

RAPPORT DE STAGE

Modélisation 2D de l'équation du transfert radiatif et reconstruction de la densité par un réseau de neurones

Auteur :
Roussel Desmond NZOYEM

Maitres de stage :
Emmanuel FRANCK
Laurent NAVORET
Vincent VIGON

Enseignant référent :
Christophe PRUD'HOMME

*Stage realise dans le cadre du Master CSMI
du 15 juin 2020 au 15 aout 2020
a l'initiative de l'equipe MOCO
au sein de l'UFR de mathématiques et d'informatique*

Annee academique 2019 - 2020

15 août 2020

Remerciements

Je tiens à remercier dans un premier temps mes maîtres de stages MM. Emmanuel FRANCK, Laurent NAVORET, et Vincent VIGON de m'avoir permis d'effectuer un stage scientifique très enrichissant dans les meilleurs conditions possibles, compte tenu de la situation sanitaires liée au COVID-19. Je tiens particulièrement à les remercier pour leur patience et leur compréhension devant ma lenteur et mon manque de maîtrise de la matière à plusieurs reprises.

Je remercie aussi mes camarades de CSMI Guillaume STEIMER et Léo BOIS pour leurs conseils qui ont été instrumentaux dans l'amélioration de mes résultats.

Table des matières

11

ETR Equation (du) Transfert Radiatif

ETL Equilibre Thermique Local

UFR Unite deFormation et de Recherche de l'Universite de Strasbourg

Liste des symboles

ρ	densite du milieu	kg m^{-3}
σ_a	opacite d'absorption	m^{-1}
σ_c	opacite de diffusion (de scattering)	m^{-1}
Symbol	Name	Unit
c	vitesse de la lumiere	m s^{-1}

Chapitre 1

Introduction

En 2015, le réseau de neurones vainqueur de l'ILSVRC¹ obtient une précision de 97.3% ce qui conduit les chercheurs à postuler que les machines peuvent identifier les objets dans des images mieux que les humains (**Reference1**). Depuis lors, le domaine du machine learning a continué à prendre de l'ampleur et ses applications et se multiplient dans les domaines industriels, académiques, du divertissement, et plus important, en tomographie médicale.

Grâce à son unité mixte de recherche l'IRMA, L'UFR de math-info est un pôle de recherche en matière de calcul scientifique et probabilités. À travers ses équipes MOCO et Probabilités, l'IRMA s'intéresse aux problèmes de modélisation des EDP et de Machine Learning, raison pour laquelle j'ai choisi d'y effectuer mon stage de master 1 CSML. Au cours de ce stage qui s'est déroulé du 15 juin au 15 août 2020, j'ai pu m'intéresser au problème inverse de reconstruction de la densité d'un domaine par une architecture de réseau de neurone.

Ce stage a été suivi par les enseignants-chercheurs MM. Emmanuel FRANCK, Laurent NAVORET, et Vincent VIGON et s'inscrit dans la continuation d'un projet effectué du 20 Mars au 9 Mai 2020 au sein de l'IRMA (et encadré par la même équipe). Le projet consistait en l'élaboration d'un logiciel de résolution du modèle P1 de l'équation du transfert radiatif (ETR)² en 1D. Le stage quant à lui a consisté en la résolution du modèle P1 en 2D, et en la reconstruction la densité du domaine par une architecture de réseau de neurones. Plus généralement, ce stage a été l'opportunité pour moi d'apprendre sur les EDP et les réseaux de neurones tout en me familiarisant avec l'API de la fameuse librairie de réseaux de neurones Keras.

Les tissus cancéreux ont des propriétés optiques différentes de celles des tissus aux alentours³. Étant donné un domaine et un signal qui s'y propage, reconstruire la densité à l'aide du signal temporel mesuré sur les bords de ce domaine constitue un problème inverse. Ces problèmes sont très importants et se rencontrent dans beaucoup d'autres domaines (radar, vision, imagerie informatique). Ils sont malheureusement très difficiles à résoudre car ils nécessitent l'utilisation d'algorithmes d'optimisation avancés. L'utilisation des réseaux de neurones se présente comme une méthode potentiellement moins coûteuse et plus rapide.

En vue de rendre compte de manière fidèle des deux mois passés au sein de l'UFR, il apparaît logique de présenter en titre de préambule le cadre du stage et son environnement technique. Ensuite il s'agira de présenter les différentes missions et tâches que j'ai pu effectuer. Enfin je présenterais un bilan du stage, incluant les différents apports et enseignements que j'ai pu en tirer.

1. ImageNet Large Scale Visual Recognition Challenge

2. l'équation (1) rappelle l'ETR et l'équation (2) définit sa simplification en modèle P1

3. les tissus cancéreux sont généralement plus denses que les tissus sains

Chapitre 2

Présentation de l'IRMA

Les informations presentes dans cette section sont entierement issues du [site web de l'IRMA](#)

Cree en 1966 par Jean Frenjel et Gearges Reebe, l'IRMA¹ est une unité mixte de recherche (UMR 7501) sous la double tutelle du CNRS (a travers l'INSMI²) et de l'Université de Strasbourg (UFR de Mathématique et d'Informatique).

Dirigee par le professeur Philippe Helluy, l'IRMA comporte environ 130 membres. On y compte enviro 87 chercheurs et enseignants-chercheurs permanent et une quarantaine de non permanents repartis en 8 equipes de recherche.

Les activites majeures de l'organisme sont l'organisation des seminaires, des journees, des colloques et des conferences. Ces activites sont renforcees par les nombreux partenariats qu'elle maintient dans les secteurs academique(Cemosis, Labex IRMIA, etc..) et indutriel (AxesSIM, Electis, etc..).

2.1 Structure de l'organisation

L'organigramme de l'entreprise representant ses sections majeures est donne a la figure ??).

2.2 Les équipes MOCO et Probabilite

L'equipe MOCO³ se compose de specilistes des EDP, de la theorie deu controle, ddu calcul scientifique et haute performance et des statistiques. Ses activites s'étendent a l'internationale et dans l'indutriel (REF, ...). Les enseignants-chercehurs MM. Emmanuel Franck et en Laurent Navoret y sont responsables des seminaires en equations aux derivees partielles.

L'equipe probabilite est composee d'experts en calcul de probabilite. Ses membres se retrouvent regulierement lors du "seminaire stochatique". A cette equipe appartient M. Vincent Vigon.

Je tiens une fois de plus a remercier les trois chercheurs mentiones ci-hauts qui ont encadrer ce stage. La combinaison de ces deux equipes dont ils font partie a permis de faire face aux deux aspects de ce stage. Premièrement la modelisation d'EDP et finalement l'utilisation des reseaux de neurones.

1. Institut de Recherche Mathématique Avancée

2. Institut National des Sciences Mathématiques et de leurs Interactions

3. MOdélisation et COntrole



FIGURE 2.1 – Organigramme représentant l'organisation de l'IRMA au mois de mars 2020 (Reference7)

Chapitre 3

Modelisation de l'EDP en 2D

Ayant resolu le modele en 1D durant le stage, on procede dans cette partie a sa modelisation en 2D. Il s'agit de resoudre le probleme direct du transfer radiatif avant de passer au probleme inverse dans la partie suivante. On rappelle brevement le modele considere avant de decrite l'implementation que utilisee.

3.1 Le transfert radiatif

On considere un rayonnement transporte par des particules de masse nulle appeles photons. Lorsqu'ils se trouvent en presence de la matiere, les photons inteassgissent avec celle. Trois phono-mees sont preponderant (figure ??) :

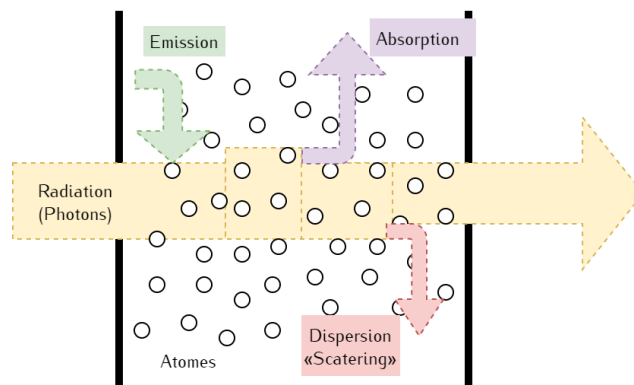


FIGURE 3.1 – Illustration des interaction entre radiation et matiere.

- l'émission : Les photons sont emis en reponse aux electrons excites descendants a des niveaux d'énergie plus bas. Ce phenomenes est caraterise par l'opacite d'émission σ_e . Il s'agit de l'inverse du libre parcours d'emission¹. Plus la temperature matiere est elevee, plus ce phenomene est important.
- l'absorption : A l'inverse, certains photons sont absorbes par la matiere. Ce pehenomene se caraterise par l'opacite d'absorption σ_a . Lorsqu'on est a l'equilibre thermique, $\sigma_e = \sigma_e$.
- la dispersion (ou "scattering" ou parfois diffusion) : Certains photons sont devies de leur trajectoire par la matiere. Ce phenomme se caraterise non seulement par son opacite de scatering σ_c ², mais aussi par une fonction de distribution angulaire decrivant la maniere dont les photons sont devies.

1. Le libre par cours moyen d'émission represente la distance moyenne entre deux emissions de photons. Les libres parcours d'absorption et de dispersion sont definis de maniere similaire

2. σ_a et σ_c sont definis de maniere similaire a σ_e

L'équation du transfert radiatif (ETR) (équation ??) représente un bilan d'énergie lié au rayonnement au niveau microscopique. Nous nous placerons dans le cas particulier d'équilibre thermodynamique local (ETL)³. L'équilibre radiatif⁴ quant à lui sera considéré comme condition initiale pour les simulations.

$$\begin{aligned} \frac{1}{c} \frac{\partial}{\partial t} I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) + \boldsymbol{\Omega} \cdot \nabla_{\mathbf{x}} I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) &= \sigma_a(\rho, \boldsymbol{\Omega}, \nu) (B(\nu, T) - I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu)) \\ &+ \frac{1}{4\pi} \int_0^\infty \int_{S^2} \sigma_c(\rho, \boldsymbol{\Omega}, \nu) p(\boldsymbol{\Omega}' \rightarrow \boldsymbol{\Omega}) (I(t, \mathbf{x}, \boldsymbol{\Omega}', \nu) - I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu)) d\boldsymbol{\Omega}' d\nu \end{aligned} \quad (3.1)$$

Où I représente l'intensité spécifique de radiation et p (telle que $\oint p(\boldsymbol{\Omega}' \rightarrow \boldsymbol{\Omega}) d\boldsymbol{\Omega}' = 1$) est la fonction de distribution angulaire de "staring". Les autres termes sont définis dans la table des symboles.

Il est possible de modéliser l'ETR à travers plusieurs modèles. Le modèle P1 est un modèle macroscopique⁵ aux moments (d'ordre 2), linéaire et hyperbolique. Vu que l'énergie du rayonnement n'est pas conservée durant son interaction avec la matière, il faut coupler le modèle P1 avec une équation régissant l'énergie de la matière. On utilisera une équation d'énergie matière simplifiée qui ne tient compte que des termes d'échange avec le rayonnement. Le modèle P1 couplé à la matière est présenté ci-bas (**Reference2**) :

$$\begin{cases} \partial_t E + c \operatorname{div} \mathbf{F} = c\sigma_a (aT^4 - E) \\ \partial_t \mathbf{F} + c \nabla E = -c\sigma_c \mathbf{F} \\ \rho C_v \partial_t T = c\sigma_a (E - aT^4) \end{cases} \quad (3.2)$$

Dans l'équation ??, σ_a et σ_c sont écrits sans indiquer leurs arguments⁶ afin de faciliter les notations. $E, \mathbf{F},$ et T représentent respectivement l'énergie des photons, le flux de photons, et la température radiative. Partant de l'équation ??, E et T sont définies de la manière suivante :

$$\begin{aligned} E(t, \mathbf{x}) &= \frac{4\pi}{c} \int_0^\infty \int_{S^2} I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) d\boldsymbol{\Omega} d\nu \\ \mathbf{F}(t, \mathbf{x}) &= \frac{4\pi}{c} \int_0^\infty \int_{S^2} \boldsymbol{\Omega} I(t, \mathbf{x}, \boldsymbol{\Omega}, \nu) d\boldsymbol{\Omega} d\nu \end{aligned}$$

Comme on peut le voir à travers la définition de E et \mathbf{F} , notre modèle est dit "gris" car nous l'intégrons sur tout le spectre de fréquence. En effet, nous ne nous intéressons qu'au rayonnement à travers son bilan d'énergie transporté par le flux radiatif. Sur ce point, la version du modèle P1 que nous avons utilisé est moins précise qu'un modèle microscopique basé soit sur une méthode Monte-Carlo ou une méthode des ordonnées discrètes. Néanmoins notre modèle présente l'avantage d'être très peu coûteux et relativement facile à implémenter (**Reference3**).

3.2 Schéma de splitting

Le modèle P1 tend vers une équation de diffusion lorsque les opérateurs d'absorption (σ_a) et de dispersion (σ_c) sont élevés (de façon à ce que $c/\sigma_a = 1$). Les schémas classiques tels

3. état dans lequel on peut définir une température pour chaque point du domaine, et l'émission est décrite par la fonction de Planck (**Reference3**)

4. il se produit si la matière est à l'équilibre avec le rayonnement. Si on est dans l'ETL, les photons sont émis suivant la fonction de Planck à la température de la matière

5. ils ne prennent en compte que les variables d'espace et de temps et sont obtenus par intégration des termes microscopiques tels que I par rapport à la fréquence et la direction

6. juste ρ et T si on se place dans l'ETL

que le schema de Rusanov ne sont pas assez precis pour capturer cette propriete (**Reference4**).

$$\partial_t \left(aT^4 + \rho C_v T \right) - \text{div} \left(\frac{acT^3}{\sigma_a} \nabla T \right) = O \left(\frac{1}{c} \right) \quad (3.3)$$

Le schema en 2 etape (ou de Splitting) propose par (franck) est assez precis pour traduire la limite de diffusion. Les deux etapes sont resumees co-bas.

3.2.1 Etape 1

La premiere etape (dite etape de couplage ou d'equilibre, ou etape de relaxation de la temperature) permet de regler la temperature sur chaque maille (independamment des autres mailles). On ne considere que les equations ou la temperature est impiquee (equations 1 et 3 du modele P1 ??), en fixant la valeur du flux sur chaque maille. Il s'agit d'une methode de point fixe qui est toujours definie. (**Reference2**)

Le domaine rectangulaire est suppose discretise en $N \times M$ mailles uniformes. On se trouve sur la maille j (Figure ??) a l'etape d'iteration n .

On pose donc $\Theta = aT^4$ et on obtient lw sytème :

$$\begin{cases} \frac{E_j^{q+1} - E_j^n}{\Delta t} = c\sigma_a(\Theta_j^{q+1} - E_j^{q+1}) \\ \rho_j C_v \mu_q \frac{\Theta_j^{q+1} - \Theta_j^n}{\Delta t} = c\sigma_a(E_j^{q+1} - \Theta_j^{q+1}) \end{cases}$$

Ou $\mu_q = \frac{1}{T^{3,n} + T^n T^{2,q} + T^q T^{2,n} + T^{3,q}}.$

L'etape revient a resoudre un systme de Cramer. On obtient au final :

$$\begin{cases} E_j^{q+1} = \frac{\alpha E_j^n + \beta \gamma \Theta_j^n}{1 - \beta \delta} \\ \Theta_j^{q+1} = \frac{\gamma \Theta_j^n + \alpha \delta E_j^n}{1 - \beta \delta} \end{cases} \quad (3.4)$$

Avec $\alpha = \frac{1}{\Delta t \left(\frac{1}{\Delta t} + c\sigma_a \right)}, \quad \beta = \frac{c\sigma_a}{\frac{1}{\Delta t} + c\sigma_a}, \quad \gamma = \frac{\rho_j C_v \mu_q}{\Delta t \left(\frac{\rho_j C_v \mu_q}{\Delta t} + c\sigma_a \right)} \quad \text{et} \quad \delta = \frac{c\sigma_a}{\frac{\rho_j C_v \mu_q}{\Delta t} + c\sigma_a}.$

On itere ainsi sur q jusqu'a ce que E et Θ convergent vers E^* et Θ^* . F reste inchange dutant cette etape.

3.2.2 Etape 2

Il s'agit ici de resoudre les deux EDP hyperboliques en 1 et 2. Avast d'qtter le schema schema de splitting, on note que les equations 1 et 2 du modele P1 (equation ??) sont hyperboliques et que La methode des volumes finis est donc adaptee pour les resoudre. On se place sur une maille j caracterisee par son volule Ω_j .

$$\begin{cases} \partial_t \int_{\Omega_j} E + c \int_{\Omega_j} \text{div} \mathbf{F} = 0 \\ \partial_t \int_{\Omega_j} \mathbf{F} + c \int_{\Omega_j} \nabla E = -c\sigma_c \int_{\Omega_j} \mathbf{F} \end{cases}$$

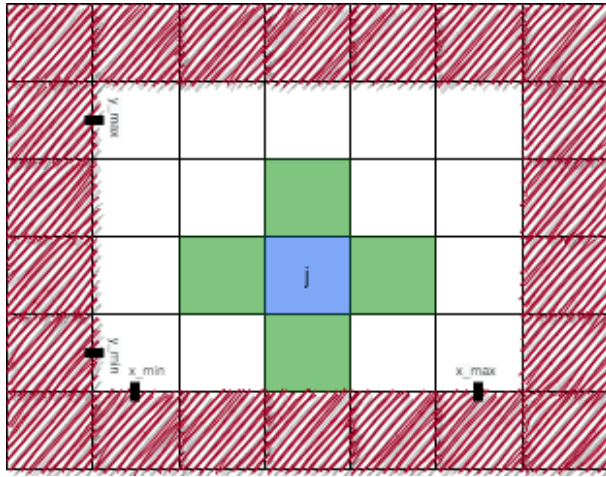
On definit une normale \mathbf{n}_j a la surface Ω_j (voir figure ?? B) et on applique le theoreme de la divergence. On moyenne les integrales sur chaque mailles pour obtenir :

$$\begin{cases} \partial_t E_j + \frac{c}{|\Omega_j|} \int_{\partial\Omega_j} (\mathbf{F}, \mathbf{n}_j) = 0 \\ \partial_t \mathbf{F}_j + \frac{c}{|\Omega_j|} \int_{\partial\Omega_j} (E, \mathbf{n}_j) = \frac{c\sigma_c}{|\Omega_j|} \int_{\Omega_j} \mathbf{F} \end{cases} \quad (3.5)$$

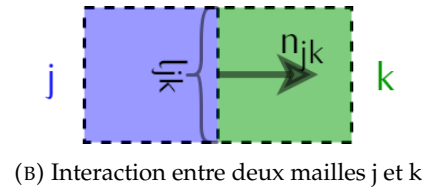
Avec

$$E_j(t) = \frac{1}{|\Omega_j|} \int_{\Omega_j} E(t, \mathbf{x}) \quad \text{et} \quad \mathbf{F}_j(t) = \frac{1}{|\Omega_j|} \int_{\Omega_j} \mathbf{F}(t, \mathbf{x})$$

Nous retournons donc sur le maillage discretise en definissant les differents flux numeriques implicites. Durant cette etape, il faut considerer l'ajout de mailles fantomes, ce qui porte le nombre total de volumes a $(N + 2) \times (M + 2)$.



(A) Discretisation du domaine



(B) Interaction entre deux mailles j et k

FIGURE 3.2 – Discretisation du maillage 2D. Sur la figure (A), on peut observer les mailles dites "fantomes" hachurees en rouge. Les quatre mailles voisines d'une maille j sont indiquees en vert. Le volume de la maille j est defini par Ω_j . Le nombre de mailles suivant l'horizontale M est choisi telle que le maillage soit uniforme i.e $\Delta x = \frac{x_{max}-x_{min}}{N} = \frac{y_{max}-y_{min}}{M} = \Delta y$. Sur la figure (B), on observe la definition de la normale sortante \mathbf{n}_{jk} de la mailles j . On peut aussi observer la longueur caracteristique l_{jk}

Les flux numeriques a l'etape d'iteration n sont definit entre une maille j et sa maille voisine k comme suit (**Reference4**) :

$$\begin{aligned} (\mathbf{F}_{jk}, \mathbf{n}_{jk}) &= l_{jk} M_{jk} \left(\frac{\mathbf{F}_j^n \cdot \mathbf{n}_{jk} + \mathbf{F}_k^n \cdot \mathbf{n}_{jk}}{2} - \frac{E_k^n - E_j^n}{2} \right) \\ (E_{jk}, \mathbf{n}_{jk}) &= l_{jk} M_{jk} \left(\frac{E_j^n + E_k^n}{2} - \frac{\mathbf{F}_k^n \cdot \mathbf{n}_{jk} - \mathbf{F}_j^n \cdot \mathbf{n}_{jk}}{2} \right) \mathbf{n}_{jk} \end{aligned}$$

En posant :

$$\begin{aligned}\mathbf{S}_j &= - \left(\sum_k M_{jk} \sigma_{jk} \right) \mathbf{F}_j^{n+1} \\ \mathbf{S}'_j &= \frac{1}{|\Omega_j|} \left(\sum_k l_{jk} M_{jk} \mathbf{n}_{jk} \right) E_j^n \\ M_{jk} &= \frac{2}{2 + \Delta x \sigma_{jk}} \\ \sigma_{jk} &= \frac{1}{2} \left(\sigma_c(\rho_j, T_j^n) + \sigma_c(\rho_k, T_k^n) \right)\end{aligned}$$

On peut donc ecrire cette etape du schema sous la forme :

$$\begin{cases} \frac{E_j^{n+1} - E_j^*}{\Delta t} + \frac{c}{|\Omega_j|} \sum_k (\mathbf{F}_{jk}, \mathbf{n}_{jk}) = 0 \\ \frac{\mathbf{F}_j^{n+1} - \mathbf{F}_j^*}{\Delta t} + \frac{c}{|\Omega_j|} \sum_k (E_{jk}, \mathbf{n}_{jk}) - c \mathbf{S}'_j = c \mathbf{S}_j \end{cases}$$

Qui se reecrit comme suit :

$$\begin{cases} E_j^{n+1} = E_j^* + \alpha \sum_k (\mathbf{F}_{jk}, \mathbf{n}_{jk}) \\ \mathbf{F}_j^{n+1} = \beta \mathbf{F}_j^* + \gamma E_j^n + \delta \sum_k (E_{jk}, \mathbf{n}_{jk}) \end{cases} \quad (3.6)$$

Avec

$$\begin{aligned} \alpha &= -\frac{c \Delta t}{|\Omega_j|}, \quad \beta = \frac{1}{\Delta t} \left(\frac{1}{\Delta t} + c \sum_k M_{jk} \sigma_{jk} \right)^{-1}, \quad \gamma = \frac{c}{|\Omega_j|} \left(\frac{1}{\Delta t} + c \sum_k M_{jk} \sigma_{jk} \right)^{-1} \left(\sum_k l_{jk} M_{jk} \mathbf{n}_{jk} \right) \\ \text{et } \delta &= -\frac{c}{|\Omega_j|} \left(\frac{1}{\Delta t} + c \sum_k M_{jk} \sigma_{jk} \right)^{-1} \end{aligned}$$

La condition de CFL $\Delta t < \frac{\Delta x}{c}$ est necessaire pour assurer la stabilite du shema. Lors de l'implementation en C++, on remarquera qu'en pratique, il faut prendre $\Delta t < 0.5 \times \frac{\Delta x}{c}$.

3.3 Implementation en C++

Le dode de calcul a ete developpe durant la 4 eme semaine du stage. Etant donne des parametre du probleme, il permet d'exporter les signaux temporels E, F et T sur les quatres bords du domaine. Pour raisons de visualisation, il permet aussi dd'exporter les signaux sur l'entierete du domaine en tout temps. Ces signaux peuvent ensuite etre visualiser sous forme d'une animation a l'aide d'un notebook construit a cet effet.

L'executable se nomme `transfer` et est disponible avec le reste du code sur le repository Github [projet-inverse-2d](#).

3.3.1 Configuration du modele

L'exécutable nécessite un fichier de Configuration pour s'exécuter (.CFG, .TXT, etc.). Les paramètres à définir sont indiqués ci-dessous. Les détails supplémentaires sont donnés en annexe ???. La manière dont un fichier de configuration pourrait être indiquée à la figure ??.

```
x_min 0
x_max 1
y_min 0
y_max 1
N 90

c 299
a 0.01372
C_v 0.14361

CFL 0.5
precision 1e-6
t_0 0
t_f 0.01

rho crenau(0.5,0.5,0.1,10)
sigma_a rho*T
sigma_c rho*T

E_0 0.01372*(5^4)
F_0_x 0
F_0_y 0
T_0 5

E_u neumann
F_u_x neumann
F_u_y neumann
T_u neumann

E_d neumann
F_d_x neumann
F_d_y neumann
T_d neumann

E_l ponctuel(0.4,0.6)
F_l_x 0
F_l_y 0
T_l 5

E_r neumann
F_r_x neumann
F_r_y neumann
T_r neumann

export_file data/df_simu.csv
export_mode dataframe
write_mode truncate
```

FIGURE 3.3 – Exemple d'un fichier de configuration au format texte. Ici ne sont représentés que les 38 paramètres obligatoires pour faire tourner une simulation et l'exporter. Le résultat produit par ce fichier est présenté aux figures ?? et ??

3.3.2 Sauvegarde des données

Comme mentionné ci-haut, on dispose de deux options pour sauvegarder les résultats de la simulation :

- sous le format CSV : Ce mode permet une visualisation facile des résultats à l'aide du notebook. Il est très coûteux en espace mémoire et nécessite la Bibliothèque Pandas pour le lire en forme de dataframe. Cette opération prend une quantité non négligeable de RAM, ce qui peut nuire à l'usage qu'on veut faire des données.
- sous le format SDS⁷ : Ce format binaire ne sauvegarde que les informations les plus importantes de la simulation. En l'occurrence la source utilisée, la densité du domaine, et les différents signaux sur les bords du domaine. Il est particulièrement intéressant pour générer les données nécessaires à l'apprentissage. Les détails concernant ce format sont donnés en annexe ??.

3.4 Résultats

Quelques résultats obtenus sont présentés ici. Les images ci-bas sont obtenues avec le fichier de configuration (IMAGE DF SIMU). La source est une onde sinusoïdale placée en E sur la gauche.

7. source-densite-signal

La densitee en particulier a la forme d'un signal en crenau egale a 10 sur le crenau et 0.1 en dehors. Les opacites d'absorbes sont proportionnelles a la densite.

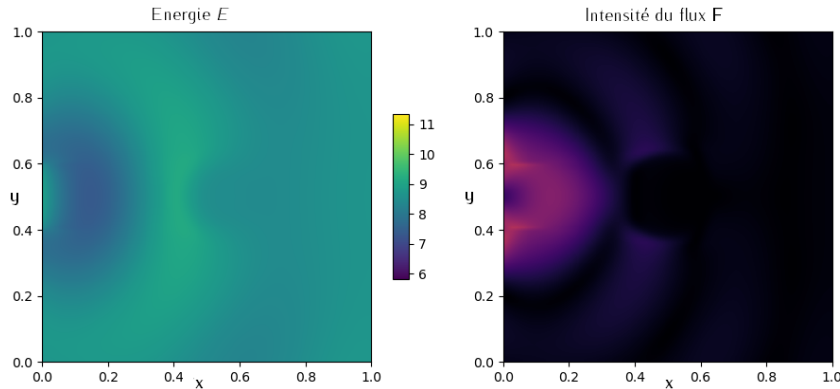


FIGURE 3.4 – Visualisation de l'energie et de l'intensite du flux des photons au temps final pour un domaine avec une densite en forme de crenau circulaire ((vu du haut)).

Cette figure correspond aux resultat obtenus avec la simulation ??

La figures ?? permet d'observer une asorption presque totale du signal au niveau du crenau du a la forte valeur des opacite d'absorption et de dispersion. En ce sens, le saut de densite agit comme un obstacle a la propagation du signal. l'evolution de l'energie sur les bords du domaine (figure ??) traduit l'effet qu'a la densite sur la propagation du signal.

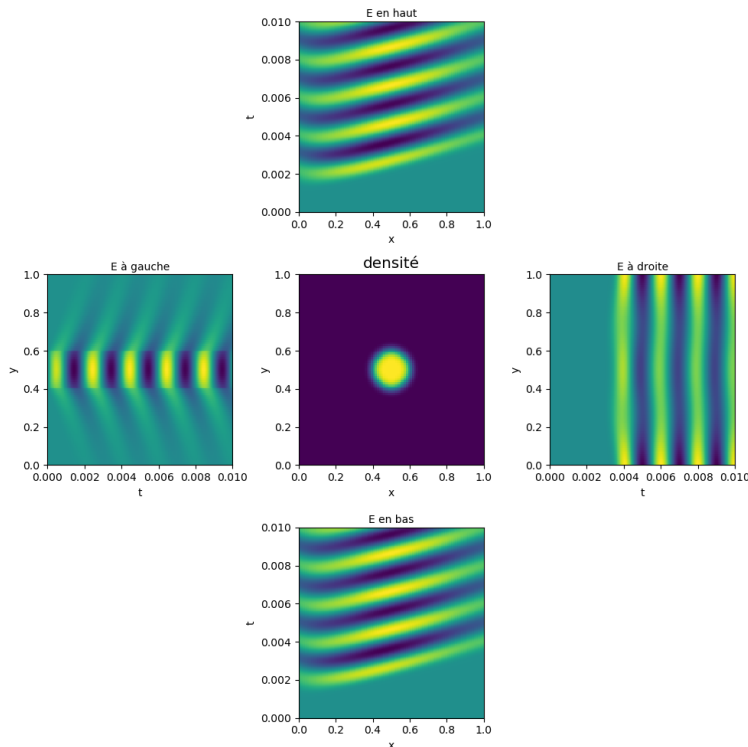


FIGURE 3.5 – Evolution de l'energie sur les bord (vue du haut). La cause du probleme direct (la densite) est illutree au milieu de l'image, et les effets (E) sont presntes aux alentours. Les indices x , y , et t representent repectivement l'abscisse, l'ordonne et le tempps. Les autres figures associees a ce cas sont ?? et ??.

Nous testons ensuite notre modèle sur le cas très particulier de l'approximation de diffusion (figures ??, ?? et ??) , un atout important du schéma de splitting que nous avons implémenté. Le bord gauche est continuellement chauffé ($E_{left} = a * ((T_{left} + 1)^4)$). La densité est un signal en forme de créneau rectangulaire (vu du haut) valant 10. En dehors de ce créneau, la densité vaut 0.1. Théoriquement, la limite de diffusion s'observe pour $\frac{c}{\sigma_c} = 1$ (voir EQUATION ...), mais en pratique cela pose des problèmes de visualisation avec les coefficients que nous avons choisis. On prend donc $\sigma_a = \sigma_c = 100 \times \rho$ ce qui donne environ $\frac{c}{\sigma_c} = 30$ en dehors de l'obstacle.

On confirme effectivement l'effet de diffusion du signal dans le domaine. Nous pouvons à présent passer à l'apprentissage. Pour l'apprentissage, nous nous placerons essentiellement dans des conditions proches de celles de la configuration ??. Une section de l'apprentissage traitera aussi du problème 1D non modélisé ici⁸.

```

x_min 0
x_max 1
y_min 0
y_max 1
N 90

c 299
a 0.01372
C_v 0.14361

CFL 0.5
precision 1e-6
t_0 0
t_f 0.01

rho crenau(0.5,0.5,0.1,10)
sigma_a rho*100
sigma_c rho*100

E_0 0.01372*(5^4)
F_0_x 0
F_0_y 0
T_0 5

E_u neumann
F_u_x neumann
F_u_y neumann
T_u neumann

E_d neumann
F_d_x neumann
F_d_y neumann
T_d neumann

E_l 0.01372*(6^4)
F_l_x 0
F_l_y 0
T_l 5

E_r neumann
F_r_x neumann
F_r_y neumann
T_r neumann

export_file data/df_simu.csv
export_mode dataframe
write_mode truncate

```

FIGURE 3.6 – Configurations utilisées pour illustrer la limite de diffusion. L'option nécessaire pour obtenir un obstacle en forme de rectangle n'est pas insérable dans le fichier de configuration, cela se fait directement dans le code de calcul.

8. le problème a été résolu pendant le projet de CSMI

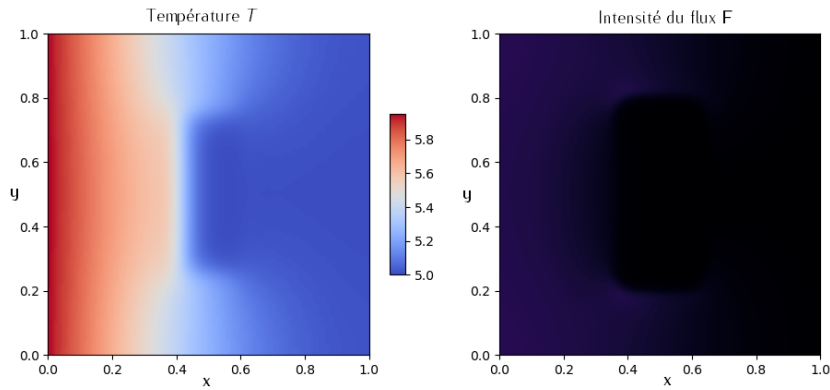


FIGURE 3.7 – Visualisation de la temperature et de l'intensité du flux des photons au temps final pour la limite de diffusion. La densité a une forme de crenau rectangulaire comme representee sur l'image du milieu de la figure ??.

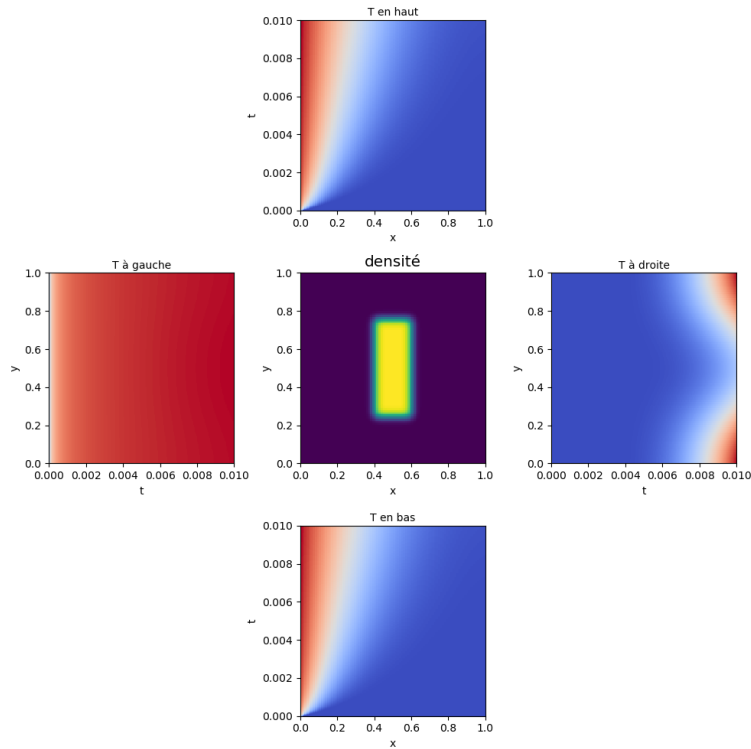


FIGURE 3.8 – Evolution de la temperature sur les bord illustrant l'effet de diffusion. Tout comme a la figure ??, l'expression des opacités pour ces simulation est $\sigma_a = \sigma_c = 100 \times \rho$ afin d'obtenir un maximum de diffusion au en dehors de l'obstacle et une absorption totale sur l'obstacle.

Chapitre 4

Apprentissage

L'objectif de cette section est de reconstruire la densité en connaissant l'énergie E , le flux F , et la température sur les bords du domaine aux cours du temps. Dans la suite, nous ferons une simplification majeure : la densité est supposee un signal en creneau (la forme du creneau vu du haut pouvant etre non connue). Aisni, reconstruire la densité revient juste a predire la position et la hauteur du creneau. La valeur de la densité en dehors du creneau sera aussi supposee connue. Nous recherchons une fonction f^{-1} invese de f (fonction definissant le probleme direct) telle que $y = f^{-1}(X)$. OU y represente la densité (plus precisement les attribut de son aut de densité), et X la les signaux sur les bords. Mais le caractere naturellement mal pose des problem inverse rend difficile la determination de f^{-1} . On procede donc a une approximation de f^{-1} a l'aide d'un reseau de neurones artificiel (ANN) notee \hat{f}^{-1} . En notant θ les parametres de l'ANN, on cherche \hat{y} telle que

$$\hat{y} = \hat{f}^{-1}(X, \theta).$$

4.1 Description des entrees/sorties

4.1.1 En 1D

Les entree sont onpossee des signaux etmporels E , F , et T . A chaque fois, il faut normaliser avant de les nourrir au reseau de neurones. Comme mentionne plus hautm nous avons faitr quelques simplifications sur la nature des sorties. Il s'agit uniquement d'un vecteur de scalaires representants l'abscisse et de la hauteur du saut de densité.

De facon visuelle, une entree et une sortie on l'aspect represente a la figure ??.

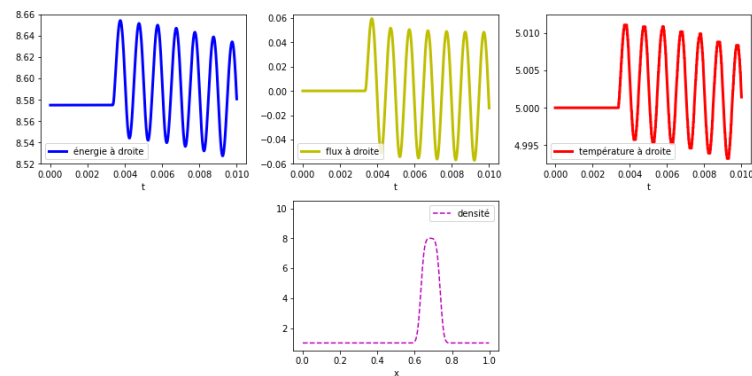


FIGURE 4.1 – Visualisation d'une entree (en haut) et d'une sortie (en bas) en 1D. Seul le signal sur la droite est utilise. Les 3 canaux E , F et T sont representes ici.

En 1D, les entrees ne sont constituees que du signal recuperer sur le bord droit du domaine. La forme d'une exemple en presentee a la figure ??.

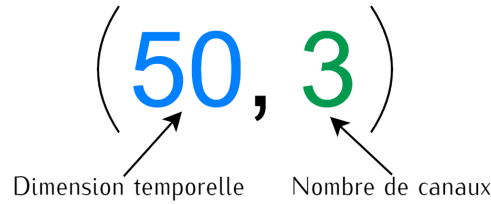


FIGURE 4.2 – Forme d'une entree en 1D. La dimension spatiale a ete reechantillonne de 907 a 50 iterations. Les 3 canaux designent les signaux E, F et T.

4.1.2 En 2D

Une entree 2D contient considerableme plus d'informations. Les 3 signaux E, F et T sur les 4 bords y sont inclus. On y inclu aussi un signal correspondant a l'une des quatre positions de la source dans chacun de ces cas. En effet, une entree correspond a 4 simulations effectuees chacune avec la source a une position differente comme on peut le voire a la figure . Comparer a la 1D, il faut rajouter l'ordonnee du saut de densite pour obtenir la densite.

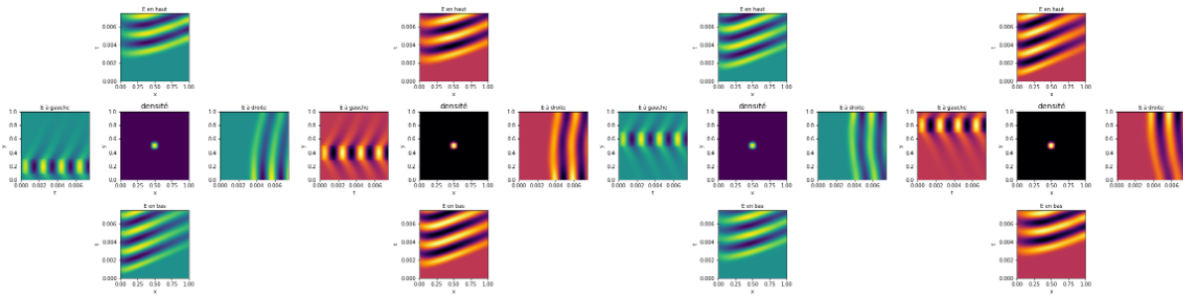


FIGURE 4.3 – Visualisation d'une entree (aux alentours des quatre images) et d'une sortie (aux mileiux) en 2D. On peut voir les postions des 4 sources utilisees a tour de role pour former une seule entree. Ici n'est representee que l'energie (qui constitue 1/3 des canaux) sur les 4 bords. Seuls les attributs de la densite (abscisse, ordonnee et hauteur) consituent la sortie

La forme d'une entree est representee a la figure ?. Des jeux de donnes complets 1D/2D ont ete sauvegardee et les details pour els recuperer et les traiter sont donnees en Annexe ?.

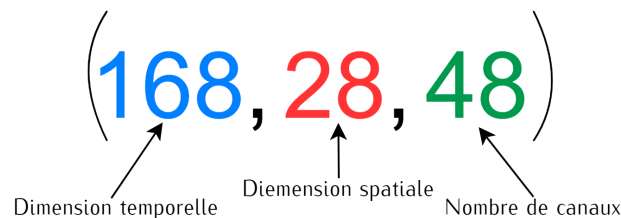


FIGURE 4.4 – Forme d'une entree en 2D. Contrairement a la 1D, il faut tenir compte de la dimention spatiale qui correspond au nombre de mailles sur chaque bords du domaine ($N=M=28$). Le nombre de canaux augmente donc considerableme. On passe a 48 car il faut tenir compte des 3 canaux originaux (E, F, et T), ensuite de chacun de 4 bords du domaine, et enfin des 4 positions de la source.

4.2 Architecture generale

Un reseau de neurones artificiel¹ est un systeme computationnel base sur le reseau de neurones biologique. L'apprentissage profond² permet de resoudre des problemes en Machine Learning³ que les methodes telles que la regression lineaire, etc.. ne peuvent pas. Il reussit cela en introduisant des representations des donnees qui s'expriment sous forme d'autres representations, plus simples cette fois. Les reseaux profonds en aval⁴ (ou MLP) constituent l'exemple typique en apprentissage profond. Il s'agit juste d'une fonction (composition de differentes fonctions) faisant correspondre une serie d'entree a une serie de sortie $f^{-1} = \text{composition de } f_1, f_2, \text{ etc.}$.

Un MLP est constitue de plusieurs couche (assimilables aux fonction f_1, f_2, \dots precedentes) apprenant chacune un aspect particulier des donnees (voir figure). On distingue une couche d'entree, une ou plusieurs couches cachees, et une couche de sortie ??).

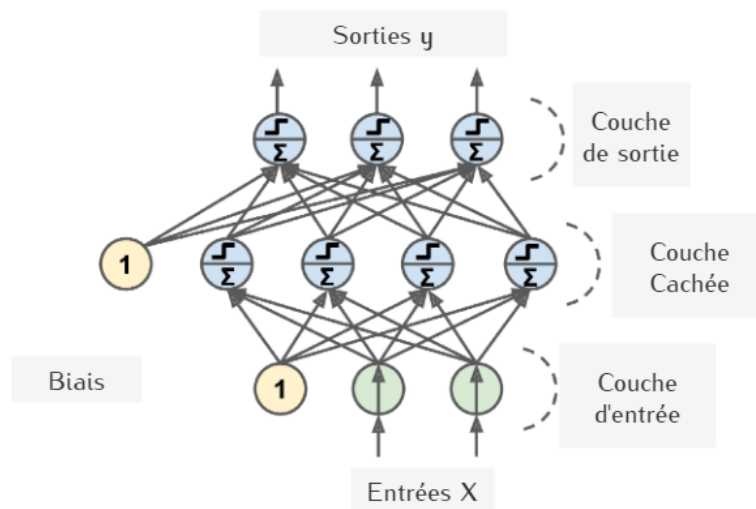


FIGURE 4.5 – Illustration d'un MLP avec une couche dense comme couche cachée. Le nombre de couche cachée peut être élevé ce qui conduit aux réseaux de neurones profonds. Le 1 représente le biais (**Reference8**).

Les réseaux de neurones convolutifs sont une forme de MLP spécialisée dans le traitement des données qui ont une forme de grille. Par exemple des séries en temps qui peuvent être vues comme des grilles 1D (l'axe de temps) prenant des données (vecteur de données) à interval de temps régulier (**Reference5**). Ils sont donc particulièrement adaptés à la reconstruction de la densité partant des signaux temporels E, F, et T. L'architecture de base a été proposée par M. Vigon. Nous utilisons deux variantes : DRNN⁵ 1 (figure ??) et DRNN 2 (figure ??). Les architectures seront implémentées sous la bibliothèque de machine learning Keras (avec Tensorflow backend). Les différentes couches présentes seront détaillées dans la suite. Nous indiquerons aussi en quoi elles sont importantes pour notre apprentissage.

1. nous y ferezons référence dans la suite juste par réseau de neurones
2. définition du nombre de couche
3. définition
4. en opposition à un réseau de neurones récurrent qui réutilise les résultats des modèles pour s'améliorer
5. Density Reconstruction Neural Network
6. Density Reconstruction Neural Network
7. les détails concernant le pooling seront donnés plus tard

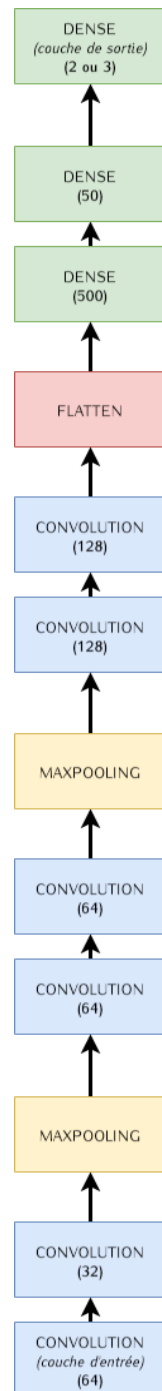


FIGURE 4.6 – Première architecture (nommée DRNN 1⁶). Le nombre de neurones utilisés pour chaque couches est indiqués entre parenthèses. Le nombre de neurones de la couche de sortie dépend qu'on soit en 1D ou en 2D.

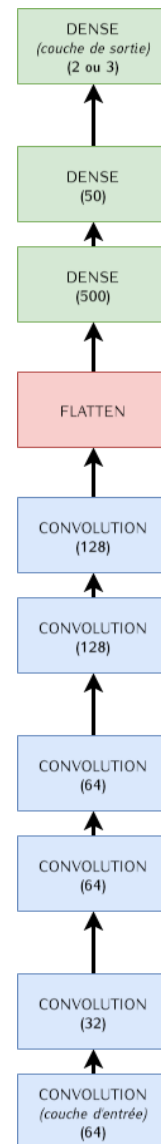


FIGURE 4.7 – Deuxième architecture utilisée (nommée DRNN 2). Ce modèle ne contient pas de couches de Pooling⁷

4.3 Les couches utilisées

4.3.1 Les couches de convolution

la convolution est l'opération fondamentale d'un CNN. Il s'agit d'une opération linéaire qui combine deux signaux pour en extraire un troisième. En général, une opération de convolution se définit par la formule suivante (i est le signal d'entrée et k est le noyau de la convolution)

$$s(t) = (i * k)(t) = \int i(x)k(t - x) dx$$

En pratique, les signaux temporels ne sont pas continus, ils sont discrétisés par interval de temps Δt . Dans ce contexte, la convolution 1D se définit par la formule :

$$s(t) = \sum_{x=-\infty}^{\infty} i(x)k(t - x) \quad (4.1)$$

Cette formule doit aussi être adaptée en 2D vu que nos inputs sont 2D. La formule devient donc :

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (4.2)$$

L'opération de convolution est commutative grâce à l'inversion du noyau relativement au signal d'entrée. Cette propriété, bien qu'importante d'un point de vue théorique, ne présente pas d'avantages majeurs du point de vue computationnel. C'est la raison pour laquelle on dispose de l'opération de cross-correlation qui est convolution sans inversion du noyau. En 2D elle se présente comme ceci :

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (4.3)$$

On remarque aussi que le parcours des indices se fait suivant l'input. Il se trouve que c'est plus direct et rapide ainsi, parcequ'il y a moins de variation dans la plage de valeurs valides pour n et m .

Plusieurs bibliothèques de machine learning implémentent la cross-correlation mais l'appellent convolution. C'est le cas de Keras lorsqu'elle utilise le backend Tensorflow⁸ (**Reference6**).

Dans les architectures de CNN typiques, la couche de convolution est généralement suivie d'une étape dite de détection. Dans cette étape, les résultats linéaires de la convolution sont passés à une fonction non linéaire au niveau d'une couche de d'activation. Nous détaillerons les détails de l'activation dans les sections suivantes. Après cette étape de détection, le Pooling est généralement appliqué pour modifier les résultats encore plus profondément.

4.3.2 Le MaxPooling

L'opération de Pooling permet de réduire la taille des données (downsampling). Une fonction de pooling transforme les entrées voisines par une fonction d'aggrégation statistique. Plusieurs fonctions d'aggrégations peuvent être utilisées. Par exemples, le maxpooling renvoie le maximum parmi les entrées sur un domaine (rectangulaire en 1D et rectangulaire en 2D).

En général, l'opération de pooling permet de rendre la représentation approximativement invariante aux petites variations dans l'input. Parlant de l'identification d'objets dans une image par

8. Lorsqu'on utilise Theano, les convolutions sont effectivement des convolution comme définie en ?? et ??

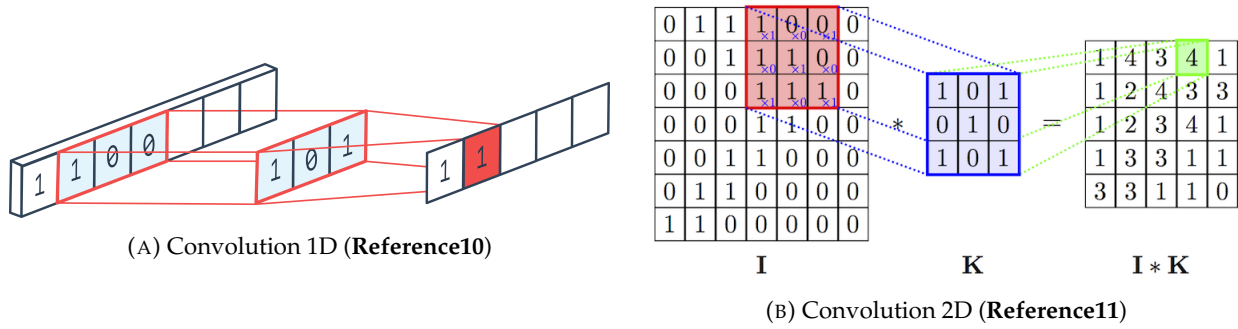


FIGURE 4.8 – Illustration d’une convolution (cros-corelation) 1D/2D en mode "valide" (aucun padding de 0 ne sera ajoute et la sortie S aura une taille inferieure a l’entree I). La taille du noyaux que nous utiliserons sera de 3 (1D) et (6,2) (2D) . Un autre parametre important pour reduire la taille de la sortie est le "stride", il s’agit de l’ecart entre deux applications du noyaux de convolution K. Nous le prenons egale a 1 (1D) et (1,1) (2D) de facon a couvrir tous les indices valides de l’entree I.

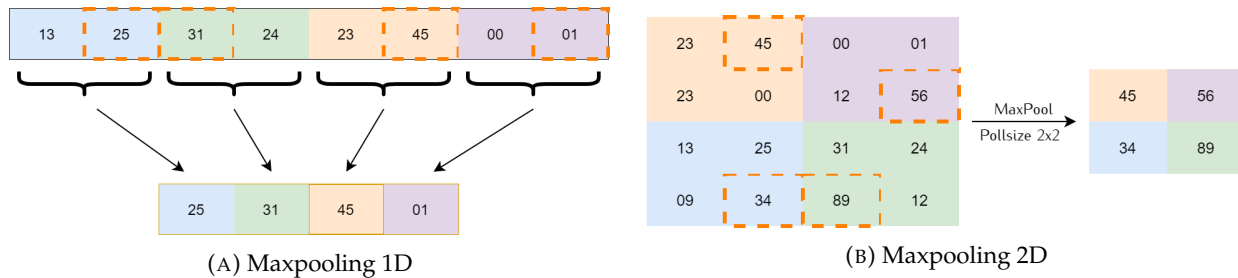


FIGURE 4.9 – Operation de maxpooling en 1D/2D avec un "pool size" de 2 (en 1D) et de 2x2 (en 2D)

exemple, l’invariance par translations locales (petites translations) peut etre utile si on est plus interesse par la presence de l’object que par sa localisation exacte(**Reference5**).

Dans le probleme inverse que nous resolvons, on est aimerai non seulement detecter la presence du saut de densite, mais aussi ses coordonnees exactes. Cela nous amenera donc a considerer dans un premier temps une architecture avec pooling (figure ??), et dans un dexieme temps, sans Pooling (figure ??).

4.3.3 Flatten

L’operation d’aplatissage permet de transformer les donnees en quittant de la forme tensorielle (2D avec plusieurs canaux) a une forme vectorielle. Il s’agit en realite d’une etape de preparation a une couche complement connectee.

4.3.4 Les couches denses

Dans cette couche, tous les neurones sont connectes a tous les neurones de la couche precedente. Une couche dense prend les resultats d’une convolution/pooling et en ressort des points. Les couches de convolution ayant appris des aspects particuliers des donnees, la couche est un moyen facile d’apprendre des combinaisons non lineaires de ces dernieres.

Si f_2 designe la fonction representant une couche dense. l’operation effectuee est la suivante : $f_2 = \phi(XW + b)$, ou X represente les entrees de la couche, b le bias , W la matrice des points (une

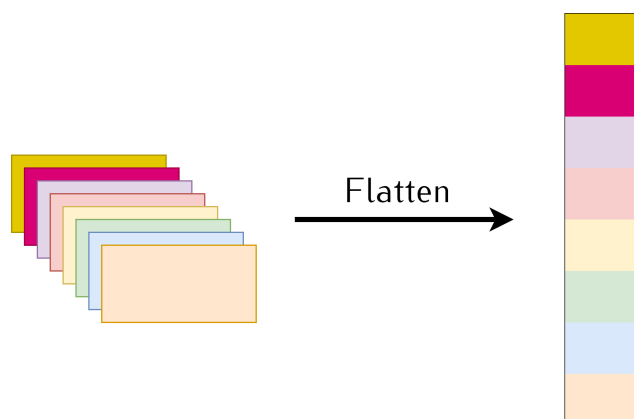


FIGURE 4.10 – Illustration d'une operation de flatten). Qu'on soit en 1D ou en 2D, le flatten assure que la sortie est un vecteur sur lequel une couche Dense peut operer.

ligne par neurone d'entree et une colonne par neurone de cette couche, excepte ceux du biais), et ϕ designe la fonction d'activation que nous detaillerons plus tard. (**Reference8**)

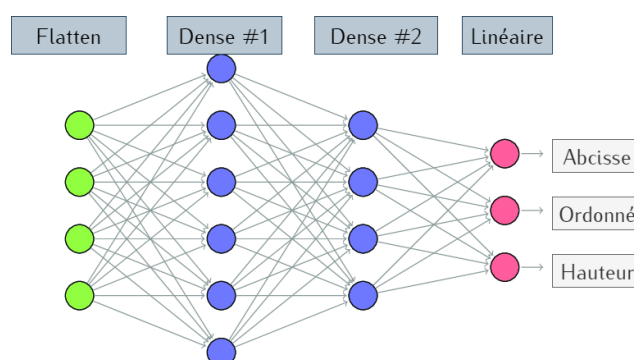


FIGURE 4.11 – Serie de 3 couches denses "fully connected". La couche (en vert) represent le resultat de l'operation de Flatten. L'utilisation des couches denses constitue la derniere etape des reseaux ?? et ??(Image adaptee de (**Reference9**))

4.4 Configurations de l'entrainement sous Keras

Keras proposent une multitude d'options et d'hyperparametres pour tuner le modele. Les plus importants sont detaillés dans les sections suivantes.

4.4.1 Les hyper-parametres

4.4.1.1 L'optimiseur

L'optimization est une methode d'accélération de l'entrainement. L'optimiseur Adam⁹ combine les proprietés de deux autres algorithmes d'entrainement (AdaGrad et RMSProp).

9. Adaptive moment estimation

4.4.1.2 Activation RELU

La fonction d'activation introduit une non-linearité entre les couches. L'avantage majeur de l'activation ReLU¹⁰ par rapport aux autres fonctions d'activation c'est qu'il n'active pas tous les neurones en même temps. D'un point de vue computationnel, elle est très efficace tout en produisant des résultats satisfaisants.

4.4.1.3 le taux d'apprentissage

Il s'agit du paramètre le plus influant pour notre apprentissage. Il contrôle à quelle vitesse le modèle¹¹ s'adapte au problème en déterminant de quelle quantité les poids des neurones seront mis à jour après l'algorithme de backpropagation. S'il est très élevé, il risquerait de conduire à une solution non optimale; s'il est très faible, le modèle peut rester figé (il faudra alors un nombre élevé d'époques pour le débloquent).

Avec un taux d'apprentissage égal à $1e-4$, nous n'avons été capable que de détecter la hauteur du créneau en 1D; et une réduction supplémentaire entraîne la divergence du modèle. En 2D, il a fallu descendre jusqu'à $1e-5$ pour déterminer avec précision l'abscisse, l'ordonnée, et la hauteur du créneau.

4.4.1.4 le batch size

Il s'agit de la taille de chaque paquet de données¹² passées au modèle durant une époque. Un batch size faible apporte du bruit au modèle vu qu'une partie aléatoire des données est utilisée pour mettre à jour les poids des neurones. Ceci permet une meilleure généralisation du modèle tout en permettant de limiter la quantité de données chargées dans la RAM à chaque époque.

4.4.1.5 Early stopping

La technique d'early sera notre moyen primaire de lutte contre le sur-apprentissage. Pour l'implémenter de façon efficace sous Keras, il nous faut une patience. Il s'agit du nombre d'époques à attendre avant d'arrêter l'apprentissage de façon précoce. Nous arrêterons nos entraînements dès que le score R2 (voir paragraphe ??) sur le jeu de validation n'aura pas augmenté pendant 10 époques.

4.4.2 Les métriques

4.4.2.1 Loss MSE

Pendant la génération des données, on a pris soin de ne pas introduire de données aberrantes. La MSE qui est plus élevée sur les valeurs aberrantes que la MAE est donc plus adaptée ici. Si les \hat{y}_i désignent les prédictions et y_i les véritables cibles (valeurs observées), la MSE se définit par :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.4)$$

10. Rectified Linear Unit

11. les poids des neurones sont initialisés de façon aléatoire

12. nombre d'instances d'entraînement sélectionnées aléatoirement

4.4.2.2 Coefficient de determination R2

Le Coefficient de determination R2 est tres important en statistique. On peut l'obtenir par la formule :

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (4.5)$$

Avec

$$SS_{res} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad \text{et} \quad SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$$

Ou $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ represente la moyenne des valeurs observees.

On peut remarquer que :

- Si le modele predit les valeurs attendues (observees), le score R2 vaut 1.
- SI le modele predit toujours la valeur moyenne \bar{y} , le score R2 vaut 0
- Si les predictions sont pires que la moyenne, le score R2 est negatif

En general On voit que si les predictions et les valeurs observees sont tres correles (sans etre egaux), on aura un score R2 ce qui n'est pas carateritique des resultats. En effet, pour des taches de regression il se definit comme etant le caree du coefficient de correlation entre les valeurs predites et les valeurs observees.

Dans la suite de ce rapport, le score R2 sera presente sous forme de pourcentage.

4.4.2.3 Un score personnalise

On definit donc un nouveau score particulieremtn adapte a nos donnees. On decalre qu'une prediction est correcte si elle est suffisamment proche du label :

- au dixième près pour la position (suivant x ou y) car le domaine de d'étude est $[0, 1] \times [0, 1]$
- à l'unité près pour la hauteur car les hauteurs observees sont comprises entre 0 et 10

Le score personnalise est un score severe (pourcentage de prediction correctes) qui qui recompense les prediction qui sont a la fois precisent en hauteurs et en position. La prediction de la position etant le probleme majeure auquel nous avons fait face, il cause generalement des valeurs faibles pour le score personnalise.

4.5 Resultats

Nous resumons la sections precedentes en specifiant les paramtres (et leurs noms) utilises pour entrainer le modele sous Keras.

Nous avons entraine les architectures en 1D (fig ...) et par la suite en 2D (fig ...) en ajustant les dimensions des couches de neurones connablement.

4.5.1 Régression

4.5.1.1 En 1D

On obtient de tres bonnes predictions sur la hateur de l'obstacle qui affecte directement l'amplitude des signaux sur le bord droit du domaine. Ce score n'est pas assez indicatif vu que les predictions sur la position du crenaux ne sont pas assez precises. Le score personnalise permet de capturer ce defaut et donne environ 25%.

Le modele avec MaxPoling correspond a la figure ??, et celui sans MaxPooling a la figure ??.

TABLE 4.1 – Liste des parametres majeurs utilises pour l'entrainement. L'activation relu est utilisee sur les couches cachees et "linear" sur la couche de sortie

Parametre	Definition	Valeur 1D / 2D
learning_rate	taux d'apprentissage	1e-4 / 1e-5
batch_size	taille d'un batch a chaque epoque	32
optimizer	algorithme d'optimisation	Adam
activation	type de fonction d'activation	"relu" ou "linear"
patience	patience pour l'early_stopping	10
epochs	nombre d'epoques	100
kernel_size	taille du noyau de convolution	3 / (6,2)

TABLE 4.2 – Resultats obtenus sur le jeu de test en 1D. Ces valeurs representent une moyenne obtenue sur plusieurs cycles d'entrainement/prediction

Score	Avec MaxPooling	Sans MaxPooling
R2	99.49 %	99.50 %
personalise	26.50 %	28.21 %

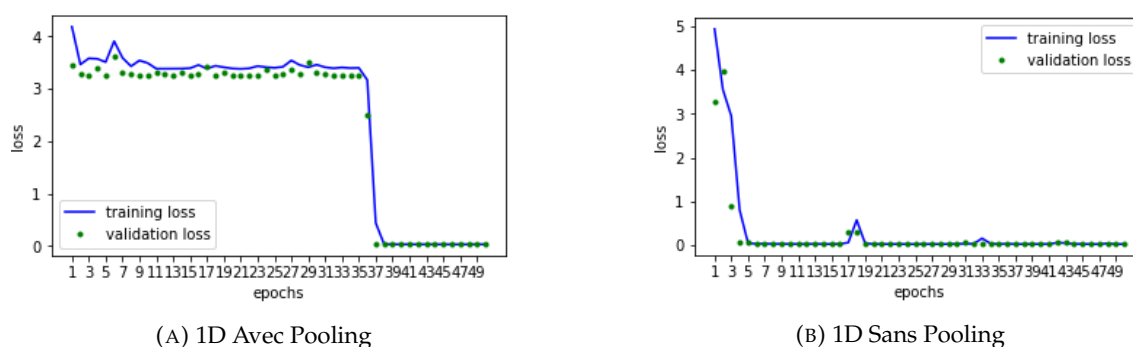


FIGURE 4.12 – Comparaison de la vitesse de décroissance de la loss en 1D

Observons les meilleures predictions du modele ((saans null doute du au hazard)

(AFFICHER DES MEILLEURES PREDICTIONS 1D)

Ci-dessous sont quelques unes des pires predictions du modele.

(AFFICHER LES PIRES PREDICTIONS 1D)

Le score personnelise autour de 25% indique que les prediction de la position se font quasiment aleatoirement. Pour remedier a ce probleme, il faut passer en 2D. En effet, le probleme inverse est naturellement mal defini dans le sens ou plusieurs entree peuvent donner la meme sortie. En 1D, on ne peut mesurer la sortie que sur un seul bord du domaine, ce qui limite beaucoup notre apprentissage.

4.5.1.2 En 2D

Comme attendu, le reseau est capable de detecter non seulement la hauteur de l'obstacle, mais aussi son abscisse et son ordonnee. Notre score personnelise nous permet de confirmer cela dans le

tableau ??.

TABLE 4.3 – Resultats obtenus sur le jeu de test en 2D

Score	Avec MaxPooling	Sans MaxPooling
R2	94.80 %	98.81 %
personalise	55.75 %	93.50 %

On constate que les modele sans l'operation de MaxPooling est globalement meilleure que son homologue avec MaxPooling du a l'application de l'early stopping. La figure ?? permet d'observer cela a travers la vitesse de convergence du modele.

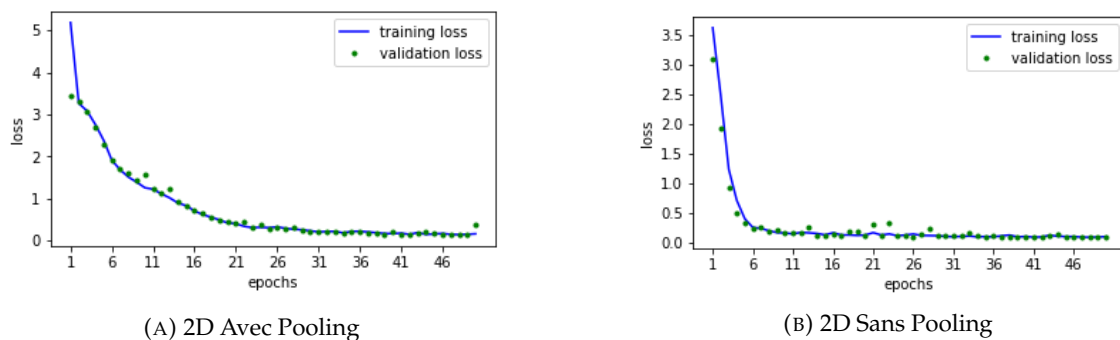


FIGURE 4.13 – Comparaison de la vitesse de decroissance de la loss en 2D

Pour observer les pires et meilleures predictions du modle, nous utilison naturellement le mo-
dele sans MaxPooling (voir figure).

(AFFICHER LES MEilleure PREDICTIONS 2D)

(AFFICHER LES PIRES PREDICTIONS 2D)

En ce qui concerne la generalisation du modle a d'autres formes d'obstacles, je n'ai pas eu le temps de comparer les modeles avec et sans MaxPooling. Le modele sans couches de MaxPooling risque d'etre moins performant conformement a la theorie (voir paragraphe ??). Sous Keras, le modele a prouver etre capable d'apprendre en continu, du moment que les entrees soient toutes normalisee et ayant la meme forme.

4.5.2 Classification

Durant le stage, il a fallu effectuer une classification multilabel sur les donnees en 2D. Qui permet de placer l'obstacle dans une categorie definie a partir de la source. La classification permet de detecter juste l'ordonne de l'obstacle. La structure des entrees est quasiment la meme que pour la regression, sauf qu'il manque les signaux sur la gauche (ou se trouve la source). En ce qui concerne les sorties, l'image ci-dessous decrit mieux leur structure :

Les donnees utilisees pour la classification ont une shape differente des autres. On a moins d'iterations en temps (40 au lieu de 168) mais une maille beaucoup plus fine (90x90 au lieu de 28x28). L'architecture du modele utilise ressemble a celle de la regression. Une majeure difference est qu'on utilise une activation "sigmoid" a la place de l'activation "lineaire". On obtient donc en sortie des probabilites qu'il faut classer par categories. Le modele est entraine avec des hyper-parametres identiques a ceux utilisees durant la regression. Les resultats sont presentes a la table ??

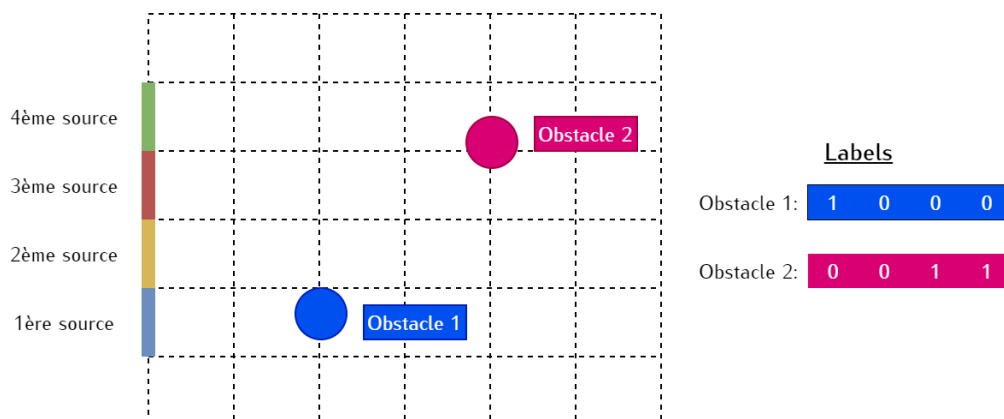
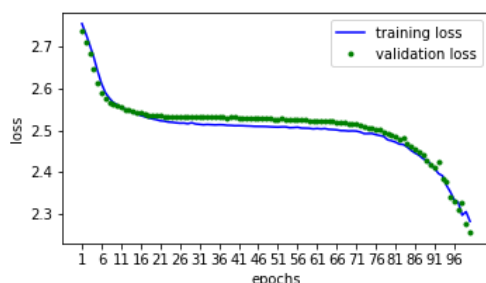


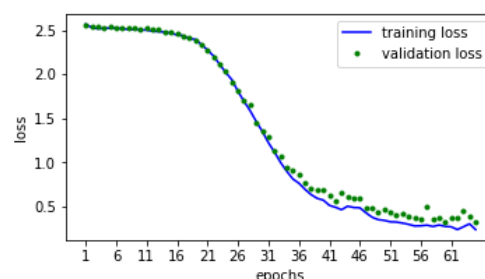
FIGURE 4.14 – Description des labels pour la classification multilabel en 2D. Un label est marqué 1 si l’obstacle se trouve dans le champ de la source correspondante

TABLE 4.4 – Resultats obtenus pour la classification en 2D. Le score severe favorise les predictions qui sont exactes au veritable sur tous les 4 colones . Le seuillage permet d’augmenter la precision des resultats en se fixant un nouveau seuil a partir duquel les interpreter les sorties du reseau de neurones. Le score de Binary Accuracy procure par keras est calcule avant seuillage.

Score	Avec MaxPooling	Sans MaxPooling
Binary accuracy	75.00 %	98.86 %
Score severe apres seuillage	27.27 %	95.45 %



(A) 2D Avec Pooling



(B) 2D Sans Pooling

FIGURE 4.15 – Comparaison de la vitesse de décroissance de la loss (multilabel cross-entropy) pour la classification en 2D

Dans ce cas aussi, le modele sans Maxpooling seble meilleure. Cependant nous ne disposons pas d’axxez de donnes pour verifier celui qui se generalise mieux. Ceci etant une classification, les meilleures prediction du meilleur modele sont exactement les labels attendus et les pires predictions ne varient que de peu (de 1) de leurs cibles. Il est interressnat de constater qu’il n’y a aucune prediction aberante, par exemple : un obstacle se trouve en face des sources 1 et 3 sans recontrer la source 2.

Chapitre 5

Bilan du stage

5.1 Ressources utilisées

Les ressources utilisées durant le stage varient en nature et en fonction.

5.1.1 Ecriture du Code

- VSCode : Pour l'édition du (principalement C++) grâce à ses fonction. L'extension majeure ici est CMake.
- Google Colab : Facilitation de l'apprentissage sous Keras grâce à ses GPU. Les librairies majeures ici sont Numpy, pandas, et Keras.
- Jupyter : Pour les tâches en Python ne nécessitant pas trop de ressource (visualisation, sauvegarde en format PQT). Les librairies majeures utilisées ici sont Numpy, Pandas et Matplotlib.
- Kile : Pour l'écriture du rapport en Latex
- Draw.io : Pour les illustrations

5.1.2 Communication

Les communications se sont effectuées principalement par messagerie électronique. J'ai aussi eu l'occasion de communiquer avec les professeurs en présentiel à 3 reprises.

5.2 Journal de bord

5.2.1 Semaine 1 et 2

- 15 juin : Réunion de début de Stage par Google Meet
- 16 juin : Demande aux professeurs de vérifier un exemple de simulation 1D, avant de me lancer la génération des données
- 17 juin : Remarque du problème d'apparition du crenau sur l'énergie
- 18 juin : Rédaction d'un nouveau schéma par M. Franck (pour l'étape 1) qui devrait conserver l'équilibre
- 22 juin : Détection de la source du problème du crenau sur E, et redefinition des termes.
- 23 juin : Confirmation de l'exactitude des simulations 1D et début de la génération des données avec 500 mailles.
- 25 juin : Demande d'aide à M. Vigon pour la configuration de la fonction d'activation de la couche de sortie

5.2.2 Semaine 3 et 4

- 3 juillet : Rencontre avec M. Navoret pour discuter des avancements. Prise de connaissance de d'une des raisons potentielles du probleme de mauvaise prediction de la position du crenau sur la densite en 1D. Proposition de plusieurs solutions par M. Navoret, entre autre de partir d'un signal stationnaire sinusoïdal et d'introduire l'onde a un temps $t^* > 0$.
- 6 juillet : Nouvelles simulations effectuees en vue d'observer la difference entre les effets de deux densites differentes. Continuation vers des nouvelles simulations avec 300 mailles.
- 8 juillet : decroissance du taux d'apprentissage a la suggestion de M. Franck mais non amelioration des resultats d'apprentissage.
- 9 juillet : Passage aux reseaux convolutif grace a M. Vigon
- 11 juillet : Plot du debut des oscillations, des maximum, des minimum a la demande de M. Vigon, afin de mieux observer les effets de deux creneaux de densite differents.

5.2.3 Semaine 5 et 6

- 13 juillet : Rencontre avec M. Navoret et M. Franck a la fac. Devant la persistance du probleme de non detection de la position du crenau, l'implementation du probleme en 2D semble etre la solution appropriee.
- 14 juillet reformulation 2D du schema de splitting et adaptation du code 1D en 2D
- 19 juillet : fin du codage 2D et presentation des resultats
- 25 juillet : ajustement de la gamme de couleurs pour les visualisations et passage a la generation des donnees sur 90x90 mailles.

5.2.4 Semaine 7 et 8

- 5 aout : Rencontre avec M. Franck a la fac. Proposition de solutions pour la non detection de la position du crenau en 2D par resolution d'un systeme proche de l'eq de la chaleur, apres affichage par ligne de niveau. La possibilite d'adopter un obstacle s'etendant sur toute la verticale est envisagee. Prise de connaissance des delais pour la redaction du rapport.
- 6 aout : Redaction et envoi du plan du rapport de stage.
- 7 aout : Proposition de reduction drastique de resolution spatiale par M. Vigon, et proposition de nouvelles idees par M. Vigon, entre autre la consideration d'un obstacle considerable plus opaque.
- 8 aout : Nouvel apprentissage avec des simplifications majeures qui fonctionnent. Melioration des resultats et continuation du rapport.

5.3 Difficultés rencontrées et solutions apportées

5.3.1 Apparition d'un crenau sur E

Au tout debut du stage, un crenau se formait puis se propageait sur l'energie E , le flux F et la temperature T . Grace a mes encadrants, ce probleme a ete resolu par rajout d'un terme au niveau de la deuxieme equation du schema de splitting.

5.3.2 Detection de la position du crenau

La detection de la position du saut de densite a ete un probleme majeur durant le stage. A la fin du stage, aucune solution (si elle existe) n'a ete trouvee pour le probleme inverse en 1D. Cependant en 2D le probleme a ete resolu essentiellement par augmentation du nombre d'epoques et diminution du taux d'apprentissage a $1e-5$. Il est bien connu que les problemes de machine peuvent diverger si le taux d'apprentissage est trop eleve. Quand au nombre d'epoques, je n'en faisais pas

suffisamment pour voir le modèle converger. Une solution bien plus rapide aurait été d'automatiser la recherche des hyper-paramètres, chose que je n'ai apprise qu'à la fin du stage.

5.3.3 Gestion du temps

La gestion du temps durant le stage n'a pas été facile. Au moment d'implémenter le schéma en 2D (ce qui n'était pas initialement prévu), j'ai longement hésité sur l'option la plus rapide. J'ai pu compter sur les conseils de M. Navoret pour surmonter cet obstacle.

Aussi, je me suis rendu compte des délais bien en retard, j'ai dû me débrouiller pour améliorer les résultats et terminer l'apprentissage. Cela dit, je n'ai pas réussi à faire une partie essentielle qui consiste à vérifier comment un modèle avec MaxPooling se généralise mieux qu'un modèle sans.

5.4 Les apports du stage

Ce stage a été enrichissant pour moi sur plusieurs fronts :

5.4.1 Expérience en développement

J'ai gagné de l'expérience en développement C++ et Python, tout en me développant un portfolio. J'ai beaucoup appris sur l'API de Pandas, Matplotlib, et plus important encore, celle de Keras. J'ai à présent une large base de données de code réutilisable pour d'autres tâches.

5.4.2 Équations aux dérivées partielles

J'ai pu observer directement quelques astuces utilisées par mes maîtres de stages pour vérifier la validité de la modélisation d'une EDP. Pour l'équation du transfert radiatif, j'ai compris la nécessité de partir d'un état d'équilibre radiatif.

5.4.3 Réseau de neurones

Ce stage m'a permis de percevoir la puissance des réseaux de neurones. J'ai appris à quel point le taux d'apprentissage est important. Comme mentionné dans le livre de référence Deep learning *The learning rate is perhaps the most important hyperparameter. If you have time to tune only one hyperparameter, tune the learning rate* (Reference5).

J'en ressort aussi avec quelques questions concernant le batch size. Lors de l'apprentissage, il a fallu entraîner le modèle en utilisant la méthode d'augmentation du batch size pour obtenir les premiers "bons" résultats. Cette méthode référencée ici (LIEN RETROUVABLE DANS LES MAIL) montre que beaucoup de questions restent à résoudre dans le domaine du deep learning.

5.4.4 Expérience de recherche

En tant que première expérience dans un environnement de recherche tel que l'UFR, j'ai pu me familiariser avec le milieu. J'ai notamment appris que les résultats ne doivent pas toujours être ceux auxquels on s'attend, du moment que l'on a une explication de l'échec.

Chapitre 6

Conclusion

Pour conclure, j'ai effectué mon stage de master 1 en tant que stagiaire en calcul scientifique/data scientist au sein de l'UFR de mathématiques et d'informatiques de l'Unistra. Lors de ce stage de 2 mois, j'ai pu mettre en pratique mes connaissances en développement C++ et Python acquises durant ma formation en calcul scientifique et mathématiques de l'information. Je me suis confronté au problème inverse de reconstruction de la densité d'un domaine par un CNN après avoir modéliser la propagation du signal dans ce dernier en 2D.

Ce stage fut très enrichissant pour moi car il m'a permis d'approfondir mon savoir théorique sur la backpropagation et la descente de gradient utilisés dans les réseaux de neurones malgré le fait que j'y ai passé relativement peu de temps. J'ai aussi gagné beaucoup d'expérience de développement logiciel et je me suis familiarisé avec l'environnement de Keras. Par contre je n'ai pas eu l'occasion de mieux comprendre la théorie des EDP, en particulier l'intuition derrière la définition des flux numériques dans les schémas. Ce stage m'a aussi permis de comprendre le déroulement d'une activité de recherche, et à quel point une bonne organisation et un certain degré d'autonomie sont importants. Ce stage a donc conforté mon projet de m'orienter vers un poste de ...

Cette expérience de stage fut centrée autour de la problématique de l'apport des réseaux de neurones dans la résolution des problèmes inverses, spécialement dans la détection des tumeurs¹. Les réseaux de neurones sont capables de détecter des sauts de densité bien variés, sous des conditions variées (opacités d'absorption différentes, maillages variés, etc.), tout ceci avec un coût de calcul relativement faible.

Fort de cette expérience et de ses nombreux enjeux, j'aimerais beaucoup par la suite, via un prochain stage, affiner les résultats à l'aide d'un apprentissage en continu en passant à la détection de plusieurs sauts de densité par exemple. Tout ceci pourrait conduire ultimement à la création d'un tomographe et un déploiement en milieu médical.

1. les tumeurs sont assimilables à des creux, des sauts de densité, ou obstacles

Annexe A

Comment reproduire les resultats ?

A.1 Execution du code 1D/2D

Les codes de calculs 1D et 2D se trouvent dans deux repositoy different :

- 1D : LIEN (Developpe pendant le projet mais utilise pour l'apprentissage durant ce stage)
- 2D : LIEN

Pour compiler le code dans les deux cas, il faut passer par CMake soit directemetn, soit par par un conteneur Docker :

```
rm -rf build
cmake -H. -Bbuild
cmake --build build
```

Ensuite il faut lancer l'exécutable avec un fichier de configuration (COMMANDE). La definition des paramtres 1D et 2D sont definis sur leurs repositories respectifs. Nous rappelons ceux pour la 2D ci-dessous.

(RAPPELLER L'ASPECT D'UN FICHIER CONFIG 2D)

A.2 Lecture du format binaire

Les donnes peuvent etre sauvegardees au format binaires a l'aide du parametre `export_type binary`. Lorsque c'est la cas, il faut se servir de la fonction ci-dessous en Python pour la lire.

(FONCTION BINARY)

A.3 Execution des notebook

Deux categories de notebooks ont ete crees dans le repository. Les deux premiers sont executables directement apres clonage du repository. Il s'agit de :

- Visualisation : Il permet de visualiser les resultats d'une simualtion exportee en CSV dans le fichier `data/df_simu.csv`
- Sauvegarde : Il permet de tranformer des donnes du foramt CSV au format binaire PARQUET rapidement lisible par Pandas.

Les autres notebook sont executables sur Google Colab. Il ne sont executables que si ont dispose des donnes d'entrainement, de test et de train. Il faut alternativement telecharger le modele deja entraine. Du a leur tailles condiderable, ces donnes ne sont pas diposbles sur le repository et doivent etre telechargee separement sur (LIEN POUR),

Annexe B

Comment faire des predictions avec ce modele ?

Pour faire des prediction, il suffit de disposer d'un donee ayant une forme bien particuliere decrite a la figure (). Ilf aut en suite charger le modele a l'aide de Keras et la compiler.

B.1 Normalisation des donnees

En plus d'avoir la forme .. , les donnees doivent etre normalisee. Y toute les energies doivent etre divisees par leur maximum (en valeur absolue). Il en est de meme pour le flux et la temperature.

B.2 Chargement du modele

Le modele a ete sauvegarde sous la convention SavedModel de Tensorflow. Apres l'avoir charger, il faut inclure imperativement la fonction de calcul du score R2 indiquee ci-dessous lors de la compilation.

La commande est la suivante :

B.3 Entrainer le modele en continu

Le modele peut etre entrainer en continue. Apres l'avoir charger, on peut l'apprendre a detecter d'autres formes d'obstacles sous differentes conditions. Il suffit de disposer de telles donnees. Pour l'apprentissage en continue, la (TRAIN ON BATCH) et peut etre interessante.