impact sur la qualité des résultats obtenus, une étude de convergence étant à chaque fois nécessaire afin de trouver les pas δ_z et δ_t adaptés à la simulation numérique désirée.

1.2 Présentation des phénomènes physiques

L'une des forces de l'équation de Schrödinger non-linéaire est que c'est un modèle relativement simple qui permet pourtant de décrire les phénomènes complexes ayant lieu au cours la propagation d'une onde dans un milieu non-linéaire. En effet, elle contient le fait que la vitesse de groupe dépend de la fréquence (dispersion de vitesse de groupe), qu'une impulsion se déphase au cours de sa propagation proportionnellement à son intensité (auto-modulation de phase), et que la compétition entre ces deux phénomènes donne lieu à des instabilités (instabilité de modulation) et à l'existence d'ondes particulières, périodiques spatialement et/ou temporellement : les solitons.

1.2.1 Élargissement induit par la dispersion

Comme annoncé précédemment, la dispersion de vitesse de groupe est à l'origine d'un élargissement de l'impulsion lumineuse qui se propage dans une fibre dispersive (au moins un des coefficients β_k est non-nul). Comme nous cherchons à étudier les effets de la dispersion sur le comportement de l'impulsion lumineuse, il est préférable de traiter ce problème en étudiant l'ESNL normalisée (1.63), lorsque seuls les effets dispersifs sont pris en compte ($\gamma = 0$) :

$$\frac{\partial U}{\partial z} = i\hat{H}U \quad (1.74)$$

où l'opérateur \hat{H}

$$\hat{H} = \sum_{k=2}^{+\infty} \frac{i^k \beta_k}{k! T_0^k} \frac{\partial^k}{\partial \tau^k} \quad (1.75)$$

est auto-adjoint ($\hat{H}^* = \hat{H}$). La transformée de Fourier \tilde{U} de U satisfait alors l'équation

$$\frac{\partial \tilde{U}}{\partial z} = i \sum_{k=2}^{+\infty} \frac{(-1)^k \beta_k}{k!} \omega^k \tilde{U} \quad (1.76)$$

qui a pour solution

$$\tilde{U}(z, \omega) = \tilde{U}(0, \omega) \exp \left(i \sum_{k=2}^{+\infty} \frac{(-1)^k \beta_k}{k!} \omega^k z \right) \quad (1.77)$$

On voit alors que chaque composante spectrale est déphasée au cours de la propagation par un facteur dépendant de la pulsation ω . Ce déphasage induit par la dispersion laisse le profil d'intensité spectrale $\tilde{U}(0, \omega)$ inchangé, mais peut néanmoins altérer l'impulsion temporelle, et notamment sa largeur.

La résolution générale exacte de cette équation étant trop compliquée à exprimer, cette dernière est caractérisée par ses moments $\langle \tau^n \rangle$ définis comme [33]

$$\langle \tau^n \rangle = \frac{\int_{-\infty}^{+\infty} U^*(z, \tau) \tau^n U(z, \tau) d\tau}{\int_{-\infty}^{+\infty} |U(z, \tau)|^2 d\tau} \quad (1.78)$$

où $\langle \tau^n \rangle$ est le moment d'ordre n de l'impulsion, et U^* est l'adjoint de U . La mesure de l'élargissement de l'impulsion se fait alors grâce à son écart-type

$$\sigma = \sqrt{\langle \tau^2 \rangle - \langle \tau \rangle^2} \quad (1.79)$$

Sachant que l'onde normalisée U est telle que l'intégrale temporelle de son module au carré vaut un, on peut montrer par récurrence que

$$\frac{\partial^k \langle \tau^n \rangle}{\partial z^k} = i^k \int_{-\infty}^{+\infty} U^* [\tau^n, \hat{H}]_k U d\tau = i^k \langle [\tau^n, \hat{H}]_k \rangle \quad (1.80)$$

où le commutateur d'ordre k $[\tau^n, \hat{H}]_k$ est défini par

$$\begin{cases} [\tau^n, \hat{H}]_0 = \tau^n \\ [\tau^n, \hat{H}]_1 = [\tau^n, \hat{H}] = \tau^n \hat{H} - \hat{H} \tau^n \\ [\tau^n, \hat{H}]_{k+1} = [[\tau^n, \hat{H}]_k, \hat{H}] \end{cases} \quad (1.81)$$

et on peut également montrer que

$$\frac{\partial^{n+1} \langle \tau^n \rangle}{\partial z^{n+1}} = 0 \quad (1.82)$$

et donc que $\langle \tau^n \rangle$ que est un polynôme d'ordre n de forme générale

$$\langle \tau^n \rangle = \sum_{k=0}^n c_{n,k} z^k \quad (1.83)$$

où les coefficients $c_{n,k}$ sont déterminés en fonction des conditions initiales $U(0, \tau)$

$$c_{n,k} = \frac{i^k}{k!} \int_{-\infty}^{+\infty} U^*(0, \tau) [\tau^n, \hat{H}]_k U(0, \tau) d\tau \quad (1.84)$$

Ainsi, **l'élargissement (temporel) de l'impulsion** mesuré par son écart-type vaut

$$\sigma^2 = (c_{2,2} - c_{1,1})z^2 + (c_{2,1} - 2c_{1,1}c_{1,0})z + c_{2,0} - c_{1,0}^2 \quad (1.85)$$

et **varie quadratiquement par rapport à la distance de propagation**, quelque soit le nombre de termes considérés pour la dispersion, et indépendamment du profil initial de l'onde. En particulier, dans le cas d'une impulsion gaussienne simple, l'écart-type vaut

$$\sigma^2 = \sigma_0^2 + \left(\frac{\beta_2 z}{2\sigma_0} \right)^2 \quad (1.86)$$

où $\sigma_0 = T_0/\sqrt{2}$ est la largeur initiale de l'onde, et pour une impulsion en sécante hyperbolique

$$\sigma^2 = \sigma_0^2 + \left(\frac{\pi \beta_2 z}{6\sigma_0} \right)^2 \quad (1.87)$$

où $\sigma_0 = \pi T_0/\sqrt{12}$.

1.2.2 Auto-modulation de phase

Le profil de l'impulsion peut être également altéré par les effets non-linéaires de la fibre. Ainsi, afin de mettre cela en évidence, il est bon d'étudier l'ESNL en l'absence d'effets dispersifs ($\beta_k = 0$) :

$$\frac{\partial U}{\partial z} = i\gamma P_0 e^{-\alpha z} |U|^2 U \quad (1.88)$$

En posant

$$U(z, \tau) = |U(z, \tau)| e^{i\varphi(z, \tau)} \quad (1.89)$$

on a

$$\frac{\partial |U|}{\partial z} + i|U| \frac{\partial \varphi}{\partial z} = i\gamma P_0 e^{-\alpha z} |U|^3 \quad (1.90)$$

et en identifiant les parties réelles et imaginaires, on obtient

$$\begin{cases} \frac{\partial |U(z, \tau)|}{\partial z} = 0 \\ \frac{\partial \varphi(z, \tau)}{\partial z} = \gamma P_0 e^{-\alpha z} |U(z, \tau)|^2 \end{cases} \quad (1.91a)$$

$$\quad (1.91b)$$

Ce système d'équations donne donc, en imposant $\varphi(0) = 0$

$$\varphi(z, \tau) = \frac{\gamma P_0 |U(0, \tau)|^2}{\alpha} (1 - e^{-\alpha z}) \quad (1.92)$$

et donc, en posant

$$L_{\text{eff}} = \frac{1 - e^{-\alpha z}}{\alpha} \quad (1.93)$$

la longueur efficace de la fibre, on a

$$U(z, \tau) = U(0, \tau) e^{i|U(0, \tau)|^2 L_{\text{eff}}/L_{NL}} \quad (1.94)$$

On constate que l'onde se déphase au cours de sa propagation, que ce déphasage est proportionnel au coefficient non-linéaire γ , et surtout à l'intensité initiale de l'onde $|U(0, T)|^2$; la forme de l'onde, elle, reste inchangée au cours de la propagation. **La phase de l'onde est modulée proportionnellement à son intensité initiale; c'est ce qu'on appelle l'auto-modulation de phase.**

De manière cruciale, cette phase φ induite par la non-linéarité dépend du temps, au travers de l'impulsion initiale $U(0, \tau)$, et donc la fréquence instantanée de l'onde $\delta\omega$ également :

$$\delta\omega(\tau) = -\frac{\partial \varphi}{\partial \tau} \quad (1.95)$$

$$= -\frac{L_{\text{eff}}}{L_{NL}} \frac{\partial |U(0, \tau)|^2}{\partial \tau} \quad (1.96)$$

Lorsque cette variation de la fréquence instantanée, appelée chirp, est non-nulle, elle a pour effet de générer de nouvelles composantes fréquentielles tout au long de la propagation, et donc d'**élargir le spectre de l'onde.**

1.2.3 Instabilité de modulation

Il n'est pas rare qu'un système non-linéaire comme celui décrit ici présente des instabilités, qui conduisent à la modulation de l'état stationnaire. Ce phénomène est en effet présent dans de nombreux domaines de la physique, tels que la mécanique des fluides, la physique des plasmas, ou encore l'optique non-linéaire, et conduit à des comportements riches dont l'étude est souvent complexe et révèle les limites des outils d'analyse moderne.

Il est alors instructif d'étudier la stabilité de l'état stationnaire lorsque ce dernier est perturbé; cela permet de saisir l'influence du bruit et autres incertitudes (qui sont inhérents à toute mesure physique) sur l'évolution d'une onde. L'enveloppe normalisée $U(z, \tau)$ est dans le régime stationnaire lorsqu'elle ne dépend plus du temps ($\partial^k U / \partial \tau^k = 0$), et l'équation 1.63 devient alors, en négligeant les pertes,

$$\frac{\partial U}{\partial z} = i\gamma P_0 |U|^2 U \quad (1.97)$$

Comme nous l'avons vu dans la section précédente, une solution de cette équation peut se mettre sous la forme

$$U(z) = |U(z)| e^{i\varphi(z)} \quad (1.98)$$

avec

$$\begin{cases} \frac{\partial |U(z)|}{\partial z} = 0 & (1.99a) \\ \frac{\partial \varphi(z)}{\partial z} = \gamma P_0 |U(z)|^2 & (1.99b) \end{cases}$$

où $|U|$ et φ ne dépendent pas du temps puisque l'étude porte sur le régime stationnaire. On a donc $|U(z)| = |U(0)| = 1$, et $\varphi(z) = \gamma P_0 z$.

En perturbant l'amplitude U de cette onde stationnaire par une perturbation (complexe) u telle que $|u|^2 \ll 1$, on a

$$U(z) = e^{i\varphi(z)} \rightarrow (1 + u(z, \tau)) e^{i\varphi(z)} \quad (1.100)$$

et en injectant cette onde perturbée dans l'équation 1.63, on obtient que u vérifie l'équation suivante

$$\frac{\partial u}{\partial z} \simeq i \left(\sum_{k=2}^{+\infty} \frac{i^k \beta_k}{k! T_0^k} \frac{\partial^k u}{\partial \tau^k} + \gamma P_0 (u + \bar{u}) \right) \quad (1.101)$$

où \bar{u} est le complexe conjugué de u . En décomposant la perturbation en partie réelle et imaginaire

$$u(z, \tau) = u^{(r)}(z, \tau) + i u^{(i)}(z, \tau)$$

l'équation 1.101 fournit deux équations (une pour la partie réelle et une pour la partie imaginaire) et le système ainsi obtenu peut être résolu, mettant en évidence l'existence de solutions à croissance exponentielle. L'existence de telles solutions est appelée **instabilité de modulation**; la perturbation (modulation) u devient instable et croît exponentiellement du fait de l'interaction entre les effets dispersifs et non-linéaires. Cette interaction conduit donc à l'émergence de

phénomènes nouveaux qu'on ne peut décrire en négligeant l'un ou l'autre de ces effets ; parmi ces phénomènes, celui des solitons en est un tout particulièrement important (il sera détaillé à la section suivante).

Dans le cas particulier où seuls les effets dispersifs d'ordre 2 sont retenus (donc dans le cas particulier de l'ESNL), la condition d'émergence de ces solutions instables est

$$-\frac{\beta_2}{2}\omega^2 \left(\frac{\beta_2}{2}\omega^2 + 2\gamma P_0 \right) > 0 \implies \omega^2 \in \left[0 ; -\frac{4\gamma P_0}{\beta_2} \right] \quad (1.102)$$

La quantité ω^2 étant positive, si γ (et donc n_2) l'est également, alors cette inégalité ne peut être vérifiée que lorsque $\beta_2 < 0$ (on a donc $\beta_2 = -|\beta_2|$), et on définit alors la pulsation critique d'instabilité

$$\omega_c^2 = \frac{4\gamma P_0}{|\beta_2|} = \frac{4}{|\beta_2| L_{NL}} \quad (1.103)$$

Le gain associé à cette amplification exponentielle est alors

$$g = 2\sqrt{|\beta_2|\gamma P_0\omega^2 - \frac{|\beta_2|^2}{4}\omega^4} \quad (1.104)$$

$$= |\beta_2\omega|\sqrt{\omega_c^2 - \omega^2} \quad (1.105)$$

Ce gain est maximal en

$$\omega_{max} = \pm \frac{\omega_c}{\sqrt{2}} = \sqrt{\frac{2\gamma P_0}{|\beta_2|}} \quad (1.106)$$

et vaut

$$g_{max} = g(\omega_{max}) = 2\gamma P_0 \quad (1.107)$$

et la fréquence d'amplification maximale vaut donc

$$f_{max} = \frac{\omega_{max}}{2\pi} \quad (1.108)$$

Comme $\beta_2 < 0$, d'après l'équation 1.53, on a

$$\frac{\partial v_g}{\partial \omega} > 0 \quad (1.109)$$

Cette dernière condition indique que l'instabilité de modulation ne peut avoir lieu que lorsque les basses fréquences se propagent plus lentement que les hautes fréquences ; on parle alors de **vitesse de groupe anormale**.

Ainsi, lorsque l'impulsion d'entrée a une durée infinie (on parle d'onde continue ou *continuous wave*) et qu'elle est bruitée, certaines fréquences de ce bruit sont amplifiées exponentiellement selon la courbe de gain décrite par g et visible sur la figure 1.3 [34]. Cependant, même lorsque l'impulsion initiale est de durée finie, cette dernière a tendance à s'aplatir en son sommet, créant ainsi une "onde continue locale", et qui est donc soumise au phénomène d'instabilité de modulation ; on parle alors d'**instabilité de modulation impulsienne**. Ces deux régimes d'instabilité seront étudiés dans cette thèse.

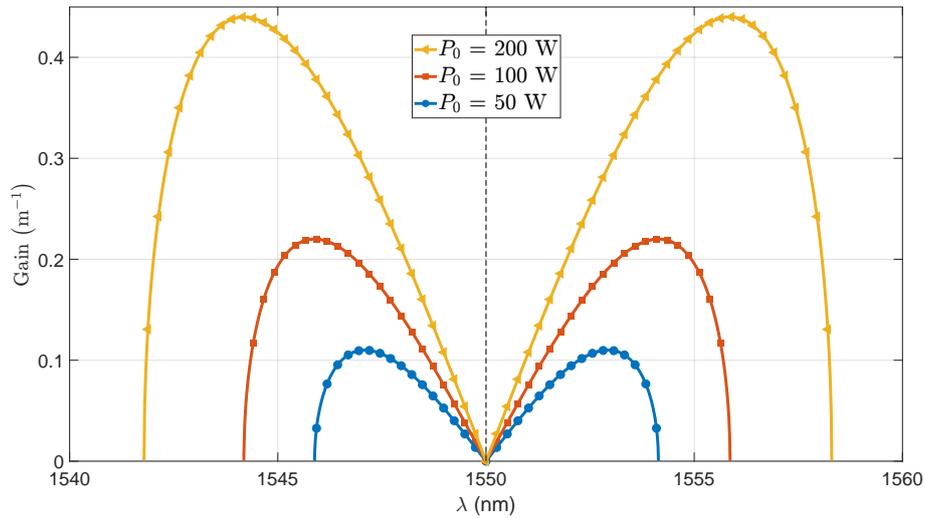


FIGURE 1.3 – Courbes de gain obtenues avec les paramètres de la simulation CW MI pour différentes puissances P_0 : $\beta_2 = -2,1 \times 10^{-26} \text{ s}^2 \text{ m}^{-1}$, $\gamma = 0,0011 \text{ W}^{-1} \text{ m}^{-1}$ et $\lambda_0 = 1550 \text{ nm}$.

1.2.4 Solitons

Parmi les différents phénomènes physiques émergeant lors de la propagation d'une onde dans une fibre, le soliton en est un tout particulièrement fascinant, le terme soliton faisant référence à un type d'onde qui se propage sans déformation, le rendant très intéressant pour le transport d'énergie/information. La première observation de ce phénomène a d'abord été faite dans un contexte océanographique dans la première moitié du XIX^e siècle [1]. Les raisons de ce phénomène ne furent pleinement comprises que près de cent-cinquante ans plus tard [5] grâce à la méthode de la transformée de Fourier non-linéaire (*inverse scattering*).

Il fût alors compris que l'émergence de ces solitons est due à l'interaction entre les effets dispersifs et non-linéaires du matériau. En effet, nous avons vu que la dispersion de vitesse de groupe avait pour effet d'altérer la largeur de l'impulsion (par déphasage de chaque composante spectrale) tout en laissant l'intensité spectrale inchangée (1.85), et que l'auto-modulation de phase avait pour effet d'altérer l'intensité spectrale (par la génération de nouvelles composantes spectrales) tout en laissant la largeur de l'impulsion inchangée (1.94); lorsque l'impulsion est suffisamment étroite et que les deux grandeurs L_D et L_{NL} sont comparables, ces deux phénomènes coexistent et interagissent pour donner naissance aux solitons.

D'un point de vue physique, les fréquences générées par auto-modulation de phase (à cause de la non-linéarité) se déplacent à différentes vitesses (à cause de la dispersion de vitesse de groupe), ce qui a pour effet de modifier la largeur de l'onde. Lorsque la dispersion a pour effet de faire aller les basses fréquences plus vite que les hautes fréquences (dispersion normale), la dispersion et la non-linéarité se renforcent l'une l'autre, ce qui étale d'autant plus l'onde; lorsque la dispersion a pour effet de faire aller les basses fréquences plus lentement que les hautes fréquences (dispersion anormale), **la dispersion et la non-linéarité se compensent l'une l'autre**, ce qui maintient l'onde dans un état stable. Cette dernière se propage alors de manière périodique spatialement et/ou temporellement, avec une période potentiellement infinie, générant dans ce cas un soliton localisé.

D'un point de vue analytique, la forme précise de ces solitons est étudiée dans le cas de l'ESNL (seuls les effets dispersifs d'ordre deux sont retenus), et est obtenue grâce à la méthode de la transformée de Fourier non-linéaire (*inverse scattering transform*). Pour cela, il est plus pratique de réécrire l'équation 1.65 en normalisant la variable spatiale z ; on pose alors

$$\begin{cases} \xi = \frac{z}{L_D} \\ u = NU \end{cases} \quad (1.110a)$$

$$(1.110b)$$

et l'ESNL devient

$$\frac{\partial u}{\partial \xi} = i \left(\frac{-\text{sgn}(\beta_2)}{2} \frac{\partial^2 u}{\partial \tau^2} + |u|^2 u \right) \quad (1.111)$$

qui est la forme analytique adimensionnée standard de l'ESNL utilisée dans de nombreux contextes physiques [35].

La méthode de transformée de Fourier non-linéaire (*inverse scattering*) consiste à trouver un système d'équations linéaires équivalent au problème à résoudre. Ceci est effectué en transformant le problème initial en un problème aux valeurs propres appliqué à de nouveaux opérateurs qui constituent une paire de Lax. Dans le cas de l'équation 1.111 en régime de dispersion anormale ($\text{sgn}(\beta_2) = -1$), le système linéaire aux valeurs propres s'avère être [36]

$$\begin{cases} \frac{\partial v_1}{\partial \tau} = i(uv_2 - \zeta v_1) \\ \frac{\partial v_2}{\partial \tau} = i(\bar{u}v_1 + \zeta v_2) \end{cases} \quad (1.112)$$

où ζ est une valeur propre, complexe en général, avec $\zeta = \zeta^{(r)} + i\zeta^{(i)}$. En appliquant la démarche d'*inverse scattering*, il est possible de reconstruire u à partir de données issues du problème de *scattering* direct [36], ce qui donne

$$u(\xi, \tau) = \sum_{j=1}^{\lfloor N \rfloor} \phi_j \quad (1.113)$$

où ϕ_j se calcule en résolvant un système d'équations non-linéaires. La partie entière de N (1.66) correspond donc à l'ordre du soliton considéré.

Pour $N = 1$, on obtient **le soliton fondamental**

$$u(\xi, \tau) = \zeta^{(i)} \text{sech}(\zeta^{(i)}(\tau - \tau_s + \zeta^{(r)}\xi)) e^{i((\zeta^{(i)2} - \zeta^{(r)2})\xi/2 - \zeta^{(r)}\tau + \varphi_s)} \quad (1.114)$$

où τ_s , $\zeta^{(r)}$, $\zeta^{(i)}$, et φ_s sont respectivement la position, la fréquence, l'amplitude, et la phase à l'origine du soliton. En choisissant correctement l'origine temporelle, la fréquence de la porteuse, et en imposant $u(0, 0) = 1$, on obtient $\tau_s = 0$, $\zeta^{(r)} = 0$, et $\zeta^{(i)} = 1$, d'où

$$u(\xi, \tau) = \text{sech}(\tau) e^{i\xi/2} \quad (1.115)$$

la phase à l'origine n'ayant aucune pertinence ici. Ce soliton fondamental est une onde localisée temporellement, qui évolue de manière invariante tout au long de la fibre (figure 1.4a) et d)). Il est d'une importance capitale pour l'analyse des différents phénomènes qui ont lieu lors de la propagation de la lumière dans une fibre optique. Le soliton d'ordre deux ($N = 2$), lui, a pour

expression analytique [37]

$$u(\xi, \tau) = 4 \frac{(\cosh(3\tau) + 3e^{4i\xi} \cosh(\tau))}{\cosh(4\tau) + 4 \cosh(2\tau) + 3 \cosh(4\xi)} e^{i\xi/2} \quad (1.116)$$

qui est périodique selon ξ , de période $\pi/2$. Ramenée en unité physique, la période est

$$z_0 = \frac{\pi}{2} L_D \quad (1.117)$$

Les solitons d'ordres supérieurs exhibent tous une périodicité spectrale et temporelle de période z_0 (figure 1.4b,c,e et f); ces solitons d'ordre N sont les éléments de base de l'analyse qui va suivre concernant le régime supercontinuum [38].

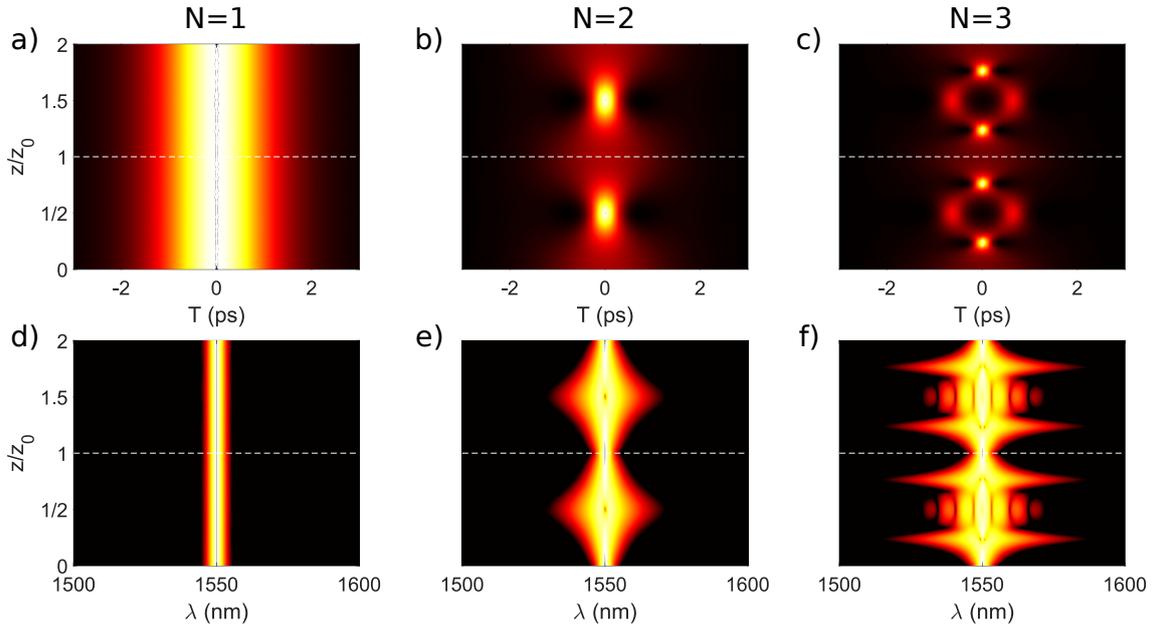


FIGURE 1.4 – Simulation numérique de solitons d'ordre 1, 2 et 3 (respectivement a), b) et c)) dans le domaine temporel (en haut) et spectral (en bas). Les paramètres de la simulation numérique sont les mêmes que pour le CW MI où la puissance P_0 est ajustée selon l'équation 1.66 pour obtenir le soliton d'ordre désiré : $\beta_2 = -2,1 \times 10^{-26} \text{ s}^2 \text{ m}^{-1}$, $\gamma = 0,0011 \text{ W}^{-1} \text{ m}^{-1}$, $\lambda_0 = 1550 \text{ nm}$ et FWHM = 2 ps ($T_0 = 1,13 \text{ ps}$)

Cette périodicité est similaire en nature à la récurrence de Fermi-Pasta-Ulam-Tsingou. L'expérience de FPUT avait pour but d'étudier le comportement d'une corde vibrante discrète présentant une non-linéarité; l'équation d'étude était de la forme [39]

$$\frac{\partial^2 x_i}{\partial t^2} = (x_{i+1} - 2x_i + x_{i-1}) + \alpha ((x_{i+1} - x_i)^2 - (x_i - x_{i-1})^2) \quad (1.118)$$

où x_i représente la position du i^{eme} nœud, et α est une perturbation non-linéaire. La corde est initialement excitée sur son premier mode de vibration, et la première intuition consisterait à prédire qu'un effet de thermalisation devrait répartir équitablement l'énergie dans chaque mode de vibration, aboutissant ainsi à un état de vibration désordonné et chaotique. Or, la simulation numérique effectuée à l'époque (la première de l'Histoire) mit en évidence un comportement quasi-périodique; après l'installation de vibrations erratiques, la corde revient sur son premier mode de vibration et recommence le cycle. C'est plus tard, lorsque le lien avec l'équation de

Korteweg-de Vries a été fait (comme analogue continue de FPU), que la relation entre solitons et périodicité a été pleinement comprise. Les différents solitons émergent au cours du processus se propagent et interagissent entre eux. Lorsque deux de ces solitons se rencontrent, ils semblent se passer au travers sans que leurs formes soient altérées ; leurs amplitudes et vitesses sont modifiées au moment de l'interaction mais sont ensuite rétablies. Il apparaît alors que, malgré l'apparent désordre, ces solitons finissent par tous être quasiment en phase et ainsi quasiment reconstruire l'impulsion initiale [6]. La propagation invariante des solitons ainsi que la manière dont ils interagissent sont donc les éléments clés qui permettent l'émergence de dynamiques périodiques complexes.

De plus, il a été montré que l'équation 1.111 possède une famille de solutions facilement paramétrable, présentant une périodicité soit spatiale soit temporelle ; ce sont les *soliton on finite background* (SFB) [40] :

$$u(\xi, \tau) = e^{i\xi} \left(1 + \frac{2(1 - 2a) \cosh(b\xi) + ib \sinh(b\xi)}{\sqrt{2a} \cos(\omega_c \tau) - \cosh(b\xi)} \right) \quad (1.119)$$

où ω_c est la pulsation critique d'instabilité normalisée (1.102), et avec

$$\begin{cases} b = \sqrt{8a(1 - 2a)} \\ \omega_c = 2\sqrt{1 - 2a} \end{cases} \quad (1.120)$$

Cette famille de solutions contient la sous-famille des *breathers* d'Akhmediev qui exhibe une périodicité purement temporelle ($a < 0,5$) ainsi que celle des *breathers* de Kuznetsov-Ma qui, eux, exhibent une périodicité purement spatiale ($a > 0,5$). Le cas limite où $a = 0,5$ aboutit au soliton de Peregrine qui est un soliton localisé spatialement et temporellement, comme le montre la figure 1.5. Bien que ces expressions analytiques aient été découvertes dans les années 1970, leurs confirmations expérimentales n'ont été réalisées que plusieurs décennies plus tard, au début des années 2010 [41, 40].

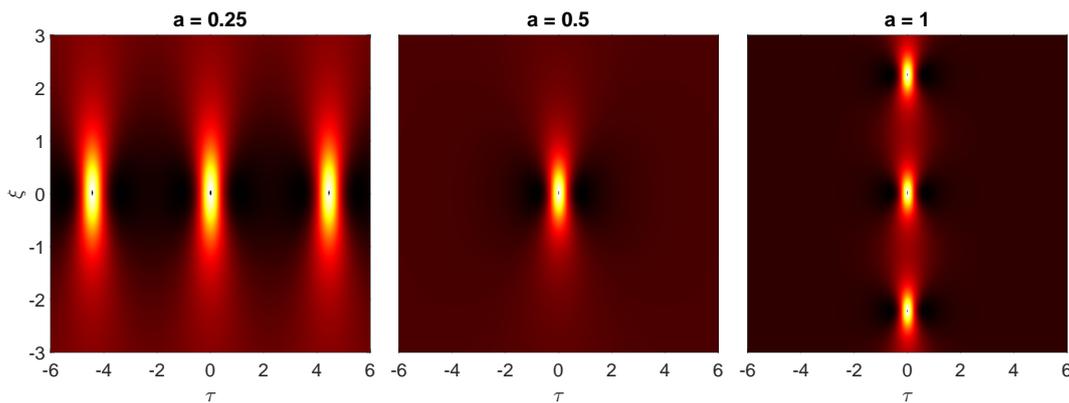


FIGURE 1.5 – Simulation numérique de solitons sur fond continu (SBF) en unité normalisée. La figure de gauche montre un *breather* d'Akhmediev ($a = 0,25$), qui présente une périodicité temporelle, la figure droite montre un *breather* de Kuznetsov-Ma ($a = 1$), qui présente une périodicité spatiale, et la figure centrale montre un soliton de Peregrine ($a = 0,5$), qui est localisé spatialement et temporellement ; ce dernier est donc le cas limite des deux *breathers* précédents.

1.2.5 Génération de supercontinuum

Jusqu'ici, l'étude des phénomènes physiques s'est concentrée sur l'analyse de l'équation 1.32 qui ne tient pas compte de la diffusion Raman, et l'analyse portant sur les solitons ne s'est faite qu'en présence de l'effet dispersif d'ordre deux (seul β_2 non-nul). Or cette diffusion ainsi que les effets dispersifs d'ordres supérieurs jouent un rôle central dans la génération de supercontinuum, c'est-à-dire d'impulsions dont la largeur spectrale finale est grande par rapport à la largeur initiale. Ces impulsions supercontinuum sont le résultat de l'interaction complexe entre plusieurs phénomènes, rendant leur description laborieuse; ces phénomènes clefs que sont la dispersion, l'auto-modulation de phase, et la dynamique solitonique, ont été présentés dans les sections précédentes. De plus, le phénomène de mélange à quatre ondes contribue également à altérer le spectre de l'onde considérée.

Le mélange à quatre ondes est le processus au cours duquel trois ondes de pulsation a priori différentes interagissent afin d'en former une quatrième, dont la pulsation dépend des conditions d'accord de phase des trois ondes incidentes. De manière générale,

$$\omega_4 = \omega_1 \pm \omega_2 \pm \omega_3 \quad (1.121)$$

et une onde avec une telle pulsation ω_4 n'est générée que si

$$k_4 = k_1 \pm k_2 \pm k_3 \quad (1.122)$$

Lorsque l'accord de phase est tel que $\omega_1 + \omega_2 = \omega_3 + \omega_4$, et que $\omega_1 = \omega_2$ (on parle de cas dégénéré), on aboutit au régime de mélange à quatre ondes prépondérant dans les fibres optiques, où une seule impulsion pompe de pulsation ω_1 est capable de générer deux nouvelles impulsions, de pulsation ω_3 et ω_4 ; comme $2\omega_1 = \omega_3 + \omega_4$, les pulsations des impulsions créées sont forcément réparties de manière symétrique par rapport à ω_1 ($|\omega_1 - \omega_3| = |\omega_1 - \omega_4|$). Par analogie avec l'effet Raman, l'impulsion de pulsation inférieure à ω_1 est appelée impulsion Stokes, et l'autre est appelée anti-Stokes. Ce phénomène a donc pour effet de générer des fréquences de part et d'autre de celles déjà existantes; dans ce cas dégénéré, le processus de mélange à quatre ondes n'est en réalité que le pendant fréquentiel de l'instabilité de modulation.

Ainsi, une impulsion de très courte durée, i.e. dont la FWHM est de l'ordre de quelques centaines de femto-secondes au plus, sera soumise au phénomène d'auto-modulation de phase, qui aura pour effet d'élargir son spectre, bien que cet élargissement soit relativement faible. Le mélange à quatre ondes générera alors une dynamique solitonique avec des solitons d'ordres relativement élevés (jusqu'à l'ordre dix par exemple). Ces solitons subiront des effets de compression temporelle (et donc d'élargissement spectral), et les effets non-linéaires d'ordres supérieurs finiront par fissionner ces solitons d'ordres élevés en plusieurs solitons fondamentaux.

En effet, il peut être montré que la solution de l'ESNL normalisée (1.111) obtenue par la méthode d'*inverse scattering* (décrite par l'équation 1.113), est une superposition de $[N]$ solitons fondamentaux dont les durées et puissances sont données par

$$\left\{ \begin{array}{l} T_k = \frac{T_0}{2(N-k) + 1} \\ P_k = P_0 \left(\frac{2(N-k) + 1}{N} \right)^2 \end{array} \right. \quad (1.123a)$$

$$\left. \right\} \quad (1.123b)$$

où k varie de 1 à $[N]$. Tant que ces solitons se déplacent à la même vitesse, l'onde formée par leur superposition reste intacte, mais lorsque leurs vitesses respectives se mettent à diffé-

rer (à cause des effets dispersifs), les solitons fondamentaux individuels se séparent ; c'est le phénomène de **fission solitonique**.

De plus, le soliton spectral le plus intense se voit décalé vers le rouge (vers les grandes longueurs d'ondes) par effet de *self-frequency shift* ("auto-décalage spectral"), comme le montre la figure 1.6b). On y voit clairement le soliton spectral le plus intense se détacher du groupe (aux alentours de 0,1 m) et se décaler vers les plus hautes longueurs d'ondes. Temporellement, la figure 1.6a) montre bel et bien plusieurs solitons émerger de l'impulsion initiale par fission solitonique, dont un qui se décale fortement vers les temps positifs ; sa trajectoire est quasiment parabolique. On voit également apparaître une onde dispersive dans la zone de dispersion normale ($\beta_2 > 0$), i.e. pour les longueurs d'ondes inférieures à la longueur d'onde de dispersion nulle, environ égale à 1270 nm pour la silice fondue (voir figure 1.2).

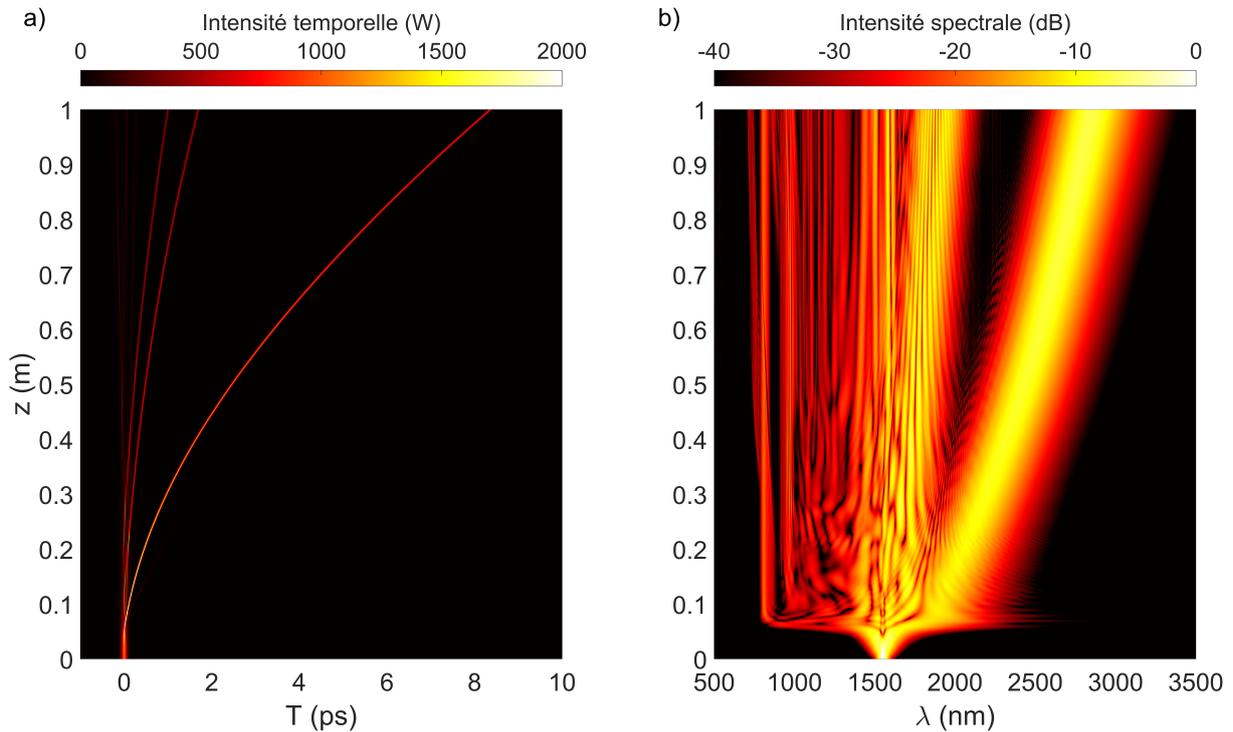


FIGURE 1.6 – Simulation numérique d'un supercontinuum. $\beta_2 = -5.9 \times 10^{-27} \text{ s}^2\text{m}^{-1}$, $\beta_3 = 4.21 \times 10^{-41} \text{ s}^3\text{m}^{-1}$, $\beta_4 = -1.25 \times 10^{-55} \text{ s}^4\text{m}^{-1}$, $\beta_5 = 2.45 \times 10^{-70} \text{ s}^5\text{m}^{-1}$, $\gamma = 0.1 \text{ W}^{-1}\text{m}^{-1}$, $\lambda_0 = 1550 \text{ nm}$, $P_0 = 1 \text{ kW}$ et FWHM = 100 fs ($T_0 = 57 \text{ fs}$). Avec ces paramètres, $[N]=7$.

Le combinaison du phénomène de *self-frequency shift* et de la génération d'onde dispersive aboutit à un spectre dont la largeur peut s'étendre sur plusieurs octaves, et qui trouve des applications diverses en spectroscopie, en métrologie ou encore en imagerie (tomographie).

1.2.6 Vagues scélérates optiques

Lorsqu'une onde se propage dans une fibre, de nombreux paramètres restent non-maîtrisés (bruit dans les conditions initiales, impuretés de la fibre, ...), et ainsi, l'évolution de plusieurs ondes dans la même fibre et dans les "mêmes" conditions présente des fluctuations. En effet, il est possible d'observer l'émergence d'impulsions de très grande amplitude, appelées vagues scélérates par analogie avec l'océanographie [42], qui sont des événements extrêmes, et donc rares. Ainsi, la caractérisation de telles vagues n'est possible qu'en réalisant un grand nombre de mesures dans les "mêmes" conditions, et en mesurant le pic d'intensité à chaque réalisation ; on observe alors une distribution de ces pics d'intensité qui a une longue traîne en forme de "L". La mesure de ces événements extrêmes requiert l'utilisation d'outils de mesure avancés tels que la transformée de Fourier dispersive (DFT) pour pouvoir caractériser en temps réel les spectres d'où émergent ces vagues scélérates, permettant ainsi de mesurer les fluctuations de pic d'intensité spectrale d'une impulsion à l'autre [28, 43].

1.3 Problématique

Depuis quelques dizaines d'années, de nouvelles techniques de mesure en temps réel ont permis de mieux cerner les phénomènes physiques en jeu dans les fibres optiques lors d'une propagation non-linéaire. Parmi ces techniques, la lentille temporelle et la transformée de Fourier dispersive (DFT) ont contribué à révéler/confirmer les détails de la dynamique de la lumière lors de la formation d'instabilité de modulation ou encore de supercontinuum, ce qui a permis d'augmenter considérablement le crédit accordé aux modèles numériques développés jusqu'ici sans réelles confirmations expérimentales.

Cependant, ces techniques expérimentales restent limitées : la lentille temporelle est très compliquée à mettre en place et n'est réalisable qu'à certaines longueurs d'ondes, et la transformée de Fourier dispersive ne permet pas d'obtenir des mesures avec des plages dynamiques supérieures à 25 dB et ne fournit que l'intensité spectrale (sans la phase) ce qui ne permet pas (a priori) de reconstruire l'information désirée sur les solitons. Seules les mesures en spectroscopie restent simples à réaliser tout en ayant une grande plage dynamique (de l'ordre de 60 dB) nécessaire à l'observation et donc l'analyse des instabilités se produisant dans les fibres optiques.

Ainsi, il est pertinent de se demander s'il est possible de reconstruire une partie de l'information temporelle ne connaissant que l'intensité spectrale, qui est la seule propriété simple à mesurer. Les méthodes classiques ayant échoué à cela, nous nous sommes naturellement tournés vers de nouvelles méthodes prometteuses et ayant déjà fait leurs preuves dans de nombreux domaines : les méthodes d'**apprentissage machine**.

Chapitre 2

Apprentissage machine

"Il n'est rien qui se produise dans l'univers sans qu'il y ait un rapport avec le maximum ou le minimum."

– Leonhard EULER, *Methodus Inveniendi Lineas Curvas Maximi Minimive proprietate gaudentes*, 1744

Comme il a été dit en conclusion du chapitre précédent, les outils d'apprentissage machine ont récemment bouleversé le champ de vision de la science moderne en permettant d'accomplir certaines tâches qui semblaient jusqu'alors inextricables, comme la classification, la reconnaissance et/ou la reconstruction d'image, de texte ou de son, le contrôle et/ou l'optimisation de systèmes physiques paramétrés, l'analyse de larges ensembles de données, ... En effet, ces algorithmes ne sont pas basés sur l'exécution de procédures explicites, prédéterminées, et adaptées au problème étudié, mais sur l'exécution d'une unique procédure générale et paramétrée, et dont les paramètres s'adaptent au problème étudié. L'architecture la plus répandue étant celle des réseaux de neurones, c'est celle-là qui a été utilisée au cours de cette thèse, et qui est présentée en détail dans ce chapitre.

2.1 Introduction

Les outils d'apprentissage machine recouvrent une large gamme d'algorithmes ayant pour fonction d'exécuter des tâches complexes que les méthodes traditionnelles (analytiques, analogiques ou numériques) ne parviennent pas à traiter (reconnaissance d'images, classification, ...). Ces algorithmes d'apprentissage reposent tous sur un même principe fondamental : il s'agit de choisir un modèle numérique paramétré et d'optimiser ses paramètres internes (le modèle et ses paramètres étant de différente nature selon l'algorithme utilisé) sur la base de données fournies (expérimentales ou générées numériquement), **afin que le modèle rende ces données les moins surprenantes possibles** (la notion de surprise étant à définir pour chaque modèle, mais souvent réduite à une notion de distance), l'objectif étant par la suite de se servir de ce modèle (après optimisation) sur de nouvelles données auxquelles il n'a jamais été confronté.

Par exemple, pour classer des individus possédant deux caractéristiques, et donc pouvant être représentés dans un plan, en n groupes, un algorithme de classification pourrait modéliser les groupes par des gaussiennes bidimensionnelles, dont les paramètres seraient la position des centres (donc leur moyenne) et leurs écarts-types. L'algorithme optimiserait alors ces paramètres pour faire en sorte que les individus proches entre eux soient dans le même groupe et que les individus éloignés les uns des autres soient dans des groupes différents. Il est possible

de modéliser les groupes par d'autres fonctions que des gaussiennes, et ce modèle aura alors d'autres paramètres à optimiser. L'algorithme créera donc des frontières de décision qui permettront de classer adéquatement de nouveaux individus qui n'ont pas servi lors de la phase d'optimisation; l'algorithme décrit s'apparente à un modèle de mélange gaussien et est schématisé dans la figure 2.1.

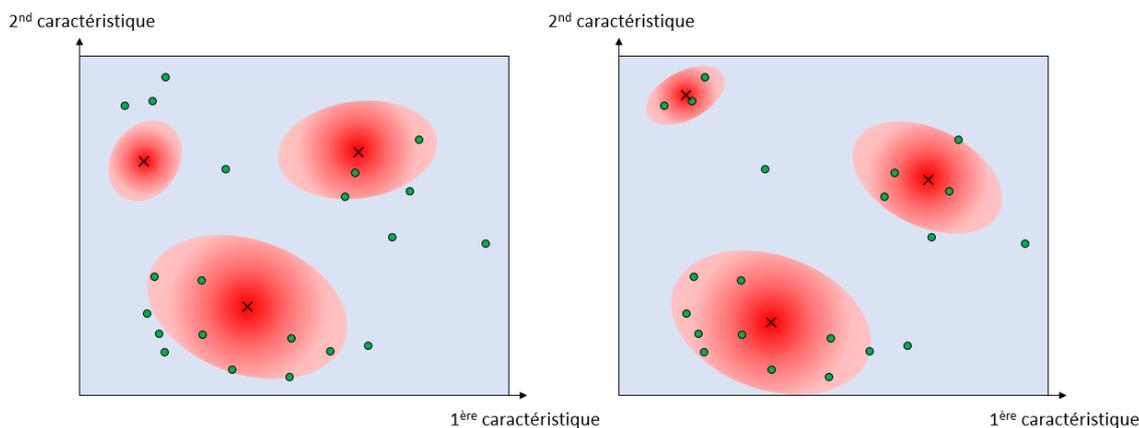


FIGURE 2.1 – Schéma de mixture gaussienne : dans l'espace des caractéristiques des individus, des groupes d'individus semblent naturellement exister, et on cherche à trouver les paramètres de trois gaussiennes qui pourraient classifier ces groupes. La figure de droite rend les données moins surprenantes que la figure de gauche.

Ces approches d'apprentissage machine diffèrent des méthodes traditionnelles de manière cruciale : l'objectif de l'algorithme n'est plus simplement d'exécuter les règles décrites par le modèle mais d'ajuster le modèle lui-même (ou du moins ses paramètres) en fonction des données. L'implication directe de ce changement de conception est que le statut des données change; au lieu d'être une conséquence directe du modèle choisi, elles deviennent une contrainte/un objectif auquel ce modèle doit se plier, et doivent donc être en **quantité** suffisante pour contraindre le modèle comme désiré. Or les données utilisées pour l'ajustement du modèle ne représentant qu'un échantillon infime de l'ensemble des données possibles, et donc un traitement statistique/probabiliste est indispensable afin de gérer au mieux le biais inhérent à cet échantillonnage; ainsi, a priori, plus le nombre de données d'entraînement fourni est grand, plus le modèle sera pertinent.

Or, la **qualité** des données fournies à un algorithme d'apprentissage machine lors de sa formation est également l'une des notions clef de cet apprentissage, puisque des données biaisées dans un sens contraindront le modèle de manière biaisé dans ce même sens, et altéreront alors ses capacités à fournir des résultats pertinents une fois confronté à de nouvelles données; les données d'entraînement doivent donc être représentatives de l'espace des données que l'on cherche à modéliser, alors que ce dernier nous est a priori inconnu. Ce concept de qualité des données d'entraînement est encore à l'heure actuelle source de grandes discussions au sein de la communauté des chercheurs en intelligence artificielle, la mesure, et même la définition, de la qualité d'une donnée étant très complexe.

Dans le contexte des algorithmes de génération de texte par exemple (tels que ChatGPT), qui sont fondamentalement basés sur les mêmes principes de réseau de neurones qui sera présenté dans la section suivante, la taille gigantesque des modèles à optimiser (de l'ordre de 10^{12} paramètres pour les modèles les plus imposants [44]) et l'abondance de données non-contrôlées (issues de forum par exemple) rend très compliquée la tâche de fournir des données

d'entraînement pertinentes ; le modèle se comporte alors comme un "perroquet" qui ne fait que répéter de manière plus ou moins déformée/sophistiquée ce qu'il a "entendu" [45]. Ce constat s'étend bien sûr à tous les modèles d'apprentissage machine, ce principe d'équilibre entre la complexité du modèle (et donc sa fidélité aux données d'entraînement) et sa capacité à généraliser à de nouvelles données (et donc sa pertinence) pouvant être synthétisé dans ce qui est communément appelé le "**compromis biais/variance**" [46] ; un modèle très complexe sera peu biaisé par rapport à ses données d'entraînement (il s'adaptera très bien à elles), mais produira des résultats très variables une fois confronté à de nouvelles données (il sera très sensible aux "aspérités"/bruit des données d'entrée), et inversement, un modèle avec moins de paramètres (donc moins "souple") sera plus robuste aux variations statistiques intrinsèques des données d'entraînement, mais pourra être insensible à certaines caractéristiques fondamentales des données.

Malgré toutes ces considérations, les outils d'apprentissage machine se sont révélés être très performants dans de nombreux domaines, y compris celui des sciences où leur utilisation est de plus en plus présente. En optique et photonique [20], les premières applications de ces méthodes algorithmiques ont surtout pris la forme d'algorithmes génétiques pour la reconnaissance des formes [47], la reconstruction d'images [48], les corrections d'aberrations [49] ou la conception de composants optiques [50]. Des travaux plus récents se sont concentrés sur l'analyse de grands ensembles de données [51] et sur des problèmes inverses où les capacités de l'apprentissage machine à classer les données, à identifier les structures sous-jacentes et à traiter un grand nombre de degrés de liberté ont permis d'obtenir de nombreux résultats, par exemple dans les domaines de la microscopie à super résolution [52], de l'optique quantique [53, 54] ou encore de la communication optique [55].

2.2 Le perceptron

Parmi les différents algorithmes d'apprentissage machine, un d'entre eux s'est particulièrement démarqué par ses nombreux succès : **le réseau de neurones artificiels (RN)**. En effet, ce dernier est capable d'effectuer des tâches de reconnaissance d'image, de classification, de régression, ... avec de très bonnes performances ; c'est donc un algorithme très versatile. Mais quelque soit l'utilisation faite du réseau de neurones, le modèle sous-jacent conserve la même architecture globale, et l'algorithme d'optimisation de ce modèle également (les modèles plus sophistiqués n'étant que des extensions/améliorations de cette architecture de base). L'appellation de cet algorithme remonte à son origine au début des années 1960 ; en effet, les avancées de l'époque sur l'étude de l'apprentissage humain, et plus particulièrement sur le fonctionnement des neurones biologiques, ont conduit les chercheurs à tenter de mimer le comportement de ces neurones, d'abord avec des systèmes analogiques, dont le pionnier fût le "Mark I perceptron" de Rosenblatt en 1958 [15], puis numériques [56].

Le perceptron vise donc à mimer la biologie des neurones (ou du moins ce qu'on en connaissait à l'époque), et notamment leur structure en réseau. Cependant, les premières tentatives d'élaboration de réseau de neurones artificiels datant des années 1950, les limitations techniques de l'époque ne permettaient pas de modéliser des structures aussi complexes que celles observées en biologie (figure 2.2). Ainsi, dans sa forme la plus simple, le perceptron consiste en un (grand) ensemble de nœuds d'entrée (le "Mark I perceptron" en comprenait quatre-cents) tous connectés à un nœud booléen de sortie (ce nœud de sortie doit valoir un si l'entrée appartient à la classe désirée, et zéro sinon) ; par analogie avec la biologie, ces nœuds sont appelés neurones artificiels.

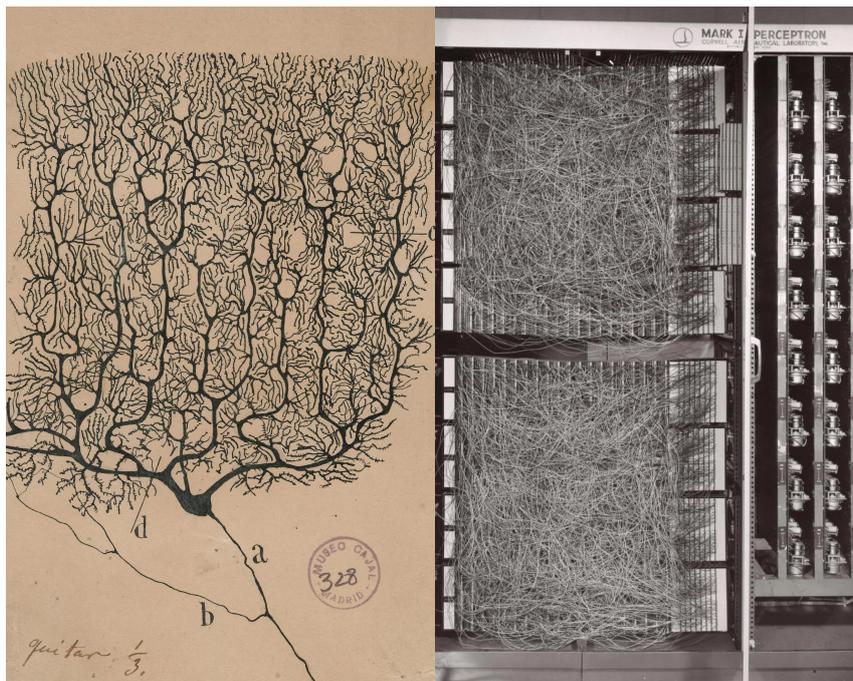


FIGURE 2.2 – À gauche : “*Purkinje neurons from the human cerebellum*” (Cajal Institute, Madrid), dessin de Santiago Ramón y Cajal datant de 1899 représentant des cellules de Purkinje présentes dans le cervelet humain ; on y voit la structure en réseau qui a inspiré les chercheurs en intelligence artificielle. À droite : photo du “Mark I perceptron” de Rosenblatt datant de 1958 ; on y voit un ensemble de connexions à gauche, et, à droite, les potentiomètres qu’il fallait ajuster afin de pondérer ces connexions et obtenir un classifieur binaire performant. Le perceptron de Rosenblatt était capable (une fois ajusté) de reconnaître un homme d’une femme lorsqu’une photo de visage lui était présentée.

Ce perceptron est donc un classifieur binaire permettant de déterminer l’appartenance ou non de l’entrée à une classe donnée. L’objectif étant d’attribuer la bonne classe à la bonne entrée, cette dernière est transformée de manière paramétrée en son “classifieur” (le booléen de sortie) afin de pouvoir ajuster l’attribution de manière adaptée. Une première transformation naïve pourrait être une simple somme pondérée ; en notant x_i la valeur du $i^{\text{ème}}$ neurone d’entrée, y la sortie calculée à partir des x_i , et w_j le coefficient de pondération (le poids) de x_j , on aurait

$$y = w_1x_1 + w_2x_2 + \dots \quad (2.1)$$

En l’état, ce modèle simple ne permet pas grand chose pour une raison majeure : il est linéaire (ce n’est rien de plus qu’une régression multi-linéaire) et est donc trop simpliste pour exécuter des tâches intéressantes. Ainsi, dans l’algorithme de perceptron, la sortie est calculée de la manière suivante

$$y = H(w_1x_1 + w_2x_2 + \dots - b) \quad (2.2)$$

où H est la fonction de Heaviside, et b est un poids supplémentaire (appelé biais) permettant d’autoriser la sortie à être non-nulle lorsque l’entrée, elle, l’est. La fonction de Heaviside agit ici comme un filtre non-linéaire qui ne laisse passer que les valeurs supérieures à b , et joue alors le rôle d’une “fonction d’activation”, l’analogie avec les neurones biologiques étant que l’on considère ces derniers comme actifs lorsqu’ils émettent un signal, et inactif sinon ; b agit donc comme un seuil d’activation.

Dans ce modèle, il ne reste alors plus qu'à trouver les poids w_j et le biais b appropriés pour que le booléen y devienne un bon classifieur. Cette tâche est au cœur de la démarche "d'apprentissage" et a longtemps été un obstacle. En effet, plusieurs algorithmes ont été proposés au fil des années pour optimiser ces coefficients, mais ils étaient souvent compliqués voire impossibles à implémenter en pratique ; l'algorithme désormais omniprésent de descente stochastique du gradient (ce dernier sera présenté en détail dans la section suivante) n'était pas à un stade de développement/performance mathématique et algorithmique suffisant [57] (il faudra attendre les années 1970 pour cela) ; par exemple, dans le perceptron de Rosenblatt (figure 2.2), qui était un ordinateur analogique, les paramètres à optimiser étaient incarnés dans des potentiomètres, et leur optimisation se faisait à l'aide de moteurs électriques [58].

Quelque soit la méthode, une fois les coefficients trouvés, ce modèle permet d'effectuer des tâches de classification diverses, telles que les portes logiques "ET", "OU" et "NON" par exemple ; la porte "ET" peut être représentée par un perceptron à deux neurones d'entrée, et dont les paramètres sont

$$\begin{cases} w_1 = w_2 = 1 \\ b = 1,5 \end{cases} \quad (2.3)$$

Cependant, les capacités de ce modèle sont limitées puisqu'il ne permet de classifier que des données qui sont linéairement séparables, c'est-à-dire dont la frontière d'appartenance à une classe est linéaire [59]. Il est en effet connu depuis 1969 que ce modèle de perceptron ne permet pas de rendre compte de la porte logique "XOR" par exemple [60] (figure 2.3).

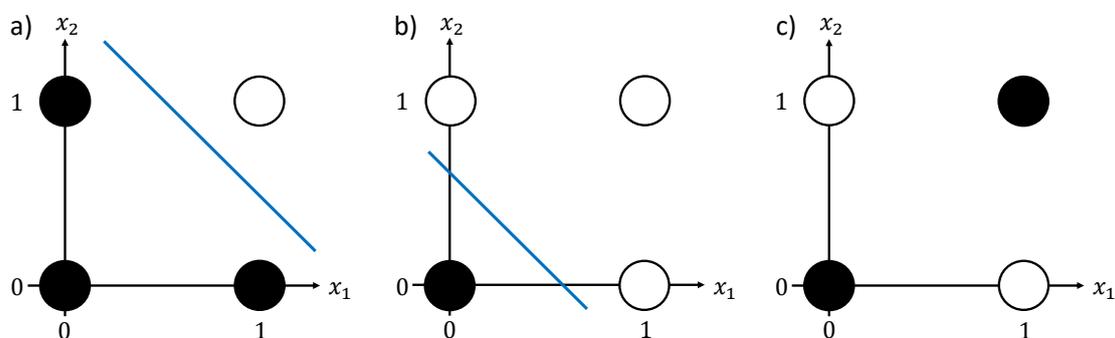


FIGURE 2.3 – Portes logiques "ET", "OU" et "XOR". L'abscisse et l'ordonnée x_1 et x_2 sont les entrées booléennes des portes logiques, et la couleur des cercles représente la valeur de la sortie : le cercle est noir lorsque la sortie vaut zéro et blanc lorsqu'elle vaut un. Les portes "ET" et "OU" sont linéairement séparables, ce qui signifie qu'un hyperplan (une ligne droite en deux dimensions) permet de classifier la sortie (séparer les ronds blancs des ronds noirs), mais la porte "XOR" ne l'est pas.

Ainsi, il fût nécessaire d'améliorer ce modèle de perceptron pour pouvoir effectuer des tâches plus complexes ; ces améliorations donnèrent naissance au réseau de neurones à propagation avant (*feed-forward*) que nous connaissons aujourd'hui.

2.3 Réseau de neurones artificiels à propagation avant

Comme nous l'avons vu, le modèle de réseau de neurones se calque sur la biologie des neurones, mais cette dernière est très compliquée à mimer. Ainsi, les améliorations au modèle naïf de perceptron furent apportées notamment par les développements mathématiques et algorithmiques de la méthode d'optimisation des poids et des biais [57]. L'une des améliorations principales est l'implémentation de couches intermédiaires entre l'entrée et la sortie (appelées couches cachées), permettant alors de traiter des données qui n'étaient pas linéairement séparables.

L'architecture générale d'un réseau de neurones consiste alors en un ensemble de nœuds (les neurones artificiels) connectés les uns aux autres par des flèches (arrêtes), faisant de ce dernier un graphe orienté. Dans un réseau de neurones à propagation avant, ces neurones sont arrangés en plusieurs couches de telle sorte que les neurones appartenant à une même couche ne sont pas connectés entre eux, et les neurones d'une couche ne sont connectés qu'aux neurones de la couche suivante (figure 2.4) ; la première couche est la couche d'entrée et la dernière celle de sortie. Les flèches de connections lient les neurones de la couche n à la couche $n + 1$ et pointent alors toutes dans la même direction : de l'entrée vers la sortie.

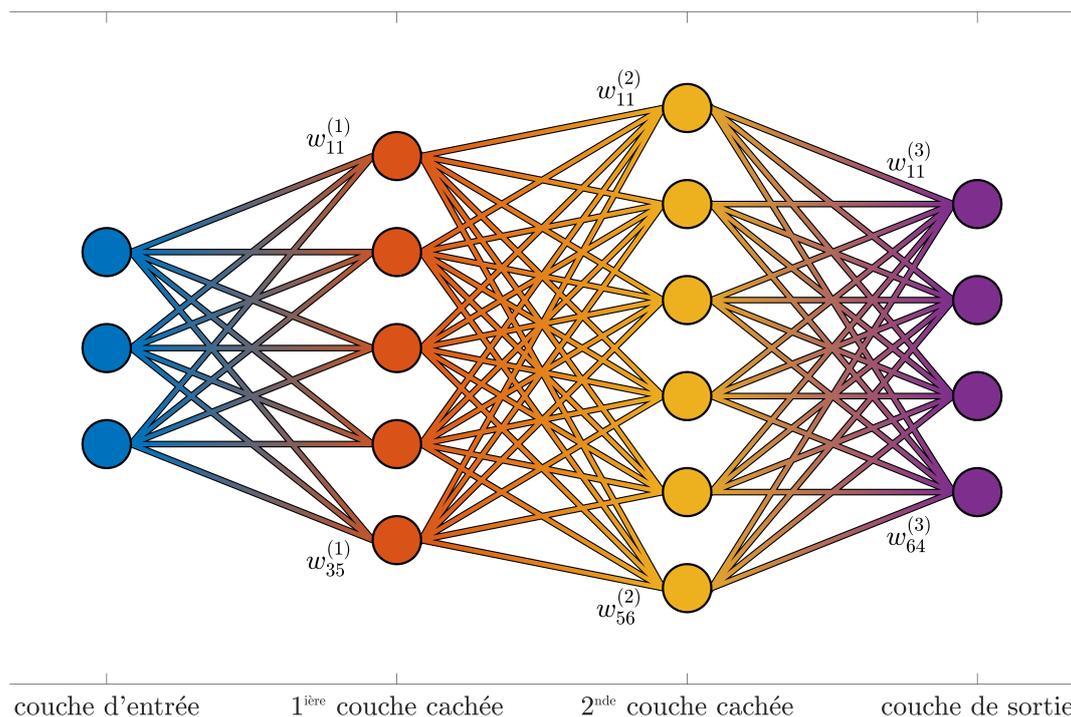


FIGURE 2.4 – Schéma de réseau de neurones artificiels à propagation avant multi-couche. Les liens entre les différentes couches ne sont pas fléchés pour ne pas perdre en lisibilité.

Dans un perceptron multi-couches, afin de calculer la valeur d'un neurone de la couche $n + 1$ connaissant les valeurs des neurones de la couche n , on effectue une somme pondérée de ces dernières. En notant $x_i^{(n)}$ la valeur du $i^{\text{ème}}$ neurone de la couche n , $d^{(n)}$ le nombre de neurones que contient cette même couche, et $w_{ij}^{(n+1)}$ le coefficient de pondération (le poids) de $x_i^{(n)}$ qui

permet de calculer $x_j^{(n+1)}$, on a

$$x_j^{(n+1)} = H \left(w_{1j}^{(n+1)} x_1^{(n)} + w_{2j}^{(n+1)} x_2^{(n)} + \dots - b_j \right) \quad (2.4)$$

$$= H \left(\sum_{i=1}^{d^{(n)}} w_{ij}^{(n+1)} x_i^{(n)} - b_j \right) \quad (2.5)$$

où H correspond toujours à la fonction de Heaviside, et $x^{(0)}$ est le vecteur d'entrée connu. Une telle architecture permet un gain de complexité colossal. En effet, dans le cas de la porte logique "XOR", le perceptron décrit dans la section précédente fournit la transformation

$$y = H(w_1 x_1 + w_2 x_2 - b) \quad (2.6)$$

alors qu'un perceptron avec une couche cachée contenant deux neurones fournit la transformation

$$y = H \left(w_{11}^{(2)} H \left(w_{11}^{(1)} x_1^{(1)} + w_{21}^{(1)} x_2^{(1)} - b_1^{(1)} \right) + w_{21}^{(2)} H \left(w_{12}^{(1)} x_1^{(1)} + w_{22}^{(1)} x_2^{(1)} - b_2^{(1)} \right) - b_1^{(2)} \right) \quad (2.7)$$

et l'espace des paramètres passe alors d'un espace de dimension trois dans l'équation 2.6 à un espace de dimension neuf dans l'équation 2.7. Avec un bon choix de paramètres, il est alors possible d'obtenir une classification équivalente à la porte "XOR" :

$$\begin{cases} w_{11}^{(1)} = w_{21}^{(1)} = w_{12}^{(1)} = w_{22}^{(1)} = w_{11}^{(2)} = -w_{21}^{(2)} = 1 \\ b_1^{(1)} = b_1^{(2)} = \frac{1}{2} \\ b_2^{(1)} = \frac{3}{2} \end{cases} \quad (2.8)$$

L'ajout d'une couche intermédiaire peut s'interpréter comme un gain en dimensionnalité du problème traité, rendant linéairement séparable certains problèmes qui ne l'étaient pas (figure 2.5), la contrepartie étant qu'il y a potentiellement beaucoup plus de paramètres libres à ajuster. Bien que de nombreux résultats avaient déjà montré les capacités des réseaux de neurones depuis les années 1970 (avec des architectures plus complexes que celles présentées jusqu'ici) [56, 61], ce n'est qu'au début des années 1990 qu'il a été montré que l'ajout d'une seule couche intermédiaire avait pour effet de faire du perceptron à une couche cachée un **approximateur universel** [62, 63]; quelque soit la fonction continue que l'on cherche à étudier, le modèle de perceptron à une couche cachée peut l'approximer avec une précision arbitrairement grande, le nombre de neurones de ladite couche, bien que fini, nécessitant possiblement d'être arbitrairement grand. Ce résultat mathématique fût d'une grande importance dans la motivation et la crédibilité des systèmes de type réseau de neurones, mais en pratique, de nouvelles méthodes algorithmiques de propagation et d'optimisation des coefficients avaient déjà été mises en place afin de contourner les limitations techniques de l'époque. En effet, bien qu'un perceptron à une seule couche cachée puisse théoriquement approximer n'importe quelle fonction continue, le temps et la puissance de calcul nécessaires à l'optimisation les coefficients deviennent rapidement hors de portée.

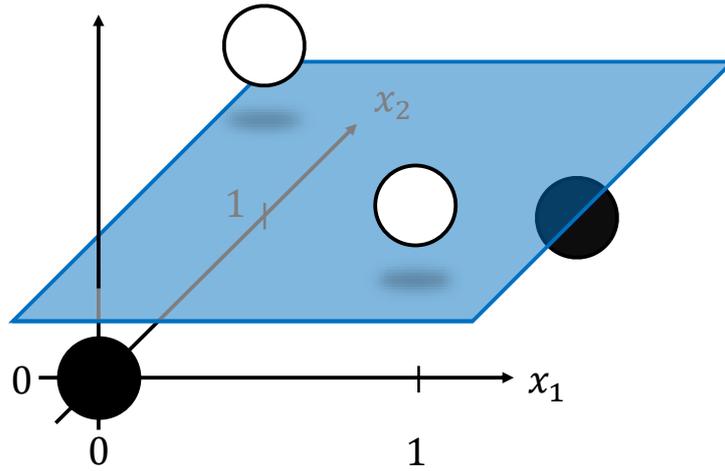


FIGURE 2.5 – Porte logique "XOR" linéairement séparable dans un espace de dimension supérieure à l'espace initial des entrées booléennes x_1 et x_2 , ce qui signifie qu'un hyperplan (un plan en trois dimensions) permet de classifier la sortie (séparer les ronds blancs des ronds noirs)

Ainsi, les évolutions du modèle de perceptron virent donc le jour rapidement, en même temps que ce dernier continuait d'être exploré, et des modèles à plusieurs couches cachées et avec d'autres fonctions d'activation furent développés. L'une des principales évolutions fût la méthode de **descente stochastique du gradient**, qui a été efficacement décrite et implémentée pour des réseaux de neurones plus complexes que le perceptron dès 1970 [64] et fût alors largement employée, au point de devenir le standard actuel. Cette méthode, basée sur une première étape de propagation et une seconde, dite de rétro-propagation, est une implémentation algorithmique efficace du théorème de dérivation des fonctions composées introduit par Leibniz en 1673 [65]. En effet, un réseau de neurones peut être interprété comme étant une fonction très complexe qui transforme un vecteur (le vecteur d'**entrée**) en un autre vecteur (le vecteur de **sortie**), que l'on espère être suffisamment similaire au vecteur désiré (le vecteur **cible**), la notion de similarité restant à définir, souvent réduite à une notion de distance.

Il est donc bon de se représenter un réseau de neurones comme une fonction paramétrée (de paramètres θ) qui opère sur un espace de (très) grande dimension. Dans le cas particulier où l'espace d'entrée et de sortie sont des vecteurs de nombres réels, en notant RN_θ cette fonction, X le vecteur d'entrée de dimension d_X , et Y le vecteur de sortie de dimension d_Y , on a

$$\begin{aligned} RN_\theta: \mathbb{R}^{d_X} &\longrightarrow \mathbb{R}^{d_Y} \\ X &\longmapsto Y \end{aligned}$$

Cette fonction n'est pas définie explicitement, mais plutôt par la description du processus itératif qui permet d'obtenir Y connaissant X ; en conservant les mêmes notations que précédemment, le mécanisme de propagation donne donc

$$x_j^{(n+1)} = f^{(n+1)} \left(\sum_{i=1}^{d^{(n)}} w_{ij}^{(n+1)} x_i^{(n)} + b_j^{(n+1)} \right) = f^{(n+1)} \left(s_j^{(n+1)} \right) \quad (2.9)$$

avec

$$s_j^{(n)} = \sum_{i=1}^{d^{(n-1)}} w_{ij}^{(n)} x_i^{(n-1)} + b_j^{(n)} \quad (2.10)$$

et où $f^{(n)}$ est une fonction non-linéaire (l'exposant ne fait pas référence à la dérivée $n^{ième}$ de f mais à l'indice de la couche considérée); le modèle présenté jusqu'ici avait la fonction de Heaviside H en guise de fonction non-linéaire. La fonction RN_θ est donc une fonction en couches imbriquées, dont la sortie peut s'exprimer comme une composition de plusieurs fonctions non-linéaires; c'est une succession de transformations élémentaires linéaires et non-linéaires réalisées par les neurones de chaque couche (figure 2.6).

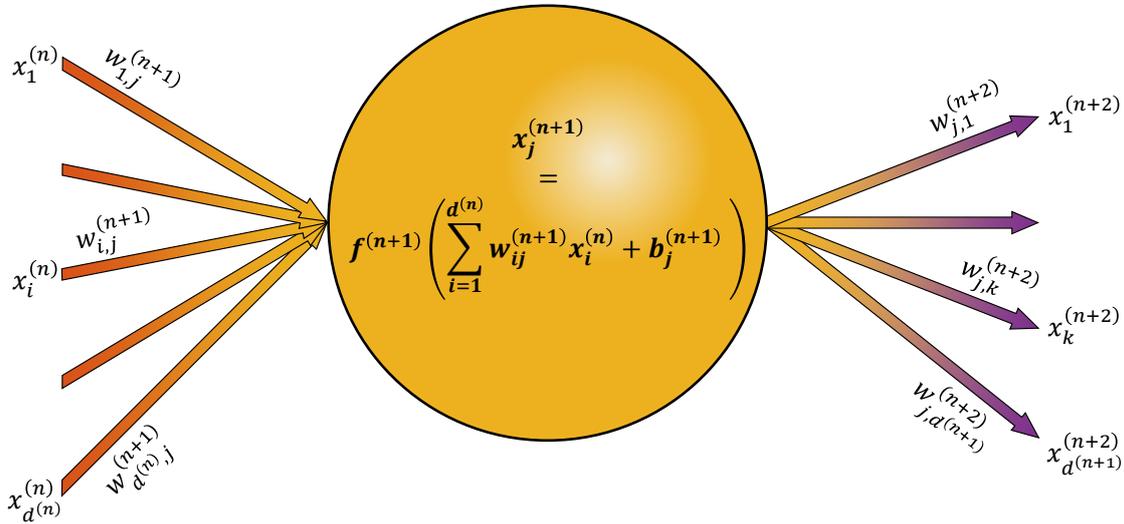


FIGURE 2.6 – Opération élémentaire effectuée par un neurone d'un réseau de neurones à propagation avant (tel que celui de la figure 2.4).

L'avantage de ce modèle est qu'il permet de décrire des comportements très complexes grâce aux fonctions non-linéaires, tout en étant **simple à paramétrer**; le (très) grand nombre de paramètres libres est à la fois un avantage et un inconvénient, puisqu'il permet de rendre compte de comportements riches, mais nécessite un temps d'optimisation (beaucoup) plus long. Dans une architecture de réseau de neurones telle que celle présentée ici, le nombre de paramètres se calcule grâce à la formule

$$N_\theta = \sum_{k=0}^{N_c-1} d^{(k)} d^{(k+1)} + \sum_{k=0}^{N_c-1} d^{(k+1)} \quad (2.11)$$

$$= \sum_{k=0}^{N_c-1} (d^{(k)} + 1) d^{(k+1)} \quad (2.12)$$

où N_c est le nombre de couches du réseau de neurones (en comptant la couche d'entrée, la couche de sortie et les couche cachées), et où la somme de gauche compte les poids d'une couche, et celle de droite les biais; par exemple, un réseau de neurones prenant en entrée un vecteur de taille 10, cherchant à prédire un scalaire, et comportant deux couches cachées de respectivement 9 et 5 neurones contient donc (déjà) 155 paramètres.

Le choix de la fonction $f^{(n)}$ est a priori arbitraire, mais de nombreuses expérimentations ont permis de sélectionner certaines familles de fonctions qui semblent être pertinentes. La première d'entre elles a été une forme "adoucie" de la fonction de Heaviside : la fonction logistique, aussi appelée sigmoïde (figure 2.7) :

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.13)$$

Le choix de cette dernière a été motivé par le fait qu'elle a l'avantage d'avoir une dérivée continue et non-nulle contrairement à la fonction de Heaviside (les raisons pour lesquelles c'est un avantage seront expliquées dans la section suivante), et qu'elle était déjà largement répandue en statistique comme fonction de régression logistique. Cependant, il peut être nécessaire de produire une sortie non-pas comprise entre 0 et 1 mais entre -1 et 1 ; c'est ce qu'accomplit la fonction tangente hyperbolique (figure 2.7) :

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.14)$$

$$= 2\sigma(2x) - 1 \quad (2.15)$$

Cette fonction a également l'avantage d'être centrée en zéro, ce qui, pour des raisons statistiques, est préférable lors de la formation d'un réseau de neurones [66]. Ces fonctions ont toutes deux un comportement similaire (en forme de "S"), mais d'autres fonctions ont été développées et ont empiriquement puis théoriquement démontré leur intérêt ; c'est le cas par exemple de la fonction ReLU (*Rectified Linear Unit*) qui est désormais l'une des fonctions non-linéaires les plus utilisées [67] [68] (figure 2.7) :

$$\text{ReLU}(x) = \max(0, x) \quad (2.16)$$

L'ensemble des fonctions qui viennent d'être présentées est visible dans la figure suivante

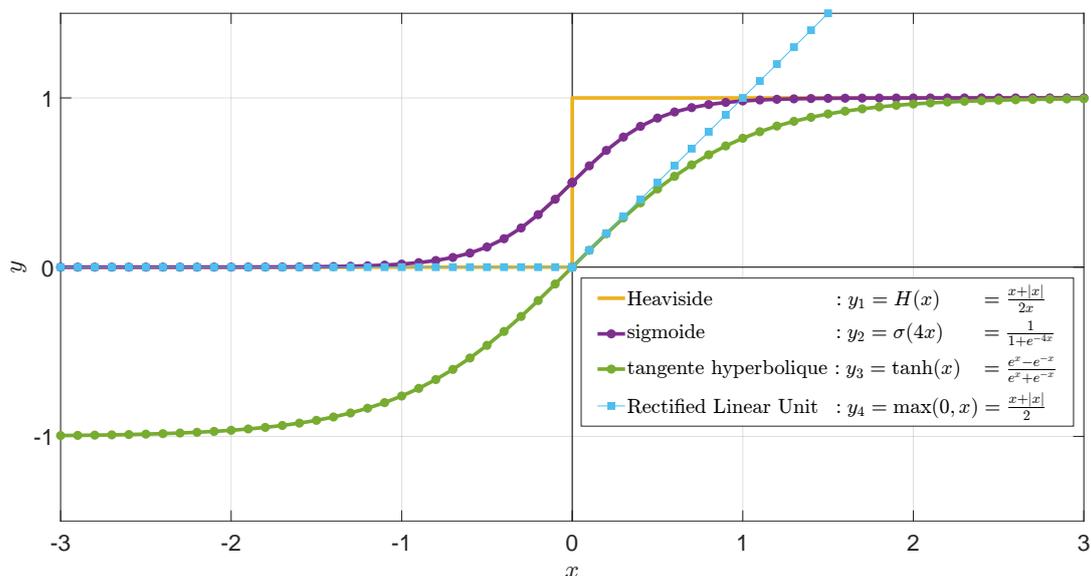


FIGURE 2.7 – Exemple de fonctions d'activation usuelles : fonction de Heaviside (montrée à titre de comparaison, elle n'est quasiment jamais utilisée de nos jours), fonction logistique, tangente hyperbolique, et fonction ReLU (*Rectified Linear Unit*).

2.4 Descente stochastique du gradient

Dans le modèle général de réseau de neurones à propagation avant, l'entrée et la sortie ayant des longueurs arbitraires, une donnée complète est une paire de vecteurs (le vecteur d'entrée et le vecteur de sortie) qui peuvent être de natures diverses et variées (le vecteur d'entrée peut contenir des mots et le vecteur de sortie des entiers par exemple); quelque soit la nature des vecteurs, ces derniers seront numérisés pour pouvoir effectuer des opérations arithmétiques avec eux (par exemple le plongement lexical ou *word-embedding* est une méthode utilisée pour numériser les mots). Tout le problème est donc de trouver de bons paramètres $w_{ij}^{(n)}$ et $b_j^{(n)}$ qui vont rendre les données le moins surprenantes possible. Pour cela, il faut d'abord définir ce que l'on entend par "surprenantes". La démarche présentée ici suit [69].

L'algorithme classique utilisé dans un réseau de neurones à propagation avant consiste à mettre à jour les poids et les biais de manière à ce qu'ils minimisent une fonction d'erreur ou fonction coût C choisie (erreur quadratique moyenne par exemple), cette dernière quantifiant l'écart entre la sortie attendue et celle fournie par le réseau; une donnée est donc surprenante si le vecteur de sortie prédit par le réseau est "éloigné" du vecteur de sortie attendu au sens de cette fonction coût. Cela sous-entend donc que cet algorithme a besoin de données complètes (dont on connaît la sortie) pour pouvoir mettre à jour les paramètres du modèle; c'est donc un algorithme supervisé. Ces données dites d'entraînement serviront donc à former le réseau, puis des données dites de test serviront à mesurer la capacité du réseau à extrapoler, le but étant au final de ne fournir au réseau que des vecteurs d'entrée dont on ignore la sortie afin qu'il complète ces entrées; on veut que le réseau devienne **un bon modèle prédictif**.

Pour mettre à jour les paramètres du réseau, il faut un moyen de quantifier la sensibilité du modèle à ces derniers. L'objectif de l'algorithme étant de minimiser le niveau de surprise des données qui lui sont présentées, cela revient à quantifier la variation de ce niveau de surprise pour une donnée par rapport aux paramètres. Une bonne mesure de cette sensibilité est donc (en première approche) la dérivée partielle de la fonction coût C par rapport aux poids et aux biais, et son extension : le gradient. En effet, le vecteur gradient est un vecteur qui, en chaque point de l'espace considéré (ici l'espace des paramètres), pointe dans la direction de plus grande pente locale positive; il suffit donc d'aller dans la direction opposée à celle du gradient pour aller dans la direction de plus grande pente locale négative, et donc se rapprocher d'un minimum (local !). La figure 2.8 permet de visualiser ce processus d'optimisation en montrant l'évolution d'une fonction coût arbitraire en fonction de seulement deux paramètres w_1 et w_2 , ainsi que quatre trajectoires avec différents paramètres d'optimisation; l'objectif est donc d'optimiser les paramètres w_1 et w_2 afin de minimiser la fonction coût C .

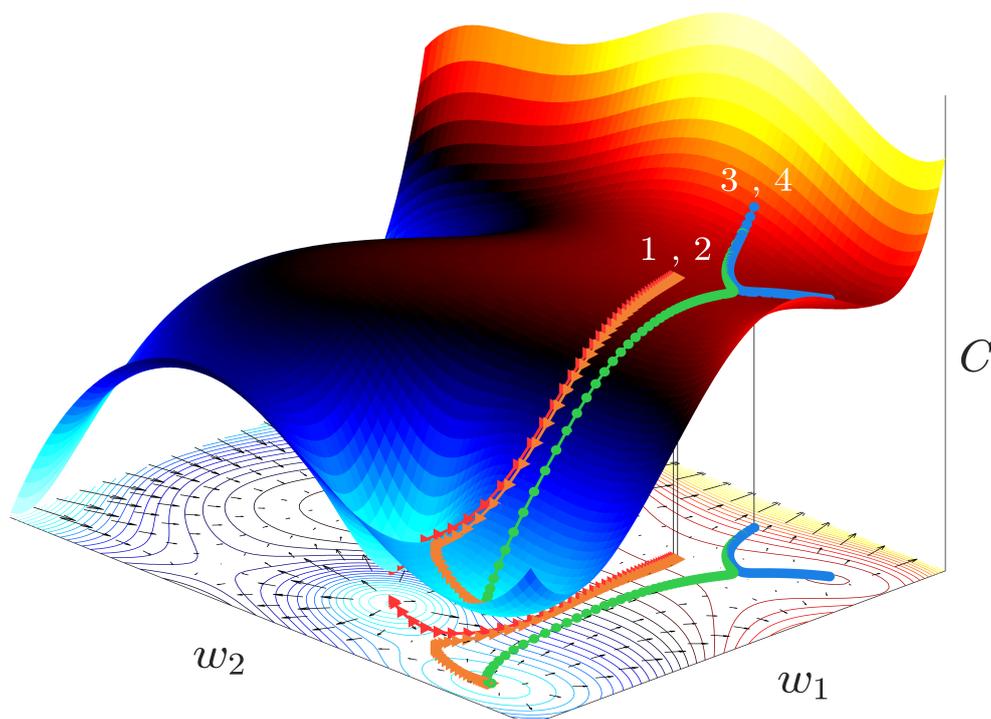


FIGURE 2.8 – Visualisation de l'algorithme de descente du gradient. Ce schéma présente quatre processus d'optimisation dans l'espace des paramètres : les processus numérotés "1" et "2" (courbes orange et rouge avec des triangles) qui partent de deux endroits très légèrement différents (au niveau du point d'inflexion de la surface), et les processus numérotés "3" et "4" (courbes bleue et verte avec des cercles) qui partent du même endroit (relativement élevé) mais mis à jour avec des taux d'apprentissages différents ; pour chacun de ces processus, l'algorithme de descente du gradient met à jour les paramètres selon la règle 2.18. Les flèches noires représentent le gradient en différents points de l'espace des paramètres. Ceci illustre bien l'importance cruciale du choix des hyper-paramètres (le taux d'apprentissage ici) ainsi que des conditions initiales des paramètres (poids et biais).

On veut donc mettre à jour les poids et les biais selon

$$\begin{cases} w_{ij}^{(n)} \rightarrow w_{ij}^{(n)} - \frac{\partial C}{\partial w_{ij}^{(n)}} \\ b_j^{(n)} \rightarrow b_j^{(n)} - \frac{\partial C}{\partial b_j^{(n)}} \end{cases} \quad (2.17)$$

Il est en réalité préférable de pondérer la dérivée partielle par un coefficient η , appelé taux d'apprentissage (*learning rate*) afin de maîtriser la taille de l'incrément de mise à jour et ainsi éviter un apprentissage trop lent ou qui ne converge jamais ; on a donc

$$\begin{cases} w_{ij}^{(n)} \rightarrow w_{ij}^{(n)} - \eta \frac{\partial C}{\partial w_{ij}^{(n)}} \\ b_j^{(n)} \rightarrow b_j^{(n)} - \eta \frac{\partial C}{\partial b_j^{(n)}} \end{cases} \quad (2.18)$$

et pour cela, il faut donc calculer

$$\begin{cases} \frac{\partial C}{\partial w_{ij}^{(n)}} = \frac{\partial C}{\partial s_j^{(n)}} \frac{\partial s_j^{(n)}}{\partial w_{ij}^{(n)}} = \frac{\partial C}{\partial s_j^{(n)}} x_i^{(n-1)} \\ \frac{\partial C}{\partial b_j^{(n)}} = \frac{\partial C}{\partial s_j^{(n)}} \frac{\partial s_j^{(n)}}{\partial b_j^{(n)}} = \frac{\partial C}{\partial s_j^{(n)}} \end{cases} \quad (2.19)$$

La structure en couches et le grand nombre de paramètres pourraient rendre ces calculs fastidieux, mais la nature récursive de la propagation dans le réseau de neurones à propagation avant (2.9) rend la tâche plus simple. En notant

$$\delta_j^{(n)} = \frac{\partial C}{\partial s_j^{(n)}} \quad (2.20)$$

on a

$$\delta_j^{(n)} = \frac{\partial C}{\partial s_j^{(n)}} \quad (2.21)$$

$$= \sum_{k=1}^{d^{(n+1)}} \frac{\partial C}{\partial s_k^{(n+1)}} \frac{\partial s_k^{(n+1)}}{\partial s_j^{(n)}} \quad (2.22)$$

$$= \sum_{k=1}^{d^{(n+1)}} \delta_k^{(n+1)} \frac{\partial s_k^{(n+1)}}{\partial s_j^{(n)}} \quad (2.23)$$

par application de la dérivation des fonctions composées. Or

$$s_k^{(n+1)} = \sum_{j=1}^{d^{(n)}} w_{jk}^{(n+1)} x_j^{(n)} + b_k^{(n+1)} \quad (2.24)$$

$$= \sum_{j=1}^{d^{(n)}} w_{jk}^{(n+1)} f^{(n)} \left(s_j^{(n)} \right) + b_k^{(n+1)} \quad (2.25)$$

et donc

$$\frac{\partial s_k^{(n+1)}}{\partial s_j^{(n)}} = w_{jk}^{(n+1)} f^{(n)'} \left(s_j^{(n)} \right) \quad (2.26)$$

On obtient finalement, toujours en notant N_c le nombre de couches du réseau de neurones

$$\begin{cases} w_{ij}^{(n)} \rightarrow w_{ij}^{(n)} - \eta \delta_j^{(n)} x_i^{(n-1)} \\ b_j^{(n)} \rightarrow b_j^{(n)} - \eta \delta_j^{(n)} \end{cases} \quad (2.27)$$

avec

$$\left\{ \begin{array}{l} \delta_j^{(N_c)} = f^{(N_c)'}(s_j^{(N_c)}) \frac{\partial C}{\partial x_j^{(N_c)}} \end{array} \right. \quad (2.28a)$$

$$\left\{ \begin{array}{l} \delta_j^{(n)} = f^{(n)'}(s_j^{(n)}) \sum_{k=1}^{d^{(n+1)}} w_{jk}^{(n+1)} \delta_k^{(n+1)} \quad \text{si } n < N_c \end{array} \right. \quad (2.28b)$$

La ressemblance de la formule 2.28b avec la formule 2.10 ainsi que sa nature itérative ont contribué à donner le nom de **rétro-propagation** à ce processus de mise à jour des paramètres.

Ces formules ayant été établies, il apparaît maintenant clairement que les dérivées des fonctions non-linéaires choisies jouent un rôle majeur dans les performances de l'algorithme d'optimisation des paramètres, et donc dans les performances du réseau de neurones. Ainsi, les fonctions de type sigmoïde présentent l'inconvénient d'avoir des dérivées qui décroissent rapidement vers zéro, bloquant la mise à jour des paramètres, puisque cette dernière se fait proportionnellement à ces dérivées (2.27, 2.28b); ce problème est connu sous le nom de *vanishing gradient* (extinction du gradient) [70], et est visible sur la figure 2.9. Le phénomène inverse appelé *exploding gradient* peut également avoir lieu lorsque la dérivée de la fonction non-linéaire devient trop grande, résultant en une optimisation qui ne converge jamais.

La fonction ReLU n'a aucun de ces inconvénients, et c'est l'une des raisons avancées pour expliquer ses performances [68, 71]. Cependant, il serait faux de conclure que les performances de cette fonction surpassent systématiquement celles des autres (les sigmoïdes). En effet, le fait qu'elle soit nulle pour des valeurs négatives peut entraîner le phénomène de "mort des neurones" (problème plus connu sous le nom de *dying ReLU*), c'est-à-dire le phénomène selon lequel l'optimisation stagne lorsque celle-ci avait préalablement conduit des valeurs négatives en sortie de couche ReLU [72]. De plus, sa moyenne est elle aussi non-nulle, ce qui introduit un biais dans les couches suivantes qu'il est difficile de contrer [73]; la fonction tangente hyperbolique par exemple ne présente pas ce dernier inconvénient.

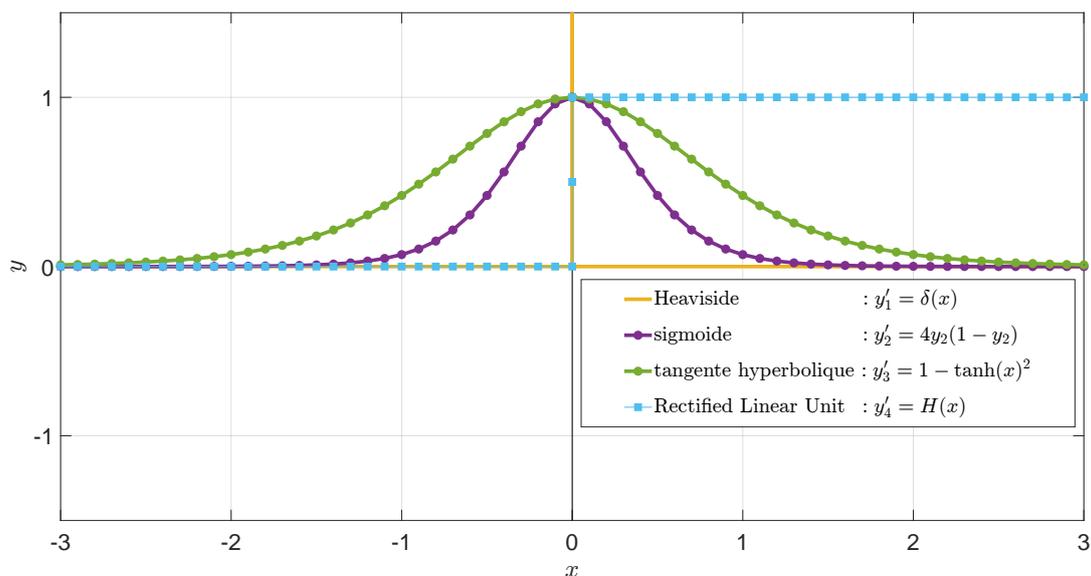


FIGURE 2.9 – Exemple de dérivées de fonctions d'activation usuelles : fonction de Heaviside (montrée à titre de comparaison, elle n'est quasiment jamais utilisée de nos jours), fonction logistiquie, tangente hyperbolique, et fonction ReLU (*Rectified Liner Unit*).

Ainsi, ces problèmes de convergence de l'algorithme d'optimisation (*vanishing gradient*, *exploding gradient* ou *dying ReLU*) dus aux propriétés des fonctions non-linéaires choisies, et à cause desquels le processus d'optimisation peut s'avérer infructueux, trouvent en partie leur solution dans la normalisation des données d'entrée (voire des résultats de sortie de chaque couche). Cette normalisation des données est un des aspects cruciaux de l'entraînement d'un réseau de neurones. En effet, il se peut très bien que le vecteur d'entrée soit constitué de valeurs de natures différentes (on pourrait vouloir prédire la taille d'un individu connaissant son âge, son sexe et sa localisation géographique par exemple), et donc, si aucune mesure n'est prise, les informations contenant les plus grandes valeurs numériques auraient un impact artificiellement gonflé, rendant l'optimisation des paramètres inutilement fastidieuse, voire impossible; dans ce cas-là, le problème serait alors mal conditionné. Cependant, face à la complexité du modèle que constitue un réseau de neurones, un critère d'optimalité pour le conditionnement des données est impossible à fournir; ainsi, seules quelques règles empiriques faisant office de bonnes pratiques sont communément admises comme nécessaires (dans la plupart des cas) pour pouvoir valider ou rejeter une architecture donnée. Parmi ces bonnes pratiques, il est conseillé de **centrer et réduire** chaque donnée d'entrée $X^{(0)}$ et de sortie $X^{(N_c)}$, afin que toutes les valeurs soient à "la même échelle" [66].

De plus, afin de maximiser les chances de convergence de l'algorithme d'optimisation, le choix des valeurs initiales des paramètres à optimiser, c'est-à-dire leurs positions initiales dans l'espace des paramètres, est une étape à ne pas négliger. En effet, comme le montre la figure 2.8, un mauvais choix de conditions initiales peut conduire à une sous-performance du réseau de neurones; les processus 1 et 2 subissent exactement la même mise à jour (2.18) mais partent de deux endroits très légèrement différents, et il en résulte qu'ils ne convergent pas vers le même minimum local, celui de la courbe rouge étant plus intéressant. Cependant, l'espace des paramètres d'un réseau de neurones, même relativement petit, devient très rapidement hors de portée (2.12), et il est donc extrêmement difficile de fournir des critères garantissant de bonnes performances. Ainsi, la valeur initiale des paramètres à optimiser nécessite une approche statistique, qui permet de conclure qu'il est préférable que ces derniers soient tirés aléatoirement selon une distribution gaussienne de moyenne nulle et d'écart-type égal à l'inverse de la racine du nombre de neurones d'entrée de la couche considérée (la variance est donc égale à l'inverse du nombre de neurones d'entrée de la couche) [69] :

$$w_{ij}^{(n)} \sim \mathcal{N}\left(0, \frac{1}{d^{(n-1)}}\right) \quad (2.29)$$

Cette initialisation est appelée "initialisation de Xavier" (*Xavier initialization*), du prénom du chercheur qui a popularisé cette méthode [74].

Il est important de noter que la taille du réseau (nombre de couches et nombre de neurones par couche) ainsi que le paramètre η sont à différencier des paramètres du modèle $w_{ij}^{(n)}$ et $b_j^{(n)}$. En effet, ils ne varient pas au cours du processus d'optimisation (η peut varier avec des algorithmes plus avancés mais il n'en est pas moins un paramètre de l'algorithme d'optimisation et pas un paramètre du modèle), et leurs valeurs sont donc laissées au libre choix de l'utilisateur. Bien que l'optimisation se fasse quelque soit leurs valeurs, cette optimisation converge plus ou moins bien en fonction de ces paramètres comme le montre la figure 2.8; les processus 3 et 4 partent du même endroit de l'espace des paramètres et sont tous les deux mis à jour selon la règle 2.18, mais avec des taux d'apprentissage (*learning rate*) η différents (celui utilisé pour la courbe bleue, qui converge vers un minimum local très proche du point de départ, est trois fois plus petit que celui utilisé pour la courbe verte, qui, elle, converge vers un minimum local bien plus bas). Ces paramètres qui ne sont pas mis à jour lors du processus de formation du réseau

de neurones sont appelés **hyper-paramètres**, et leur ajustement est donc une étape cruciale de l'algorithme d'apprentissage. Malheureusement, comme pour le conditionnement des données, il n'existe pas d'algorithme performant permettant de déterminer de bonnes valeurs d'hyper-paramètres pour un problème donné, et seules de bonnes pratiques et des essais/erreurs plus ou moins sophistiqués permettent de trouver un ensemble d'hyper-paramètres conduisant à une optimisation convaincante [75].

Enfin, la mise à jour des paramètres décrite jusqu'ici se fait en ne tenant compte que d'une seule donnée complète (paire entrée/sortie); en effet, dans l'équation 2.28a, la dérivée de la fonction coût C ne s'effectue que par rapport à un seul élément de sortie $x_j^{(N_c)}$. Or il a été observé qu'il est intéressant de mettre à jour les paramètres sur la base de plusieurs données à la fois [76]; des mises à jour trop fréquentes ne permettent pas de profiter pleinement des capacités de parallélisation des bibliothèques d'algèbre linéaire modernes (sur CPU ou GPU), et peuvent donc rendre le processus d'optimisation plus lent et erratique. Ainsi, il est préférable de mettre à jour les paramètres en tenant compte du gradient de la fonction coût moyenné sur plusieurs données. En notant $x_i^{(n),D}$ la valeur du $i^{\text{ème}}$ neurone de la couche n et de la donnée D , le système 2.27 devient alors

$$\begin{cases} w_{ij}^{(n)} \rightarrow w_{ij}^{(n)} - \frac{\eta}{N_D} \sum_{D=1}^{N_D} \delta_j^{(n),D} x_i^{(n-1),D} \\ b_j^{(n)} \rightarrow b_j^{(n)} - \frac{\eta}{N_D} \sum_{D=1}^{N_D} \delta_j^{(n),D} \end{cases} \quad (2.30)$$

où N_D est le nombre de données considérées pour la mise à jour. Le paquet de données choisies pour effectuer la mise à jour est appelé un *batch* de données; la méthode qui était présentée jusqu'à maintenant était donc la descente du gradient avec un *batch* de taille un, méthode dite de *online learning*.

Comme expliqué plus haut, le *online learning* présente l'inconvénient de ne pas tirer profit des capacités de parallélisation des ordinateurs modernes (qui peuvent paralléliser la somme effectuée sur les données, comme il sera expliqué dans la section suivante), et il est donc intéressant de choisir une valeur de N_D plus grande. L'autre extrême, consistant à prendre N_D égal au nombre total de données d'entraînement disponibles, est appelé *full batch learning*. Cependant, cette méthode nécessite de charger en mémoire vive toutes les données d'entraînement (qui sont a priori très nombreuses), ce qui a pour effet de ralentir le processus total d'optimisation. De plus, pour des raisons encore assez peu comprises, une taille de *batch* trop grande peut avoir pour effet de dégrader les performances de généralisation du modèle [77]. Donc, bien que que le *full batch learning* permette d'évaluer la fonction coût et son gradient de manière très précise (puisque l'erreur calculée du modèle tient compte de toutes les données), cette méthode ne constitue pas un choix optimal aussi bien en terme de performance que de vitesse. Ainsi, le choix de la taille du *batch* est une affaire de compromis entre le gain de temps de calcul occasionné par la parallélisation, la perte de temps et d'espace mémoire occasionnés par le grand nombre de données d'entraînement, et la précision du calcul du gradient de la fonction coût; lorsque la taille du *batch* n'est égale ni à un, ni au nombre total de données d'entraînement, le *batch* est alors renommé en *mini-batch*, et on parle alors de *mini-batch learning*. On appelle alors *epoch* l'incrément qui compte le nombre de fois que le modèle a parcouru toutes les données d'entraînement disponibles; en *full batch learning*, une *epoch* correspond à une seule itération d'optimisation (puisque toutes les données d'entraînement sont parcourues

en une seule itération), et en *online learning*, il faut attendre autant d'itérations d'optimisation que de données d'entraînement pour qu'une *epoch* se soit écoulée (puisque une seule donnée d'entraînement est parcourue par itération).

De plus, comme il a été expliqué en introduction de ce chapitre, le biais dû à l'échantillonnage est inévitable, faisant apparaître un compromis entre le biais et la variance du réseau de neurones, ce qui se traduit encore par le phénomène de **sur-interprétation** (*overfitting*). Ce phénomène apparaît lorsque le modèle fournit de bonnes performances sur les données d'entraînement, mais de moins bonnes sur les données de test, indiquant qu'il a été "trop" optimisé, et qu'il finit par apprendre les détails des données d'entraînement, plutôt que d'en extraire le comportement global/statistique; un modèle surentraîné peine donc à généraliser ses prédictions sur des données pour lesquelles il n'a pas été formé (les données de test). En d'autres termes, les données d'entraînement n'étant qu'une partie infime des données possibles, un biais d'échantillonnage est inévitable, et donc le minimum global de la fonction coût n'est pas (a priori) l'objectif à atteindre; il faut plutôt converger vers un minimum local qui serait commun à tous les jeux de données d'entraînement possible, i.e. un minimum invariant par changement de données d'entraînement. Pour éviter cette sur-interprétation, il est donc préférable de mélanger les données d'entraînement de manière aléatoire afin d'éviter au maximum l'apparition de structure non-désirées que le réseau de neurones pourrait "apprendre"; en faisant cela, le sous-échantillon de données d'entraînement que constitue le *mini-batch* étant construit aléatoirement, l'algorithme de descente du gradient devient, de fait, stochastique. Le gradient ainsi calculé n'est qu'une estimation du "vrai" gradient (celui qui aurait été calculé en prenant en compte toutes les données d'entraînement), mais il a été montré que cette approximation du gradient ne dégradait pas les performances finales du réseau de neurones [78, 79, 80]. Cependant, cela ne suffit généralement pas, et d'autres techniques sont mises en place; les méthodes visant à limiter la sur-interprétation et donc maximiser les capacités de généralisation d'un modèle sont appelées méthodes de régularisation.

Parmi elles, la régularisation de la fonction coût a pour but de modifier artificiellement cette dernière, ce qui a pour effet de sur-contraindre artificiellement les paramètres à optimiser, dans le but qu'ils ne puissent pas explorer de régions spécifiques de l'espace des paramètres. Il a été observé que la sur-interprétation avait lieu lorsque les paramètres prenaient des valeurs trop élevées, donc les méthodes de régularisation les plus utilisées sont celles qui contraignent ces paramètres à avoir des valeurs qui n'explorent pas. Parmi ces méthodes, la plus fréquente est celle de Tikhonov (appelé aussi régression de Ridge) [81], où une pénalisation sur la norme des poids est ajoutée à la fonction coût; les normes le plus souvent utilisées sont la norme L_1 , la norme L_2 (parfois la norme L_∞). La nouvelle fonction C' s'exprime alors

$$C' = C + \lambda \sum_w w^2 \quad (2.31)$$

où la somme s'effectue sur tous les poids du réseau. Cette nouvelle fonction coût croît à mesure que les poids augmentent, et l'algorithme d'optimisation cherchera donc à minimiser la valeur de ces poids (tout en continuant de minimiser la fonction coût initiale C) pour minimiser la fonction coût globale C' .

Afin de limiter la sur-interprétation, il est également possible de raccourcir le temps d'optimisation, réduisant ainsi les chances de l'algorithme de s'adapter aux détails des données d'entraînement; cette technique est appelée *early stopping* (arrêt précoce). Il est également possible de rendre le modèle robuste au bruit, donc là encore, moins sensible aux détails des données d'entraînement, en bruitant artificiellement les données d'entrée (tout en conservant leurs sorties); cela a pour effet d'entraîner le modèle à savoir regrouper dans une même catégo-

rie des données très légèrement différentes, ce qui est donc très utile lorsqu'un réseau de neurones est utilisé comme classifieur. Enfin, l'une des méthodes les plus utilisées pour s'assurer de la bonne généralisation du réseau est le "**dropout**" [82]. Cette technique est assez particulière en ce qu'elle modifie l'architecture même du réseau. Elle consiste à choisir aléatoirement un ensemble de neurones du réseau et de les rendre temporairement inactifs (un tel neurone est alors "mort"), le temps d'un cycle d'optimisation (propagation/rétro-propagation) effectué sur un *mini-batch*; la proportion de neurones choisis parmi l'ensemble de ceux disponibles constituant donc un nouvel hyper-paramètre. En passant au *mini-batch* suivant, les neurones qui avaient été désactivés sont réactivés, et un nouvel ensemble de neurones est choisi aléatoirement pour être désactivé à son tour le temps d'un cycle d'optimisation. Cette méthode de régularisation a pour but de mimer une perte aléatoire d'information, et permet, d'un certain point de vue, de former plusieurs réseaux de neurones à la fois; en effet, à chaque tirage aléatoire de neurones à désactiver, c'est une nouvelle architecture qui est sondée et optimisée, si bien que la prédiction finale du réseau entraîné peut s'interpréter comme une moyenne des prédictions des différents "sous-réseaux" qui ont été entraînés. Cette méthode constitue donc une manière très efficace d'obtenir un modèle capable de bien généraliser.

Toutes ces contraintes inhérentes au processus d'optimisation d'un modèle avec un très grand nombre de paramètres rendent la tâche de convergence vers un modèle pertinent très fastidieuse en pratique; de longues phases d'essai/erreur, guidées par des méthodes générales [66, 83], sont inévitables et n'aboutissent pas toujours à un résultat convainquant.

2.4.1 Implémentation

L'implémentation des formules qui viennent d'être présentées se doit d'exploiter au mieux les capacités de parallélisation des ordinateurs modernes pour pouvoir traiter un grand nombre de données en un temps raisonnable. L'implémentation en MATLAB de ce qui va être présenté dans cette section se trouve dans l'annexe A.

2.4.1.1 Propagation et rétro-propagation

Pour implémenter l'algorithme de réseau de neurones efficacement, il faut passer au formalisme matriciel, ce dernier permettant de bénéficier de la parallélisation des bibliothèques d'algèbre linéaires des langages de programmation (tel que MATLAB). Ainsi, on remarque que l'équation de propagation dans le réseau (2.9) peut s'écrire sous forme matricielle

$$\begin{pmatrix} x_1^{(n+1)} \\ x_2^{(n+1)} \\ \vdots \\ x_{d^{(n+1)}}^{(n+1)} \end{pmatrix} = f^{(n+1)} \left(\begin{pmatrix} w_{11}^{(n+1)} & w_{21}^{(n+1)} & \cdots & w_{d^{(n)}1}^{(n+1)} \\ w_{12}^{(n+1)} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ w_{1d^{(n+1)}}^{(n+1)} & \cdots & \cdots & w_{d^{(n)}d^{(n+1)}}^{(n+1)} \end{pmatrix} \begin{pmatrix} x_1^{(n)} \\ x_2^{(n)} \\ \vdots \\ x_{d^{(n)}}^{(n)} \end{pmatrix} + \begin{pmatrix} b_1^{(n+1)} \\ b_2^{(n+1)} \\ \vdots \\ b_{d^{(n)}}^{(n+1)} \end{pmatrix} \right) \quad (2.32)$$

soit, de manière compacte

$$X^{(n+1)} = f^{(n+1)} (W^{(n+1)} X^{(n)} + B^{(n+1)}) \quad (2.33)$$

où $X^{(n)}$ est le vecteur contenant les $x_i^{(n)}$, $W^{(n+1)}$ la matrice contenant les $w_{ij}^{(n+1)}$, et $B^{(n+1)}$ le vecteur contenant les $b_j^{(n+1)}$; la fonction $f^{(n+1)}$ s'applique alors à chaque élément du vecteur $W^{(n+1)} X^{(n)} + B^{(n+1)}$.

Une fois la propagation effectuée, la rétro-propagation s'effectue de manière similaire ; en effet, l'équation 2.28b peut s'écrire

$$\begin{pmatrix} \delta_1^{(n)} \\ \delta_2^{(n)} \\ \vdots \\ \delta_{d^{(n)}}^{(n)} \end{pmatrix} = f^{(n)'} \left(\begin{pmatrix} s_1^{(n)} \\ s_2^{(n)} \\ \vdots \\ s_{d^{(n)}}^{(n)} \end{pmatrix} \right) \odot \begin{pmatrix} w_{11}^{(n+1)} & w_{12}^{(n+1)} & \cdots & w_{1d^{(n+1)}}^{(n+1)} \\ w_{21}^{(n+1)} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ w_{d^{(n)}1}^{(n+1)} & \cdots & \cdots & w_{d^{(n)}d^{(n+1)}}^{(n+1)} \end{pmatrix} \begin{pmatrix} \delta_1^{(n+1)} \\ \delta_2^{(n+1)} \\ \vdots \\ \delta_{d^{(n+1)}}^{(n+1)} \end{pmatrix} \quad (2.34)$$

où " \odot " est le produit de Hadamard (produit termes à termes) ; de manière compacte, on a

$$\begin{cases} \Delta^{(N)} = f^{(N)'}(S^{(N)}) \odot \frac{\partial C}{\partial X^{(N)}} \\ \Delta^{(n)} = f^{(n)'}(S^{(n)}) \odot W^{(n+1)\top} \Delta^{(n+1)} \quad \text{si } n < N \end{cases} \quad (2.35)$$

où $\Delta^{(n)}$ est le vecteur contenant les $\delta_j^{(n)}$, et où la dérivée partielle de C par rapport à $X^{(N)}$ est le vecteur des dérivées partielles de C par rapport à chaque composante de $X^{(N)}$. La mise à jour des paramètres se fait alors grâce à

$$\begin{cases} W^{(n)} \rightarrow W^{(n)} - \eta \nabla_W^{(n)} C = W^{(n)} - \eta X^{(n-1)} \Delta^{(n)\top} \\ B^{(n)} \rightarrow B^{(n)} - \eta \nabla_B^{(n)} C = B^{(n)} - \eta \Delta^{(n)} \end{cases} \quad (2.36)$$

où $\Delta^{(n)\top}$ est le vecteur transposé de $\Delta^{(n)}$; $\nabla_W C$ et $\nabla_B C$ sont respectivement les gradients de la fonction coût par rapport aux poids et aux biais.

La fonction coût en présence de facteur de régularisation (2.31) s'exprime pour la norme L_1 et L_2

$$\begin{cases} C' = C + \lambda \|W\|_1 \\ C' = C + \lambda \|W\|_2^2 \end{cases} \quad (2.37)$$

où W est le vecteur contenant tous les poids du réseau de neurones.

2.4.1.2 Mini-batch

L'implémentation présentée ici ne tient compte que d'une seule donnée d'entraînement (le vecteur X). Cependant, il a été expliqué dans la section précédente qu'il est préférable de mettre à jour les paramètres du modèle en tenant compte de plusieurs données à la fois (mais pas trop !), ce qui résulte en une évaluation plus précise de la fonction coût et de son gradient, tout en accélérant le processus d'optimisation. La manière naïve d'implémenter l'équation 2.30 serait d'effectuer une boucle sur les différentes données, mais là encore, la parallélisation de la librairie d'algèbre linéaire ne serait pas exploitée, et donc il est préférable de coder la propagation de manière matricielle également

$$\begin{pmatrix} x_1^{(n+1),1} & \cdots & x_1^{(n+1),n_D} \\ x_2^{(n+1),1} & \cdots & x_2^{(n+1),n_D} \\ \vdots & \ddots & \vdots \\ x_{d^{(n+1)}}^{(n+1),1} & \cdots & x_{d^{(n+1)}}^{(n+1),n_D} \end{pmatrix} = f^{(n+1)} \left(\begin{pmatrix} w_{11}^{(n+1)} & \cdots & w_{d^{(n+1)}1}^{(n+1)} \\ w_{12}^{(n+1)} & & \vdots \\ \vdots & \ddots & \vdots \\ w_{1d^{(n+1)}}^{(n+1)} & \cdots & w_{d^{(n+1)}d^{(n+1)}}^{(n+1)} \end{pmatrix} \begin{pmatrix} x_1^{(n),1} & \cdots & x_1^{(n),n_D} \\ x_2^{(n),1} & \cdots & x_2^{(n),n_D} \\ \vdots & \ddots & \vdots \\ x_{d^{(n)}}^{(n),1} & \cdots & x_{d^{(n)}}^{(n),n_D} \end{pmatrix} + \begin{pmatrix} b_1^{(n+1)} \\ b_2^{(n+1)} \\ \vdots \\ b_{d^{(n)}}^{(n+1)} \end{pmatrix} \right) \quad (2.38)$$

soit, de manière compacte

$$X^{(n+1)} = f^{(n+1)} (W^{(n+1)} X^{(n)} + B^{(n+1)}) \quad (2.39)$$

où cette fois $X^{(n)}$ est la matrice contenant les $x_i^{(n),1:n_D}$. L'équation 2.28b devient alors

$$\begin{pmatrix} \delta_1^{(n),1} & \cdots & \delta_1^{(n),n_D} \\ \delta_2^{(n),1} & \cdots & \delta_2^{(n),n_D} \\ \vdots & \ddots & \vdots \\ \delta_{d^{(n)}}^{(n),1} & \cdots & \delta_{d^{(n)}}^{(n),n_D} \end{pmatrix} = \frac{1}{n_D} f^{(n)'} \left(\begin{pmatrix} s_1^{(n),1} & \cdots & s_1^{(n),n_D} \\ s_2^{(n),1} & \cdots & s_2^{(n),n_D} \\ \vdots & \ddots & \vdots \\ s_{d^{(n)}}^{(n),1} & \cdots & s_{d^{(n)}}^{(n),n_D} \end{pmatrix} \right) \odot \begin{pmatrix} w_{11}^{(n+1)} & \cdots & w_{1d^{(n+1)}}^{(n+1)} \\ w_{21}^{(n+1)} & & \vdots \\ \vdots & \ddots & \vdots \\ w_{d^{(n+1)}1}^{(n+1)} & \cdots & w_{d^{(n+1)}d^{(n+1)}}^{(n+1)} \end{pmatrix} \begin{pmatrix} \delta_1^{(n+1),1} & \cdots & \delta_1^{(n+1),n_D} \\ \delta_2^{(n+1),1} & \cdots & \delta_2^{(n+1),n_D} \\ \vdots & \ddots & \vdots \\ \delta_{d^{(n+1)}}^{(n+1),1} & \cdots & \delta_{d^{(n+1)}}^{(n+1),n_D} \end{pmatrix} \quad (2.40)$$

soit, de manière compacte

$$\begin{cases} \Delta^{(N)} = \frac{1}{n_D} f^{(N)'} (S^{(N)}) \odot \frac{\partial C}{\partial X^{(N)}} \\ \Delta^{(n)} = \frac{1}{n_D} f^{(n)'} (S^{(n)}) \odot W^{(n+1)\top} \Delta^{(n+1)} \quad \text{si } n < N \end{cases} \quad (2.41)$$

où cette fois $\Delta^{(n)}$ est la matrice contenant les $\delta_i^{(n),1:n_D}$, et où la dérivée partielle de C par rapport à $X^{(N)}$ est toujours la matrice des dérivées partielles de C par rapport à chaque composante de $X^{(N)}$.

2.4.1.3 Mise en forme des données

Comme nous l'avons vu plus haut, afin de maximiser les performances d'une architecture donnée, il est préférable de bien conditionner les données d'entrée ainsi que les valeurs cibles. Face à la complexité du modèle que constitue un réseau de neurones, un critère d'optimalité pour le conditionnement des données est impossible à fournir; ainsi, seules quelques règles empiriques faisant office de bonnes pratiques sont communément admises comme nécessaires (dans la plupart des cas) pour pouvoir valider ou rejeter une architecture donnée.

Parmi ces bonnes pratiques, il est conseillé de **centrer et réduire** chaque donnée d'entrée $X^{(0)}$ et de sortie $X^{(N)}$, ce qui donne la transformation suivante

$$\begin{cases} X'^{(0)} = \frac{X^{(0)} - \bar{X}^{(0)}}{\sigma(X^{(0)})} \\ X'^{(N)} = \frac{X^{(N)} - \bar{X}^{(N)}}{\sigma(X^{(N)})} \end{cases} \quad (2.42)$$

où $\bar{X}^{(0)}$ est la moyenne de $X^{(0)}$ et $\sigma(X^{(0)})$ son écart-type. Il peut parfois être également pertinent de centrer les données sans les réduire, mais plutôt en les confinant dans l'intervalle $[-1; 1]$; il faut alors les normaliser par leur valeur maximale :

$$\begin{cases} X'^{(0)} = \frac{X^{(0)} - \bar{X}^{(0)}}{\max(|X^{(0)} - \bar{X}^{(0)}|)} \\ X'^{(N)} = \frac{X^{(N)} - \bar{X}^{(N)}}{\max(|X^{(N)} - \bar{X}^{(N)}|)} \end{cases} \quad (2.43)$$

Dans les deux cas, les valeurs relatives des données sont conservées.

2.4.1.4 Toolbox utilisées

Les langages de programmation modernes incluent désormais tous des toolbox d'intelligence artificielle, et plus particulièrement de réseau de neurones. Nous allons voir ici comment créer un modèle de réseau de neurones à propagation avant en MATLAB et en Python. Le modèle créé sera à deux couches cachées, contenant respectivement 80 et 20 neurones, utilisant d'abord la fonction non-linéaire tangente hyperbolique (entre la couche d'entrée et la première couche cachée), puis ReLU (entre la première couche cachée et la seconde); il n'y a pas de fonction non-linéaire entre la seconde couche cachée et la couche de sortie. Les valeurs des hyper-paramètres (le nombre de couches cachées et de neurones par couche, le taux d'apprentissage η , le *momentum* μ , et le nombre d'*epochs*) sont données à titre d'exemple. L'importation des données ainsi que leur traitement ne sont pas montrés.

En MATLAB, la toolbox "Deep Learning" permet de rapidement créer des modèles de réseau de neurones à propagation avant ; huit lignes suffisent à créer le modèle décrit plus haut.

```

1 % crée le réseau de neurones à propagation avant avec nombre
2 % de neurones par couche cachée et algorithme d'optimisation
3 net = feedforwardnet([80 20] , 'traingdm') ;
4
5 % définit les fonctions non-linéaires
6 net.layers{1}.transferFcn = 'tansig' ; % tanh
7 net.layers{2}.transferFcn = 'poslin' ; % ReLU
8
9 % définit le learning rate, le momentum et le nombre d'epochs
10 net.trainParam.lr = 0.01 ;
11 net.trainParam.mc = 0.2 ;
12 net.trainParam.epochs = 250 ;
13
14 % effectue l'entraînement
15 [trained_net,tr] = train(net , Donnee_entree,Donnee_cible) ;
16
17 % effectue le test
18 Sortie = sim(trained_net , Donnee_test) ;

```

En python la toolbox "TensorFlow" permet également de créer des modèles performants en seulement quelques lignes de code ; pour cela, il est possible d'utiliser l'API "Keras".

```

1 import tensorflow as tf
2 from tensorflow.keras import optimizers
3 from tensorflow.keras.layers import Dense
4
5 # crée un réseau de neurones à propagation avant vierge
6 model = tf.keras.models.Sequential()
7
8 # définit les deux couches cachées
9 model.add(Dense(80 , activation = 'tanh'))
10 model.add(Dense(20 , activation = 'relu'))
11 # définit la couche de sortie
12 model.add(Dense(dim_sortie , activation = 'linear'))
13
14 # définit la fonction coût et l'algorithme d'optimisation
15 C = 'mean_squared_error'
16 optim = optimizers.SGD(learning_rate = 0.01 , momentum = 0.2)
17
18 # compile le modèle puis effectue l'entraînement
19 model.compile(loss = C , optimizer = optim)
20 model.fit(Donnee_entree,Donnee_cible , epochs = 250)
21
22 # effectue le test
23 Sortie = model.predict(Donnee_test)

```

2.4.2 Variantes de l'algorithme

L'algorithme qui vient d'être présenté a été développé il y a plusieurs années, et ses accomplissements ont poussé la communauté scientifique à perfectionner les modèles et les algorithmes d'optimisations associés, rendant les modèles plus généraux et/ou mieux adaptés aux problèmes qu'ils doivent traiter, et les algorithmes d'optimisation plus rapides, robustes, ...

2.4.2.1 Variantes du modèle

De nombreuses variations du modèle initial existent, et chacune d'entre elles présente ses avantages et inconvénients selon la situation. Toutes ces variations reposent sur le même concept de base : appliquer des transformations linéaires paramétrées puis non-linéaires sur les données d'entrées afin de produire des sorties les moins surprenantes possible, sur la base de données d'entraînement. Les variations les plus fréquemment utilisées sont donc des variations concernant les connexions du réseau de neurones, c'est-à-dire la structure de son graphe. En effet, comme nous l'avons vu, un réseau de neurones à propagation avant est un graphe orienté acyclique complet par couche, i.e. où chaque neurone d'une couche donnée est connecté à tous les neurones de la couche suivante. Il est donc possible de modifier ce modèle en rendant le graphe cyclique et/ou incomplet par exemple.

En rendant le graphe cyclique, mais toujours complet par couche, où certains neurones sont connectés à eux-mêmes (aux instants précédents), le modèle devient alors un réseau de neurones récurrent. Cette architecture est très utilisée pour analyser des données temporelles [84]. En rendant le graphe incomplet, mais toujours acyclique, où chaque neurone d'une couche donnée n'est connecté qu'à une partie des neurones de la couche suivante, le modèle devient alors un réseau de neurones convolutionnel. Cette architecture est très utilisée pour l'analyse d'images (reconnaissance ou classification) ou encore pour traitement du langage [85]. La méthode de *dropout* peut être considérée comme une manière de rendre le graphe incomplet (toujours acyclique) en faisant en sorte que certains neurones ne soient connectés à aucun autre. Un bestiaire de la plupart des architectures utilisées de nos jours est présenté en annexe B.

2.4.2.2 Variantes de l'algorithme d'optimisation

L'un des problèmes de l'algorithme de mise à jour 2.18 est qu'il lui est impossible de sortir d'un minimum local une fois tombé dedans. L'idée, inspirée de la physique, est donc de conférer à la trajectoire d'optimisation une "inertie" (*momentum*) qui lui permettra de pouvoir "escalader" les "vallées" de la fonction coût, dans l'espoir de converger vers un minimum plus intéressant [86]. En notant $\theta_{(i)}^{(n)}$ les paramètres de la couche n à optimiser à l'itération i et $\Theta_{(i)}^{(n)}$ le vecteur contenant ces paramètres, une première amélioration possible de mise à jour est donc

$$\begin{cases} v_{(i+1)}^{(n)} = \mu v_{(i)}^{(n)} - \eta \nabla_{(i)}^{(n)} C \\ \Theta_{(i+1)}^{(n)} = \Theta_{(i)}^{(n)} + v_{(i+1)}^{(n)} \end{cases} \quad (2.44)$$

où $v_{(i)}$ est le vecteur "vitesse", $\nabla_{(i)}^{(n)} C$ est le vecteur gradient à l'itération i , c'est-à-dire le vecteur des dérivées de C par rapport aux $\theta_{(i)}^{(n)}$. Le paramètre μ (qui doit être compris entre zéro et un) constitue donc un nouvel hyper-paramètre à ajuster pour maximiser les performances de l'algorithme d'optimisation.

De plus, comme il a été expliqué dans la section précédente, l'algorithme d'optimisation de descente du gradient consiste à partir d'un point de départ dans l'espace des paramètres, choisir une direction de mise à jour, puis effectuer cette mise à jour dans cette direction avec un certain pas (le taux d'apprentissage η), et potentiellement une certaine inertie (le *momentum* μ). Dans l'algorithme de descente du gradient, la direction choisie est celle opposée au gradient de la fonction coût. L'algorithme de descente du **gradient conjugué** cherche à améliorer cette direction d'optimisation ; cette dernière est alors choisie de manière à être conjuguée aux directions des itérations précédentes, la conjugaison étant une forme d'orthogonalité. Ainsi, toujours en notant $\Theta_{(i)}^{(n)}$ le vecteur des paramètres à l'itération i , la formule 2.36 devient

$$\begin{cases} D_{(i+1)}^{(n)} = \beta_{(i)}^{(n)} D_{(i)}^{(n)} - \nabla_{(i)}^{(n)} C \\ \Theta_{(i+1)}^{(n)} = \Theta_{(i)}^{(n)} + \eta D_{(i+1)}^{(n)} \end{cases} \quad (2.45)$$

où $D_{(i)}$ est le vecteur de direction, $\nabla_{(i)} C$ le vecteur gradient à l'itération i , et $\beta_{(i)}$ un coefficient à déterminer à chaque itération. La détermination de ce dernier est assez libre a priori, mais l'une des formules les plus utilisées est celle de Fletcher-Reeves [87]

$$\beta_{(i)}^{(n)} = \frac{\|\nabla_{(i)}^{(n)} C\|}{\|\nabla_{(i-1)}^{(n)} C\|} \quad (2.46)$$

Lorsque $\beta_{(i)}$ est constant (égal à μ/η), on retrouve la méthode de descente du gradient avec *momentum*, et en particulier, si $\beta_{(i)} = 0$, alors on retrouve la méthode de descente du gradient classique.

L'algorithme de réseau de neurones à propagation avant ayant été présenté en détail, ce dernier peut maintenant être mis à contribution dans l'analyse de données concernant la propagation lumineuse dans les systèmes fibrés.

Chapitre 3

État de l'art et vérification

*"There are two kinds of ultrashort-laser pulse measurement :
those with independent checks and those that are wrong."*

– John M. DUDLEY, *correspondance avec Rick TREBINO, 1999*

L'objectif des travaux qui vont être présentés est d'obtenir un modèle prédictif pertinent capable d'établir une relation entre le profil d'intensité spectrale (sans la phase) et le maximum d'intensité temporelle associé. En effet, les évènements extrêmes, et plus particulièrement les vagues scélérates, ont fait l'objet de nombreuses études ces dernières années aussi bien dans le domaine de l'optique que de l'océanographie [88]. Ces évènements sont, par définition, rares et imprévisibles (avec les outils actuels), mais également compliqués à mesurer à cause de la plage dynamique élevée nécessaire à capturer l'ensemble de la dynamique.

3.1 Motivation

3.1.1 Mesures expérimentales

De nombreux outils et techniques expérimentales permettent de mesurer des profils spectraux et/ou temporels à l'issue d'une propagation dans un système fibré ; ils ont cependant tous leurs limitations. Les analyseurs de spectre optique (OSA) permettent d'obtenir des informations sur le spectre de l'onde mesurée. Un spectrographe classique envoie la lumière sur un réseau de diffraction permettant ainsi de séparer les longueurs d'onde qui constituent l'impulsion à analyser (à la résolution près), et ainsi mesurer l'intensité de chacune d'entre elles à l'aide d'un photodétecteur. Un interféromètre de Fabry-Pérot, lui, fait passer l'onde dans une cavité résonnante créée par deux miroirs parallèles. L'espacement entre ces deux miroirs permet de sélectionner la fréquence de résonance de cette cavité et donc la fréquence de la composante spectrale avec le maximum de transmittance ; l'intensité de cette composante spectrale est alors détectée. Cependant, bien que ces méthodes puisse être configurées pour avoir une résolution et une plage dynamique très élevées, elles reposent généralement sur un moyennage temporel, et ne sont donc pas adaptées à la mesures d'instabilités coup par coup.

Des méthodes plus sophistiquées telles que la transformée de Fourier dispersive (DFT) permettent d'obtenir des mesures rapides en temps réel, mais la plage dynamique de ces mesures ne dépasse pas 25 dB, et seule l'intensité spectrale (sans la phase) est observable, ce qui ne permet pas (a priori) de reconstruire l'information désirée sur les solitons.

Les systèmes de type lentille temporelle permettent d'obtenir des mesures très précises directement dans le domaine temporel, ce qui permet d'avoir accès à toute l'information désirée sur les solitons; le principe est d'appliquer une phase quadratique temporelle au signal mesuré de manière analogue à une véritable lentille qui applique une phase quadratique spatiale. Cependant, ce protocole expérimental est très compliqué à mettre en œuvre et a une bande passante limitée, si bien que peu de mesures sont réalisées avec cette technique en pratique.

Néanmoins, il a récemment été montré que les méthodes spectrographiques pouvaient être adaptées pour effectuer des mesures en temps réel avec des plages dynamiques de l'ordre de 60 dB [89], ouvrant la possibilité d'effectuer des mesures spectrales coup par coup de manière relativement simple. Ainsi, la problématique à laquelle nous faisons face est d'être capable d'extraire des informations temporelles n'ayant accès qu'à l'intensité spectrale, cette dernière étant la seule quantité facilement mesurable.

3.1.2 Simulation numérique

Comme nous l'avons vu dans le chapitre précédent, le réseau de neurones à propagation avant est un modèle assez simple et très performant qui permet d'effectuer un grand nombre de tâches. Parmi elles, la régression non-linéaire est d'un intérêt tout particulier en physique puisque l'aboutissement d'une telle régression fournit un modèle prédictif rapide et robuste au bruit permettant d'établir une correspondance entre des variables d'entrée (observables/mesurables) et des variables de sortie (difficilement mesurables); un tel modèle peut donc prétendre à se substituer à un modèle physique (connu ou non) pour ce qui est de sa capacité prédictive. Cependant, l'inconvénient majeur des réseaux de neurones est leur incapacité (en l'état) à fournir des résultats interprétables, allant à l'encontre de ce qui est désiré par les chercheurs.

L'utilisation d'un réseau de neurones à propagation avant en condition d'apprentissage supervisé nécessite d'avoir des données d'entraînement complètes (entrées et cibles) grâce auxquelles on peut optimiser les paramètres du réseau. Dans notre cas, ces données sont obtenues par intégration numérique de l'équation de Schrödinger non-linéaire (ESNL), parfois généralisée (ESNLG). L'intégration numérique peut être faite par un algorithme d'intégration d'équation différentiel ordinaire (O.D.E.) classique (Runge-Kutta, ...) ou encore grâce à la méthode *split-step Fourier*. Pour intégrer cette équation, il faut donner les différents paramètres physiques du système laser, à savoir : la longueur L de la fibre en mètres, les coefficients de dispersion β_k en fs^k/mm , le coefficient non-linéaire γ en $\text{W}^{-1}.\text{m}^{-1}$, la longueur d'onde centrale de l'impulsion initiale λ_0 en nm, sa puissance crête P_0 en W, et sa largeur à mi-hauteur (FWHM) en ps; des paramètres supplémentaires peuvent être ajoutés selon le système fibré considéré.

L	β_2	...	β_k	γ	λ_0	P_0	FWHM
[m]	[fs^2/mm]		[fs^k/mm]	[$\text{W}^{-1}.\text{m}^{-1}$]	[nm]	[W]	[ps]

TABLE 3.1 – Paramètres physiques d'une simulation de système fibré : paramètres de la fibre à gauche et paramètres de l'impulsion d'entrée à droite

En plus de fournir les paramètres physiques, il faut également bien choisir les paramètres numériques de la simulation afin que les résultats obtenus soient pertinents. Les simulations faites visant à intégrer l'ESNL à une dimension, il faut spécifier la grille spatiale et temporelle, la grille fréquentielle en découlant naturellement. Ainsi, pour chaque simulation, une étude préalable de convergence doit être effectuée afin de s'assurer que les résultats obtenus approximativement au mieux la réalité.

3.2 Instabilité de modulation et supercontinuum

Dans cette section, les résultats présentés sont ceux issus de la littérature récente concernant la prédiction des maxima d'intensité temporelle en régime d'instabilité de modulation et de supercontinuum. Ces résultats ont été obtenus en 2018 [89] et 2020 [90], par l'équipe de recherche de l'université de Tampere (Finlande) et les premiers objectifs de ma thèse ont été de reproduire ces résultats à l'aide du code de réseau de neurones que j'ai moi-même développé en MATLAB en suivant la démarche expliquée dans le chapitre précédent.

3.2.1 Instabilité de modulation

Les premiers résultats ont concerné l'étude de fibres laser en régime d'instabilités de modulation impulsionnelle. Comme expliqué dans le chapitre 1, l'instabilité de modulation est un phénomène dû à l'interaction entre dispersion et non-linéarité, qui cause l'amplification de certaines fréquences; en cas de dispersion anormale (les fréquences plus élevées vont plus vite), une impulsion de durée finie exhibera au bout d'un certain temps une instabilité de modulation (impulsionnelle), ce qui générera des profils d'intensité spectrale et temporelle très structurés et complexes. Les auteurs de [89] ont alors mis en place un système de mesure à haute plage dynamique (environ 60 dB) permettant d'observer précisément ces structures du profil d'intensité spectrale, ce qui a permis de corroborer les résultats de simulation numérique avec des mesures en temps réel pour la première fois (figure 3.1).

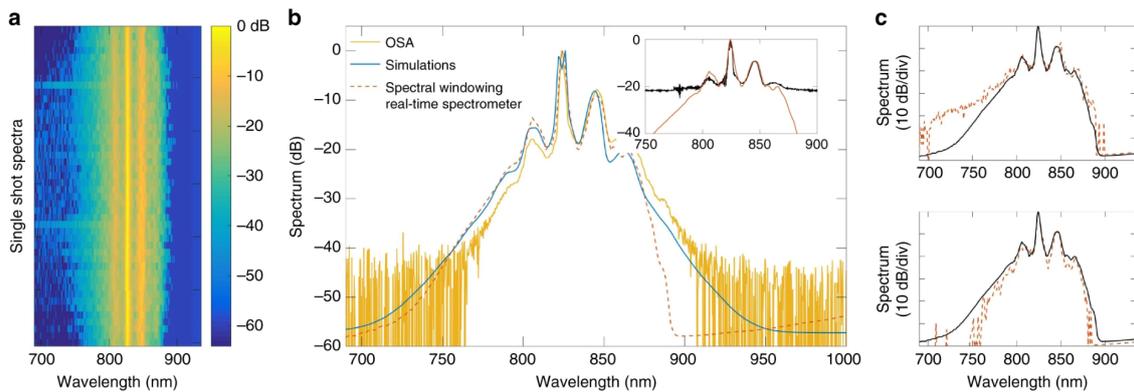


FIGURE 3.1 – Spectres MI enregistrés expérimentalement avec une plage dynamique élevée. (a) Série de 60 spectres enregistrés empilés dans le sens vertical pour illustrer les fluctuations d'un coup à l'autre observées avec le MI. (b) Spectre moyen mesuré avec la technique en temps réel à plage dynamique élevée (pointillé rouge), spectre moyen mesuré avec un analyseur de spectre optique (jaune), et spectre moyen simulé (bleu). L'encadré montre la comparaison entre la mesure spectrale avec une plage dynamique de 60 dB (pointillé rouge) et celle utilisant une DFT conventionnelle avec seulement 22 dB de plage dynamique (noir). (c) Deux spectres de sélectionnés à partir d'expériences (pointillé rouge) comparés au spectre expérimental moyen (noir). Figure 6 de [89]

Dans ces travaux, le système étudié est excité par une impulsion sécante hyperbolique de quelques picosecondes de FWHM dans une fibre à cristaux photoniques (PCF). L'utilisation d'une telle fibre est motivée par le fait que ces dernières sont spécialement conçues pour diminuer l'aire du mode spatial de l'onde \mathcal{A}_m , et ainsi augmenter l'influence du coefficient non-linéaire γ , stimulant donc l'émergence des phénomènes non-linéaires. Les données d'entraîne-

ment (paires entrées/sorties) sont obtenues par intégration numérique de l'ESNLG (1.48) où les effets dispersifs ont été retenus jusqu'à l'ordre six, et en l'absence de pertes

$$\frac{\partial A}{\partial z} = i \left(\sum_{k=2}^6 \frac{i^k \beta_k}{k!} \frac{\partial^k A}{\partial T^k} + \gamma \left(1 + \frac{i}{\omega_0} \frac{\partial}{\partial T} \right) \left(\int_0^{+\infty} R(T') |A(z, T - T')|^2 dT' A \right) \right) \quad (3.1)$$

avec $f_R = 0,18$. Les caractéristiques du système sont présentées dans le tableau suivant :

L	β_2	β_3	β_4	β_5	β_6	γ	λ_0	P_0	FWHM
[m]	[fs ² /mm]	[fs ³ /mm]	[fs ⁴ /mm]	[fs ⁵ /mm]	[fs ⁶ /mm]	[W ⁻¹ .m ⁻¹]	[nm]	[W]	[ps]
0,68	-10,3	47,4	23,5	-117	-907	0,1	825	175	3

TABLE 3.2 – Paramètres physiques des simulations en régime d'instabilité de modulation stimulée par une impulsion sécante hyperbolique dans une fibre à cristaux photoniques (PCF) : paramètres de la fibre à gauche et paramètres de l'impulsion d'entrée à droite.

Avec ces caractéristiques, le système présente une dispersion anormale forte, si bien que les profils spectraux et temporels observés sont caractéristiques du phénomène d'instabilité de modulation.

L'objectif est donc d'obtenir un modèle prédictif pertinent capable d'établir une relation entre le profil d'intensité spectrale (sans la phase) et le maximum d'intensité temporelle associé, i.e. à la même distance z dans la fibre. Il a été montré [89] qu'un réseau de neurones à deux couches cachées comportant respectivement 30 et 10 neurones et avec des fonctions non-linéaires en tangente hyperbolique (schématisé figure 3.2) était capable de fournir une telle relation avec un haut niveau de confiance.

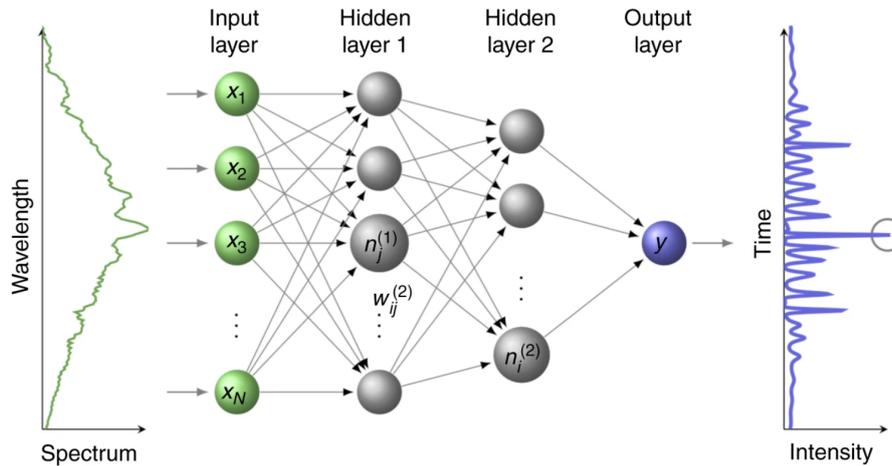


FIGURE 3.2 – Schéma du réseau de neurones utilisé pour analyser les données de simulation d'instabilité de modulation impulsionnelle. Le profil d'intensité spectrale est en entrée du réseau de neurones (à gauche), puis est transformé par les couches successives, jusqu'à aboutir à la prédiction du maximum d'intensité temporelle (à droite). Figure 2 de [89]

Afin de mesurer les performances du réseau, les auteurs ont affiché la prédiction de ce dernier en fonction de la valeur qu'il aurait dû prédire (i.e. la valeur issue de simulation numérique). Le nuage de points obtenu est censé être aligné sur la diagonale ; idéalement, il faudrait

que le réseau prédise exactement la valeur du maximum d'intensité temporelle pour tous les spectres. Bien sûr, la réalité diverge du cas idéal et le nuage de points n'est pas parfaitement aligné sur la diagonale ; les points sont regroupés autour de celle-ci avec un certain étalement plus ou moins grand.

Lorsque la variable prédite est discrète, il est commun d'afficher les performances du réseau dans une matrice de confusion [91]. Or pour une variable continue telle que celle qui nous intéresse, il est impossible d'obtenir une telle matrice. Ainsi, un histogramme est construit à partir du nuage de points, afin d'obtenir l'équivalent d'une matrice de confusion pour une variable continue. Les points contenus dans une même case carrée de longueur fixée sont moyennés puis normalisés afin de créer une carte de densité ; la longueur est choisie pour que chaque case contienne assez de points tout ayant une carte suffisamment définie. L'étalement des points autour de la diagonale est quantifié par le coefficient de corrélation de Pearson noté ρ ; lorsque ce dernier vaut un, cela signifie que la prédiction est à tous les coups égale au résultat attendu, et lorsqu'il vaut zéro, cela signifie que la prédiction n'est pas plus performante qu'un tirage aléatoire. Un tel histogramme a été obtenu dans l'article [89] comme le montre la figure 3.3.

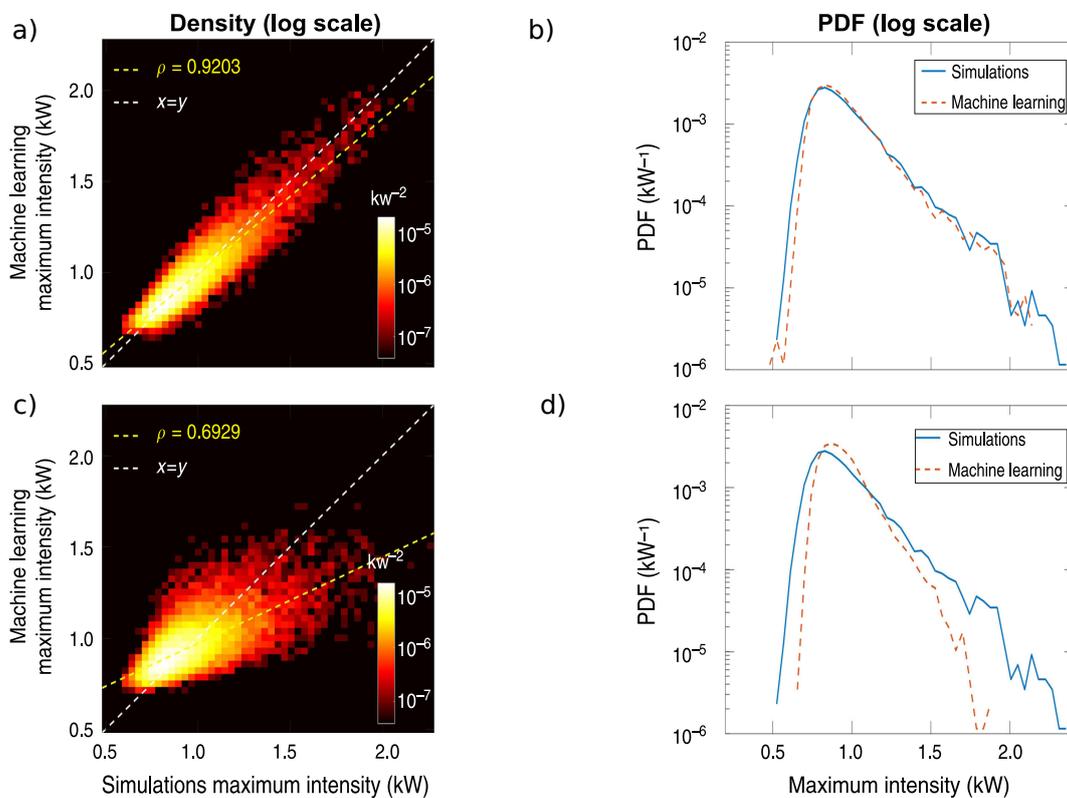


FIGURE 3.3 – Analyse par apprentissage supervisé de spectres MI simulés (30000 données d'entraînement et 20000 pour le test). a) et c) montrent l'intensité maximale prédite par le réseau de neurones en fonction de la valeur à prédire, celle-ci étant issues de simulations. Le panneau supérieur (respectivement inférieur) correspond à la troncature des données spectrales d'entrées à 60 dB (respectivement 25 dB). Les résultats en a) et c) sont présentés sous la forme d'un histogramme en échelle logarithmique qui montre la densité normalisée des points regroupés dans des classes de surface constante. La ligne blanche en pointillés marque la correspondance 1 à 1 entre l'intensité maximale (puissance de crête instantanée) prédite par le réseau et la valeur à prédire. b) et d) montrent les densité de probabilité (PDF) des intensités temporelles maximales prédites par l'algorithme d'apprentissage automatique (ligne rouge) comparée aux PDF calculées à partir des données temporelles simulées (ligne bleue). Figure 3 de [89]

Ces résultats ont pu être **reproduits fidèlement** grâce au code que j'ai développé en MATLAB (figure 3.4); les quelques différences de performances de mon algorithme (légèrement meilleures pour les données tronquées à 60 dB et légèrement moins bonnes pour les données tronquées à 25 dB) sont très certainement dues à la différence d'algorithme d'optimisation utilisé. En effet, dans l'article original, la toolbox MATLAB été utilisée avec la méthode du gradient conjugué avec mise à jour de Fletcher-Reeves (2.45), alors que mon implémentation n'utilise qu'une simple descente stochastique du gradient avec *momentum* (2.44), a priori plus rudimentaire, mais qui s'avère être suffisante pour le problème traité ici; bien qu'étant censée être plus performante en général, il se peut que dans ce cas particulier la méthode du gradient conjugué sous-performe quelque peu, ou bien encore que certains hyper-paramètres (le taux d'apprentissage initial par exemple) n'aient pas été tout à fait suffisamment ajustés.

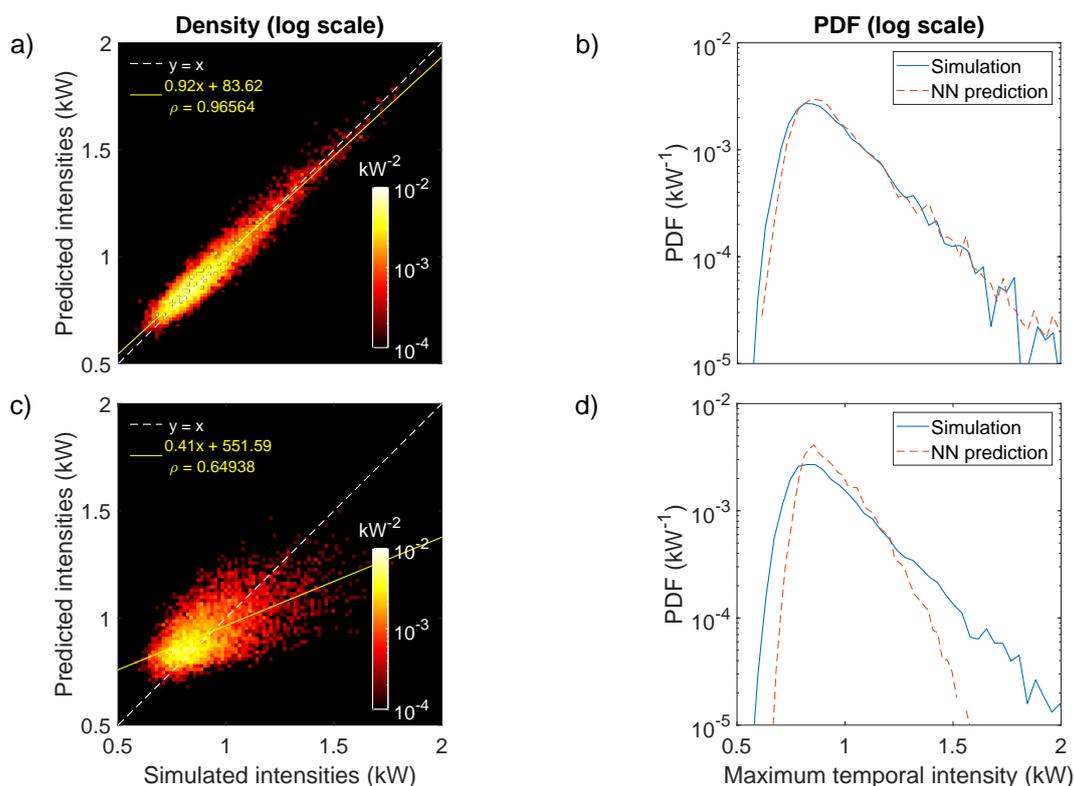


FIGURE 3.4 – Analyse par apprentissage supervisé de spectres MI simulés (20000 données d'entraînement et 10000 pour le test). Cette figure est la reproduction des résultats de la figure 3.3 à l'aide de mon code de réseau de neurones. Des performances supérieures sont obtenues pour les données tronquées à 60 dB (figures a) et b)), et des performances légèrement inférieures pour les données tronquées à 25 dB (figures c) et d)).

Mes résultats sont néanmoins très proches des résultats originaux, confirmant la bonne utilisation de l'algorithme d'intégration numérique (choix des paramètres de l'intégration et étude de convergence), la bonne implémentation de l'algorithme de réseau de neurones (propagation et rétro-propagation), ainsi que la bonne mise en place du protocole de test du modèle (histogramme et densité). Il est à noter qu'une reproduction exacte des résultats est illusoire, puisqu'à chaque nouvel entraînement, de nouveaux paramètres initiaux du réseau (poids et biais) sont tirés aléatoirement, ce qui altère la convergence finale de l'algorithme d'optimisation, et donc le résultat issu de cette convergence (voir figure 2.8). Afin de compléter cette étape de validation, l'état de l'art effectué en régime supercontinuum a également été reproduit.

3.2.2 Supercontinuum

Les résultats précédents ayant été un succès, l'étude d'un système fibré générant des profils spectraux et temporels plus structurés et chaotiques a été menée par la même équipe de recherche de l'université de Tampere ; cette étude a donc été faite dans le régime de supercontinuum [90].

Comme expliqué dans le chapitre 1, le régime supercontinuum est atteint lorsque les effets dispersifs d'ordre élevés (jusqu'à l'ordre dix environ) ainsi que l'effet Raman sont stimulés. Dans ce régime, la fission solitonique a pour effet de casser l'impulsion initiale en une série de solitons fondamentaux de très courte durée, de l'ordre de quelques dizaines de femtosecondes ; ces solitons étant très fins, ils se manifestent par un élargissement spectral de l'onde qu'ils constituent. Cette situation physique ne diffère pas de celle présentée dans la section précédente ; l'émergence d'une dynamique supercontinuum vient simplement du fait que l'impulsion se propage désormais dans une fibre suffisamment longue pour que l'effet Raman soit notable.

Ainsi, comme précédemment, les données ont été générées par intégration numérique de l'ESNLG (1.48) avec des termes dispersifs jusqu'à l'ordre sept et sans atténuation :

$$\frac{\partial A}{\partial z} = i \left(\sum_{k=2}^7 \frac{i^k \beta_k}{k!} \frac{\partial^k A}{\partial T^k} + \gamma \left(1 + \frac{i}{\omega_0} \frac{\partial}{\partial T} \right) \left(\int_0^{+\infty} R(T') |A(z, T - T')|^2 dT' A \right) \right) \quad (3.2)$$

avec $f_R = 0,18$. Les caractéristiques du système sont présentées dans le tableau suivant :

L	β_2	β_3	β_4	β_5	β_6	β_7	γ	λ_0	P_0	FWHM
[m]	[fs ² /mm]	[fs ³ /mm]	[fs ⁴ /mm]	[fs ⁵ /mm]	[fs ⁶ /mm]	[fs ⁷ /mm]	[W ⁻¹ .m ⁻¹]	[nm]	[W]	[ps]
0,85	-12,4	89,4	25,4	-701	2280	-2210	0,1	810	400	2

TABLE 3.3 – Paramètres physiques des simulations en régime de supercontinuum : paramètres de la fibre à gauche et paramètres de l'impulsion d'entrée à droite

L'observation des données a permis de mettre en évidence l'existence d'une relation entre les caractéristiques spectrales et temporelles des solitons Raman (solitons les plus décalés vers le rouge) émergeant au cours de cette propagation ; la transformée de Fourier effectuée sur la partie spectre au-delà de 950 nm semble permettre d'isoler le pic temporel associé à ce soliton Raman comme le montre la figure 3.5.

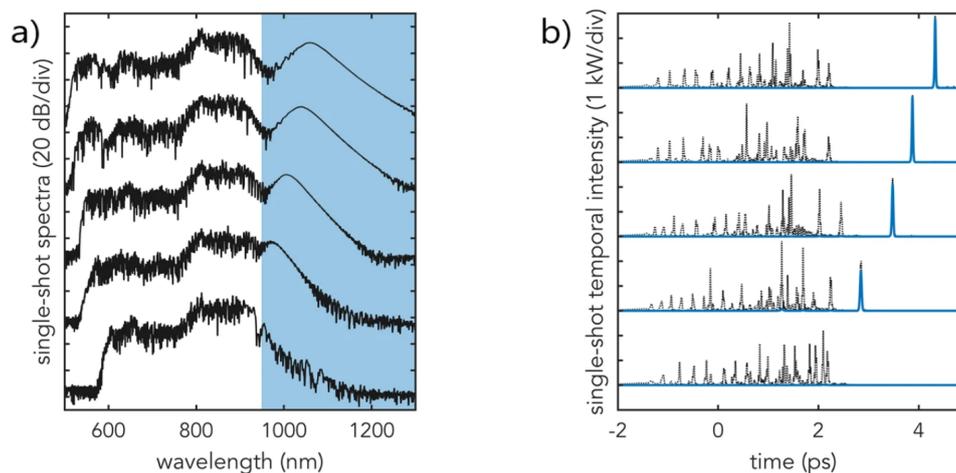


FIGURE 3.5 – Sélection de 5 spectres de supercontinuum aléatoires et profils d'intensité temporelle correspondants (noir). La courbe bleue en b) montre l'intensité temporelle du soliton "scélérat" associé aux composantes spectrales filtrées avec un filtre passe-bas (zone bleue) en a) tandis que la ligne pointillée noire montre l'intensité complète dans le domaine temporel. Figure 1 de [90]

Cependant, la transformée de Fourier nécessite de connaître l'intensité et la phase spectrale de l'impulsion, or la méthode de DFT ne permettant d'avoir accès qu'à l'intensité spectrale, des métriques simples ont été cherchées afin de prédire les caractéristiques temporelles (position ou FWHM par exemple) du soliton Raman. Cette recherche s'est avérée infructueuse, si bien que les chercheurs se sont tournés vers les réseaux de neurones. Il a ainsi été montré qu'un réseau de neurones bien choisi était capable de prédire avec une très bonne confiance ($\rho > 0,9$) l'intensité, la largeur à mi-hauteur (FWHM), et la position du pic temporel associé au soliton Raman (figures 3.6 et 3.7).

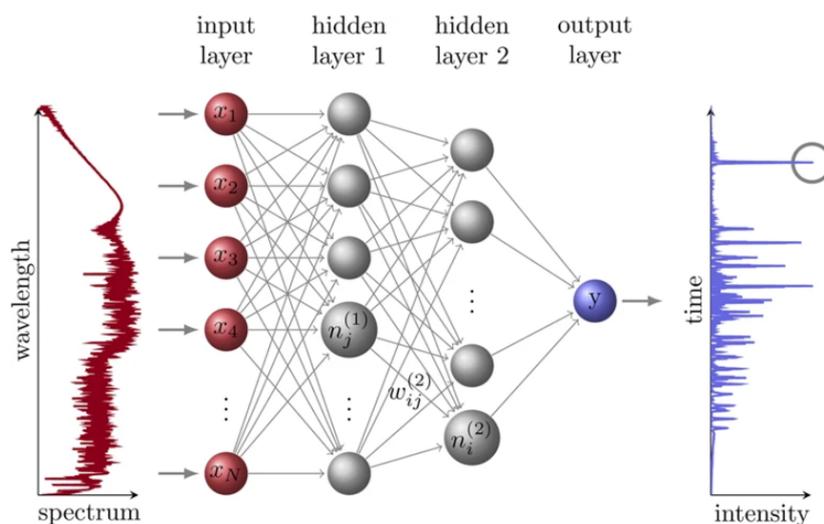


FIGURE 3.6 – Schéma du réseau de neurones utilisé dans ce travail. (a) L'entrée du réseau est un vecteur d'intensité spectrale de supercontinuum $X_n = [x_1, x_2, \dots, x_N]$ conduisant à la sortie du réseau y qui correspond à la puissance de crête, à la durée ou au retard temporel des solitons errants. Le réseau se compose de deux couches cachées et d'un seul neurone de sortie. Figure 2 de [90]

La méthode de caractérisation des performances du réseau est la même que dans la section précédente; un histogramme est construit à partir du nuage de points affichant les prédictions du réseau en fonction des valeurs qu'il est censé prédire, puis le coefficient de corrélation de Pearson sert d'indicateur de l'étalement des prédictions autour de la diagonale, et donc d'indicateur des performances du réseau.

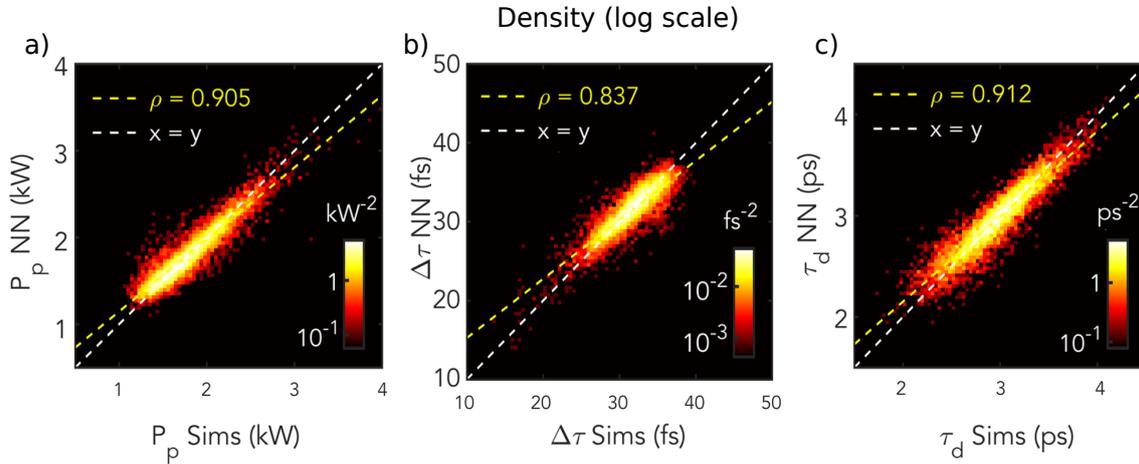


FIGURE 3.7 – Résultats montrant la prédiction des caractéristiques des solitons scélérats par le réseau de neurones pour un ensemble de 10000 SC générés avec des impulsions d'entrée identiques (à l'exception de la graine de bruit). (a) compare l'intensité maximale prédite du soliton le plus décalé vers le rouge avec la valeur exacte des profils simulés dans le domaine temporel. (b) compare la durée prédite $\Delta\tau$ des solitons avec la valeur exacte des simulations. (c) compare le retard prédit τ_d des solitons avec la valeur exacte des simulations. Pour toutes les sous figures, les histogrammes en échelle logarithmique montrent la densité normalisée des points groupés en cases de surface constante. Dans chaque cas, la ligne blanche en pointillés marque la correspondance un à un, et la ligne jaune en pointillés est un ajustement linéaire aux prédictions avec le coefficient de corrélation de Pearson ρ . Figure 3 de [90]

Ces résultats ont été reproduits grâce au code de réseau de neurones que j'ai développé en MATLAB. L'architecture utilisée diffère légèrement de l'architecture originale puisqu'il s'agit un réseau de neurones à trois couches cachées contenant respectivement 80, 20 et 10 neurones au lieu de n'avoir que deux couches cachées contenant 80 et 20 neurones; cette architecture a été trouvée après quelques essais, afin d'optimiser les performances du réseau. Les trois propriétés temporelles (maximum d'intensité temporelle, FWHM, et décalage temporel) ont été prédites en utilisant à chaque fois la même architecture et les mêmes hyper-paramètres. Les résultats obtenus sont bien similaires à ceux présentés dans l'article original, avec cependant une nette amélioration de la corrélation obtenue pour la FWHM $\Delta\tau$ (figure 3.8 b)). En plus du changement d'architecture, cela peut provenir de deux facteurs principaux : la manière de calculer cette FWHM, et l'algorithme d'optimisation utilisé.

En effet, la détermination algorithmique de la FWHM de profils si structurés est complexe, puisqu'il faut réussir à déterminer ce qui définit un pic, et ce qui permet de le distinguer des autres (deux pics quasiment confondus doivent être discernés l'un de l'autre pour ne pas surestimer la largeur du pic d'intérêt); les deux critères permettant cela sont la hauteur et la prééminence limites en deçà desquelles on ignore un pic; la valeur calculée de la FWHM dépend alors du choix de ces deux critères. Par ailleurs, comme pour la section précédente, l'utilisation de l'algorithme de descente stochastique du gradient avec *momentum* plutôt que la descente du gradient conjugué avec mise à jour de Fletcher-Reeves a pu jouer sur les performances.

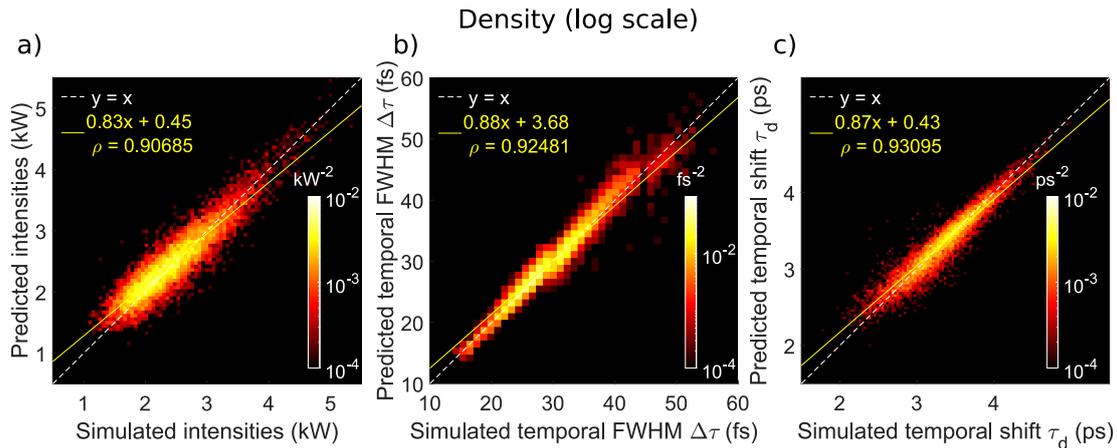


FIGURE 3.8 – Reproduction des résultats de la figure 3.7 avec mon code de réseau de neurones. Là encore, des performances similaires, et même légèrement supérieures sont obtenues avec mon code.

Le processus de formation du réseau est visible dans une interface visuelle que j'ai codé (figure 3.9) afin de pouvoir suivre les performances du réseau et ainsi arrêter le processus rapidement en cas de non-convergence avérée ou de sur-apprentissage par exemple.

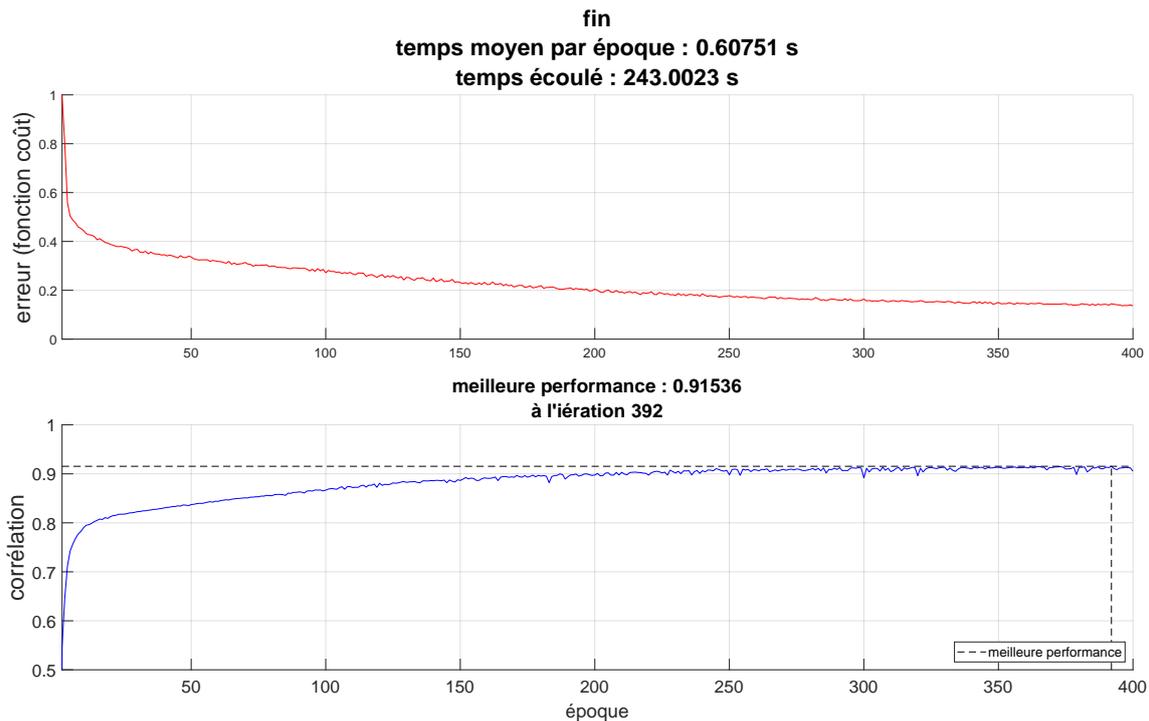


FIGURE 3.9 – Courbe d'évolution de la fonction coût (en haut) et du coefficient de corrélation de Pearson (en bas) au cours du processus d'optimisation qui a permis d'aboutir à la figure gauche de 3.8.

L'ensemble des données d'entraînement est séparé en deux groupes : les données servant à réellement former le réseau, et celles servant à estimer rapidement les capacités de ce dernier, afin de ne pas perdre de temps à former une architecture complètement inadaptée. Ce deuxième groupe de données constitue les données de validation, et ne sont pas utilisées pour la mise à jour des paramètres du réseau; elles ont le même rôle que les données de test mais ne sont

qu'un tout petit échantillon des données d'entraînement et ne fournissent qu'une estimation, potentiellement grossière, des performances de ce dernier. Mon interface montre donc l'évolution de la fonction coût C au cours des itérations d'optimisation (en temps réel), ainsi que la valeur du coefficient de corrélation de Pearson ρ calculé sur des données de validation ; la figure 3.9 montre en particulier l'évolution des performances du réseau lors de la prédiction du maximum d'intensité temporelle.

On observe tout d'abord une décroissance rapide de la fonction coût, qui se traduit par une augmentation tout aussi rapide du coefficient de corrélation. Arrivé à environ dix itérations (*epochs*), l'optimisation se fait plus lente. Aux alentours de 250 *epochs*, la fonction coût semble avoir atteint un minimum stable, et le coefficient de corrélation n'évolue plus non-plus. La corrélation finale obtenue est de 0,913 (sur les données de validation), or on peut lire dans la figure gauche de 3.8 que la corrélation obtenue sur les données de test est de 0,907 ; il y a donc eu un très léger sur-apprentissage, puisque le réseau fournit de meilleures performances sur les données de validation que sur celles de test (ce sur-apprentissage est néanmoins très faible).

De plus, nous avons testé si un réseau de neurones pouvait être formé à prédire les trois caractéristiques désirées en même temps. En effet, jusqu'ici, chaque caractéristique avait été prédite indépendamment en formant à chaque fois un nouveau réseau de neurones (en conservant la même architecture mais en trouvant de nouveaux paramètres optimaux à chaque fois) ; la sortie du réseau était donc un scalaire. Nous avons donc cherché à savoir s'il était possible de prédire directement un vecteur contenant les trois quantités désirées. Il s'avère que cela est possible et que les performances obtenues sont similaires à celles obtenues en formant les réseaux indépendamment ; les coefficients de corrélation obtenus sont 0,89, 0,92 et 0,91 pour chacune des trois quantités à prédire. Ainsi, un même jeu de paramètres peut prédire à la fois l'intensité, la FWHM et le retard du pic temporel le plus intense, sur la seule base du profil d'intensité spectrale.

La reproduction de l'état de l'art à l'aide de mon code ainsi que de la toolbox MATLAB m'a permis de valider mon implémentation de l'algorithme de réseau de neurones présenté au chapitre précédent (de la génération de données au test du modèle), ce qui a rendu possible l'exploration de cet outil dans de nouveaux contextes physiques, tel que le régime *noise-like pulse*.

Chapitre 4

Résultats

"J'ai travaillé dur. Tous ceux qui travaillent aussi dur que moi auront les mêmes résultats."

– Jean-Sébastien Bach, 1735

Suite aux succès des réseaux de neurones à propagation avant dans l'analyse de corrélations entre intensité spectrale et maximum d'intensité temporelle, nous avons voulu continuer ces travaux dans d'autres régimes de propagation. Une première étude a été menée sur le régime d'instabilité de modulation pure (et non impulsionnelle), puis sur le régime de cavité *noise-like pulse* où les profils spectraux et temporels sont hautement chaotiques et structurés. Enfin, une collaboration avec l'université de Tampere a permis d'explorer l'utilisation des réseaux de neurones en tant qu'algorithme dit *model-free*. Ces travaux ont été réalisés à l'aide de mon code développé en MATLAB, de la toolbox MATLAB, ainsi que de la toolbox Python "TensorFlow".

4.1 Instabilité de modulation : onde continue

4.1.1 Propagation

Les travaux précédents ont été effectués dans le régime d'instabilité de modulation dit impulsionnel, dans lequel le phénomène est stimulé par une impulsion en entrée de la fibre. Cependant, aucune étude n'avait encore été menée sur le régime d'onde continue (*continuous wave*), où l'instabilité de modulation résulte de l'amplification préférentielle spontanée de certaines fréquences d'une onde continue bruitée (modulée), dû à l'interaction entre la dispersion de la fibre et la non-linéarité, comme expliqué dans le chapitre 1. Ce régime est le premier étudié historiquement, tous les autres régimes (*supercontinuum*, *noise-like pulse*, ...) étant décrit par le prisme de celui-ci ; il est donc celui où les résultats théoriques les plus importants ont été formulés. De plus, il existe de fortes analogies avec d'autres domaines de la physique tels que l'océanographie [88], la physique des plasmas [92], ou encore les condensats de Bose-Einstein [93] ; son analyse par apprentissage machine semblait donc pertinente [94].

Comme il a été vu au chapitre 2, un réseau de neurones en condition d'apprentissage supervisé a besoin de données complètes pour pouvoir optimiser ses paramètres internes (poids et biais). Ici, ces données sont obtenues par intégration numérique de l'ESNL (1.58) sans pertes

$$\frac{\partial A}{\partial z} = i \left(-\frac{\beta_2}{2} \frac{\partial^2 A}{\partial T^2} + \gamma |A|^2 A \right) \quad (4.1)$$

et excitée par une onde continue (*continuous wave*) bruitée, de puissance P_0 .

Les caractéristiques du système sont données dans le tableau suivant :

L	β_2	γ	λ_0	P_0	FWHM
[m]	[fs ² /mm]	[W ⁻¹ .m ⁻¹]	[nm]	[W]	[ps]
400	-21	0,0011	1550	50	∞

TABLE 4.1 – Paramètres physiques des simulations en régime d’instabilité de modulation stimulée par une onde continue : paramètres de la fibre à gauche et paramètres de l’onde continue d’entrée à droite

Ce système a donc été étudié dans son régime transitoire (sur une distance de propagation relativement courte) et stationnaire (sur une distance de propagation très grande), ce dernier étant le régime d’étude privilégié dans lequel les résultats théoriques sont obtenus grâce à la méthode d’*inverse scattering* (1.112) [95]. Dans le régime transitoire, l’évolution d’une onde continue bruitée a été étudiée sur une distance totale de 450 m. La figure 4.1 illustre ce régime de cette dynamique d’instabilité de modulation spontanée, où la propagation consiste en une phase initiale de développement de l’instabilité sur environ 200 m, suivie de quatre cycles de croissance et de décroissance de la dynamique du *breather* sur les 250 m suivants.

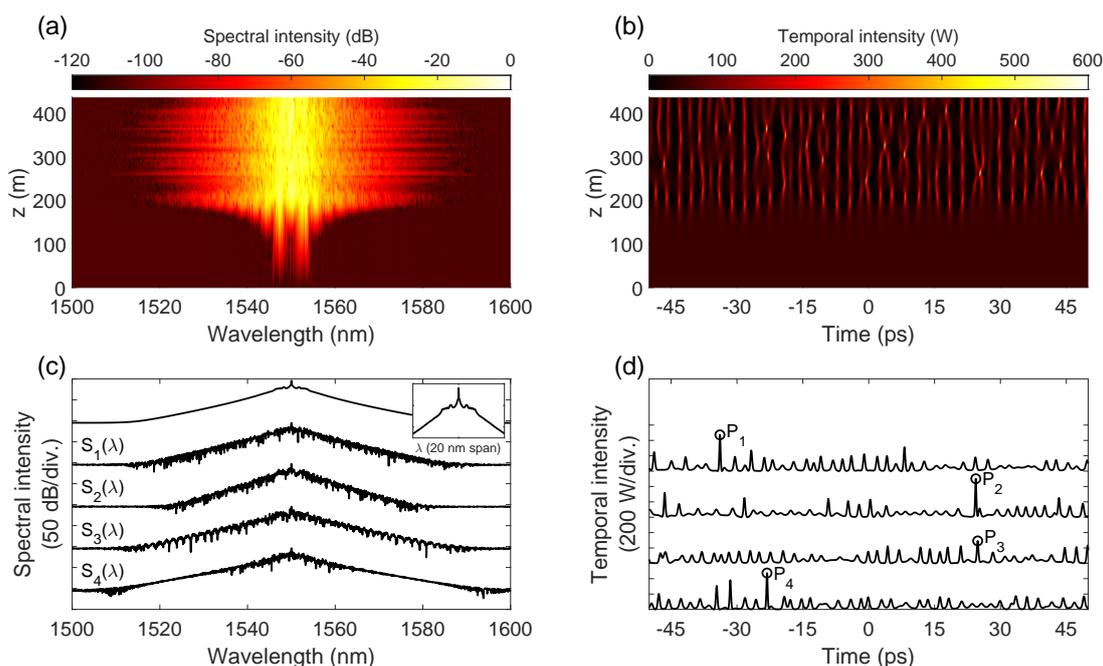


FIGURE 4.1 – Simulation numérique d’instabilité de modulation excitée par une onde continue bruitée dans le régime transitoire. Les figures (a) et (b) montrent respectivement l’évolution spectrale et temporelle sur 450 m issues d’une seule simulation. Les figures (c) et (d) illustrent les fluctuations en intensité spectrale et temporelle à la sortie de la fibre pour 4 simulations différentes. (c) montre également la moyenne calculée (en haut) et un zoom de cette dernière (encart) au voisinage de la pompe ; cet encart est tracé sur un intervalle de 20 nm et sur une plage dynamique de 50 dB. Les pics d’intensité temporelle P_k , représentés en (d) par un cercle pour chaque simulation, sont les grandeurs que nous cherchons à prédire à partir des spectres correspondants S_k , représentés en (c). Afin d’afficher clairement les différents résultats de simulations, un décalage vertical artificiel entre les différents profils est ajouté en (c) et (d).

Les figures 4.1(a) et (b) montrent une évolution spatio-spectrale et spatio-temporelle typiques, les résultats en (b) montrant en particulier l'émergence d'une série complexe de pics temporels issus du bruit, évoluant de manière aléatoire. On peut y observer des structures solitoniques classiques telles que celles décrites dans le chapitre 1 ; on observe des *breathers* d'Akhmediev ainsi que des solitons de Kuznetsov-Ma qui se propagent et interagissent pour former des solitons d'ordres supérieurs. La figure 4.1(a) présente bien une amplification spectrale telle que celle décrite dans le chapitre 1 ; les deux bandes caractéristiques de l'instabilité de modulation situées autour de la longueur d'onde centrale λ_0 sont bien notables, et dont la largeur d'environ 5 nm coïncide avec celle obtenue analytiquement (figure 1.3).

Suivant l'approche utilisée précédemment, un grand nombre de simulations (10000) a été effectué en utilisant différentes graines (*seeds*) de bruit aléatoire, et en sauvegardant les profils d'intensité temporelle et spectrale à la fin du segment de propagation (450 m). Les figures 4.1(c) et (d) montrent quatre résultats issus de ces simulations, où l'on voit comment les caractéristiques spectrales et temporelles en sortie varient de manière significative d'une simulation à l'autre. Il est alors possible d'observer plus finement la structure locale (à z donné) de cette dynamique, et de se rendre ainsi compte de la nature chaotique du processus d'instabilité de modulation. En effet, chaque profil spectral et temporel est très complexe et est constitué de nombreux pics dont l'espacement est plutôt régulier (associé à la fréquence f_{max} , voir équation 1.108) et d'amplitudes très variables. De plus, les variations d'une simulation à l'autre, où seule la perturbation (bruit) initiale change, sont également remarquables, puisque seulement quelques cycles de *breathers* suffisent à rendre le profil de sortie de fibre complètement imprévisible (a priori) ; la position et l'amplitude du maximum d'intensité temporelle par exemple ne semblent pas suivre de loi particulière, et les spectres, bien qu'ayant tous une forme triangulaire (en échelle logarithmique), ont une largeur très variable.

Le spectre moyen (sur les 10000 simulations) est représenté par la courbe supérieure dans la figure 4.1(c), et l'encadré de cette figure montre une vue élargie sur 20 nm autour de la pompe pour mettre en évidence les pics de la bande latérale ; cette structure de bande latérale n'apparaît pas visuellement sur les profils individuels mais est bien présente sur le spectre moyen, et reflète les premiers stades de l'évolution. L'objectif est donc, là encore, de tenter de prédire le maximum d'intensité temporelle P_k à l'intérieur de la fenêtre temporelle sur la seule base des profils d'intensité spectrale $S_k(\lambda)$; ces points sont représentés par des cercles sur la figure 4.1(d).

La méthode numérique utilisée est la technique standard de *split-step Fourier*. En règle générale, pour les simulations de MI, la fenêtre temporelle doit être suffisamment grande pour pouvoir accéder à une variété de scénarios solitoniques, et ainsi observer toute la gamme des solutions de *solitons on finite background* (1.119) telles que les *breathers* d'Akhmediev ou de Kuznetsov-Ma. Il faut également que la résolution temporelle soit suffisante pour observer avec précision les pics temporels, et que la fenêtre de fréquence correspondante soit telle que le spectre soit clairement discernable du bruit de fond numérique. Pour les paramètres de fibre et de puissance utilisés, il est attendu que les pics temporels soient séparés en moyenne par l'inverse de la fréquence caractéristique d'instabilité de modulation f_{max} (voir équation 1.108), ce qui suggère une séparation temporelle de l'ordre de 2,7 ps. Une fenêtre temporelle de 110 ps a donc été choisie de manière à observer une quarantaine de pics distincts, ce qui donne, avec 2048 points de grille, une résolution temporelle de 53,7 fs. Étant donné que la FWHM moyenne des pics les plus élevés a été mesurée comme étant d'environ 500 fs, la résolution choisie est suffisante pour une bonne détection des pics. Pour ces paramètres, la plage de fréquences de 18,6 THz englobe le spectre complet, qui s'étend sur une plage d'environ 11 THz avant d'atteindre le bruit de fond.

Le second régime étudié est réalisé sur une distance beaucoup plus grande de 550 km pour les mêmes conditions initiales (au bruit près). Dans ce cas, les caractéristiques spectrales et temporelles ont été analysées au-delà de la phase initiale de la propagation, dans le régime stationnaire, où l'entrée évolue vers une instabilité de modulation aléatoire (turbulente) pleinement développée, sans structure apparente de bande latérale [96, 97]. Pour cela, les spectres avant une distance de 2,7 km (environ $150 L_{NL}$) ont été écartés et, à partir de cet endroit, 10000 profils d'intensité spectrale et temporelle ont été extraits à des intervalles réguliers de 55 m (environ $3 L_{NL}$) pour former des ensembles de données d'entraînement et de test. Ces résultats sont présentés dans la figure 4.2 et sont à comparer aux résultats de la figure 4.1. Les figures 4.2(a) et (b), montrent la propagation sur une distance de 300 m seulement après être entrée dans le régime stationnaire (en particulier, la figure 4.2 montre l'évolution après 55 km, loin dans le régime stationnaire). Là encore, des dynamiques solitoniques de *breathers* sont apparentes, et les variations d'un profil spectral (ou temporel) à l'autre montrent bien la nature chaotique du système; l'entrée dans le régime de turbulence intégrable est apparente dans le spectre moyen montré dans l'encadré de la figure 4.2(c), qui ne présente aucune structure de bande latérale. Les caractéristiques statistiques de ce régime étant les mêmes que pour le régime précédent, les mêmes paramètres de simulation numérique (2048 points sur une fenêtre temporelle de 110 ps) ont été utilisés.

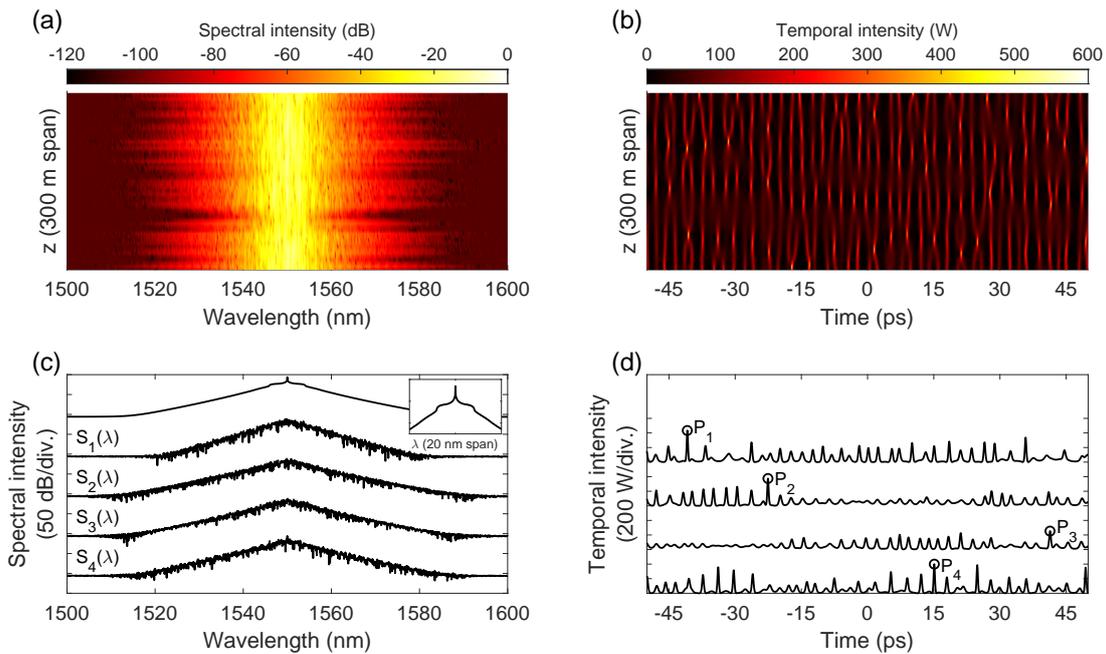


FIGURE 4.2 – Simulation numérique d'instabilité de modulation excitée par une onde continue bruitée dans le régime stationnaire. Les figures (a) et (b) montrent respectivement l'évolution spectrale et temporelle sur 300 m issues d'une seule simulation (la propagation affichée ici commence à 55 km). Les figures (c) et (d) illustrent les fluctuations en intensité spectrale et temporelle à 55 m ($\sim 3 L_{NL}$) d'intervalles. (c) montre également la moyenne calculée (en haut) et un zoom de cette dernière (encart) au voisinage de la pompe; cet encart est tracé sur un intervalle de 20 nm et sur une plage dynamique de 50 dB. Les pics d'intensité temporelle P_k , représentés en (d) par un cercle, sont les paramètres que nous cherchons à prédire à partir des spectres correspondants S_k , représentés en (c). Afin d'afficher clairement les différents résultats de simulations, un décalage vertical artificiel entre les différents profils est ajouté en (c) et (d).

4.1.2 Analyse par réseau de neurones

Afin de prédire le maximum d'intensité temporelle P_k , ne connaissant que l'intensité spectrale $S_k(\lambda)$ (sans la phase) grâce à un réseau de neurones à propagation avant, il a tout d'abord fallu trouver une architecture pertinente; il a donc fallu chercher de bons hyper-paramètres. Comme expliqué dans le chapitre 2, une telle recherche est malheureusement très compliquée, si bien que l'une des méthodes les plus employées consiste simplement à tester un grand nombre d'architectures en ne faisant varier qu'un seul hyper-paramètre à la fois; cette méthode est appelée *grid search*. Cette recherche a été effectuée en ne faisant varier que deux hyper-paramètres : le nombre de neurones de la première couche cachée, et le nombre de neurones de la seconde couche cachée. A chaque fois, les fonctions non-linéaires choisies étaient des tangentes hyperboliques. On obtient alors la carte de la figure 4.3. Cette dernière montre l'évolution du coefficient de corrélation de Pearson ρ calculé pour chaque architecture après formation du réseau de neurones. On constate qu'augmenter le nombre de neurones de la seconde couche cachée a tendance à dégrader la capacité de généralisation du réseau. Il est toujours délicat d'expliquer les raisons pour lesquelles un réseau sous-performe (c'est d'ailleurs la raison pour laquelle il est si compliqué de trouver de bons hyper-paramètres), mais une raison qui pourrait être avancée ici est que les architectures avec un trop grand nombre de neurones dans leur seconde couche cachée ont tendance à sur-interpréter (*overfitting*) les données d'entraînement, et donc à mal généraliser. Ainsi, c'est l'architecture correspondant à la corrélation maximale (point bleu de la figure 4.3b)) qui a été choisie pour l'étude qui va suivre; le réseau de neurones comporte donc une première couche cachée de 150 neurones et une seconde en comportant 10.

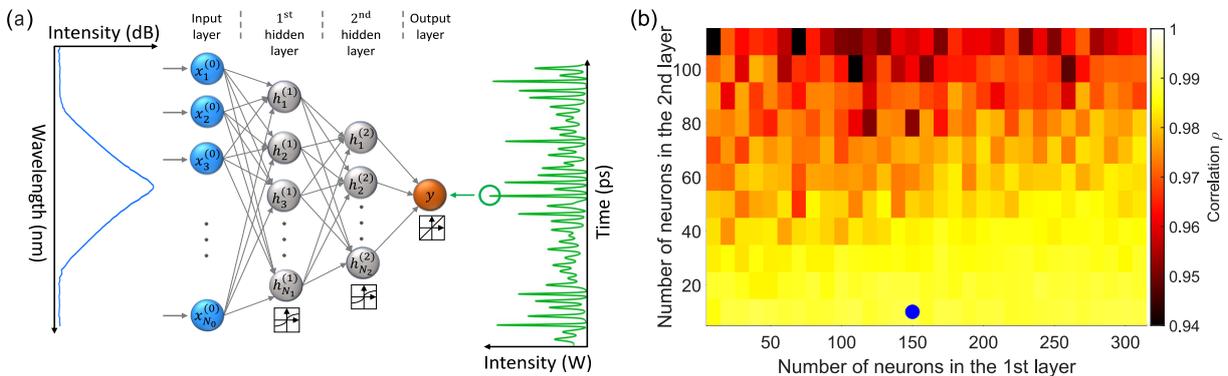


FIGURE 4.3 – (a) Représentation schématique du réseau de neurones choisi afin d'effectuer l'analyse (150 neurones en entrée, 150 neurones pour la première couche cachée, 10 neurones pour la seconde couche cachée, et un neurone en sortie). (b) *Grid search* effectué sur deux hyper-paramètres dans le régime stationnaire : le nombre de neurones de la première couche cachée en abscisse, et le nombre de neurones de la seconde couche cachée en ordonnée. La couleur indique le coefficient de corrélation de Pearson obtenu pour chaque architecture et le point bleu indique l'architecture retenue pour l'étude.

Avant d'être fournies au réseau de neurones, les données sont prétraitées afin de simuler les contraintes expérimentales; chaque profil d'intensité spectrale est convolué avec une gaussienne présentant une FWHM de 1 nm ce qui simule la réponse impulsionnelle du spectromètre, puis les points contenus dans un même intervalle de largeur 1 nm sont moyennés afin de simuler la résolution spectrale du spectromètre. On obtient alors un profil plus lisse et contenant beaucoup moins de points; on passe de 2048 points initialement à 100 après prétraitement. Ces données sont ensuite mises en échelle logarithmique (dB) afin qu'elles soient

mieux conditionnées ; en effet, il est numériquement préférable de travailler avec des valeurs qui sont toutes du même ordre de grandeur plutôt qu’avec des valeurs relatives extrêmes i.e. de plusieurs ordres de grandeurs plus petites ou grandes. Pour finir, les données sont centrées et réduites afin qu’elles soient mieux adaptées à l’analyse par réseau de neurones, comme il a été expliqué au chapitre précédent. La figure 4.4 montre les différentes étapes du processus de pré-traitement des données : la ligne fine grise est une donnée brute en dB, la ligne épaisse rouge est issue de la convolution de la donnée brute avec la gaussienne, et la ligne bleue est issue du groupement des points par classes de 1 nm. Les données sont alors séparées en deux groupes : un groupe de 7500 données pour l’entraînement et un groupe de 2500 pour tester l’architecture.

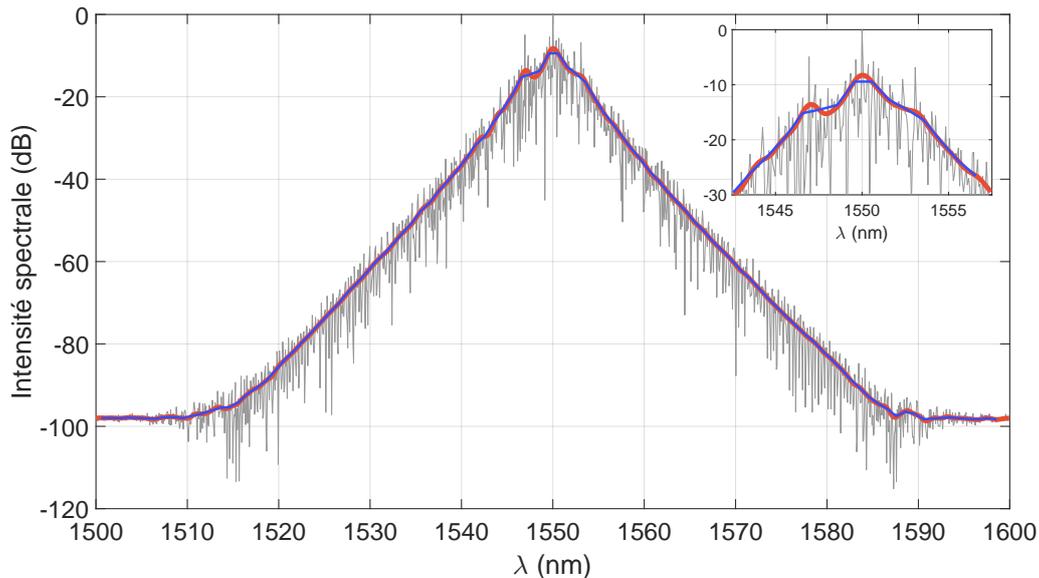


FIGURE 4.4 – Étapes de pré-traitement des données d’entrée du réseau de neurones. La ligne fine grise montre le profil d’intensité spectrale brut issu de l’intégration numérique de l’équation de Schrödinger non-linéaire (en dB). La ligne la plus épaisse en rouge montre la convolution de ce profil brut par une gaussienne de largeur (FWHM) 1 nm, et la ligne bleue montre ce profil convolué après avoir été moyenné dans des classes de largeur 1 nm ; ce sont ces dernières données qui seront fournies au réseau de neurones, après avoir été normalisées. Le profil initial contient 2048 points répartis entre 1480 et 1630 nm, et le profil final en contient 100 répartis entre 1500 et 1600 nm. L’encadré en haut à droite est un zoom sur la zone centrée sur la longueur d’onde de la pompe λ_0 .

Les performances du réseau sont là encore visualisées grâce à l’histogramme issu du nuage de points {vraie valeur ; prédiction}, et quantifiées grâce au coefficient de corrélation de Pearson ρ . Pour cette étude, nous avons fait varier la plage dynamique des vecteurs d’entrée du réseau (intensité spectrale) afin de simuler les contraintes expérimentales. La figure 4.5(a) montre les performances du réseau de neurones en régime transitoire et avec toute la plage dynamique offerte par la résolution numérique (environ 100 dB) et la figure 4.5(b) montre les performances de ce même réseau lorsque les données d’entraînement ont été tronquées à 40 dB. L’évolution de la corrélation ρ en fonction à la plage dynamique des données d’entraînement apparaît sur la figure 4.5(c).

On constate que lorsque la plage dynamique est maximale (~ 100 dB), la prédiction du réseau de neurones est quasiment parfaite, avec un coefficient de corrélation de 0,99. De plus, bien que cette prédiction se dégrade à mesure que la plage dynamique des données d’entraînement

diminue, elle demeure néanmoins raisonnablement bonne, avec un coefficient de corrélation supérieur à 0,9 jusqu'à une plage dynamique de 40 dB, ce qui est atteignable d'un point de vue expérimental. Le fait de réduire la plage dynamique a pour effet de tronquer l'information contenue dans les ailes du profil d'intensité spectrale, or c'est justement là qu'est contenue, a priori, l'information sur les pics temporels les plus intenses; en effet, ces derniers étant très fins, ils sont ceux qui contribuent le plus à l'élargissement du spectre. Ainsi, en privant le réseau de l'information de faible intensité spectrale, on le prive de l'information contenue le plus loin de la pompe, c'est-à-dire celle directement corrélée aux pics les plus fins du profil temporel, et donc de l'information concernant le maximum d'intensité temporelle; l'information contenue au niveau de la pompe (entre 1540 et 1560 nm par exemple) est un amalgame d'informations concernant tous les pics temporels, les plus larges (les moins intenses) comme les plus fins (les plus intenses), et c'est pourquoi elle n'est pas suffisante à bien prédire le maximum d'intensité temporelle.

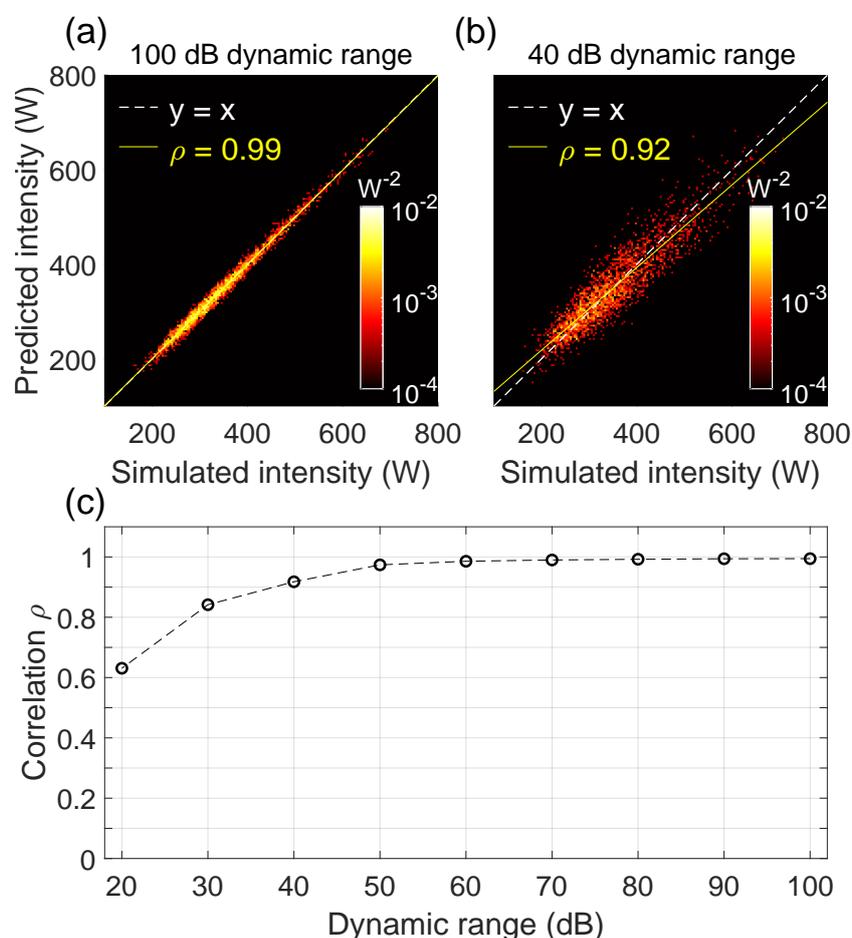


FIGURE 4.5 – Performances du réseau de neurones pour le régime transitoire, quantifiant la prédiction par le réseau des pics d'intensité temporelle de MI sur la seule base des données spectrales d'entrée. Ces résultats sont basés sur un ensemble de 2500 simulations non-utilisées dans la phase de formation. Les puissances crêtes prédites des maxima d'intensités temporelles sont comparées avec celles issues de simulations : (a) avec la plage dynamique complète de 100 dB des simulations, (b) tronquée à une plage dynamique de 40 dB. (c) Représentation graphique du coefficient de corrélation pour différentes plages dynamiques de 20 à 100 dB.

L'étude du système en régime stationnaire a mené à la figure 4.6, qui présente les mêmes grandeurs que la figure 4.5 dans ce nouveau régime.

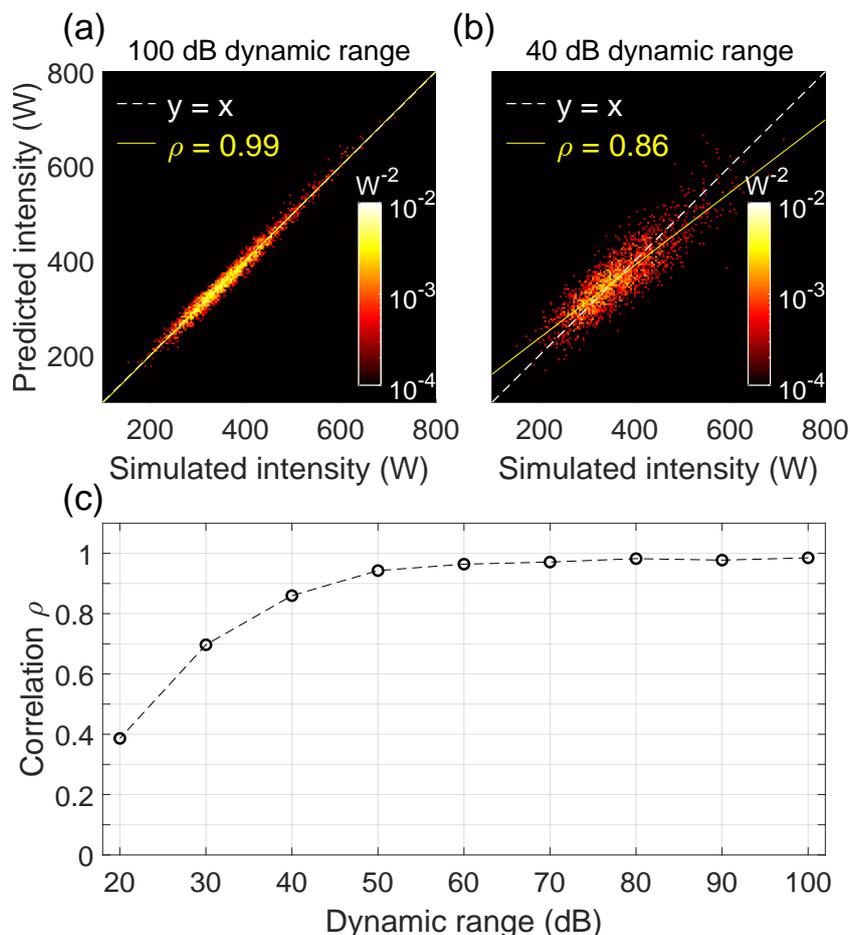


FIGURE 4.6 – Performances du réseau de neurones pour le régime stationnaire, quantifiant la prédiction par le réseau des pics d'intensité temporelle de MI sur la seule base des données spectrales d'entrée. Ces résultats sont basés sur un ensemble de 2500 simulations non-utilisées dans la phase de formation. Les puissances crêtes prédites des maxima d'intensités temporelles sont comparées avec celles issues de simulations : (a) avec la plage dynamique complète de 100 dB des simulations, (b) tronquée à une plage dynamique de 40 dB. (c) Représentation graphique du coefficient de corrélation pour différentes plages dynamiques de 20 à 100 dB.

Là encore, la prédiction du réseau est quasiment parfaite lorsque toute la plage dynamique est accessible comme le montre la figure 4.6(a), mais se dégrade néanmoins légèrement plus rapidement que précédemment à mesure que la plage dynamique diminue, comme le montre les figures 4.6(b) et (c). De plus, le fait que la corrélation reste très bonne, même dans ce régime sans structure de bande latérale, montre bien que cette dernière ne joue pas de rôle crucial dans la prédiction du maximum d'intensité temporelle. Il est à noter que des résultats similaires ont été obtenus (aussi bien dans le régime transitoire que stationnaire) avec des données issues de simulations dont la fenêtre temporelle était deux fois plus grande (220 ps) ou deux fois plus petite (50 ps).

Jusqu'ici, toutes les prédictions faites par le réseau de neurones consistaient à prédire le maximum d'intensité temporelle à une certaine position z dans la fibre ne connaissant que le profil d'intensité spectrale à cette même distance z . Nous nous sommes alors intéressés à la capacité du réseau à prédire ce maximum d'intensité temporelle sur la base d'un profil

d'intensité spectrale obtenu plus tôt dans la propagation, c'est-à-dire à une distance $z - \Delta z$, afin de savoir s'il était possible de prévoir l'émergence d'un pic d'intensité en amont. L'évolution de la corrélation en fonction de la distance Δz (normalisée en proportion de L_{NL}) pour le régime transitoire (respectivement stationnaire), tout d'abord dans le cas où la plage dynamique n'est limitée que par la précision machine, puis après avoir réduit la plage dynamique des données à un seuil de 40 dB, est visible dans la figure 4.7(a) (respectivement 4.7(b)).

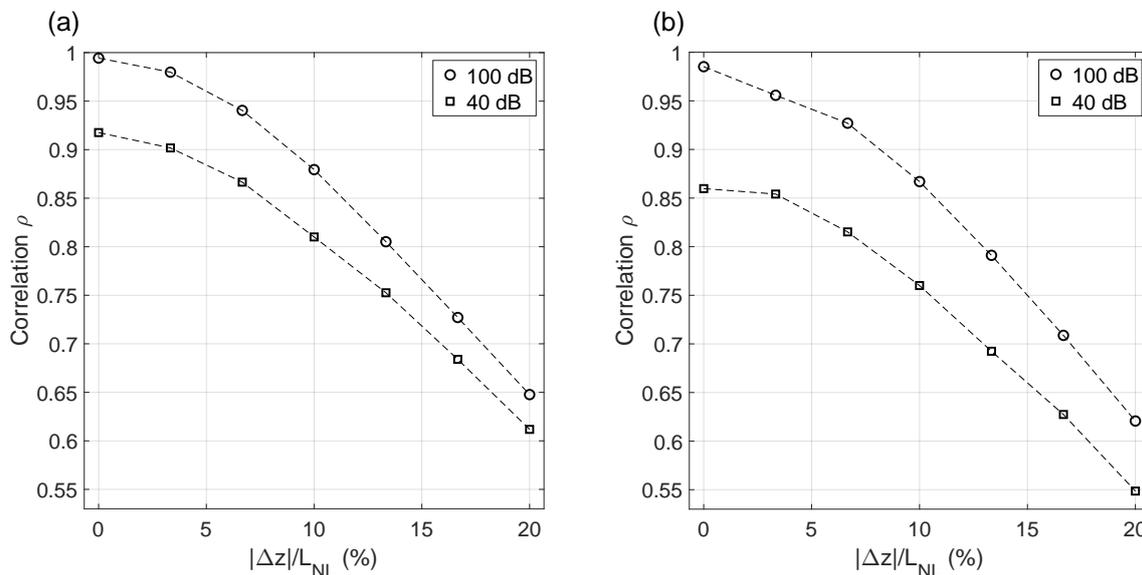


FIGURE 4.7 – Coefficient de corrélation du test dans le régime (a) transitoire et (b) stationnaire en fonction de $\Delta z/L_{NL}$, la distance entre le profil temporel cible utilisé lors de l'entraînement et celui utilisé dans ce test prédictif. Les distances sont exprimées en pourcentage de la longueur non linéaire. Les résultats sont présentés pour deux plages dynamiques 100 dB et 40 dB (comme indiqué).

Bien que la corrélation diminue rapidement avec la distance, il est possible d'établir des limites au pouvoir prédictif utile du réseau ; pour obtenir une corrélation supérieure à 0,8 dans les deux régimes, il est nécessaire que Δz soit inférieur à $0,1 L_{NL}$. Bien que, d'un point de vue prédictif, ces résultats suggèrent qu'il existe une décorrélation rapide entre les propriétés temporelles et spectrales avec la distance de propagation (ce qui était attendu de part le caractère chaotique de la dynamique étudiée), l'établissement de cette limite de manière quantitative est néanmoins utile.

4.2 Laser à soliton dissipatif

4.2.1 Propagation

Au vu des résultats positifs concernant les capacités d'un réseau de neurones bien formé à établir une correspondance entre le profil d'intensité spectrale et certaines propriétés temporelles dans un régime aussi complexe et chaotique que celui du supercontinuum, nous nous sommes intéressés à un régime plus complexe et plus chaotique encore, où les instabilités donnent naissance à des dynamiques solitoniques riches [98]. Ce système est un système en cavité dans lequel une impulsion initiale est injectée dans une fibre à gain dopée en erbium (fibre EDF) puis une fibre hautement non-linéaire (fibre HNLF) avant de passer dans un absorbant saturable couplé à un filtre passe-bande étroit centré à 1550 nm puis de retourner dans la fibre EDF, d'où l'impulsion peut refaire un tour de cavité; un schéma du système est visible à la figure 4.8.

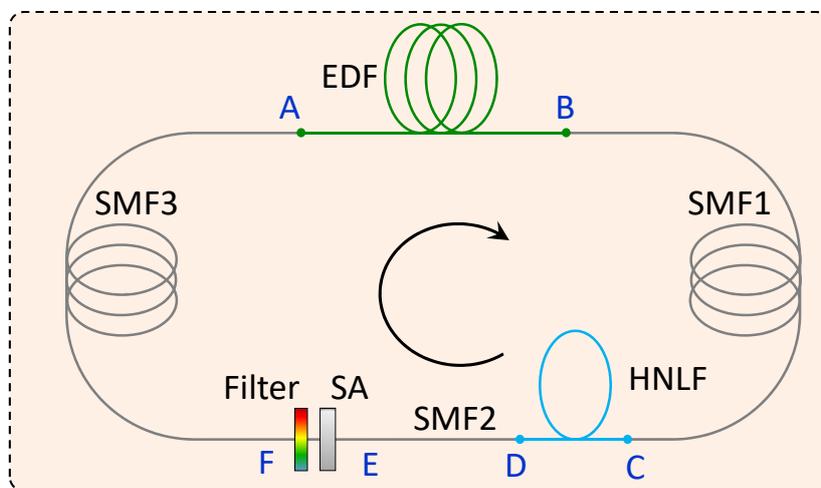


FIGURE 4.8 – Schéma du laser NLP. EDF : fibre dopée en l'Erbium. SMF : fibre mono-mode SMF28. HNLF : fibre hautement non linéaire. SA : absorbant saturable. Les paramètres de tous les éléments de la cavité sont donnés dans le texte.

Le choix d'un tel système a été motivé par le fait que la classe d'instabilité associée au fonctionnement des lasers fibrés à soliton dissipatif a fait l'objet d'études intensives en optique non linéaire ces dernières années [99, 100, 101]. Ici, l'interaction entre le gain, les pertes et la dynamique de l'équation de Schrödinger non-linéaire généralisée peut conduire à un large éventail de comportements, y compris des explosions de solitons, des molécules de solitons [102] et des intermittences chaotiques [103]. Une instabilité particulièrement complexe dans les lasers à fibre, dite de type *noise-like pulse* (NLP), a également attiré beaucoup d'attention en raison de son observation dans une large gamme de régimes de paramètres laser, y compris avec une dispersion de cavité normale et anormale [104], et avec des largeurs de bande spectrales pouvant être étroites (quelques dizaines de nm) [105] ainsi que des largeurs de bande beaucoup plus grandes que la largeur de bande de gain (quelques centaines de nm) [106].

Une étude récente du fonctionnement de cavité type NLP utilisant à la fois des mesures en temps réel par lentilles temporelles, des mesures par transformée de Fourier dispersive et des simulations numériques détaillées a mis en évidence le rôle central joué par la turbulence des solitons et la dynamique supercontinuum dans ce régime [107]. Dans ces travaux les auteurs ont mené une étude expérimentale et numérique, mettant en évidence que l'interaction

entre les phénomènes d'auto-modulation de phase et de dispersion de vitesse de groupe sont à l'origine des comportements riches et des dynamiques très complexes, chaotiques et multi-échelles, appelées *noise-like* ; c'est sur cette étude que nous nous sommes appuyé pour effectuer ces travaux.

Mon but était donc de savoir si un réseau de neurones serait capable d'établir une correspondance entre un profil d'intensité spectrale (sans la phase) issu d'une cavité laser à soliton dissipatif et certaines informations temporelles associées à ce spectre. Les données complètes nécessaires à la formation du réseau de neurones ont été obtenues par intégration numérique de l'équation de Schrödinger non-linéaire généralisée (1.48), où les pertes ont été négligées ($\alpha = 0$), et à laquelle un terme de gain \hat{g} est ajouté pour modéliser le gain de la fibre EDF :

$$\frac{\partial A}{\partial z} = -i\frac{\beta_2}{2}\frac{\partial^2 A}{\partial T^2} + \frac{\beta_3}{6}\frac{\partial^3 A}{\partial T^3} + i\gamma_0 \left(1 + \frac{i}{\omega_0}\frac{\partial}{\partial T}\right) \left(\int_0^{+\infty} R(T')|A(z, T - T')|^2 dT' A\right) + \frac{\hat{g}}{2} \quad (4.2)$$

avec

$$\hat{g}(\omega) = \begin{cases} \frac{1}{1 + \frac{E}{E_{sat}}} \times \frac{g_0}{1 + \left(\frac{\omega - \omega_0}{\omega_g}\right)^2} & \text{dans la fibre EDF} \\ 0 & \text{sinon} \end{cases} \quad (4.3)$$

et où $E = \int |A(z; T)|^2 dT$ est l'énergie de l'impulsion dans la cavité, E_{sat} l'énergie de saturation du gain en nJ, g_0 est le gain non-saturé (petit signal) en m^{-1} , et ω_g le gain à demi largeur spectrale.

Des fonctions de transfert servant à modéliser l'absorbant saturable (\mathcal{T}_{as}) couplé au filtre (\mathcal{T}_f) ont été utilisées pour modéliser la propagation de l'onde au travers de ces deux éléments :

$$\begin{cases} \mathcal{T}_{as}(T) = 1 - \frac{q_0}{1 + \frac{P(T)}{P_{sat}}} \\ \mathcal{T}_f(\omega) = c_1 e^{-\left(\frac{\omega - \omega_0}{\omega_1}\right)^{2m_1}} + c_2 e^{-\left(\frac{\omega - \omega_0}{\omega_2}\right)^{2m_2}} \end{cases} \quad (4.4)$$

où q_0 est la perte non-saturée, $P(T) = |A(z; T)|^2$ la puissance instantanée en W, P_{sat} la puissance de saturation de l'absorbant saturable en W, et c_1 , c_2 , ω_1 , ω_2 , m_1 et m_2 des coefficients optimisés pour ajuster au mieux la fonction de transfert analytique aux mesures expérimentales de la fonction de transfert réelle du filtre : les paramètres de la fonction de gain \hat{g} sont $g_0 = 0,7326 m^{-1}$, $E_{sat} = 3,5 nJ$ et $\omega_g = 20 nm$, ceux de l'absorbant saturable sont $q_0 = 0,9$ et $P_{sat} = 150 W$, et les coefficients d'ajustement sont $c_1 = 0,7036$, $c_2 = 0,2944$, $\omega_1 = 4,2584 rad.ps^{-1}$, $\omega_2 = 6,8006 rad.ps^{-1}$, $m_1 = 1,4483$ et $m_2 = 1,0034$.

Le filtre spectral appliqué après l'absorbant saturable est une super-gaussienne de 10 nm de FWHM, avec une transmittance unitaire à la longueur d'onde centrale λ_0 . Une perte linéaire intracavité globale de 7 dB a été appliquée après le filtre pour modéliser les pertes d'insertion et de couplage avec la sortie. La largeur de bande de 10 nm du filtre super-gaussien et la perte globale sont basées sur des résultats expérimentaux, le rôle du filtre étant de réduire la largeur de bande de l'impulsion après l'élargissement spectral avant la réinjection de cette dernière

dans la fibre EDF. Les paramètres physiques des simulations numériques sont résumés dans le tableau suivant :

paramètre		L	β_2	β_3	γ	λ_0	P_0	FWHM
type de fibre		[m]	[fs ² /mm]	[fs ³ /mm]	[W ⁻¹ .m ⁻¹]	[nm]	[W]	[ps]
EDF	A → B	10	40	0	0,006	1550	1	10
SMF	B → C	3	-21,7	86	0,0011	nan	nan	nan
	D → E	4						
	F → A	8						
HNLF	C → D	2	-5,23	42,76	0,0184			

TABLE 4.2 – Paramètres physiques des simulations de cavité *noise-like pulse* : paramètres des fibres au milieu et paramètres de l'impulsion d'entrée à droite

La dispersion nette de la cavité est alors de 0,075 ps² en l'absence d'HNLF et de 0,064 ps² lorsque la fibre HNLF est incluse.

L'étude a été menée sur deux montages, qui donnent lieu à deux régimes bien distincts : l'absence de fibre HNLF génère des spectres à bande relativement étroite, et sa présence génère des spectres supercontinuum à bande relativement large ; des spectres de propagation typiques sont montrés figures 4.9 et 4.10.

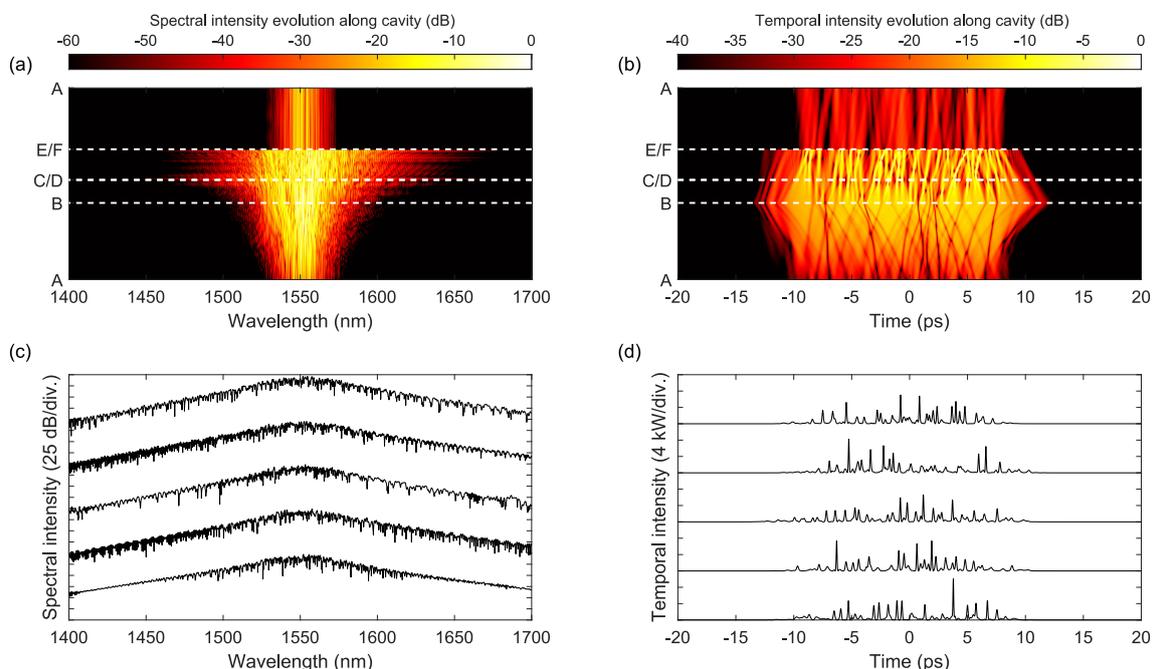


FIGURE 4.9 – Simulation numérique dans le régime à spectre étroit (sans HNLF). Les figures (a) et (b) montrent respectivement l'évolution spectrale et temporelle lors d'un tour de cavité. Les figures (c) et (d) illustrent les fluctuations entre chaque tour en affichant respectivement cinq profils d'intensité spectrale et temporelle sur cinq tours consécutifs, mesurés au point D (après le segment de fibre SMF1).

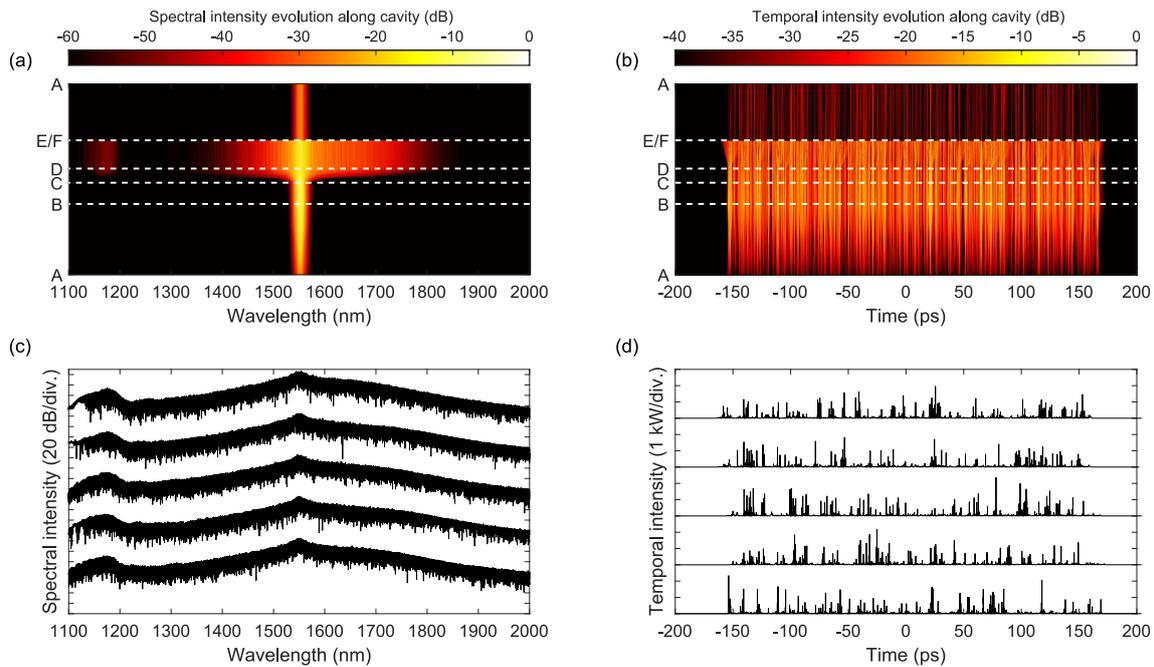


FIGURE 4.10 – Simulation numérique dans le régime à spectre large (avec HNLF). Les figures (a) et (b) montrent respectivement l'évolution spectrale et temporelle lors d'un tour de cavité. Les figures (c) et (d) illustrent les fluctuations entre chaque tour en affichant respectivement cinq profils d'intensité spectrale et temporelle sur cinq tours consécutifs, mesurés au point D (après le segment de fibre HNLF).

Le montage sans fibre HNLF génère des spectres d'une largeur de bande de 70 nm au seuil de -20 dB, et la présence de fibre HNLF en génère dont la largeur s'étend sur 200 nm à ce même seuil. Les simulations ont été faites avec une grille de 2^{13} (respectivement 2^{17}) points et une fenêtre temporelle de 120 ps (respectivement 650 ps) afin d'avoir une résolution spectrale et temporelle suffisamment bonne pour capter les détails des profils d'intensité.

Dans les deux cas, le signal subit un élargissement spectral incohérent dans la fibre EDF (segment A \rightarrow B), suivi d'une compression temporelle tout au long de la première fibre SMF (segment B \rightarrow C). La fibre HNLF (segment C \rightarrow D lorsqu'elle est présente) occasionne ensuite un élargissement supercontinuum, et l'absorbant saturable couplé au filtre passe-bande étroit ("segment" E \rightarrow F) assurent que le signal en fin de cavité puisse être réinjecté en début de cavité (point A). Les figures 4.9(c-d) et 4.10(c-d) montrent l'évolution de l'intensité spectrale (c) et temporelle (d) sur cinq tours de cavité consécutifs au point de mesure D de la cavité, situé à 13 (respectivement 15) mètres du point A. On constate que la dynamique du laser NLP conduit à des variations importantes d'un tour de cavité à l'autre pour chaque régime; on observe en effet dans les deux régimes l'émergence de nombreux solitons qui interagissent de manière très complexe, ainsi que des *dark solitons*, notamment dans le régime à spectre étroit (sans segment HNLF), résultant en des profils spectraux et temporels très structurés, contenant de nombreux pics. Ces variations temporelles et spectrales sont également associées à des fluctuations d'énergie entre les différents tours de cavité qui sont de l'ordre de 5 %, avec des énergies intracavité moyennes similaires pour le régime à bande étroite (13,4 nJ au point C) et le régime à bande large (12,3 nJ au point D).

Il est bon de noter que, bien que les deux régimes de fonctionnement aient des énergies comparables, les propriétés spectrales et temporelles sont très différentes. Plus précisément, dans le cas du régime à bande étroite, le spectre est approximativement symétrique, d'une largeur de bande de 70 nm au seuil de -20 dB, et avec une enveloppe d'impulsion d'environ

15 ps contenant approximativement 50 pics temporels d'une puissance crête médiane de 1 kW et d'une durée de 0,15 ps. En revanche, dans le régime à large bande, le spectre est fortement asymétrique avec une largeur de bande de 200 nm au seuil de -20 dB et les caractéristiques temporelles correspondantes sont associées à une enveloppe d'impulsion d'environ 300 ps, contenant approximativement 300 pics temporels de puissance crête médiane de 0,2 kW et de durée 0,05 ps. Bien que la largeur de bande à -20 dB dans le régime avec HNLF (large bande) soit de 200 nm, la plage complète de longueurs d'onde incluant l'onde dispersive générée par effet Raman s'étend de 1150 à 1700 nm (donc une plage de 550 nm).

Malgré le fait que la dynamique de cette cavité soit très différente des cas de propagation étudiés dans les sections précédentes, nous allons voir qu'un réseau de neurones convenablement formé est toujours capable de prédire l'intensité temporelle maximale des impulsions en se basant uniquement sur les intensités spectrales (sans phase), cette approche étant particulièrement robuste dans le cas de la bande étroite.

4.2.2 Analyse par réseau de neurones

25000 données complètes ont été générées, dont l'entrée est le profil d'intensité spectrale et la sortie le maximum d'intensité temporelle associé. Ces 25000 données sont issues du régime établi de la cavité, et sont donc extraites d'un ensemble plus grand (environ 26000) dont les spectres du régime transitoire ont été écartés (environ 1000 spectres). Les profils d'intensité spectrale ont été restreints à l'intervalle [1350; 1850] nm dans le régime à spectre étroit (sans HNLF) et l'intervalle [1100; 3200] nm dans le régime à spectre large (avec HNLF); ces valeurs ont été choisies comme compromis pour d'une part ne pas avoir des vecteurs d'entrée trop grands et d'autre part s'assurer que ces vecteurs contiennent suffisamment d'information. Afin de simuler au mieux les conditions expérimentales, ces profils d'intensité spectrale ont été convolués avec une gaussienne de 1 nm de FWHM, ce qui simule la réponse impulsionnelle du spectromètre et lisse les profils, puis groupé par classe de longueur 1 nm, ce qui simule la résolution du spectromètre; on aboutit alors à des vecteurs d'entrée de 500 et 2100 points. Les profils obtenus sont finalement mis en échelle logarithmique (en dB) puis centrés et réduits afin de mieux conditionner les données. La figure 4.11 montre les différentes étapes du processus de pré-traitement des données : la ligne fine grise est une donnée brute en dB, la ligne épaisse rouge est issue de la convolution de la donnée brute avec la gaussienne, et la ligne bleue est issue du groupement des points par classes de 1 nm. Les données sont alors séparées en deux groupes : un groupe de 15000 données pour l'entraînement et un groupe de 10000 pour tester l'architecture.

Après avoir affiné des hyper-paramètres par *grid search*, l'architecture retenue a été un réseau de neurones à deux couches cachées, contenant 100 et 30 neurones pour la cavité à spectre étroit et 300 et 100 neurones pour la cavité à spectre large. Les fonctions non-linéaires choisies sont des tangentes hyperboliques pour les deux couches, et la fonction coût à minimiser est la fonction quadratique. Un schéma du réseau de neurones est montré figure 4.12.

Comme pour les études précédentes, afin de modéliser au mieux les contraintes expérimentales, les spectres obtenus avec la démarche décrite plus haut ont été tronqués à différentes plages dynamiques. Après simulation, les données brutes ont une précision arbitrairement grande (à l'erreur machine près) et donc une plage dynamique elle aussi arbitrairement grande; ici, cette dernière est de l'ordre de 100 dB, ce qui est bien plus que ce qui est réalisable expérimentalement. Nous avons donc appliqué la même démarche d'optimisation (même architecture de réseau de neurones, même nombre de données d'entraînement et de test, même métrique de comparaison, ...) avec des données tronquées à différents seuils de plage dyna-

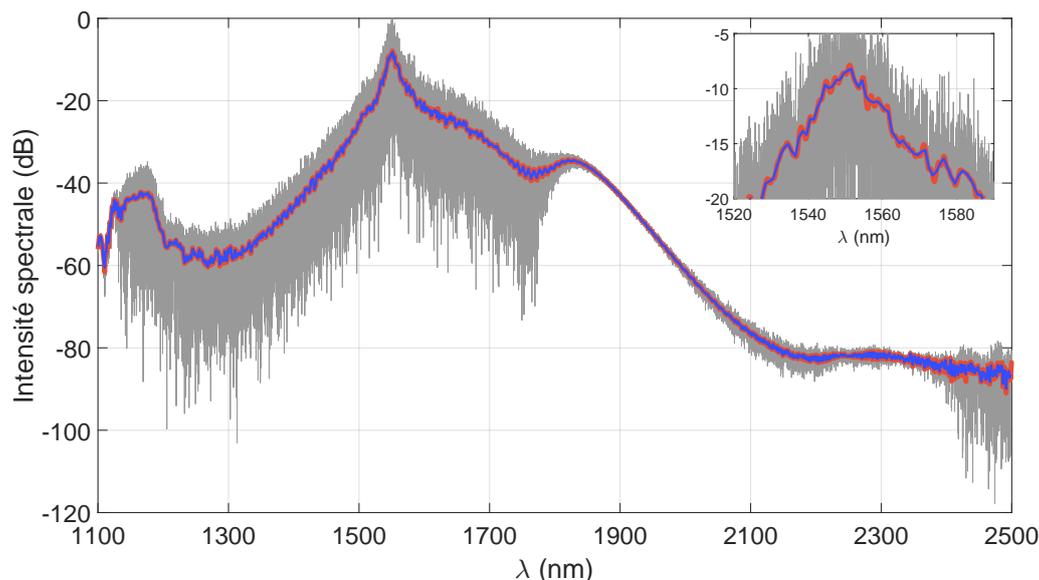


FIGURE 4.11 – Étapes de pré-traitement des données d'entrée du réseau de neurones en régime à spectre large (avec fibre HNLF). La ligne fine grise montre le profil d'intensité spectrale brut issu de l'intégration numérique de l'équation de Schrödinger non-linéaire. La ligne la plus épaisse en rouge montre la convolution de ce profil brut par une gaussienne de largeur (FWHM) 1 nm, et la ligne bleue montre ce profil convolué après avoir été moyenné dans des classes de largeur 1 nm; ce sont ces dernières données qui seront fournies au réseau de neurones, après les avoir normalisées. Le profil initial contient 2^{17} points répartis entre 1020 et 3240 nm, et le profil final en contient 2100 répartis entre 1100 et 3200 nm. L'encadré en haut à droite est un zoom sur la zone centrée sur la longueur d'onde de la pompe λ_0 .

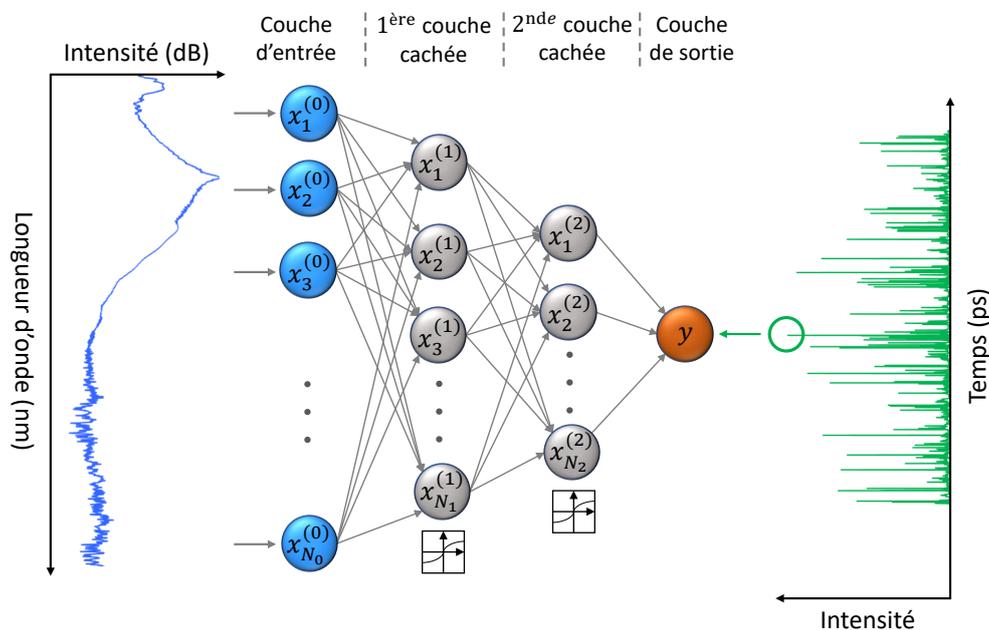


FIGURE 4.12 – Schéma du réseau de neurones à propagation avant utilisé. L'entrée est un vecteur d'intensité spectrale correspondant aux données spectrales issues de simulation après échantillonnage et convolution. Les deux couches cachées ont 100 et 30 neurones (dans le cas à bande étroite) et 300 et 100 neurones (dans le cas à bande large).

miques. Pour cela, à chaque fois que l'intensité spectrale était en dessous de la plage dynamique désirée, soit la valeur des neurones associés était mise à zéro (neurones mort), soit cette valeur était remplacée par un bruit gaussien d'amplitude 0,01 dB. Là encore, la capacité du réseau à généraliser est quantifiée en calculant le coefficient de corrélation linéaire de Pearson ρ ; celui-ci doit être le plus proche de un possible. La droite de régression linéaire du nuage de points à partir duquel est calculé l'histogramme est affichée en trait plein jaune, et les coefficients de régression doivent être idéalement le plus proche de un possible pour la pente de la droite, et de zéro pour son ordonnée à l'origine; les résultats sont présentés dans les figures 4.13 et 4.14.

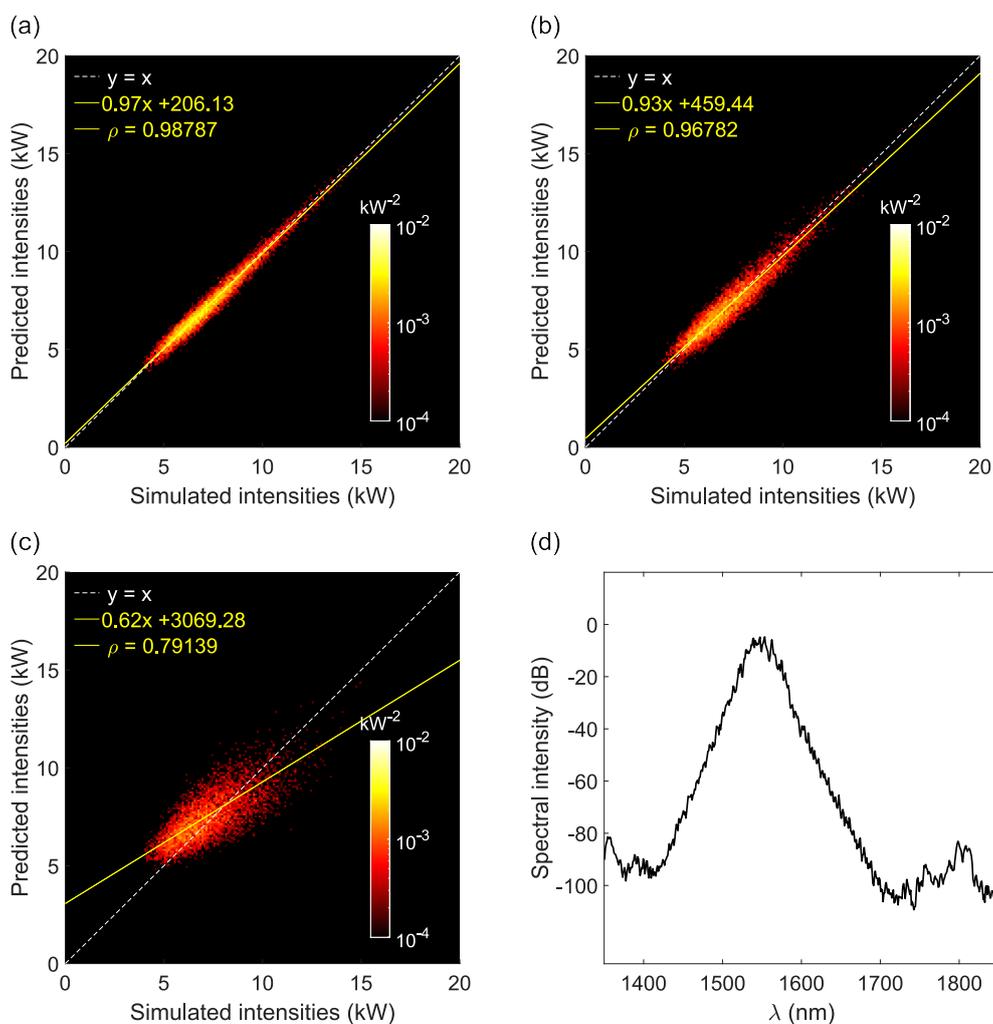


FIGURE 4.13 – Test du réseau de neurones pour la cavité à spectre étroit (sans HNLF). Prédiction du maximum d'intensité temporelle effectuée par le réseau de neurones en fonction de la valeur réelle de ce maximum. Les différentes figures correspondent à différents seuils de plage dynamique pour les données fournies au réseau : (a) montre la corrélation obtenue entre les prédictions et les valeurs réelles lorsque le réseau est formé avec des données non-tronquées, (b) montre cette corrélation lorsque les données sont tronquées à une plage dynamique de 50 dB, et (c) lorsque les données sont tronquées à une plage dynamique de 30 dB. La ligne jaune est la régression linéaire du nuage de points, et la ligne pointillée blanche est la diagonale vers laquelle doit tendre la ligne jaune. La figure (d) montre un profil d'intensité typique obtenu lors d'une simulation.

On constate que lorsque toute la plage dynamique est prise en compte (environ 100 dB), le réseau de neurones parvient très bien à prédire le maximum d'intensité temporelle associé à un profil d'intensité spectrale donné ; on voit que les points sont bien regroupés autour de la diagonale, et les coefficients de corrélation linéaire sont supérieurs à 0,9 dans les deux régimes, comme le montrent les figures 4.13(a) et 4.14(a). La diminution de la plage dynamique des entrées entraîne naturellement une baisse des performances du réseau de neurones, comme le montrent les figures 4.13(b-c) et 4.14(b-c) ; les figures 4.13(d) et 4.14(d) montrent des profils d'intensité spectrale typiques dans chaque régime.

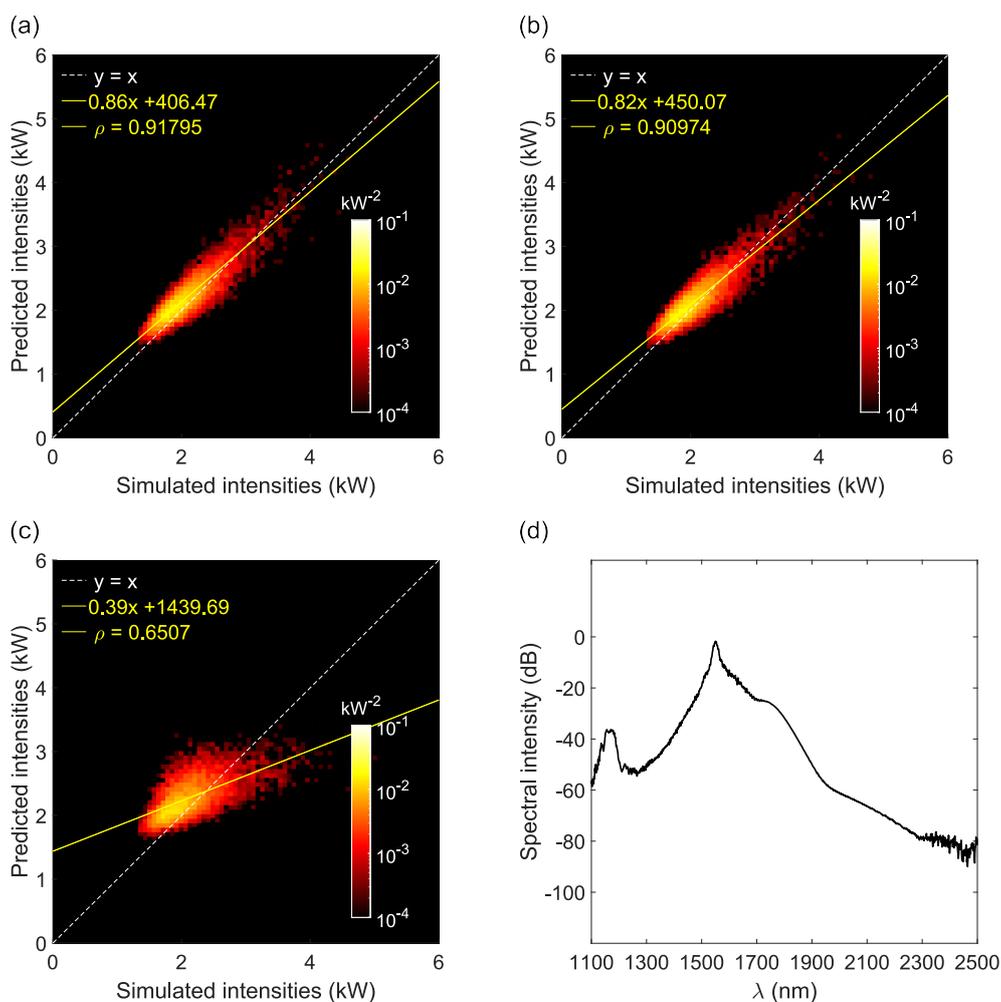


FIGURE 4.14 – Test du réseau de neurones pour la cavité à spectre large (avec HNFL). Prédiction du maximum d'intensité temporelle effectuée par le réseau de neurones en fonction de la valeur réelle de ce maximum. Les différentes figures correspondent à différents seuils de plage dynamique pour les données fournies au réseau : (a) montre la corrélation obtenue entre les prédictions et les valeurs réelles lorsque le réseau est formé avec des données non-tronquées, (b) montre cette corrélation lorsque les données sont tronquées à une plage dynamique de 80 dB, et (c) lorsque les données sont tronquées à une plage dynamique de 50 dB. La ligne jaune est la régression linéaire du nuage de points, et la ligne pointillée blanche est la diagonale vers laquelle doit tendre la ligne jaune. La figure (d) montre un profil d'intensité typique obtenu lors d'une simulation.

L'évolution du coefficient de corrélation en fonction de la plage dynamique des données est décrit plus en détail dans la figure 4.15. On constate que dans le régime à spectre étroit (sans fibre HNLF), la dégradation du coefficient de corrélation ne se fait qu'à partir d'une plage dynamique d'environ 60 dB (qui est la limite de ce qui est atteignable expérimentalement), mais que le réseau reste performant ($\rho > 0,9$) jusqu'à 40 dB environ. Dans le régime à spectre large (avec fibre HNLF), les performances se dégradent plus vite, si bien qu'elles ne sont acceptables que pour une plage dynamique de 80 dB.

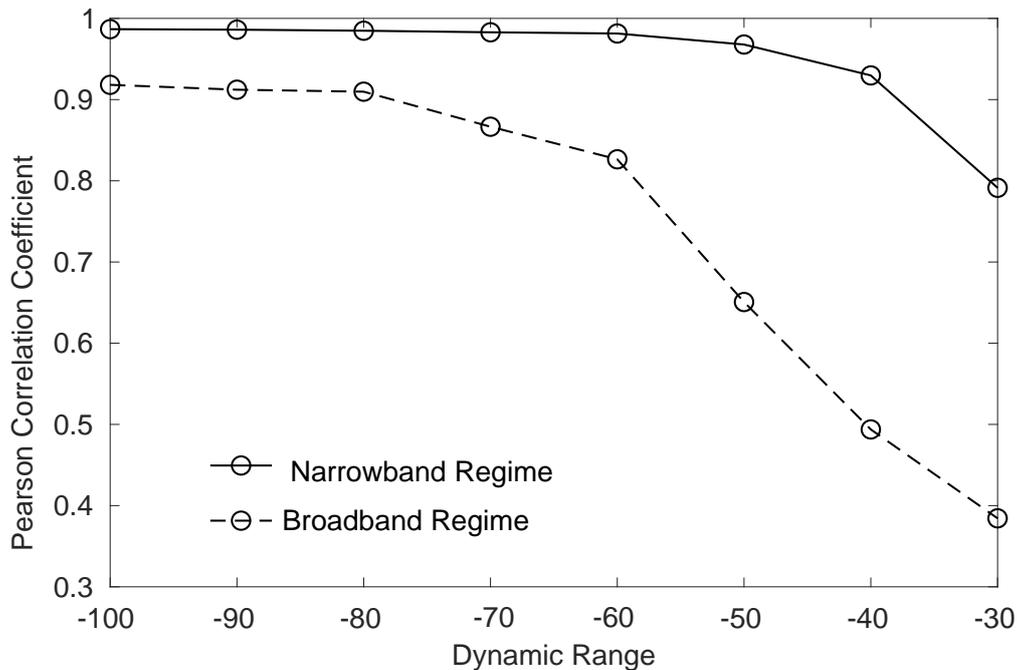


FIGURE 4.15 – Évolution du coefficient de corrélation de Pearson entre les résultats prédits par le réseau de neurones et ceux fournis par intégration numérique en fonction de la plage dynamique des données.

En plus de mesurer la capacité prédictive du réseau de neurones, nous nous sommes également intéressés à ses performances statistiques, et notamment à sa capacité à rendre compte des valeurs extrêmes de pic d'intensité temporelles ; les vagues scélérates optiques. Pour cela, nous affichons la densité de probabilité associée au maximum d'intensité temporelle calculé à partir des données brutes et à partir des prédictions du réseau. Ces résultats sont visibles dans la figure 4.16, où sont présentées les densités de probabilité des maxima d'intensité temporelle issues de l'intégration numérique et des prédictions du réseau de neurones. On y constate que le réseau de neurones parvient très bien à reconstruire l'ensemble de la densité des maxima, y compris les valeurs les plus extrêmes, et ce, jusqu'à un seuil de troncature de plage dynamique des données d'entrées de 50 dB en régime à spectre étroit et 80 dB en régime à spectre large.

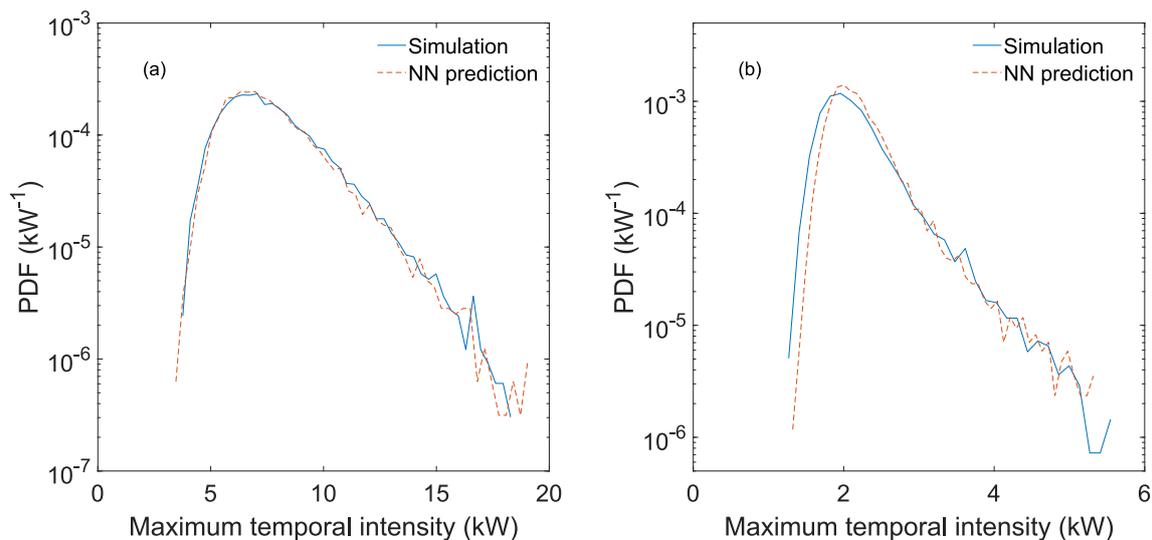


FIGURE 4.16 – Densité de probabilité des maxima d'intensité temporelle issue des simulation (ligne bleue) et reconstruite par le réseau de neurones (ligne rouge en tirets) : (a) dans le régime à spectre étroit (sans HNLF) avec un plage de dynamique de 50 dB et (b) dans le régime à spectre large (avec HNLF) avec un plage de dynamique de 80 dB.

Il convient également de noter que tous les résultats présentés ci-dessus ont été obtenus en utilisant la même largeur de bande de convolution des données et de *binning* de 1 nm. Des tests supplémentaires ont montré que la précision du réseau diminue à mesure que cette largeur de bande augmente, mais que des coefficients de corrélation acceptables de 0,9 peuvent encore être obtenus avec des largeurs de bande allant jusqu'à l'ordre de 10 nm. Il est également naturel de se demander si un réseau de neurones peut corréler les propriétés spectrales et temporelles sur une gamme plus large d'énergies d'impulsion, obtenues en faisant varier l'énergie de saturation E_{sat} . À cet égard, il a tout d'abord été observé que la diminution de E_{sat} à des valeurs proches de 1 nJ fait entrer le laser dans des régimes d'impulsions uniques et/ou d'états moléculaires qui peuvent présenter soit une dynamique stable soit des formes d'instabilité diverses [108, 107]; des réseaux de neurones ont déjà été utilisés pour corréler les propriétés spectrales et temporelles dans ces régimes [109], mais il s'agit d'un mode de fonctionnement très différent du régime NLP étudié ici. Dans le contexte des instabilités NLP, l'approche décrite jusqu'à maintenant a donc été testée pour des valeurs plus élevées de E_{sat} (jusqu'à approximativement 10 nJ), et le réseau de neurones a été formé avec succès pour donner des résultats comparables à ceux présentés plus haut. La réalisation de simulations pour générer de grands ensembles de données d'entraînement à des valeurs encore plus grandes de E_{sat} est coûteuse en termes de calcul, mais il serait surprenant que cette approche ne fonctionne plus pour d'autres valeurs de E_{sat} où la même dynamique NLP (i.e. avec des profils temporels et spectraux similaires) est observée.

4.3 Extraction de l'information

L'un des problèmes majeurs des réseaux de neurones est leur manque d'interprétabilité. En effet, comme il a été expliqué au chapitre 2, ces derniers peuvent être considérés comme des fonctions non-linéaires paramétrées très complexes qui transforment les données d'entrées pour obtenir des données de sorties, et dont on cherche à optimiser les paramètres; ce processus d'optimisation est en tout points similaire à une régression non-linéaire, le modèle de réseau de neurones étant un modèle particulièrement flexible. Or ce processus a lieu dans un espace de très haute dimension (le nombre de paramètres d'un réseau de neurones devenant rapidement très grand, voir 2.12), si bien que le nombre de paramètres utilisés ne correspond pas forcément (et c'est même une quasi-certitude) au nombre de degrés de libertés intrinsèque du système étudié, qui est plus faible (a priori) [110]; les modèles de réseau de neurones sont donc presque toujours sur-paramétrés. Cela rend l'interprétation des paramètres, et donc des performances du modèle très compliquée, puisque les-dits paramètres n'ont pas d'interprétation "physique" directe, et il est donc délicat de mesurer/quantifier quelle partie des données d'entrée est pertinente ou non dans la prédiction des sorties.

Néanmoins, afin de déceler au mieux quelle partie des spectres est la plus informative pour la prédiction du réseau de neurones, la zone spectrale d'intérêt a été réduite, c'est-à-dire que les données d'entrée ont été limitées aux points compris dans un intervalle $[\lambda_{min}; \lambda_{max}]$. Pour cela, le vecteur d'entrée est multiplié par une fonction porte centrée en λ_c et de largeur $\Delta\lambda$, et il est alors possible de sélectionner la région du profil d'intensité spectrale désirée en faisant varier ces deux paramètres; on a donc

$$\left\{ \begin{array}{l} \lambda_c = \frac{\lambda_{max} + \lambda_{min}}{2} \\ \Delta\lambda = \lambda_{max} - \lambda_{min} \end{array} \right. \iff \left\{ \begin{array}{l} \lambda_{min} = \lambda_c - \frac{\Delta\lambda}{2} \\ \lambda_{max} = \lambda_c + \frac{\Delta\lambda}{2} \end{array} \right. \quad (4.5)$$

Pour chaque paire $\{\lambda_c; \Delta\lambda\}$, le réseau de neurones est alors entraîné en conservant la même architecture à chaque fois (la même que dans la section précédente), et on mesure à quel point le réseau parvient à prédire le maximum d'intensité temporelle (avec les mêmes métriques que décrites dans la section précédente). Si les vecteurs d'entrée d'entraînement sont tronqués par la fonction porte, ce n'est pas le cas des vecteurs d'entrée de test qui sont eux fournis au réseau de neurones dans leur entièreté; tous les vecteurs d'entrée (pour l'entraînement et le test) ont néanmoins subi les étapes de pré-traitement décrites plus haut (convolution par une gaussienne et regroupement dans des classes de longueur 1 nm). Faire cela permet de vérifier la robustesse au bruit du réseau de neurones; en effet, les poids du réseau qui se retrouvent en pondération d'une valeur nulle ne seront pas mis à jour (la mise à jour d'un poids étant proportionnelle à la valeur du neurone qu'il pondère, voir 2.27), et resteront donc aléatoires. Ainsi, lors de la phase de test, les caractéristiques d'un vecteur d'entrée qui se retrouvent pondérées par un poids qui n'a pas été optimisé sont comme bruitées par ce poids, et donc l'obtention de bonnes performances du réseau malgré ce bruit est un signe de bonne capacité de généralisation et de robustesse. La figure 4.17 montre la procédure de sélection de la région dont on veut savoir si elle contient de l'information ou non.

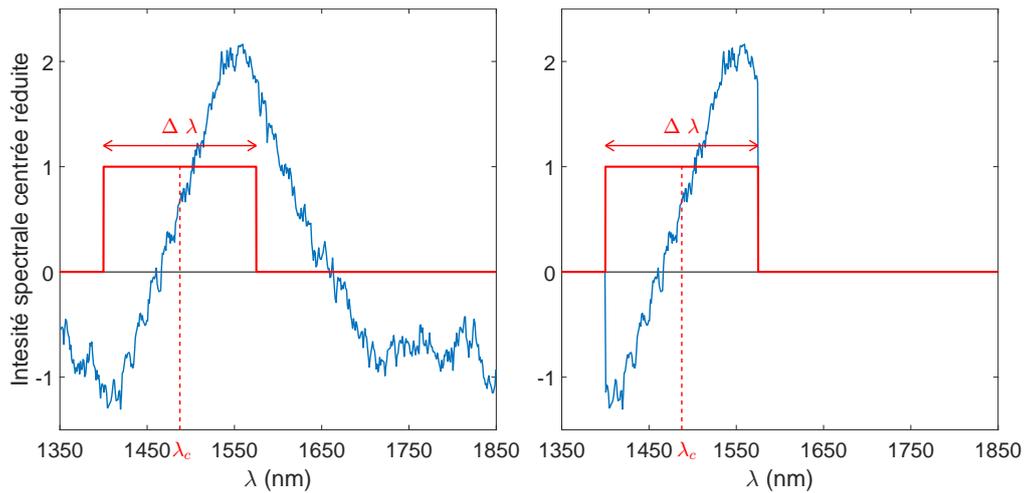


FIGURE 4.17 – Schéma de la procédure de sélection de la zone d'intérêt des données d'entrée. Le profil d'intensité spectrale est multiplié par une fonction porte centrée en λ_c et de largeur $\Delta\lambda$.

Ainsi, pour chaque paire $\{\lambda_c ; \Delta\lambda\}$, le coefficient de corrélation de Pearson ρ est calculé de la même manière que précédemment pour les trois régimes présentés : le régime d'instabilité de modulation pure, le régime de cavité *noise-like pulse* sans fibre HNLf, et le régime de cavité *noise-like pulse* avec fibre HNLf. À chaque fois, l'architecture utilisée est la même que celle utilisée lors des travaux précédents (pour chaque régime), et avec des données dont la plage dynamique n'a pas été tronquée (environ 100 dB); cela aboutit aux figures 4.18 (MI), 4.19 (NLP à spectre étroit), et 4.20 (NLP à spectre large).

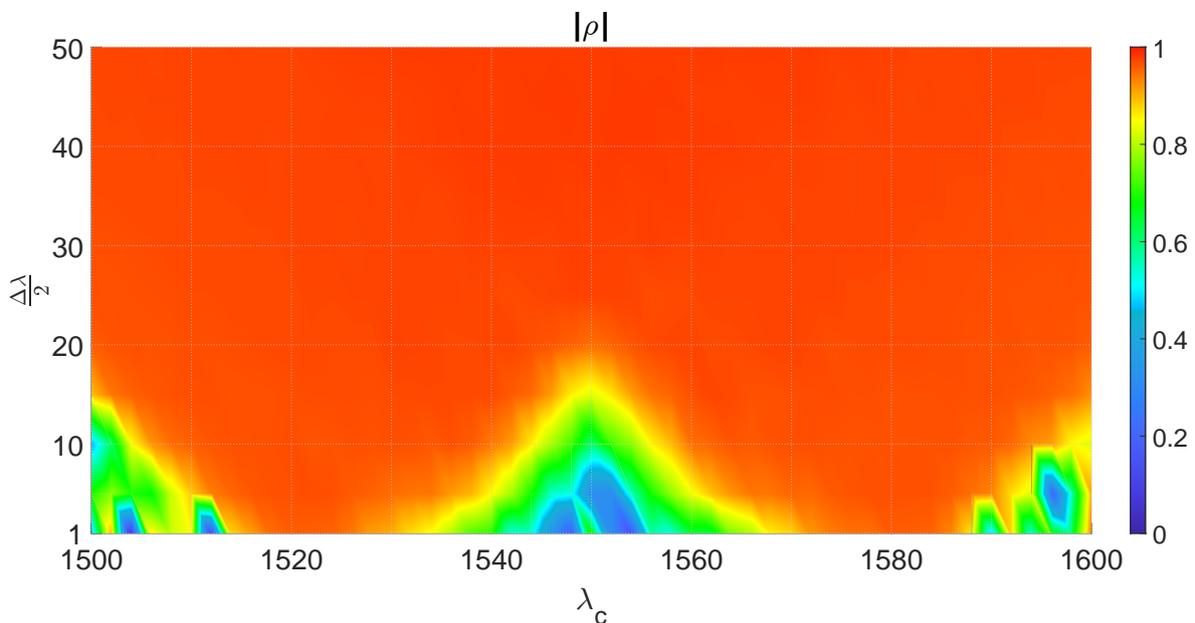


FIGURE 4.18 – Évolution du coefficient de corrélation de Pearson ρ en fonction de λ_c et $\Delta\lambda$ dans le régime d'instabilité de modulation (MI) pure. L'axe des abscisses représente la fréquence centrale λ_c de porte et l'axe des ordonnées la demi-largeur de la porte $\Delta\lambda/2$.

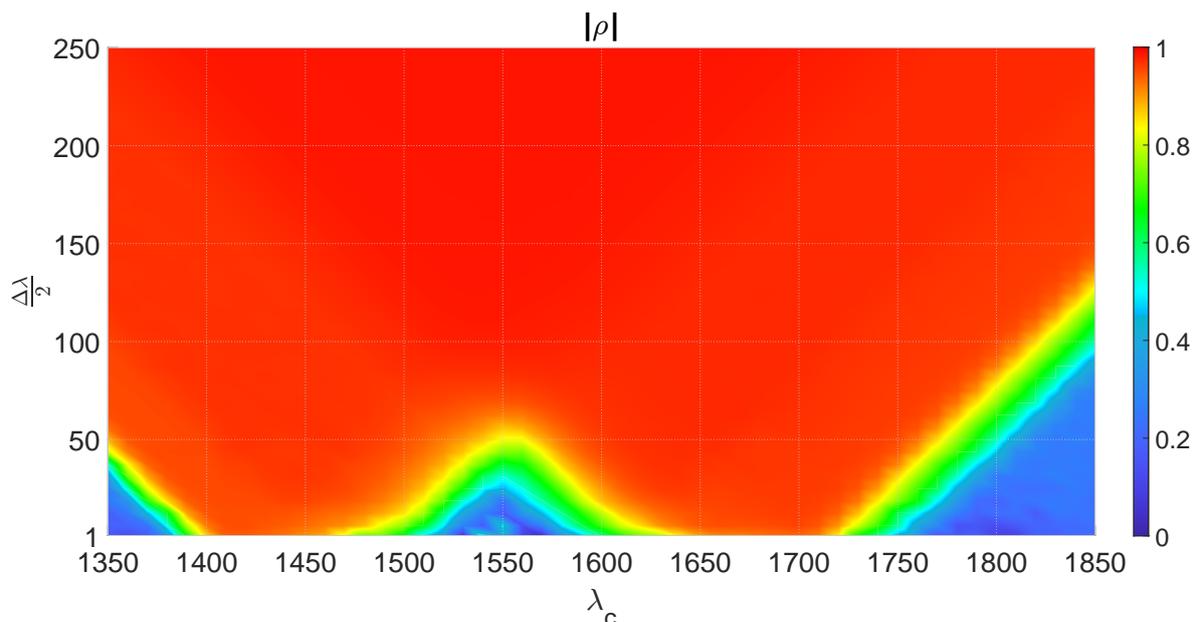


FIGURE 4.19 – Évolution du coefficient de corrélation de Pearson ρ en fonction de λ_c et $\Delta\lambda$ dans le régime *noise-like pulse* (NLP) sans HNLF. L'axe des abscisses représente la fréquence centrale λ_c de porte et l'axe des ordonnées la demi-largeur de la porte $\Delta\lambda/2$.

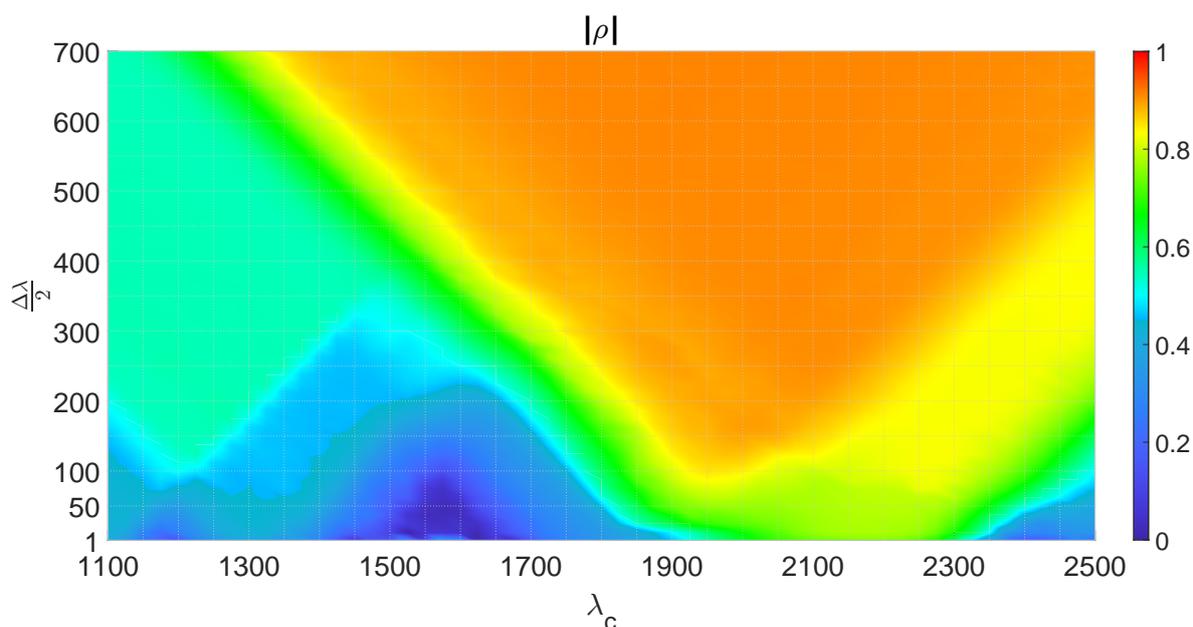


FIGURE 4.20 – Évolution du coefficient de corrélation de Pearson ρ en fonction de λ_c et $\Delta\lambda$ *noise-like pulse* (NLP) avec HNLF. L'axe des abscisses représente la fréquence centrale λ_c de la porte et l'axe des ordonnées la demi-largeur de la porte $\Delta\lambda/2$.

On constate que dans les cas de l'instabilité de modulation ainsi que de la cavité sans fibre HNLF (figures 4.18 et 4.19), les zones de faible corrélation sont la région localisée à la longueur d'onde centrale de l'impulsion λ_0 (1550 nm) et les régions localisées aux limites de la fenêtre spectrale, à savoir vers 1500 et 1600 nm pour le MI, et vers 1350 et 1850 nm pour le NLP sans fibre HNLF. L'information du profil d'intensité spectrale nécessaire et suffisante à pouvoir prédire le maximum d'intensité temporelle semble donc se situer dans les intervalles aux abords

du bruit de fond à 100 dB, à savoir la zone $I \simeq [1515 ; 1530] \cup [1570 ; 1585]$ nm pour le MI, et la zone $I \simeq [1400 ; 1450] \cup [1650 ; 1710]$ nm pour le NLP sans fibre HNLf. En effet, les régions de faible corrélation sont séparées des zones de haute corrélation par des lignes droites de corrélation environ 0,9 (lignes jaunes/orange) de pente 45° , ce qui montre que tant que l'information contenue dans la région I n'est pas prise en compte, la corrélation reste faible, et, réciproquement, que la corrélation devient élevée dès que cette information est prise en compte.

Une manière de se représenter le phénomène est d'imaginer un observateur n'ayant accès qu'à l'information située dans une zone centrée en λ_c , et dont le "champ de vision spectral" $\Delta\lambda$ grandisse à mesure que le "temps" passe. Dans cette expérience de pensée, l'observateur commence donc avec un "champ de vision" initial de 1 nm de part et d'autre de sa "position" λ_c , et n'a donc accès qu'à de l'information très locale. À mesure que le "temps" (proportionnel à $\Delta\lambda$) passe, il acquiert de l'information provenant de points du spectre situés loin de lui ; des points situés à au plus $\Delta\lambda/2$ nm de part et d'autre de sa "position". Les lignes de corrélation constante étant orientées à 45° (sur une grille carrée), cela signifie que l'information nécessaire et suffisante est contenue dans l'intervalle I décrit plus haut ; les figures 4.18 et 4.19 présentent donc ce qui peut s'interpréter comme un "cône de lumière" issu des longueurs d'onde $\{1520 ; 1580\}$ nm pour le MI et $\{1425 ; 15865\}$ nm pour le NLP à spectre étroit.

Dans le cas de la cavité avec fibre HNLf (figure 4.20), il semble à première vue que l'information utile soit principalement contenue dans l'intervalle $I \simeq [2000 ; 2300]$ nm ; cependant l'interprétation est plus compliquée. En effet, contrairement au cas précédent, la zone de bonne performance ($\rho \geq 0,9$, zone en orange) semble n'être constituée que de points qui ne contiennent pas assez d'information individuellement pour pouvoir reconstruire le maximum d'intensité temporelle. En reprenant l'expérience de pensée décrite plus haut, un observateur situé en $\lambda_c = 1950$ nm et ayant un "champ de vision" de 1 nm de part et d'autre de sa "position" ne parviendrait à générer des résultats que faiblement corrélés avec les résultats attendus ($\rho \simeq 0,55$). À mesure que son "champ de vision" grandit, il finit par avoir accès à l'information située 100 nm de part et d'autre de sa "position" ($\Delta\lambda/2 = 100$ nm), et est alors en mesure de générer des résultats bien corrélés avec les résultats attendus ($\rho \simeq 0,9$). Or aucun des points du spectre qu'il observe ne fournit de bonne prédiction individuellement ; tous les points situés dans son "champ de vision" $[1950-100 ; 1950+100]$ nm fournissent individuellement une corrélation inférieure à 0,6.

Il semble donc que dans le régime à spectre large, la capacité du réseau de neurones à générer des résultats bien corrélés avec les résultats attendus ne soit pas simplement due à sa capacité à sélectionner les régions d'intérêt dans les données d'entrées (comme dans le régime à spectre étroit), mais soit due à sa capacité à synthétiser l'information provenant de plusieurs sources, chacune d'entre-elles ne contenant pas suffisamment d'information pour atteindre l'objectif désiré (ici, prédire le maximum d'intensité temporelle). Cette capacité de synthèse de l'information est certainement l'aspect le plus intéressant et complexe des réseaux de neurones, et reste encore aujourd'hui un défi pour la recherche en intelligence artificielle.

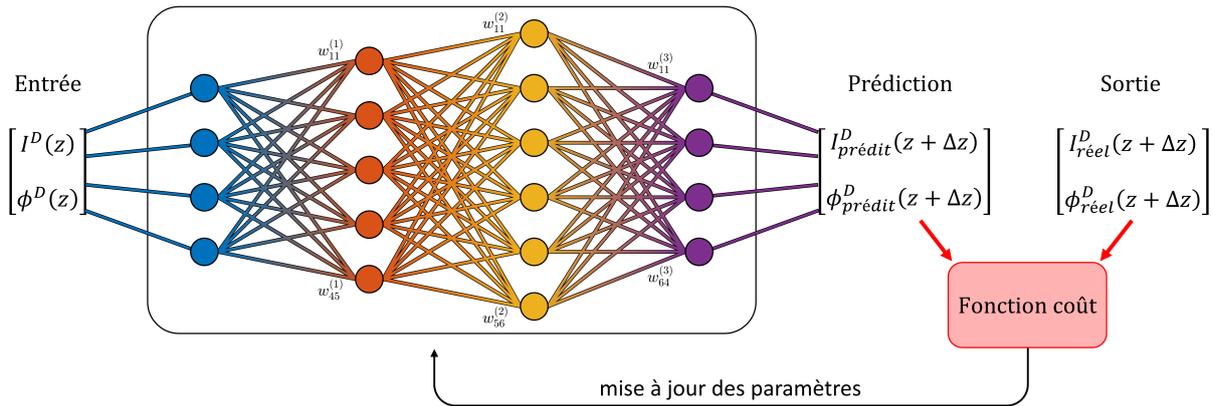
4.4 Intégration de l'ESNLG par réseau de neurones

Comme nous l'avons vu dans le chapitre 2, un réseau de neurones à propagation avant n'est rien de plus qu'une fonction sophistiquée dont les paramètres internes s'ajustent automatiquement, et qui opère dans un espace de (très) grande dimension. Or un intégrateur numérique peut également être considéré de la sorte ; un intégrateur numérique est une fonction qui prend en entrée un vecteur décrivant l'état du système à l'instant t et/ou en position \vec{x} et ressort un vecteur décrivant l'état du système à l'instant $t + dt$ et/ou en position $\vec{x} + d\vec{x}$. De plus, de nombreuses méthodes algorithmiques ont vu le jour ces dernières années utilisant des outils d'apprentissage machine pour se substituer aux méthodes traditionnelles d'intégration numérique, dans le cadre de systèmes chaotiques à haute dimensionnalité [111], ou encore dans le cadre de l'équation de Schrödinger non-linéaire [112]. Ces méthodes sont dites sans modèle (*model-free*) puisqu'elles n'utilisent aucune connaissance a priori sur le modèle physique/mathématique sous-jacent décrivant le comportement des données, contrairement aux modèles dits *physics-informed* (PINNs), pour lesquels des équations physiques (de l'information a priori) sont encodées dans la fonction coût [113]. Les travaux visant à utiliser les réseaux de neurones en lieu et place de schémas d'intégration numérique classique utilisent souvent des réseaux de neurones récurrents (évoqués succinctement dans le chapitre 2), ou des variations de ceux-ci (les *Long Short-Term Memory*, ou LSTM) [114], ces modèles de réseaux étant particulièrement adaptés à l'analyse de données temporelles [84]. Cette utilisation des réseaux de neurones en substitution des algorithmes d'intégration classique se répand de plus en plus, de nouvelles architectures et modèles étant développés et testés (*residual network*, réseaux récurrents, *neural ODE*, ...), l'intérêt de ces approches étant principalement le gain de flexibilité et de mémoire [115, 116]. C'est ainsi que l'idée d'utiliser ces méthodes d'apprentissage machine dans le cas du régime supercontinuum (régime particulièrement chaotique) semblait tout-à-fait appropriée.

Pour ce faire, un réseau de neurones récurrent est formé sur un grand nombre de données afin qu'il puisse, connaissant le profil spectral (intensité et phase) à une distance z arbitraires dans la fibre, fournir le profil spectral après un incrément de propagation Δz . Pour cela, les vecteurs d'entrée fournis sont des profils spectraux (intensité et phase) à des distance z , et les vecteurs de sorties sont les profils spectraux à la distance $z + \Delta z$ issus d'une intégration classique (figure 4.21a). Une fois le réseau formé, un profil spectral initial (sech par exemple) lui est fourni en entrée, et l'algorithme renvoie le profil spectral à une distance Δz , ou du moins ce qu'il prédit comme devant être le profil spectral. Le profil obtenu est alors réinjecté en entrée du réseau, qui fournit donc le profil spectral à $2\Delta z$. En itérant l'opération, le réseau est alors capable d'agir comme un intégrateur (figure 4.21b), dont la précision dépend essentiellement de la convergence de la phase de formation, et donc du bon choix d'architecture, d'hyperparamètres, du nombre et de la qualité des données d'entraînement.

Dans [117], les auteurs ont montré qu'il était bel et bien possible de former un réseau de neurones récurrent pour qu'il puisse se substituer à un algorithme d'intégration classique dans la génération de supercontinuum. Dans cet article, les auteurs ont fait une étude quantitative quant aux avantages et inconvénients d'utiliser une telle méthode, et il en ressort que bien que les performances soient dégradées par rapport à une intégration classique, elles ne le sont que très peu, et cette dégradation des performances est largement compensée par un gain de temps et de mémoire conséquents ; une fois formé (ce qui prend environ deux heures), le réseau est capable de fournir un spectrogramme en environ 2 millisecondes, là où un algorithme classique met environ 12 secondes. Le temps de formation du réseau ne rend ceci avantageux que si on intègre plus de 600 profils, mais une fois le réseau formé, ce dernier peut être utilisé dans d'autres configurations. Pour cela, les données d'entrée doivent être complétées de paramètres

a) Phase de formation de réseau :



b) Phase de prédiction de réseau :

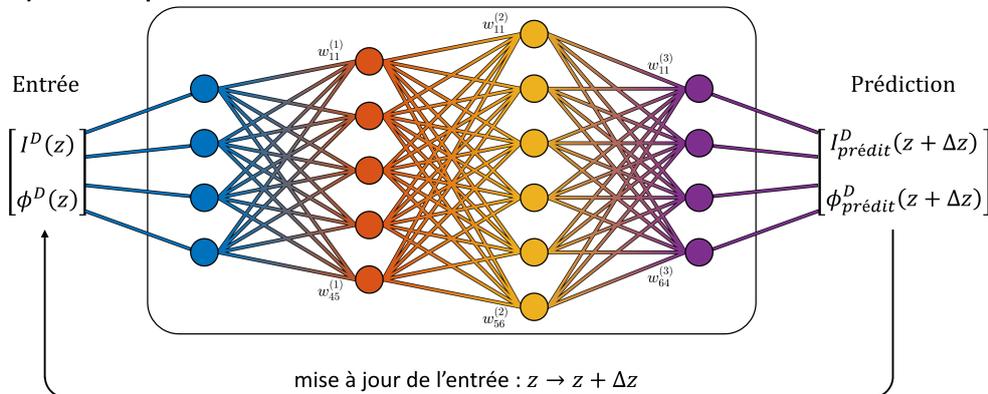


FIGURE 4.21 – Schéma du processus de formation d'un réseau de neurones à propagation avant. Lors de la phase de formation (figure a)), les paramètres du réseau (poids et biais) sont mis à jour afin que ce dernier puisse se substituer à un intégrateur numérique classique (figure b)).

décrivant les caractéristiques de l'impulsion que l'on souhaite propager (sa FWHM, la valeur de son chirp, ses coefficients de dispersions β_k , ou encore son coefficient non-linéaire γ); il est donc préférable de former le réseau sur des données issues de l'équation de Schrödinger généralisée normalisée (1.63) afin de couvrir un large panel de scénarios possibles. De plus, les ressources mémoires nécessaires pour le réseau de neurones sont quasiment 25 fois plus faibles que celles nécessaires à une intégration numérique classique par méthode *split-step Fourier*.

De plus, les mêmes auteurs ont montré que des résultats similaires pouvaient être obtenus avec un réseau de neurones à propagation avant [118]. Nous avons donc reproduit les résultats de ce papier, à la différence que la proportion utilisée entre le nombre de données d'entraînement et le nombre total de données était bien plus faible que celle utilisée par les auteurs originaux; ces derniers ont en effet généré un total de 1500 données, dont 1400 ont servi pour l'entraînement et 100 pour le test (soit un ratio de $100/1400 \simeq 0,07\%$), alors que nous avons généré un total de 1400 données, dont 1000 ont servi pour l'entraînement et 400 pour le test (soit un ratio de $400/1000 = 0,4\%$). Cette différence de proportion entre données d'entraînement et données de test joue évidemment un rôle crucial sur la mesure des performances d'un réseau de neurones, puisque ce rapport détermine la nécessité qu'a le réseau à généraliser à des données qu'il n'a jamais vu. 1400 profil spectraux (intensité et phase) de supercontinuum ont donc été générés en faisant varier la FWHM et la puissance initiale P_0 respectivement de

$100 \pm 30\% = [70;130]$ fs et de $1100 \pm 30\% = [770;1430]$ W, en accord avec [118]. La distance de propagation est de 20 cm, avec une discrétisation spatiale de 1 mm (pour un total de 200 points selon z); les paramètres de simulation sont donnés dans le tableau suivant :

L	β_2	β_3	β_4	β_5	γ	λ_0	P_0	FWHM
[m]	[fs ² /mm]	[fs ³ /mm]	[fs ⁴ /mm]	[fs ⁵ /mm]	[W ⁻¹ .m ⁻¹]	[nm]	[W]	[fs]
0,2	-5.9	42,1	-125	245	0,1	830	770 → 1430	70 → 130

TABLE 4.3 – Paramètres physiques des simulations en régime de supercontinuum : paramètres de la fibre à gauche et paramètres de l’impulsion d’entrée à droite

Cela a donc permis de fournir des données d’entraînement au réseau de neurones, une donnée d’entrée étant un vecteur contenant l’intensité et la phase spectrale à une certaine distance z , et la donnée de sortie cible associée étant un vecteur contenant l’intensité et la phase spectrale à la distance $z + \Delta z$; avec 1400 profil spectraux, sous-échantillonnés à une discrétisation spatiale de 2 mm sur une distance de 20 cm, cela représente un total de $1400 \times 100 = 140000$ données, dont 100000 ont servi pour l’entraînement et 40000 pour le test. Un exemple de résultat obtenu après formation du réseau est visible figure 4.22.

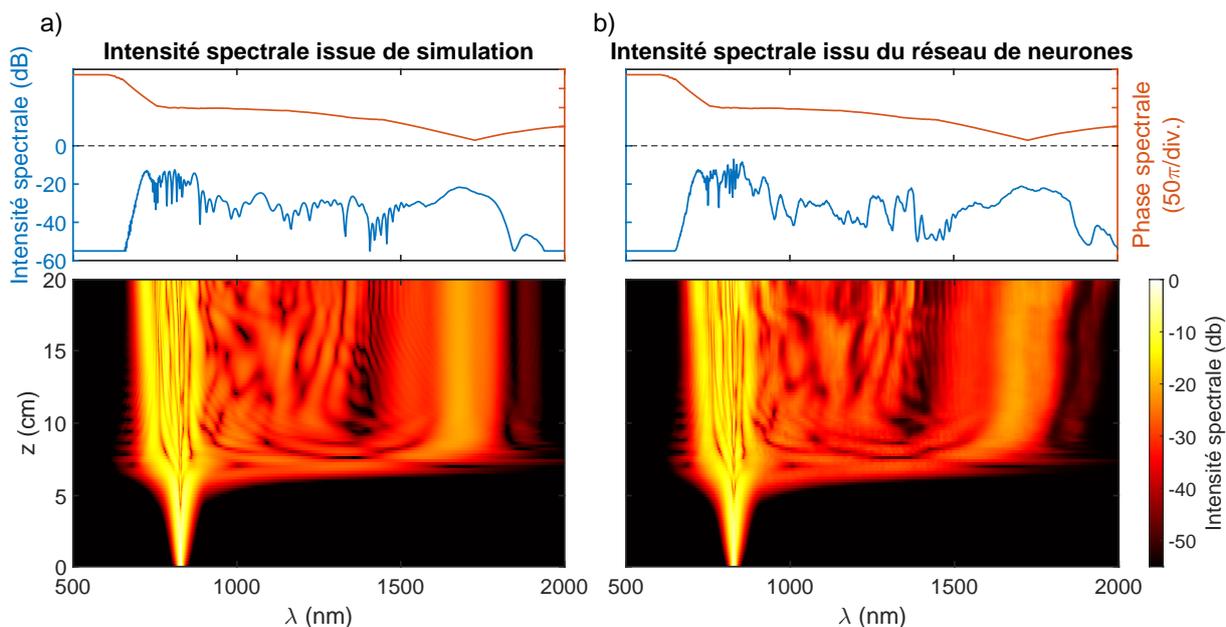


FIGURE 4.22 – Évolution spectrale le long de la fibre (en bas) et sortie de fibre (en haut) après a) une intégration numérique classique (*split-step Fourier*) et b) une intégration par réseau de neurones. Les figures du haut montrent l’intensité et la phase spectrale alors que les figures du bas ne montrent que l’intensité.

La figure 4.22a) montre l’évolution spectrale de l’impulsion le long de la fibre (en bas) et en sortie de fibre (en haut) après intégration numérique classique, alors que la figure 4.22b) montre l’évolution spectrale de l’impulsion le long de la fibre (en bas) et en sortie de fibre (en haut) après intégration par réseau de neurones. Malgré quelques dissimilarités, la correspondance entre l’intégration numérique "idéale" (intégration numérique classique par méthode de *split-step Fourier*) et celle résultante de l’intégration par réseau de neurones est frappante. Le

réseau de neurones parvient à correctement reconstruire l'élargissement spectral, l'émergence du soliton Raman, ainsi que les différentes turbulences occasionnées par l'interaction entre les solitons.

Dans l'optique de compléter et approfondir ces travaux, nous avons voulu comprendre quels facteurs pouvaient influencer le fait que certains spectres avaient des erreurs d'intégration plus faibles que d'autres. Pour cela, nous avons étudié l'évolution de cette erreur au cours de la propagation dans l'espoir de pouvoir discriminer les spectres plus facilement intégrables de ceux qui le sont moins ; une telle analyse peut s'effectuer au travers de l'étude de la racine de l'erreur quadratique moyenne (ou RMSD pour *root-mean-square deviation*). La RMSD moyennée sur la longueur de la fibre et sur les fréquences a donc été calculée pour chaque paramètre d'impulsion initiale (FWHM et P_0), ce qui a permis d'obtenir la figure 4.23

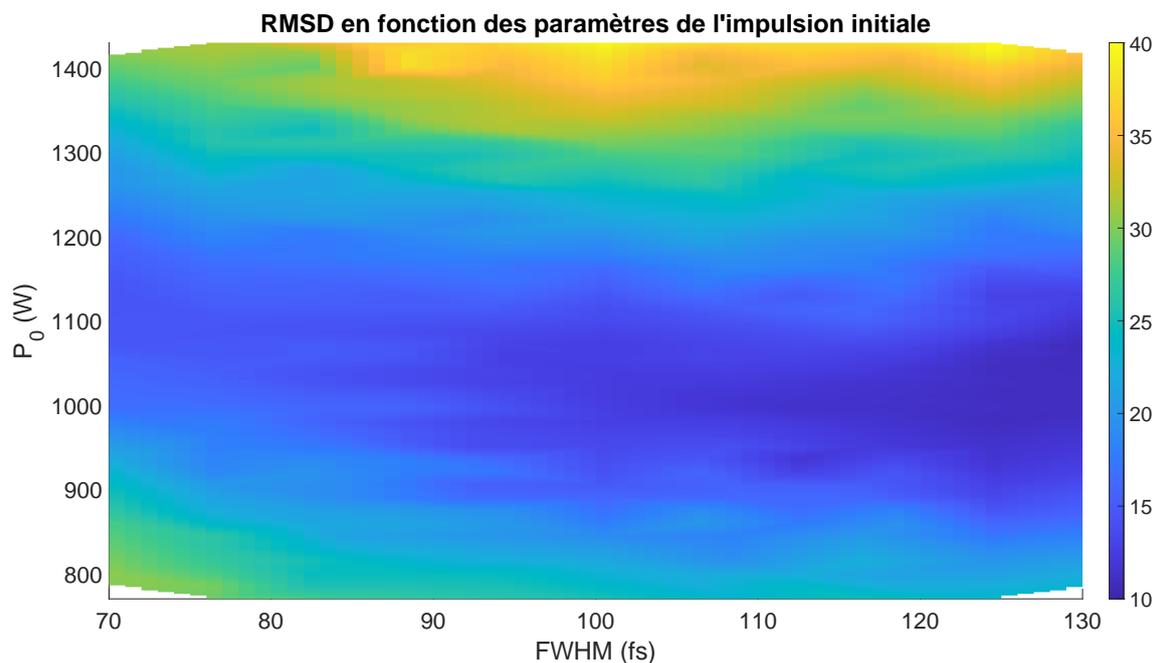


FIGURE 4.23 – Racine de l'erreur quadratique moyenne pour chaque paire $\{FWHM; P_0\}$ de paramètres de l'impulsion d'entrée. Chaque point de cette figure est une moyenne sur z et λ de la RMSD obtenue pour une impulsion.

On constate que la RMSD varie assez peu en fonction de la FWHM, mais varie assez fortement en fonction de la puissance. En effet, la région où P_0 est dans l'intervalle $[950; 1150]$ W a une RMSD inférieure à 15, et qui correspond à un bon accord entre l'intégration numérique attendue et celle effectuée par le réseau de neurones ; on remarque que plus la FWHM est élevée, plus la RMSD est faible, mais le phénomène reste assez léger. Cependant, lorsqu'on s'éloigne de cette gamme de puissance, la RMSD augmente de manière significative, aussi bien pour les puissances plus faibles que pour les plus élevées ; les puissances plus élevées fournissent néanmoins les performances les plus médiocres. Il semblerait donc que la capacité d'un réseau de neurones à propagation avant à se substituer à un algorithme d'intégration numérique classique soit plus conditionné par la gamme de puissance de l'impulsion initiale que par la largeur de cette dernière.

De plus, nous nous sommes intéressé à l'évolution des performances du réseau de neurones en fonction de la position dans la fibre et de la longueur d'onde afin de discerner quelles parties du profil spectral étaient mieux décrites. La RMSD a donc été calculée pour l'intensité et la phase à chaque position z de la fibre et à chaque longueur d'onde λ , puis moyennée sur les

différentes conditions initiales (FWHM et P_0), ce qui a aboutit à la figure 4.24. Il apparaît clairement que la région du soliton Raman est la plus compliquée à reproduire, aussi bien en intensité qu'en phase. Cela vient sûrement du fait que cette région est celle présentant la plus grande variation d'une simulation à l'autre, rendant la généralisation plus compliquée; cette variation peut se mesurer avec l'écart-type, visible dans la figure 4.25. En comparant cette dernière avec la figure 4.24, il apparaît effectivement que le variation d'intensité et de phase d'une simulation à l'autre soit un facteur influent quant aux bonnes performances du réseau de neurones, bien que certaines zones où l'écart-type est grand restent néanmoins plutôt bien reproduites; c'est le cas de la zone située à environ 5 cm, où la phase initiale du soliton Raman présente une forte variation d'une simulation à l'autre (grand écart-type), mais une faible erreur de prédiction (faible RMSD).

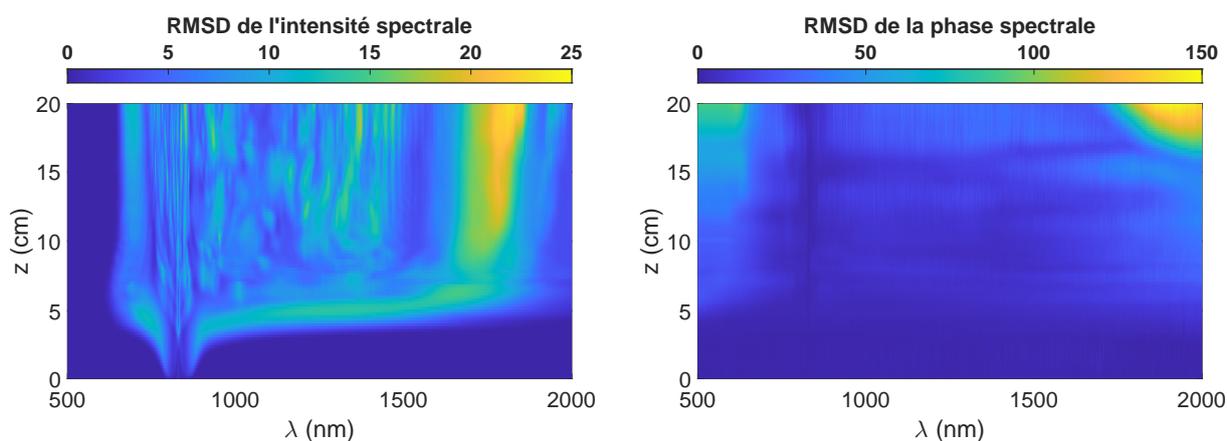


FIGURE 4.24 – Racine de l'erreur quadratique moyenne calculée à chaque position z dans la fibre et pour chaque longueur d'onde λ . Cette RMSD est moyennée sur toutes les conditions initiales (FWHM et P_0).

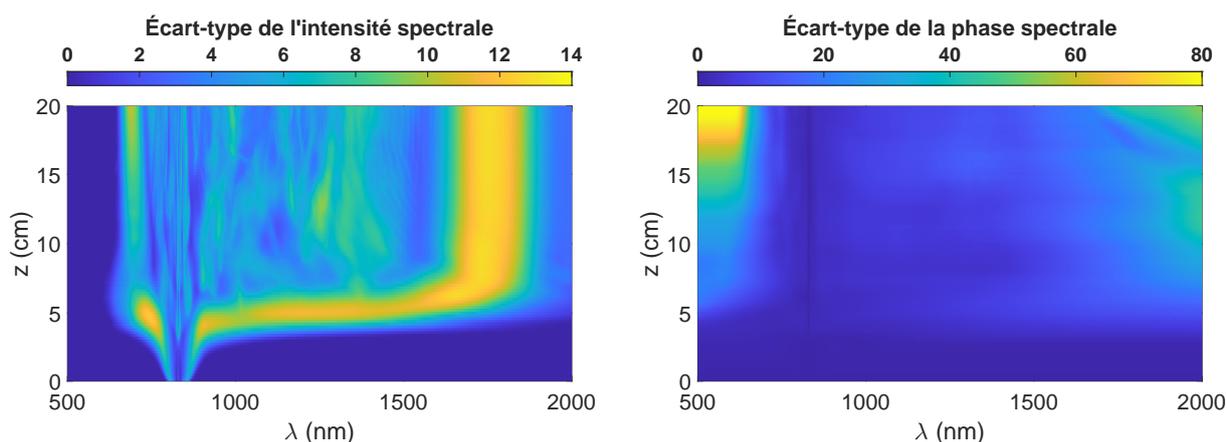


FIGURE 4.25 – Écart-type calculé à chaque position z dans la fibre et pour chaque longueur d'onde λ . Cet écart-type est moyenné sur toutes les conditions initiales (FWHM et P_0).

L'analyse détaillée des raisons qui font que le réseau parvient ou non à bien reproduire la dynamique de l'ESNL est très compliquée, mais il est bon de noter que l'étude qui vient d'être présentée a été menée sur d'autres plages de paramètres (β_2 , P_0 , ...) [118] et qu'une bonne convergence a été obtenue à chaque fois.

Ainsi, un réseau de neurones à propagation avant est capable de se substituer à un intégrateur numérique classique de manière tout à fait satisfaisante dans le régime de propagation supercontinuum, bien que les données d'entraînement semblent devoir être représentatives d'un ensemble suffisant de l'espace des paramètres physiques que l'on cherche à décrire, tous ces paramètres n'ayant pas la même influence. Il semblerait en effet qu'un réseau de neurones soit plus sensible à la puissance crête de l'impulsion d'entrée qu'à sa FWHM, puisque une même variation de 30% dans l'espace des paramètres physiques sur ces deux quantités ne donne pas les mêmes performances de prédiction/généralisation ; une légère variation de FWHM ne perturbe que peu les performances du réseau, alors qu'une légère variation de puissance les perturbe de manière significative.

Conclusion

"Les tentatives de création de machines pensantes nous seront d'une grande aide pour découvrir comment nous pensons nous-mêmes."

– Alan Turing, *Conférence à la BBC, 1951*

Au cours de cette thèse, l'exploration de ce nouvel outil qu'est le réseau de neurones a permis de prolonger les travaux qui avaient déjà été menés concernant l'analyse de données issues de la propagation de la lumière dans les diélectriques non-linéaires. Ces divers travaux avaient mis en avant les capacités des réseaux de neurones à prédire diverses propriétés physiques, notamment certaines caractéristiques temporelles compliquées à mesurer expérimentalement.

En effet, la complexité de mise en œuvre ainsi que les limitations des techniques telles que la lentille temporelle ou encore la DFT rendent les mesures temporelles en temps réel fastidieuses, là où des mesures spectrales peuvent être effectuées rapidement et avec une plage dynamique satisfaisante (jusqu'à 60 dB), la contre-partie étant que l'information sur la phase est perdue au cours du processus, rendant impossible le retour dans le domaine temporel par transformée de Fourier.

J'ai donc étudié et codé moi-même un algorithme de réseau de neurones (avec certaines extensions) duquel découle la grande majorité des architectures algorithmiques d'intelligence artificielle modernes les plus complexes. Cette étude m'a permis de me familiariser avec tous les concepts clés de cet algorithme et ainsi pouvoir l'utiliser au mieux. L'état de l'art concernant la prédiction des maxima d'intensité temporelle associés aux profils d'intensité spectrale générés par intégration numérique a ainsi pu être reproduit. Cette étape de validation du code ainsi que du protocole m'a alors mis sur la voie de la prolongation de ces travaux.

Nous nous sommes ainsi intéressés à la capacité d'un réseau de neurones à prédire le maximum d'intensité temporelle dans le cas du régime d'instabilité de modulation pure, tel qu'étudié analytiquement. Un grand nombre de données a alors été généré par intégration numérique conventionnelle (*split-step Fourier*), et l'analyse de ces données fût alors l'occasion de quantifier précisément la plage dynamique nécessaire aux données d'entrée du réseau (les profils d'intensité spectrale) pour pouvoir prédire avec une confiance suffisante le maximum d'intensité temporelle, dans le régime transitoire et stationnaire de l'instabilité de modulation. Il s'avère que cette plage dynamique seuil est aux environs de 40 dB dans le premier régime et 50 dB dans le second pour un coefficient de corrélation entre les prédictions du réseau et les valeurs issues de simulations de 0,9. De plus, la capacité à prédire l'émergence d'un événement de type vague scélérate optique a également été mesurée, et il ressort qu'une telle prédiction reste très limitée même pour ce type d'algorithme, puisqu'un spectre situé à plus de $0,1 L_{NL}$ ne fournit plus de prédiction fiable; l'aspect chaotique du régime d'instabilité de modulation rend les différents événements indépendants entre eux, à la manière d'une variable aléatoire sans mémoire.

Par la suite, nous nous sommes penchés sur le cas d'un système fibré moins conventionnel de cavité *noise-like pulse*, où l'extrême complexité des profils spectraux et temporels de ce système pouvait, a priori, laisser douter des capacités d'un réseau de neurones à analyser efficacement les données issues de cette propagation. Cependant, aussi bien dans le régime à spectre étroit (sans fibre HNLF) que dans le régime à spectre large (avec fibre HNLF), le réseau s'est montré capable de prédire là encore le maximum d'intensité temporelle avec une grande précision. Celle-ci se dégrade naturellement à mesure que la plage dynamique des profils d'intensité spectrale est réduite, et, là-encore, une quantification des performances de l'algorithme a pu être faite. De plus, dans le but de déterminer quelle partie du spectre permettait d'effectuer de bonnes prédictions, une analyse de la localisation de l'information a mis en lumière que le réseau de neurones était non-seulement capable de sélectionner la région d'intérêt du spectre, mais qu'il était également capable de synthétiser l'information, et ainsi construire une bonne prédiction à partir de la synthèse de données qui, prises individuellement, ne permettent pas une telle prédiction.

Enfin, l'utilisation d'un réseau de neurones à propagation avant en substitution d'un algorithme d'intégration classique s'est montrée fructueuse dans le cas de l'équation de Schrödinger non-linéaire généralisée. Dans cette configuration, le réseau de neurones est interprété comme une fonction de transfert qui établit une correspondance entre le vecteur d'état à une distance z et à une distance $z + \Delta z$ dans la fibre. Bien que l'intégration du réseau diffère quelque peu de l'intégration classique, ce dernier est capable, une fois formé, de fournir l'évolution complète (amplitude et phase) d'une impulsion initiale en seulement quelques millisecondes, comparées aux quelques secondes nécessaires à un algorithme classique. Une étude plus approfondie de l'erreur de la reconstruction faite par le réseau a permis de montrer que cette architecture n'est pas sensible de manière égale aux différents paramètres physiques du système, la puissance crête de l'impulsion initiale ayant une plus grande influence que la FWHM sur les performances du réseau de neurones.

Le réseau de neurones à propagation avant est donc un outil très versatile qui permet d'analyser une grande variété de données, et de fournir des corrélations intéressantes, notamment dans le domaine de l'optique non-linéaire ultra-rapide. Cependant, d'autres méthodes algorithmiques existent et ont fait leurs preuves dans l'étude de nombreux systèmes physiques, comme par exemple les algorithmes de *data-driven dominant balance*, où l'objectif est d'automatiser l'identification des processus physiques qui régissent la dynamique de différentes situations physiques et dans différents régimes [119, 120]. De plus, l'inconvénient majeur des réseaux de neurones, et de la plupart des outils d'apprentissage machine, est leur manque d'interprétabilité. En effet, lorsqu'un tel algorithme fournit un résultat (bon ou mauvais), il est compliqué de déceler quels éléments ont permis d'aboutir à ce résultat, ce qui est dommageable pour la crédibilité que l'on peut accorder aux modèles [121]; c'est cet obstacle que tente de contourner la branche de la recherche algorithmique qui se concentre sur les intelligences artificielles "explicables"/"interprétables" (*explainable AI*). Parmi les modèles prometteurs, celui basé sur la mesure de la "pertinence" (*relevance*) des caractéristiques d'une entrée d'un réseau de neurones offre des résultats particulièrement intéressants; ce modèle est communément appelé *layer-wise relevance propagation (LRP)* [122, 123]. De tels résultats sont illustrés figure 4.26 (issue de [123]), où un réseau de neurones convolutionnel bien entraîné est capable de prédire que l'image de gauche appartient à la catégorie "château", et où l'algorithme de LRP est capable de distinguer quels pixels de l'image permettent d'expliquer la prédiction; les trois formes "standard" de l'algorithme (LRP-0, LRP- ϵ et LRP- γ) fournissent des explications qui sont soit fiables, soit humainement compréhensibles, et un mélange des trois ("*composite LRP*") permet d'obtenir une explication qui soit à la fois fiable et humainement compréhensibles.

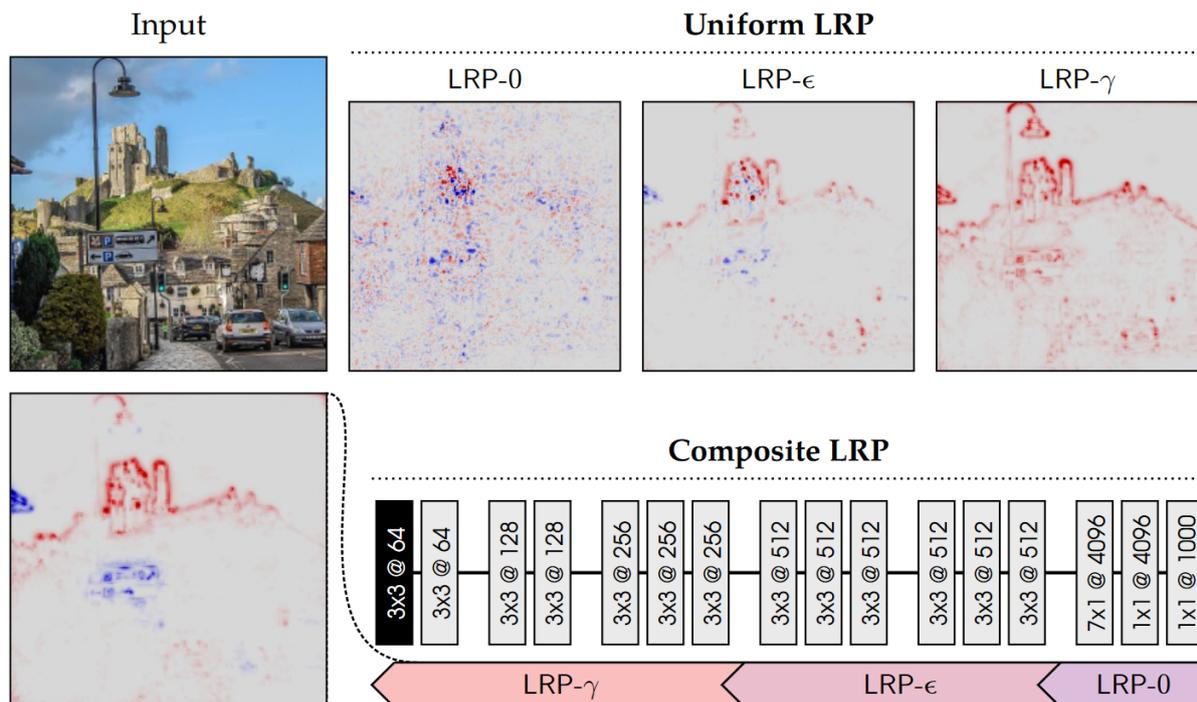


FIGURE 4.26 – Explications fournies par différentes versions de l’algorithme de *layer-wise relevance propagation* quant à la prédiction d’un réseau de neurones convolutionnel, catégorisant l’image de gauche comme étant un "château".

De même, les modèles exploitant à son maximum la nature probabiliste de la démarche de généralisation sont en plein essor, et ceux faisant usage d’une approche bayésienne [124, 125] méritent selon moi une attention toute particulière, ces derniers s’appuyant sur l’inférence bayésienne qui permet, en vertu du théorème de Cox-Jaynes [126], de formaliser au mieux la notion d’apprentissage ; dit rapidement, le théorème de Cox-Jaynes stipule que tout apprentissage (contraint par cinq "desiderata" souhaitables) y compris automatique devra nécessairement soit utiliser l’inférence bayésienne, soit donner quelque part des résultats incohérents. L’application de ce genre d’algorithmes plus interprétables serait donc une prochaine étape intéressante à mettre en place pour l’analyse de données issues de dynamiques non-linéaires.

Bibliographie

- [1] John Scott Russell. *Report of the 14th meeting of the British Association for the Advancement of Science, York, September 1844*. London, British Association, 1844.
- [2] Joseph Valentin Boussinesq. *Théorie de l'intumescence liquide, appelée onde solitaire ou de translation, se propageant dans un canal rectangulaire*. Journal de mathématiques pures et appliquées 2e série, 1871.
- [3] Lord Rayleigh. *On Waves*. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1876.
- [4] D. J. Korteweg and G. de Vries. On the change of form of long waves advancing in a rectangular canal, and on a new type of long stationary waves. *Philosophical Magazine Series 1*, 39 :422–443, 1895.
- [5] Clifford S. Gardner, John M. Greene, Martin D. Kruskal, and Robert M. Miura. Method for solving the Korteweg-deVries equation. *Phys. Rev. Lett.*, 19 :1095–1097, 1967.
- [6] N. J. Zabusky and M. D. Kruskal. Interaction of 'Solitons' in a collisionless plasma and the recurrence of initial states. *Phys. Rev. Lett.*, 15 :240–243, 1965.
- [7] Raymond Y. Chiao, E. Garmire, and Charles H. Townes. Self-trapping of optical beams. *Phys. Rev. Lett.*, 13 :479–482, 1964.
- [8] Theodore H. Maiman et al. Stimulated optical radiation in ruby. *Nature*, 187 :493–494, 1960.
- [9] E. J. Woodbury and W. K. Ng. Ruby laser operation in the near IR. *Proceedings of the IRE*, 50 :2347–2348, 1962.
- [10] P. D. Maker, R. W. Terhune, and C. M. Savage. Intensity-dependent changes in the refractive index of liquids. *Phys. Rev. Lett.*, 12 :507–509, 1964.
- [11] R. L. Carman, R. Y. Chiao, and P. L. Kelley. Observation of degenerate stimulated four-photon interaction and four-wave parametric amplification. *Phys. Rev. Lett.*, 17 :1281–1283, 1966.
- [12] Akira Hasegawa and Frederick Tappert. Transmission of stationary nonlinear optical pulses in dispersive dielectric fibers. i. anomalous dispersion. *Applied Physics Letters*, 23 :142–144, 1973.
- [13] R. R. Alfano and S. L. Shapiro. Emission in the region 4000 to 7000 Å via four-photon coupling in glass. *Phys. Rev. Lett.*, 24 :584–587, 1970.
- [14] Alan M. Turing. *Computing machinery and intelligence*. Springer, 2009.
- [15] Frank Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65 :386–408, 1958.
- [16] Gordon E. Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86 :82–85, 1998.

- [17] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, pages 604–609, 2020.
- [18] Adrià Puigdomènech Badia, Bilal Piot, Steven Kapturowski, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, and Charles Blundell. Agent57 : outperforming the Atari human benchmark. In *Proceedings of the 37th International Conference on Machine Learning*, page 11, 2020.
- [19] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence : early experiments with GPT-4. *arXiv preprint arXiv :2303.12712*, 2023.
- [20] Goëry Genty, Lauri Salmela, John M. Dudley, Daniel Brunner, Alexey Kokhanovskiy, Sergei Kobtsev, and Sergei K. Turitsyn. Machine learning and applications in ultrafast photonics. *Nature Photonics*, 15 :91–101, 2021.
- [21] Govind P. Agrawal. *Non linear fiber optics, sixth edition*. Academic Press, 2019.
- [22] Hellwarth R.W. Third-order optical susceptibilities of liquids and solids. *Progress in Quantum Electronics*, 5 :1–68, 1977.
- [23] Robert W. Boyd. *Nonlinear Optics, fourth edition*. Academic Press, 2020.
- [24] J. P. Gordon. Theory of the soliton self-frequency shift. *Opt. Lett.*, 11 :662–664, 1986.
- [25] B. Kibler, J. M. Dudley, and S. Coen. Supercontinuum generation and nonlinear pulse propagation in photonic crystal fiber : influence of the frequency-dependent effective mode area. *Applied Physics B*, 81 :337–342, 2005.
- [26] S. V. Chernikov and P. V. Mamyshev. Femtosecond soliton propagation in fibers with slowly decreasing dispersion. *Journal of the Optical Society of America B*, 8 :1633–1641, 1991.
- [27] K. J. Blow and D. Wood. Theoretical description of transient stimulated Raman scattering in optical fibers. *IEEE Journal of Quantum Electronics*, 25 :2665–2673, 1989.
- [28] John M. Dudley, Frédéric Dias, Miro Erkintalo, and Goëry Genty. Instabilities, breathers and rogue waves in optics. *Nature Photonics*, 8 :755–764, 2014.
- [29] Arnaud Couairon, Enrico Brambilla, Tommaso Corti, O de J. Ramírez-Góngora Majus, and Miroslav Kolesik. Practitioner’s guide to laser pulse propagation models and simulation : numerical implementation and practical usage of modern pulse propagation models. *The European Physical Journal Special Topics*, 199 :5–76, 2011.
- [30] Johan Hult. A fourth-order Runge–Kutta in the interaction picture method for simulating supercontinuum generation in optical fibers. *Journal of Lightwave Technology*, 25 :3770–3775, 2007.
- [31] Ronald H. Hardin. Application of the split-step fourier method to the numerical solution of nonlinear and variable coefficient wave equations. *SIAM Review*, 15 :423, 1973.
- [32] Robert A. Fisher and William K. Bischel. Numerical studies of the interplay between self-phase modulation and dispersion for intense plane-wave laser pulses. *Journal of Applied Physics*, 46 :4921–4934, 1975.
- [33] D. Anderson and M. Lisak. Analytic study of pulse broadening in dispersive optical fibers. *Phys. Rev. A*, 35 :184–187, 1987.

- [34] E. Brainis, D. Amans, and S. Massar. Scalar and vector modulation instabilities induced by vacuum fluctuations in fibers : numerical study. *Physical Review A*, 71, 2005.
- [35] Thiab R. Taha and Mark I. Ablowitz. Analytical and numerical aspects of certain non-linear evolution equations. ii. numerical, nonlinear Schrödinger equation. *Journal of Computational Physics*, 55 :203–230, 1984.
- [36] Vladimir E. Zakharov and Aleksei B. Shabat. Exact theory of two-dimensional self-focusing and one-dimensional self-modulation of waves in nonlinear media. *Sov. Phys. JETP*, 34 :62–69, 1972.
- [37] Junkichi Satsuma and Nobuo Yajima. B. initial value problems of one-dimensional self-modulation of nonlinear waves in dispersive media. *Progress of Theoretical Physics Supplement*, 55 :284–306, 1974.
- [38] J. M. Dudley, G. Genty, F. Dias, B. Kibler, and N. Akhmediev. Modulation instability, Akhmediev breathers and continuous wave supercontinuum generation. *Optics Express*, 17 :21497–21508, 2009.
- [39] E. Fermi, J. R. Pasta, S. Ulam, and M. Tsingou. Studies of the nonlinear problems. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 1955.
- [40] B. Kibler, J. Fatome, C. Finot, G. Millot, G. Genty, B. Wetzell, N. Akhmediev, F. Dias, and J. M. Dudley. Observation of Kuznetsov-Ma soliton dynamics in optical fibre. *Scientific Reports*, 2 :463, 2012.
- [41] B. Kibler, J. Fatome, C. Finot, G. Millot, F. Dias, G. Genty, N. Akhmediev, and J. M. Dudley. The Peregrine soliton in nonlinear fibre optics. *Nature Physics*, 6 :790–795, 2010.
- [42] D. R. Solli, C. Ropers, P. Koonath, and B. Jalali. Optical rogue waves. *Nature*, 450 :1054–1057, 2007.
- [43] Shanti Toenger, Thomas Godin, Cyril Billet, Frédéric Dias, Miro Erkintalo, Goëry Genty, and J. M. Dudley. Emergent rogue wave structures and statistics in spontaneous modulation instability. *Scientific Reports*, 5 :10380, 2015.
- [44] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers : scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23 :1–39, 2022.
- [45] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots : can language models be too big ? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.
- [46] Ulrike Von Luxburg and Bernhard Schölkopf. Statistical learning theory : models, concepts, and results. In *Handbook of the History of Logic*, volume 10, pages 651–706. Elsevier, 2011.
- [47] Uri Mahlab, Joseph Shamir, and H. John Caulfield. Genetic algorithm for optical pattern recognition. *Optics Letters*, 16 :648–650, 1991.
- [48] Ken D. Kihm and Donald P. Lyons. Optical tomography using a genetic algorithm. *Optics Letters*, 21 :1327–1329, 1996.
- [49] O. Albert, L. Sherman, G. Mourou, T. B. Norris, and G. Vdovin. Smart microscope : an adaptive optics learning system for aberration correction in multiphoton confocal microscopy. *Opt. Lett.*, 25 :52–54, 2000.

- [50] S. Martin, J. Rivory, and Marc Schoenauer. Synthesis of optical multilayer systems using genetic algorithm. *Applied optics*, 34 :2247–2254, 1995.
- [51] Jiajia Zhou, Bolong Huang, Zheng Yan, and Jean-Claude G. Bünzli. Emerging role of machine learning in light-matter interaction. *Light : Science & Applications*, 8 :84, 2019.
- [52] Wei Ouyang, Andrey Aristov, Mickaël Lelek, Xian Hao, and Christophe Zimmer. Deep learning massively accelerates super-resolution localization microscopy. *Nature Biotechnology*, 36 :460–468, 2018.
- [53] Adriano M. Palmieri, Egor Kovlakov, Federico Bianchi, Dmitry Yudin, Stanislav Straupe, Jacob D. Biamonte, and Sergei Kulik. Experimental neural network enhanced quantum tomography. *NPJ Quantum Information*, 6 :20, 2020.
- [54] Thomas Fösel, Petru Tighineanu and Talitha Weiss, and Florian Marquardt. Reinforcement learning with neural networks for quantum feedback. *Physical Review X*, 8 :031084, 2018.
- [55] Francesco Musumeci, Cristina Rottondi, Avishek Nag, Irene Macaluso, Darko Zibar, Marco Ruffini, and Massimo Tornatore. An overview on application of machine learning techniques in optical networks. *IEEE Communications Surveys & Tutorials*, 21 :1383–1408, 2018.
- [56] Shun’ichi Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers*, 16 :299–307, 1967.
- [57] Jürgen Schmidhuber. Deep learning in neural networks : an overview. *Neural Networks*, 81 :85–117, 2015.
- [58] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [59] Terrence J. Sejnowski. *The deep learning revolution*. MIT Press, 2018.
- [60] Marvin Minsky and Seymour Papert. *Perceptrons : an introduction to computational geometry*. MIT press, 1969.
- [61] A. G. Ivakhnenko. Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1 :364–378, 1971.
- [62] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2 :359–366, 1989.
- [63] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4 :251–257, 1991.
- [64] Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16 :146–160, 1976.
- [65] Gottfried Wilhelm Freiherr von Leibniz. *The early mathematical manuscripts of Leibniz*. Open court publishing Company, 1920.
- [66] Yann LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks : Tricks of the trade*, pages 9–50. Springer, 2002.
- [67] Kunihiko Fukushima. Cognitron : a self-organizing multilayered neural network. *Biological cybernetics*, 20 :121–136, 1975.
- [68] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [69] Michael A. Nielsen. *Neural networks and deep learning*. Determination Press San Francisco, CA, USA, 2015.

- [70] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5 :157–166, 1994.
- [71] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.
- [72] LuShin Lu, Yanhui YeonjongSu, and George Em Karniadakis. Dying ReLU and initialization : theory and numerical examples. *Communications in Computational Physics*, 28 :1671–1706, 2020.
- [73] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv preprint arXiv :1511.07289*, 2015.
- [74] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feed-forward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
- [75] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13 :281–305, 2012.
- [76] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks : Tricks of the Trade : Second Edition*, pages 421–436. Springer, 2012.
- [77] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning : generalization gap and sharp minima. *arXiv preprint arXiv :1609.04836*, 2016.
- [78] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [79] Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better : stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234, 2016.
- [80] D. Kleinberg Robert, Li Yuanzhi, and Yuan Yang. An alternative view : When does SGD escape local minima? In *International Conference on Machine Learning*, 2018.
- [81] Gene H. Golub, Per Christian Hansen, and Dianne P. O’Leary. Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, 21 :185–194, 1999.
- [82] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R ; Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv :1207.0580*, 2012.
- [83] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks : Tricks of the Trade (Second Edition)*, pages 437–478. Springer, 2012.
- [84] Benjamin Lindemann, Timo Müller, Hannes Vietz, Nasser Jazdi, and Michael Weyrich. A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, 99 :650–655, 2021.
- [85] Yann LeCun, Koray Kavukcuoglu, and Clément Faret. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

- [86] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.
- [87] Roger Fletcher and Colin M. Reeves. Function minimization by conjugate gradients. *The computer journal*, 7 :149–154, 1964.
- [88] John M. Dudley, Goëry Genty, Arnaud Mussot, Amin Chabchoub, and Frédéric Dias. Rogue waves and analogies in optics and oceanography. *Nature Reviews Physics*, 1 :675–689, 2019.
- [89] Mikko Närhi, Lauri Salmela, Juha Toivonen, Cyril Billet, John M. Dudley, and Goëry Genty. Machine learning analysis of extreme events in optical fibre modulation instability. *Nature Communications*, 9 :4923, 2018.
- [90] Lauri Salmela, Coraline Lapre, John M. Dudley, and Goëry Genty. Machine learning analysis of rogue solitons in supercontinuum generation. *Scientific Reports*, 10 :9596, 2020.
- [91] David M. W. Powers. Evaluation : from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv :2010.16061*, 2020.
- [92] Chen Xie, Vytautas Jukna, Carles Milián, Remo Giust, Ismail Ouadghiri-Idrissi, Tatiana Itina, John M. Dudley, Arnaud Couairon, and Francois Courvoisier. Tubular filamentation for laser material processing. *Scientific Reports*, 5 :8914, 2015.
- [93] Amin Chabchoub, Bertrand Kibler, Christophe Finot, Guy Millot, Miguel Onorato, John M. Dudley, and A. V. Babanin. The nonlinear Schrödinger equation and the propagation of weakly nonlinear waves in optical fibers and on the water surface. *Annals of Physics*, 361 :490–500, 2015.
- [94] Mehdi Mabed, Lauri Salmela, Andrei V. Ermolaev, Christophe Finot, Goëry Genty, and John M. Dudley. Neural network analysis of unstable temporal intensity peaks in continuous wave modulation instability. *Optics Communications*, 541 :129570, 2023.
- [95] Stéphane Randoux, Pierre Suret, and Gennady El. Inverse scattering transform analysis of rogue waves using local periodization procedure. *Scientific Reports*, 6 :29238, 2016.
- [96] Andrey Gelash, Dmitry Agafontsev, Vladimir Zakharov, Gennady El, Stéphane Randoux, and Pierre Suret. Bound state soliton gas dynamics underlying the spontaneous modulational instability. *Physical Review Letters*, 123 :234102, 2019.
- [97] François Copie, Stéphane Randoux, and Pierre Suret. The physics of the one-dimensional nonlinear Schrödinger equation in fiber optics : rogue waves, modulation instability and self-focusing phenomena. *Reviews in Physics*, 5 :100037, 2020.
- [98] Mehdi Mabed, Fanchao Meng, Lauri Salmela, Christophe Finot, Goëry Genty, and John M. Dudley. Machine learning analysis of instabilities in noise-like pulse lasers. *Optics Express*, 30 :15060–15072, 2022.
- [99] Nail Akhmediev and Adrian Ankiewicz. *Dissipative Solitons*. Springer Berlin, Heidelberg, 2005.
- [100] Philippe Grelu and Nail Akhmediev. Dissipative solitons for mode-locked lasers. *Nature Photonics*, 6 :84–92, 2012.
- [101] P. Ryczkowski, M. Närhi, C. Billet, J.-M. Merolla, G. Genty, and J. M. Dudley. Real-time full-field characterization of transient dissipative soliton dynamics in a mode-locked laser. *Nature Photonics*, 12 :221–227, 2018.

- [102] G. Herink, F. Kurtz, B. Jalali, D. R. Solli, and C. Ropers. Real-time spectral interferometry probes the internal dynamics of femtosecond soliton molecules. *Science*, 356 :50–54, 2017.
- [103] Coraline Lapre, Cyril Billet, Fanchao Meng, Piotr Ryczkowski, Thibaut Sylvestre, Christophe Finot, Göery Genty, and John M. Dudley. Real-time characterization of spectral instabilities in a mode-locked fibre laser exhibiting soliton-similariton dynamics. *Scientific Reports*, 9 :13950, 2019.
- [104] Yoonchan Jeong, Luis Alonso Vazquez-Zuniga, Seungjong Lee, and Youngchul Kwon. On the formation of noise-like pulses in fiber ring cavity configurations. *Optical Fiber Technology*, 20 :575–592, 2014.
- [105] D. Y. Tang, L. M. Zhao, and B. Zhao. Soliton collapse and bunched noise-like pulse generation in a passively mode-locked fiber ring laser. *Optics Express*, 13 :2289–2294, 2005.
- [106] Luis Vazquez-Zuniga and Yoonchan Jeong. Super-broadband noise-like pulse erbium-doped fiber ring laser with a highly nonlinear fiber for Raman gain enhancement. *IEEE Photonics Technology Letters*, 24 :1549–1551, 2012.
- [107] Fanchao Meng, Coraline Lapre, Cyril Billet, Thibaut Sylvestre, Jean-Marc Merolla, Christophe Finot, Sergei K. Turitsyn, Goëry Genty, and John M. Dudley. Intracavity incoherent supercontinuum dynamics and rogue waves in a broadband dissipative soliton laser. *Nature Communications*, 12 :5567, 2021.
- [108] Fanchao Meng, Coraline Lapre, Cyril Billet, Goëry Genty, and John M. Dudley. Instabilities in a dissipative soliton-similariton laser using a scalar iterative map. *Optics Letters*, 45 :1232–1235, 2020.
- [109] Caiyun Li, Jiangyong He, Ruijing He, Yange Liu, Yang Yue, Weiwei Liu, Luhe Zhang, Longfei Zhu, Mengjie Zhou, Kaiyan Zhu, and Zhi Wang. Analysis of real-time spectral interference using a deep neural network to reconstruct multi-soliton dynamics in mode-locked lasers. *APL Photonics*, 5 :1–3, 2020.
- [110] Chunyuan Li, Heerad Farkhor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv :1804.08838*, 2018.
- [111] Pantelis R. Vlachas, Wonmin Byeon, Zhong Y. Wan, Themistoklis P. Sapsis, and Petros Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 474 :20170844, 2018.
- [112] Junjie Jiang and Ying-Cheng Lai. Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks : role of network spectral radius. *Phys. Rev. Res.*, 1 :033056, 2019.
- [113] Maziar Raissi, Paris Perdikaris, and George E. Karniadakis. Physics-informed neural networks : a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378 :686–707, 2019.
- [114] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 :1735–1780, 1997.
- [115] Renato G. Nascimento, Kajetan Fricke, and Felipe A. C. Viana. A tutorial on solving ordinary differential equations using python and hybrid physics-informed neural network. *Engineering Applications of Artificial Intelligence*, 96 :103996, 2020.

- [116] Ricky T.Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K. Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 31, 2018.
- [117] Lauri Salmela, Nikolaos Tsipinakis, Alessandro Foi, Cyril Billet, John M. Dudley, and Goëry Genty. Predicting ultrafast nonlinear dynamics in fibre optics with a recurrent neural network. *Nature Machine Intelligence*, 3 :344–354, 2021.
- [118] Lauri Salmela, Mathilde Hary, Mehdi Mabed, Alessandro Foi, John M. Dudley, and Goëry Genty. Feed-forward neural network as nonlinear dynamics integrator for super-continuum generation. *Optics Letters*, 47 :802–805, 2021.
- [119] Jared L. Callahan, James V. Koch, Bingni W. Brunton, J. Nathan Kutz, and Steven L. Brunton. Learning dominant physical processes with data-driven balance models. *Nature communications*, 12 :1016, 2021.
- [120] Andrei V. Ermolaev, Mehdi Mabed, Christophe Finot, Goëry Genty, and John M. Dudley. Analysis of interaction dynamics and rogue wave localization in modulation instability using data-driven dominant balance. *Scientific Reports*, 13 :10462, 2023.
- [121] Zachary C. Lipton. The mythos of model interpretability : in machine learning, the concept of interpretability is both important and slippery. *Queue*, 16 :31–57, 2018.
- [122] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10, 2015.
- [123] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation : an overview. *Explainable AI : interpreting, explaining and visualizing deep learning*, pages 193–209, 2019.
- [124] Radford M. Neal. *Bayesian learning for neural networks*. Springer New York, NY, 1995.
- [125] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.
- [126] Richard Threlkeld Cox. Probability, frequency, and reasonable expectation. *American Journal of Physics*, 14 :1–13, 1946.

Table des figures

1.1	Évolution de l'indice de réfraction n et de l'indice de groupe n_g de la silice fondue en fonction de la longueur d'onde λ de l'impulsion en entrée de fibre. La ligne en pointillés noirs correspond à la longueur d'onde de 1550 nm qui est celle qui a été utilisée pour la plupart des études effectuées lors de cette thèse.	10
1.2	Évolution de la vitesse de groupe v_g (figure de gauche), du paramètre de dispersion D et de β_2 (figure de droite) de la silice fondue en fonction de la longueur d'onde λ de l'impulsion en entrée d'une fibre en silice fondue. La ligne en pointillés noirs correspond à la longueur d'onde de 1550 nm qui est celle qui a été utilisée pour la plupart des études effectuées lors de cette thèse.	15
1.3	Courbes de gain obtenues avec les paramètres de la simulation CW MI pour différentes puissances P_0 : $\beta_2 = -2,1 \times 10^{-26} \text{ s}^2\text{m}^{-1}$, $\gamma = 0,0011 \text{ W}^{-1}\text{m}^{-1}$ et $\lambda_0 = 1550 \text{ nm}$	24
1.4	Simulation numérique de solitons d'ordre 1, 2 et 3 (respectivement a), b) et c)) dans le domaine temporel (en haut) et spectral (en bas). Les paramètres de la simulation numérique sont les même que pour le CW MI où la puissance P_0 est ajustée selon l'équation 1.66 pour obtenir le soliton d'ordre désiré : $\beta_2 = -2,1 \times 10^{-26} \text{ s}^2\text{m}^{-1}$, $\gamma = 0,0011 \text{ W}^{-1}\text{m}^{-1}$, $\lambda_0 = 1550 \text{ nm}$ et FWHM = 2 ps ($T_0 = 1,13 \text{ ps}$)	26
1.5	Simulation numérique de solitons sur fond continu (SBF) en unité normalisée. La figure de gauche montre un <i>breather</i> d'Akhmediev ($a = 0, 25$), qui présente une périodicité temporelle, la figure droite montre un <i>breather</i> de Kuznetsov-Ma ($a = 1$), qui présente une périodicité spatiale, et la figure centrale montre un soliton de Peregrine ($a = 0, 5$), qui est localisé spatialement et temporellement ; ce dernier et donc le cas limite des deux <i>breathers</i> précédents.	27
1.6	Simulation numérique d'un supercontinuum. $\beta_2 = - 5.9 \times 10^{-27} \text{ s}^2\text{m}^{-1}$, $\beta_3 = 4.21 \times 10^{-41} \text{ s}^3\text{m}^{-1}$, $\beta_4 = -1.25 \times 10^{-55} \text{ s}^4\text{m}^{-1}$, $\beta_5 = 2.45 \times 10^{-70} \text{ s}^5\text{m}^{-1}$, $\gamma = 0.1 \text{ W}^{-1}\text{m}^{-1}$, $\lambda_0 = 1550 \text{ nm}$, $P_0 = 1 \text{ kW}$ et FWHM = 0.1 ps ($T_0 = 57 \text{ fs}$).	29
2.1	Schéma de mixture gaussienne : dans l'espace des caractéristiques des individus, des groupes d'individus semblent naturellement exister, et on cherche à trouver les paramètres de trois gaussiennes qui pourraient classifier ces groupes. La figure de droite rend les données moins surprenantes que la figure de gauche.	32

2.2	À gauche : “ <i>Purkinje neurons from the human cerebellum</i> ” (Cajal Institute, Madrid), dessin de Santiago Ramón y Cajal datant de 1899 représentant des cellules de Purkinje présentes dans le cervelet humain ; on y voit la structure en réseau qui a inspiré les chercheurs en intelligence artificielle. À droite : photo du "Mark I perceptron" de Rosenblatt datant de 1958 ; on y voit un ensemble de connexions à gauche, et, à droite, les potentiomètres qu’il fallait ajuster afin de pondérer ces connexions et obtenir un classifieur binaire performant. Le perceptron de Rosenblatt était capable (une fois ajusté) de reconnaître un homme d’une femme lorsqu’une photo de visage lui était présentée.	34
2.3	Portes logiques "ET", "OU" et "XOR". L’abscisse et l’ordonnée x_1 et x_2 sont les entrées booléennes des portes logiques, et la couleur des cercles représente la valeur de la sortie : le cercle est noir lorsque la sortie vaut zéro et blanc lorsqu’elle vaut un. Les portes "ET" et "OU" sont linéairement séparables, ce qui signifie qu’un hyperplan (une ligne droite en deux dimensions) permet de classifier la sortie (séparer les ronds blancs des ronds noirs), mais la porte "XOR" ne l’est pas.	35
2.4	Schéma de réseau de neurones artificiels à propagation avant multi-couche. Les liens entre les différentes couches ne sont pas fléchés pour ne pas perdre en lisibilité.	36
2.5	Porte logique "XOR" linéairement séparable dans un espace de dimension supérieure à l’espace initial des entrées booléennes x_1 et x_2 , ce qui signifie qu’un hyperplan (un plan en trois dimensions) permet de classifier la sortie (séparer les ronds blancs des ronds noirs)	38
2.6	Opération élémentaire effectuée par un neurone d’un réseau de neurones à propagation avant (tel que celui de la figure 2.4).	39
2.7	Exemple de fonctions d’activation usuelles : fonction de Heaviside (montrée à titre de comparaison, elle n’est quasiment jamais utilisée de nos jours), fonction logistique, tangente hyperbolique, et fonction ReLU (<i>Rectified Liner Unit</i>).	40
2.8	Visualisation de l’algorithme de descente du gradient. Ce schéma présente quatre processus d’optimisation dans l’espace des paramètres : les processus numérotés "1" et "2" (courbes orange et rouge avec des triangles) qui partent de deux endroits très légèrement différents (au niveau du point d’inflexion de la surface), et les processus numérotés "3" et "4" (courbes bleue et verte avec des cercles) qui partent du même endroit (relativement élevé) mais mis à jour avec des taux d’apprentissages différents ; pour chacun de ces processus, l’algorithme de descente du gradient met à jour les paramètres selon la règle 2.18. Les flèches noires représentent le gradient en différents points de l’espace des paramètres. Ceci illustre bien l’importance cruciale du choix des hyper-paramètres (le taux d’apprentissage ici) ainsi que des conditions initiales des paramètres (poids et biais).	42
2.9	Exemple de dérivées de fonctions d’activation usuelles : fonction de Heaviside (montrée à titre de comparaison, elle n’est quasiment jamais utilisée de nos jours), fonction logistique, tangente hyperbolique, et fonction ReLU (<i>Rectified Liner Unit</i>).	44

- 3.1 Spectres MI enregistrés expérimentalement avec une plage dynamique élevée. (a) Série de 60 spectres enregistrés empilés dans le sens vertical pour illustrer les fluctuations d'un coup à l'autre observées avec le MI. (b) Spectre moyen mesuré avec la technique en temps réel à plage dynamique élevée (pointillé rouge), spectre moyen mesuré avec un analyseur de spectre optique (jaune), et spectre moyen simulé (bleu). L'encadré montre la comparaison entre la mesure spectrale avec une plage dynamique de 60 dB (pointillé rouge) et celle utilisant une DFT conventionnelle avec seulement 22 dB de plage dynamique (noir). (c) Deux spectres sélectionnés à partir d'expériences (pointillé rouge) comparés au spectre expérimental moyen (noir). Figure 6 de [89] 57
- 3.2 Schéma du réseau de neurones utilisé pour analyser les données de simulation d'instabilité de modulation impulsionnelle. Le profil d'intensité spectrale est en entrée du réseau de neurones (à gauche), puis est transformé par les couches successives, jusqu'à aboutir à la prédiction du maximum d'intensité temporelle (à droite). Figure 2 de [89] 58
- 3.3 Analyse par apprentissage supervisé de spectres MI simulés (30000 données d'entraînement et 20000 pour le test). a) et c) montrent l'intensité maximale prédite par le réseau de neurones en fonction de la valeur à prédire, celle-ci étant issues de simulations. Le panneau supérieur (respectivement inférieur) correspond à la troncature des données spectrales d'entrées à 60 dB (respectivement 25 dB). Les résultats en a) et c) sont présentés sous la forme d'un histogramme en échelle logarithmique qui montre la densité normalisée des points regroupés dans des classes de surface constante. La ligne blanche en pointillés marque la correspondance 1 à 1 entre l'intensité maximale (puissance de crête instantanée) prédite par le réseau et la valeur à prédire. b) et d) montrent les densité de probabilité (PDF) des intensités temporelles maximales prédites par l'algorithme d'apprentissage automatique (ligne rouge) comparée aux PDF calculées à partir des données temporelles simulées (ligne bleue). Figure 3 de [89] 59
- 3.4 Analyse par apprentissage supervisé de spectres MI simulés (20000 données d'entraînement et 10000 pour le test). Cette figure est la reproduction des résultats de la figure 3.3 à l'aide de mon code de réseau de neurones. Des performances supérieures sont obtenues pour les données tronquées à 60 dB (figures a) et b)), et des performances légèrement inférieures pour les données tronquées à 25 dB (figures c) et d)). 60
- 3.5 Sélection de 5 spectres de supercontinuum aléatoires et profils d'intensité temporelle correspondants (noir). La courbe bleue en b) montre l'intensité temporelle du soliton "scélérat" associé aux composantes spectrales filtrées avec un filtre passe-bas (zone bleue) en a) tandis que la ligne pointillée noire montre l'intensité complète dans le domaine temporel. Figure 1 de [90] 62
- 3.6 Schéma du réseau de neurones utilisé dans ce travail. (a) L'entrée du réseau est un vecteur d'intensité spectrale de supercontinuum $X_n = [x_1, x_2, \dots, x_N]$ conduisant à la sortie du réseau y qui correspond à la puissance de crête, à la durée ou au retard temporel des solitons errants. Le réseau se compose de deux couches cachées et d'un seul neurone de sortie. Figure 2 de [90] 62

- 3.7 Résultats montrant la prédiction des caractéristiques des solitons scélérats par le réseau de neurones pour un ensemble de 10000 SC générés avec des impulsions d'entrée identiques (à l'exception de la graine de bruit). (a) compare l'intensité maximale prédite du soliton le plus décalé vers le rouge avec la valeur exacte des profils simulés dans le domaine temporel. (b) compare la durée prédite $\Delta\tau$ des solitons avec la valeur exacte des simulations. (c) compare le retard prédit τ_d des solitons avec la valeur exacte des simulations. Pour toutes les sous figures, les histogrammes en échelle logarithmique montrent la densité normalisée des points groupés en cases de surface constante. Dans chaque cas, la ligne blanche en pointillés marque la correspondance un à un, et la ligne jaune en pointillés est un ajustement linéaire aux prédictions avec avec le coefficient de corrélation de Pearson ρ . Figure 3 de [90] 63
- 3.8 Reproduction des résultats de la figure 3.7 avec mon code de réseau de neurones. Là encore, des performances similaires, et même légèrement supérieures sont obtenues avec mon code. 64
- 3.9 Courbe d'évolution de la fonction coût (en haut) et du coefficient de corrélation de Pearson (en bas) au cours du processus d'optimisation qui a permis d'aboutir à la figure gauche de 3.8. 64
- 4.1 Simulation numérique d'instabilité de modulation excitée par une onde continue bruitée dans le régime transitoire. Les figures (a) et (b) montrent respectivement l'évolution spectrale et temporelle sur 450 m issues d'une seule simulation. Les figures (c) et (d) illustrent les fluctuations en intensité spectrale et temporelle à la sortie de la fibre pour 4 simulations différentes. (c) montre également la moyenne calculée (en haut) et un zoom de cette dernière (encart) au voisinage de la pompe ; cet encart est tracé sur un intervalle de 20 nm et sur une plage dynamique de 50 dB. Les pics d'intensité temporelle P_k , représentés en (d) par un cercle pour chaque simulation, sont les grandeurs que nous cherchons à prédire à partir des spectres correspondants S_k , représentés en (c). Afin d'afficher clairement les différents résultats de simulations, un décalage vertical artificiel entre les différents profils est ajouté en (c) et (d). 68
- 4.2 Simulation numérique d'instabilité de modulation excitée par une onde continue bruitée dans le régime stationnaire. Les figures (a) et (b) montrent respectivement l'évolution spectrale et temporelle sur 300 m issues d'une seule simulation (la propagation affichée ici commence à 55 km). Les figures (c) et (d) illustrent les fluctuations en intensité spectrale et temporelle à 55 m ($\sim 3 L_{NL}$) d'intervalles. (c) montre également la moyenne calculée (en haut) et un zoom de cette dernière (encart) au voisinage de la pompe ; cet encart est tracé sur un intervalle de 20 nm et sur une plage dynamique de 50 dB. Les pics d'intensité temporelle P_k , représentés en (d) par un cercle, sont les paramètres que nous cherchons à prédire à partir des spectres correspondants S_k , représentés en (c). Afin d'afficher clairement les différents résultats de simulations, un décalage vertical artificiel entre les différents profils est ajouté en (c) et (d). 70

- 4.3 (a) Représentation schématique du réseau de neurones choisi afin d'effectuer l'analyse (150 neurones en entrée, 150 neurones pour la première couche cachée, 10 neurones pour la seconde couche cachée, et un neurone en sortie). (b) *Grid search* effectué sur deux hyper-paramètres dans le régime stationnaire : le nombre de neurones de la première couche cachée en abscisse, et le nombre de neurones de la seconde couche cachée en ordonnée. La couleur indique le coefficient de corrélation de Pearson obtenu pour chaque architecture et le point bleu indique l'architecture retenue pour l'étude. 71
- 4.4 Étapes de pré-traitement des données d'entrée du réseau de neurones. La ligne fine grise montre le profil d'intensité spectrale brut issu de l'intégration numérique de l'équation de Schrödinger non-linéaire (en dB). La ligne la plus épaisse en rouge montre la convolution de ce profil brut par une gaussienne de largeur (FWHM) 1 nm, et la ligne bleue montre ce profil convolué après avoir été moyenné dans des classes de largeur 1 nm ; ce sont ces dernières données qui seront fournies au réseau de neurones, après avoir été normalisées. Le profil initial contient 2048 points répartis entre 1480 et 1630 nm, et le profil final en contient 100 répartis entre 1500 et 1600 nm. L'encadré en haut à droite est un zoom sur la zone centrée sur la longueur d'onde de la pompe λ_0 72
- 4.5 Performances du réseau de neurones pour le régime transitoire, quantifiant la prédiction par le réseau des pics d'intensité temporelle de MI sur la seule base des données spectrales d'entrée. Ces résultats sont basés sur un ensemble de 2500 simulations non-utilisées dans la phase de formation. Les puissances crêtes prédites des maxima d'intensités temporelles sont comparées avec celles issues de simulations : (a) avec la plage dynamique complète de 100 dB des simulations, (b) tronquée à une plage dynamique de 40 dB. (c) Représentation graphique du coefficient de corrélation pour différentes plages dynamiques de 20 à 100 dB. 73
- 4.6 Performances du réseau de neurones pour le régime stationnaire, quantifiant la prédiction par le réseau des pics d'intensité temporelle de MI sur la seule base des données spectrales d'entrée. Ces résultats sont basés sur un ensemble de 2500 simulations non-utilisées dans la phase de formation. Les puissances crêtes prédites des maxima d'intensités temporelles sont comparées avec celles issues de simulations : (a) avec la plage dynamique complète de 100 dB des simulations, (b) tronquée à une plage dynamique de 40 dB. (c) Représentation graphique du coefficient de corrélation pour différentes plages dynamiques de 20 à 100 dB. 74
- 4.7 Coefficient de corrélation du test dans le régime (a) transitoire et (b) stationnaire en fonction de $\Delta z/L_{NL}$, la distance entre le profil temporel cible utilisé lors de l'entraînement et celui utilisé dans ce test prédictif. Les distances sont exprimées en pourcentage de la longueur non linéaire. Les résultats sont présentés pour deux plages dynamiques 100 dB et 40 dB (comme indiqué). 75
- 4.8 Schéma du laser NLP. EDF : fibre dopée en l'Erbium. SMF : fibre mono-mode SMF28. HNLF : fibre hautement non linéaire. SA : absorbant saturable. Les paramètres de tous les éléments de la cavité sont donnés dans le texte. 76

- 4.9 Simulation numérique dans le régime à spectre étroit (sans HNLF). Les figures (a) et (b) montrent respectivement l'évolution spectrale et temporelle lors d'un tour de cavité. Les figures (c) et (d) illustrent les fluctuations entre chaque tour en affichant respectivement cinq profils d'intensité spectrale et temporelle sur cinq tours consécutifs, mesurés au point D (après le segment de fibre SMF1). 78
- 4.10 Simulation numérique dans le régime à spectre large (avec HNLF). Les figures (a) et (b) montrent respectivement l'évolution spectrale et temporelle lors d'un tour de cavité. Les figures (c) et (d) illustrent les fluctuations entre chaque tour en affichant respectivement cinq profils d'intensité spectrale et temporelle sur cinq tours consécutifs, mesurés au point D (après le segment de fibre HNLF). 79
- 4.11 Étapes de pré-traitement des données d'entrée du réseau de neurones en régime à spectre large (avec fibre HNLF). La ligne fine grise montre le profil d'intensité spectrale brut issu de l'intégration numérique de l'équation de Schrödinger non-linéaire. La ligne la plus épaisse en rouge montre la convolution de ce profil brut par une gaussienne de largeur (FWHM) 1 nm, et la ligne bleue montre ce profil convolué après avoir été moyenné dans des classes de largeur 1 nm ; ce sont ces dernières données qui seront fournies au réseau de neurones, après les avoir normalisées. Le profil initial contient 2^{17} points répartis entre 1020 et 3240 nm, et le profil final en contient 2100 répartis entre 1100 et 3200 nm. L'encadré en haut à droite est un zoom sur la zone centrée sur la longueur d'onde de la pompe λ_0 81
- 4.12 Schéma du réseau de neurones à propagation avant utilisé. L'entrée est un vecteur d'intensité spectrale correspondant aux données spectrales issues de simulation après échantillonnage et convolution. Les deux couches cachées ont 100 et 30 neurones (dans le cas à bande étroite) et 300 et 100 neurones (dans le cas à bande large). 81
- 4.13 Test du réseau de neurones pour la cavité à spectre étroit (sans HNLF). Prédiction du maximum d'intensité temporelle effectuée par le réseau de neurones en fonction de la valeur réelle de ce maximum. Les différentes figures correspondent à différents seuils de plage dynamique pour les données fournies au réseau : (a) montre la corrélation obtenue entre les prédictions et les valeurs réelles lorsque le réseau est formé avec des données non-tronquées, (b) montre cette corrélation lorsque les données sont tronquées à une plage dynamique de 50 dB, et (c) lorsque les données sont tronquées à une plage dynamique de 30 dB. La ligne jaune est la régression linéaire du nuage de points, et la ligne pointillée blanche est la diagonale vers laquelle doit tendre la ligne jaune. La figure (d) montre un profil d'intensité typique obtenu lors d'une simulation. 82
- 4.14 Test du réseau de neurones pour la cavité à spectre large (avec HNLF). Prédiction du maximum d'intensité temporelle effectuée par le réseau de neurones en fonction de la valeur réelle de ce maximum. Les différentes figures correspondent à différents seuils de plage dynamique pour les données fournies au réseau : (a) montre la corrélation obtenue entre les prédictions et les valeurs réelles lorsque le réseau est formé avec des données non-tronquées, (b) montre cette corrélation lorsque les données sont tronquées à une plage dynamique de 80 dB, et (c) lorsque les données sont tronquées à une plage dynamique de 50 dB. La ligne jaune est la régression linéaire du nuage de points, et la ligne pointillée blanche est la diagonale vers laquelle doit tendre la ligne jaune. La figure (d) montre un profil d'intensité typique obtenu lors d'une simulation. 83

4.15	Évolution du coefficient de corrélation de Pearson entre les résultats prédits par le réseau de neurones et ceux fournis par intégration numérique en fonction de la plage dynamique des données.	84
4.16	Densité de probabilité des maxima d'intensité temporelle issue des simulation (ligne bleue) et reconstruite par le réseau de neurones (ligne rouge en tirets) : (a) dans le régime à spectre étroit (sans HNLF) avec un plage de dynamique de 50 dB et (b) dans le régime à spectre large (avec HNLF) avec un plage de dynamique de 80 dB.	85
4.17	Schéma de la procédure de sélection de la zone d'intérêt des données d'entrée. Le profil d'intensité spectrale est multiplié par une fonction porte centrée en λ_c et de largeur $\Delta\lambda$	87
4.18	Évolution du coefficient de corrélation de Pearson ρ en fonction de λ_c et $\Delta\lambda$ dans le régime d'instabilité de modulation (MI) pure. L'axe des abscisses représente la fréquence centrale λ_c de porte et l'axe des ordonnées la demi-largeur de la porte $\Delta\lambda/2$	87
4.19	Évolution du coefficient de corrélation de Pearson ρ en fonction de λ_c et $\Delta\lambda$ dans le régime <i>noise-like pulse</i> (NLP) sans HNLF. L'axe des abscisses représente la fréquence centrale λ_c de porte et l'axe des ordonnées la demi-largeur de la porte $\Delta\lambda/2$	88
4.20	Évolution du coefficient de corrélation de Pearson ρ en fonction de λ_c et $\Delta\lambda$ <i>noise-like pulse</i> (NLP) avec HNLF. L'axe des abscisses représente la fréquence centrale λ_c de la porte et l'axe des ordonnées la demi-largeur de la porte $\Delta\lambda/2$	88
4.21	Schéma du processus de formation d'un réseau de neurones à propagation avant. Lors de la phase de formation (figure a)), les paramètres du réseau (poids et biais) sont mis à jour afin que ce dernier puisse se substituer à un intégrateur numérique classique (figure b)).	91
4.22	Évolution spectrale le long de la fibre (en bas) et sortie de fibre (en haut) après a) une intégration numérique classique (<i>split-step Fourier</i>) et b) une intégration par réseau de neurones. Les figures du haut montrent l'intensité et la phase spectrale alors que les figures du bas ne montrent que l'intensité.	92
4.23	Racine de l'erreur quadratique moyenne pour chaque paire $\{\text{FWHM}; P_0\}$ de paramètres de l'impulsion d'entrée. Chaque point de cette figure est une moyenne sur z et λ de la RMSD obtenue pour une impulsion.	93
4.24	Racine de l'erreur quadratique moyenne calculée à chaque position z dans la fibre et pour chaque longueur d'onde λ . Cette RMSD est moyennée sur toutes les conditions initiales (FWHM et P_0).	94
4.25	Écart-type calculé à chaque position z dans la fibre et pour chaque longueur d'onde λ . Cet écart-type est moyenné sur toutes les conditions initiales (FWHM et P_0).	94
4.26	Explications fournies par différentes versions de l'algorithme de <i>layer-wise relevance propagation</i> quant à la prédiction d'un réseau de neurones convolutif, catégorisant l'image de gauche comme étant un "château".	99

Liste des tableaux

3.1	Paramètres physiques d'une simulation de système fibré : paramètres de la fibre à gauche et paramètres de l'impulsion d'entrée à droite	56
3.2	Paramètres physiques des simulations en régime d'instabilité de modulation stimulée par une impulsion sécante hyperbolique dans une fibre à cristaux photoniques (PCF) : paramètres de la fibre à gauche et paramètres de l'impulsion d'entrée à droite.	58
3.3	Paramètres physiques des simulations en régime de supercontinuum : paramètres de la fibre à gauche et paramètres de l'impulsion d'entrée à droite	61
4.1	Paramètres physiques des simulations en régime d'instabilité de modulation stimulée par une onde continue : paramètres de la fibre à gauche et paramètres de l'onde continue d'entrée à droite	68
4.2	Paramètres physiques des simulations de cavité <i>noise-like pulse</i> : paramètres des fibres au milieu et paramètres de l'impulsion d'entrée à droite	78
4.3	Paramètres physiques des simulations en régime de supercontinuum : paramètres de la fibre à gauche et paramètres de l'impulsion d'entrée à droite	92

Annexes

Annexe A

Code MATLAB de réseau de neurones

```
1 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Fonctions %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%
2 % fonctions d'activation
3 sigmoide = @(s) 1./(1 + exp(-s)) ; % sigmoide
4 tanH = @(s) tanh(s) ; % tanh
5 softMax = @(s) exp(s)./sum(exp(s),1) ; % softmax
6 lineaire = @(s) s ; % linéaire
7 softPlus = @(s) log(1 + exp(s)) ; % softplus
8
9 activ = {sigmoide tanH softMax lineaire softPlus} ;
10
11 % dérivées des fonctions d'activation
12 dsigmoide = @(s) sigmoide(s).*(1 - sigmoide(s)) ; % sigmoide
13 dtanH = @(s) (1 - tanH(s).^2) ; % tanh
14 dsoftMax = @(s) softMax(s).*(1 - softMax(s)) ; % softmax
15 dlineaire = @(s) ones(size(s)) ; % linéaire
16 dsoftPlus = @(s) sigmoide(s) ; % softplus
17
18 dactiv = {dsigmoide dtanH dsoftMax dlineaire dsoftPlus} ;
19
20 % fonctions coût
21 CXE = @(x,T) -sum(T.*log(x) + (1-T).*log(1-x),1) ; % entropie
    croisée
22 Clog = @(x,T) -log(x(T == 1)') ; % log-vraisemblance
23 C2 = @(x,T) sum((x - T).^2,1) ; % quadratique
24
25 C = {CXE Clog C2} ;
26
27 % dérivées des fonctions coût
28 dCXE = @(x,T) (x - T)./(x.*(1 - x)) ; % entropie croisée
29 dClog = @(x,T) -1./(x(T == 1)').*T ; % log-vraisemblance
30 dC2 = @(x,T) 2*(x - T) ; % quadratique
31
32 dC = {dCXE dClog dC2} ;
33
34
```

```

35 % fonctions de régularisation
36 L1 = @(w) sign(w) ; % régularisation L1
37 L2 = @(w) w ; % régularisation L2
38
39 L = {L1 L2} ;
40
41 % fonctions d'initialisation des poids et biais
42 I1 = @(w,Nn) w/Nn ; % initialisation de Xavier
43 I2 = @(w,Nn) zeros(length(w),1) ; % initialisation à 0
44
45 I = {I1 I2} ;
46
47 %% %%%%%%%%%%%%%%% Hyper-paramètres %%%%%%%%%%%%%%% %%
48 % Ee : entrées pour l'entraînement
49 % Ev : entrées pour la validation
50 % Et : entrées pour le test
51 % Te : cibles ("target") pour l'entraînement
52 % Tv : cibles ("target") pour la validation
53 % Tt : cibles ("target") pour le test
54
55 Nn = [size(Ee,1) 300 100 size(Te,1)] ; % architecture
56 ii = 1 ; % indice fonction d'initialisation (poids et biais)
57 ia = [2 2 4] ; % indice fonctions d'activation
58 ic = 3 ; % indice fonction coût
59 il = 2 ; % indice fonction de régularisation
60
61 eta = 0.01 ; % learning rate
62 mu = 0.2 ; % momentum
63 lambda = 1.3 ; % régularisation
64 tm = 100 ; % taille du mini-batch
65 Ne = 250 ; % nombre d'époque
66
67 Nc = size(Nn,2) - 1 ; % nombre de couches
68 Nm = size(Ee,2)/tm ; % nombre de mini-batch
69
70 %% %%%%%%%%%%%%%%% Initialisation %%%%%%%%%%%%%%% %%
71 %% %%%%%%%%%%%%%%% des paramètres %%%%%%%%%%%%%%% %%
72 s = zeros(sum(Nn(2:end)),tm) ; % vecteur des sommes pondérées
73 d = zeros(sum(Nn(2:end)),tm) ; % vecteur des gradients de C
74
75 w = randn(Nn(1:end-1)*Nn(2:end)',1) ; % poids aléatoires
76 b = randn(sum(Nn(2:end)),1) ; % biais aléatoires
77 for c = 1 : 1 : Nc
78     w(Nn(1:c-1)*Nn(2:c)'+1 : Nn(1:c)*Nn(2:c+1)') = ...
79     I{ii(c)}(w(Nn(1:c-1)*Nn(2:c)'+1:Nn(1:c)*Nn(2:c+1)'),Nn(c)) ;
80     b(sum(Nn(2:c))+1 : sum(Nn(2:c+1))) = ...
81     I{ii(c)}(b(sum(Nn(2:c))+1:sum(Nn(2:c+1))),Nn(c)) ;
82 end

```

```

83 vw = zeros(length(w),1) ; % "inerties" initiales des poids
84 vb = zeros(length(b),1) ; % "inerties" initiales des biais
85
86 corv = zeros(Ne,1) ; % correlation de validation
87
88 %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Entrainement %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% %%
89 % boucle sur les epoch
90 for t = 1 : 1 : Ne
91
92 % vecteur de permutation pour mélanger les entrées
93 r = randperm(ne) ;
94
95 % boucle sur les mini-batch
96 for m = 1 : 1 : Nm
97
98 x0m = Ee(:,r((m-1)*tm+1 : m*tm)) ; % entrées du mini-batch
99 xm = [x0m ; zeros(sum(Nn(2:end)),tm)] ;
100
101 Tm = Te(:,r((m-1)*tm+1 : m*tm)) ; % cibles du mini-batch
102
103 %% propagation : eq. 2.38/2.39
104 for c = 1 : 1 : Nc % boucle sur les couches en avant
105 wc = reshape(w(Nn(1:c-1)*Nn(2:c)'+1:Nn(1:c)*Nn(2:c+1)'),
106 [Nn(c+1) Nn(c)]) ; % matrice des poids couche c
107 s(sum(Nn(2:c))+1 : sum(Nn(2:c+1)), :) = ...
108 wc*xm(sum(Nn(1:c-1))+1 : sum(Nn(1:c)), :) + ...
109 b(sum(Nn(2:c))+1 : sum(Nn(2:c+1))) ;
110 xm(sum(Nn(1:c))+1 : sum(Nn(1:c+1)), :) = ...
111 activ{ia(c)}(s(sum(Nn(2:c))+1 : sum(Nn(2:c+1)), :)) ;
112 end
113
114 %% rétro-propagation : eq. 2.40/2.41
115 for c = Nc : -1 : 1 % boucle sur les couches en arrière
116 if c == Nc
117 d(sum(Nn(2:c))+1 : sum(Nn(2:c+1)), :) = ...
118 dC{ic}(xm(sum(Nn(1:c))+1 : sum(Nn(1:c+1)), :), Tm).* ...
119 dactiv{ia(c)}(s(sum(Nn(2:c))+1 : sum(Nn(2:c+1)), :)) ;
120 else
121 wcp1 = reshape(w(Nn(1:c)*Nn(2:c+1)'+1 : Nn(1:c+1)*
122 Nn(2:c+2)'), [Nn(c+2) Nn(c+1)]) ; % matrice des poids
123 couche c+1
124 d(sum(Nn(2:c))+1 : sum(Nn(2:c+1)), :) = ...
125 (wcp1'*d(sum(Nn(2:c+1))+1 : sum(Nn(2:c+2)), :)).* ...
126 dactiv{ia(c)}(s(sum(Nn(2:c))+1 : sum(Nn(2:c+1)), :)) ;
127 end
128 end
129

```

```

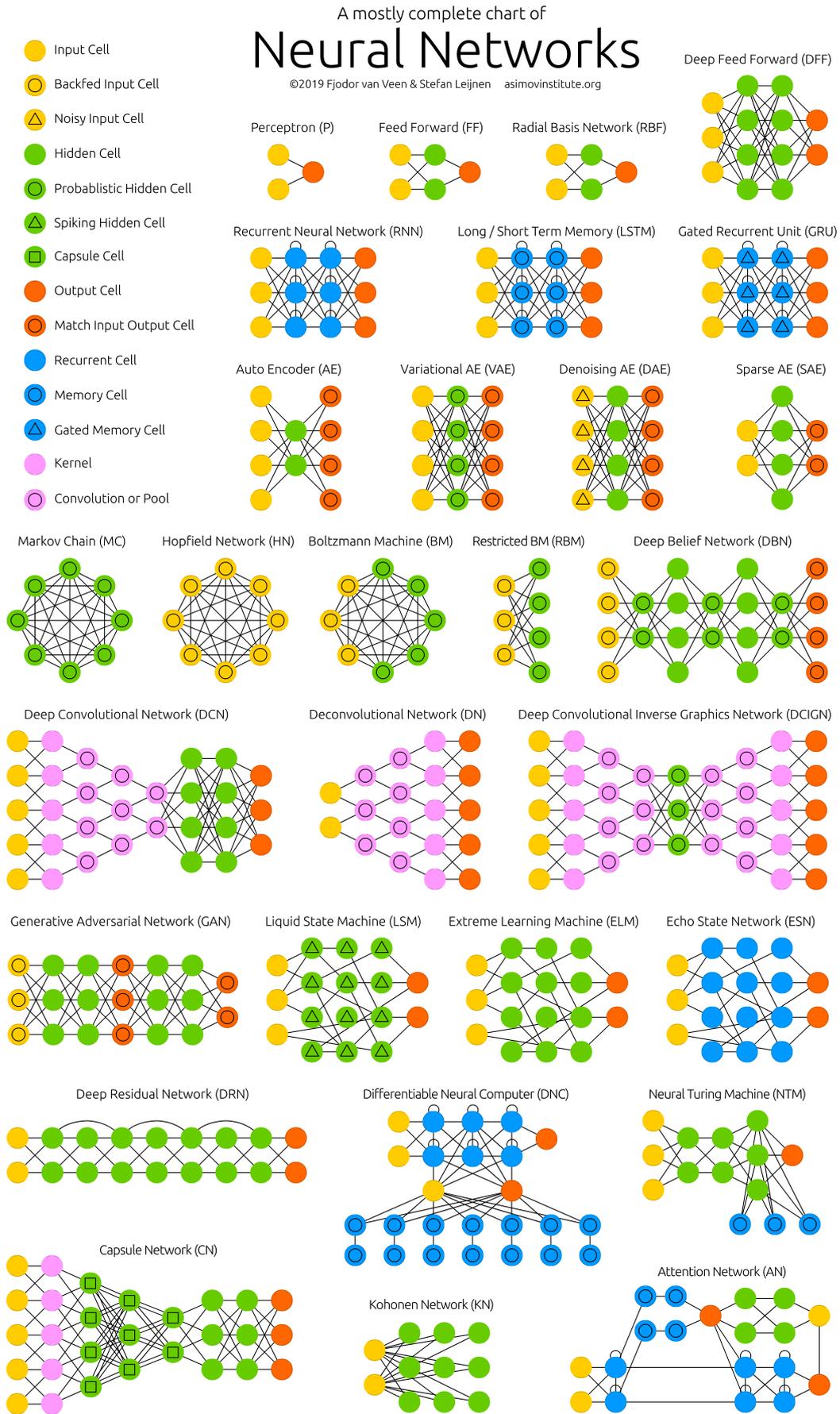
130  %% mise à jour des paramètres : eq. 2.44
131  for c = 1 : 1 : Nc % boucle sur les couches en avant
132      wc = w(Nn(1:c-1)*Nn(2:c)'+1 : Nn(1:c)*Nn(2:c+1)') ;
133      bc = b(sum(Nn(2:c))+1 : sum(Nn(2:c+1))) ;
134      vwc = vw(Nn(1:c-1)*Nn(2:c)'+1 : Nn(1:c)*Nn(2:c+1)') ;
135      vbc = vb(sum(Nn(2:c))+1 : sum(Nn(2:c+1))) ;
136
137      vw(Nn(1:c-1)*Nn(2:c)'+1 : Nn(1:c)*Nn(2:c+1)') = ...
138      mu*vwc - eta*(reshape(d(sum(Nn(2:c))+1 : sum(Nn(2:c+1)), :) *
139      xm(sum(Nn(1:c-1))+1 : sum(Nn(1:c))), :)'/tm , ...
140      [Nn(c+1)*Nn(c) 1]) + lambda*L{il}(wc)/ne) ;
141
142      vb(sum(Nn(2:c))+1 : sum(Nn(2:c+1))) = ...
143      mu*vbc - eta*sum(d(sum(Nn(2:c))+1 : sum(Nn(2:c+1)), :), 2)/tm;
144
145      w(Nn(1:c-1)*Nn(2:c)'+1 : Nn(1:c)*Nn(2:c+1)') = wc + ...
146      vw(Nn(1:c-1)*Nn(2:c)'+1 : Nn(1:c)*Nn(2:c+1)') ;
147      b(sum(Nn(2:c))+1 : sum(Nn(2:c+1))) = bc + ...
148      vb(sum(Nn(2:c))+1 : sum(Nn(2:c+1))) ;
149  end
150 end
151
152 %% validation
153 xv = Ev ; % entrées de validation
154
155 % boucle sur les couches en avant
156 for c = 1 : 1 : Nc
157     wc = reshape(w(Nn(1:c-1)*Nn(2:c)'+1:Nn(1:c)*Nn(2:c+1)') ,
158     [Nn(c+1) Nn(c)]) ; % matrice des poids couche c
159     xv = activ{ia(c)}(wc*xv+b(sum(Nn(2:c))+1:sum(Nn(2:c+1))));
160 end
161
162 corV = corrcoef(Tv,xv) ; % matrice de corrélation
163 corv(t) = corV(1,2) ; % coefficient de corrélation
164 end
165
166 %% %%%%%%%%%%% Test %%%%%%%%%%% %%
167 xt = Et ; % entrées de test
168
169 % boucle sur les couches en avant
170 for c = 1 : 1 : Nc
171     wc = reshape(w(Nn(1:c-1)*Nn(2:c)'+1:Nn(1:c)*Nn(2:c+1)') ,
172     [Nn(c+1) Nn(c)]) ; % matrice des poids couche c
173     xt = activ{ia(c)}(wc*xt+b(sum(Nn(2:c))+1:sum(Nn(2:c+1))));
174 end
175
176 corT = corrcoef(Tt,xt) ; % matrice de corrélation
177 cort = corT(1,2) ; % coefficient de corrélation

```

Annexe B

Bestiaire des architectures de réseaux de neurones

Ces dernières années ont vues l'émergence de nombreux algorithmes d'apprentissage machine, et en particulier de nombreuses architectures de réseaux de neurones. Ces derniers existent donc dans un grande variété, et ont été testées dans de nombreux scénario différents (reconnaissance d'image, génération de texte, analyse de données physique, ...). Chaque architecture a alors trouvé un ou plusieurs champ(s) d'application(s) approprié(s) ; par exemple, les réseaux de neurones convolutionnels conviennent particulièrement pour la reconnaissance d'image, et les réseaux récurrents pour l'analyse de séries temporelles. Un bestiaires des différentes architectures de réseaux de neurones existant aujourd'hui est donc présenté dans cette annexe.



Annexe C

Publications et conférences

Liste des publications :

M. Mabed, F. Meng, L. Salmela, C. Finot, G. Genty, J. M. Dudley, Machine learning analysis of instabilities in noise-like pulse lasers. *Optics Express*, 30 :15060-15072, 2022

L. Salmela, M. Hary, M. Mabed, A. Foi, J. M. Dudley, G. Genty, Feed-forward neural network as nonlinear dynamics integrator for supercontinuum generation. *Optics Letters*, 47 :802-805, 2022

A. V. Ermolaev, M. Mabed, C. Finot, G. Genty, and J. M. Dudley, Analysis of interaction dynamics and rogue wave localization in modulation instability using data-driven dominant balance. *Scientific Reports*, 13 :10462, 2023

M. Mabed, L. Salmela, A. Ermolaev, C. Finot, G. Genty, J. M. Dudley, Neural network analysis of unstable temporal intensity peaks in continuous wave modulation instability. *Opt. Commun.*, 541 :129570, 2023

Conférences nationales :

M. Mabed, F. Meng, L. Salmela, G. Genty, J. M. Dudley, Analyse des caractéristiques des solitons optiques dans la génération de supercontinuum par techniques d'apprentissage profond. *Congrès de la société française d'optique*, Optique Dijon 2021, Palais des congrès, Dijon, France, 5-9 Juillet 2021

M. Mabed, F. Meng, L. Salmela, G. Genty, J. M. Dudley, Optical solitons analysis in supercontinuum generation through deep learning techniques. *Photonics Day*, Besançon, France, 13 Décembre 2021

M. Mabed, F. Meng, L. Salmela, G. Genty, J. M. Dudley, Applications of Machine Learning to Ultrafast Nonlinear Dynamics. *Congrès de la société française d'optique*, Optique Nice 2022. Nice, France, 4-8 Juillet 2022

Conférences internationales :

C. Lapre, C. Billet, F. Meng, M. Mabed, C. Finot, L. Salmela, G. Genty, J. M. Dudley, Multipulse and Molecule states in a broadband Mamyshev oscillator around 1550 nm. *The European Conference on Lasers and Electro-Optics*, Paper CJ4.1. Virtual Event, 20-24 Juin 2021

G. Genty, L.Salmela, M. Hary, M. Mabed, C. Lapre, J. M. Dudley, Applications of Machine Learning to Ultrafast Nonlinear Dynamics. *Invited Paper SF4F.3*, CLEO Laser Science to Photonic Applications, San Jose, USA, 9-11 Mai 2022

J. M. Dudley, M. Mabed, A. Ermolaev, L. Salmela, F. Meng, C. Finot, G. Genty, Applications of Machine Learning to Ultrafast Nonlinear Dynamics. *International Conference on Photonics*, IEEE Photonics Society Malaysia Chapter, 8-10 Août 2022

J. M. Dudley, M. Mabed, A. Ermolaev, C. Lapre, F. Meng, A. Sheveleva, C. Finot, G. Genty, Recent advances understanding rogue waves and analogies in nonlinear fibre optics. *International Workshop on Rogue Waves*, University of Vermont, Burlington, Vermont, USA, 17-18 Décembre 2022

G. Genty, L. Salmela, M. Hary, M. Mabed, A. Ermolaev, J. M. Dudley, Applications of artificial intelligence to nonlinear fiber-optics. Paper PC12438-46. *AI and Optical Data Sciences V*, SPIE Photonics West 2023, San Francisco, 28 Janvier-2 Février 2023

M. Mabed, A. Ermolaev, L. Salmela, G. Genty, J. M. Dudley, Machine Learning analysis of Continuous Wave fiber Modulation Instability. *International Workshop on Optical Wave & Waveguide Theory and Numerical Modeling*, Marseille, France, 4-5 Mai 2023

Titre : Application de méthodes d'apprentissage profond à l'analyse et modélisation de la dynamique non-linéaire dans les fibres optiques

Mots clés : optique non-linéaire, apprentissage machine, optique ultra-rapide

Résumé : La propagation d'impulsions lumineuses non-linéaires dans les fibres optiques présente un large éventail de dynamiques complexes. En fonction des conditions initiales et des propriétés des fibres utilisées, il est possible d'observer des dynamiques solitoniques complexes pouvant engendrer un élargissement spectral prononcé, allant jusqu'à la génération de supercontinuum. La complexité de ces effets peut rendre leur analyse et leur modélisation difficiles, et l'objectif de cette thèse a été de montrer l'application de techniques d'apprentissage machine pour analyser différents scénarios de propagation d'impulsions non-linéaires dans des systèmes optiques fibrés régis par l'équation de Schrödinger non-linéaire. En particulier, la prédiction des propriétés temporelles des "vagues scélérates" optiques en se basant uniquement sur l'analyse de l'intensité spectrale, sans aucune information sur la phase, a été accomplie à l'aide d'un réseau de neurones à propagation avant. Ces prédictions se sont avérées très satisfaisantes en régime d'instabilité de modulation pure ainsi que dans un système de cavité laser *noise-like pulse* en régime de bande spectrale étroite et large. Enfin, l'algorithme de réseau de neurones à propagation avant s'est avéré remarquablement précis dans sa capacité de reconstruction de la dynamique de l'équation de Schrödinger non-linéaire.

Title : Application of deep learning methods to the analysis and modeling of nonlinear dynamics in optical fibers

Keywords : non-linear optics, machine learning, ultrafast optics

Abstract : Nonlinear pulse propagation in optical fibers exhibits a wide range of complex dynamics. Depending on the initial conditions and parameters of the fiber used, it is possible to observe a range of instability and solitonic dynamics leading to pronounced spectral broadening, and even supercontinuum generation. The complexity of these effects can make their analysis and modeling difficult, and the aim of this thesis has been to demonstrate the application of machine learning techniques to analyse different scenarios of nonlinear pulse propagation governed by the nonlinear Schrödinger equation. In particular, the prediction of the temporal properties of optical "rogue waves" based solely on spectral intensity analysis, without any phase information, was accomplished using a feedforward neural network. These predictions proved highly satisfactory in systems of pure modulation instability, as well as noise-like pulse laser operation in narrow- and wide-band spectral regimes. Finally, the feedforward neural network algorithm also proved remarkably accurate in reconstructing the dynamics of the nonlinear Schrödinger equation.