



HAL
open science

Développement de modèles utilisant les réseaux de neurones convolutionnels et d'une interface R-shiny pour la prédiction et la visualisation de la croissance de l'herbe d'une prairie, à partir de l'information climatique fournie

Kouamé Yannick Konan

► **To cite this version:**

Kouamé Yannick Konan. Développement de modèles utilisant les réseaux de neurones convolutionnels et d'une interface R-shiny pour la prédiction et la visualisation de la croissance de l'herbe d'une prairie, à partir de l'information climatique fournie. Intelligence artificielle [cs.AI]. 2022. hal-03782204

HAL Id: hal-03782204

<https://hal.inrae.fr/hal-03782204>

Submitted on 21 Sep 2022

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.



Distributed under a Creative Commons Attribution 4.0 International License

ÉCOLE NATIONALE DE LA STATISTIQUE
ET DE L'ANALYSE DE L'INFORMATION



STAGE STATISTIQUE 2A

Étudiant 2A
SYSLAIT

Développement de modèles utilisant les réseaux
de neurones convolutionnels et d'une interface
R-shiny pour la prédiction et la visualisation de
la croissance de l'herbe d'une prairie, à partir de
l'information climatique fournie

Étudiant :

Kouamé Yannick KONAN

Tuteurs :

M.Thomas GUYET

Mme. Anne-Isabelle GRAUX

Avril 2022

Rapport de stage 2A

21 septembre 2022

Table des matières

1	Environnement de stage	10
2	Présentation des données	10
3	Présentation de l'interface développée	11
3.1	Objectifs de l'interface, intérêts et fonctionnalités attendues	11
3.2	Aperçu de l'interface développée	13
3.2.1	Onglet d'informations générales	13
3.2.2	Onglet dédié à la prédiction et visualisation de la croissance de l'herbe	13
3.3	Guide d'utilisation	15
3.3.1	Les opérations obligatoires	15
3.3.2	Les opérations facultatives	15
3.4	Développement de l'interface	16
3.4.1	Présentation de R Shiny : l'outil de développement	16
3.4.2	Gestion des entrées de l'utilisateur	17
3.5	Limites de l'interface	21
4	Ajout d'une méthode de prédiction plus performante	22
4.1	Apprentissage automatique	22
4.1.1	Principes de l'apprentissage automatique	22
4.1.2	Descente du gradient	22
4.2	Les réseaux de neurones	23
4.2.1	L'intuition du neurone biologique : origine du neurone artificiel	23
4.2.2	Le perceptron ou le neurone artificiel : calque du neurone biologique	23
4.2.3	Des neurones montés en parallèle : réseau de neurones à une seule couche	24
4.2.4	Réseaux de neurones multicouches	25
4.3	Réseaux de neurones convolutionnels appliqués aux séries temporelles	26
4.3.1	Réseaux de neurones convolutionnels	26
4.3.2	Application à la prédiction de séries temporelles	28
4.4	Rappel des objectifs du modèle	29

4.5	Architecture du réseau	29
4.5.1	Présentation de l'architecture	29
4.5.2	Fonctionnement de la méthode	30
4.6	Expérimentations	30
4.6.1	Préparation de l'expérimentation	30
4.6.2	Résultats de l'expérimentation	31
4.6.3	Comparaison avec les autres méthodes précédemment développées (SVM, RF)	34
5	Conclusions et perspectives	35

Liste des figures

Figure 1	Vue du premier onglet de l'interface	13
Figure 2	Vue du deuxième onglet de l'interface	14
Figure 3	Code R d'une application shiny basique	16
Figure 4	Visualisation d'une application shiny basique	17
Figure 5	Diagramme d'activité de l'interface	18
Figure 6	Widget de sélection de la première variable	18
Figure 7	Vue du deuxième widget de sélection	19
Figure 8	Diagramme d'activité de la sélection des colonnes	19
Figure 9	Sélection de l'unité de la température	20
Figure 10	Sélection de la résolution temporelle	20
Figure 11	Sélection de l'année ou des années de prédiction	20
Figure 12	Sélection du modèle de prédiction	21
Figure 13	Coupe du neurone biologique	23
Figure 14	Liaison synaptique entre deux neurones	23
Figure 15	Neurone artificiel	24
Figure 16	Schéma d'une couche de quatre neurones artificiels montés en parallèle	24
Figure 17	Schéma d'un réseau de neurones multicouches :	25
Figure 18	Calcul sortie réseau de neurone multicouches	26
Figure 19	Architecture d'un réseau de neurones convolutifs	28
Figure 20	Convolution de deux images	28
Figure 21	Produit de convolution pour une série à une seule variable avec une taille de noyau de 3 et une longueur de vecteur égale à 7 et un seul vecteur de sortie	30
Figure 22	Opération de <i>padding</i> permettant de conserver la même longueur du vecteur de sortie que la longueur du vecteur d'entrée sur un vecteur de longueur 5	30
Figure 23	Epochs et nombre de batchs	31
Figure 24	Distribution de l'erreur quadratique moyenne (mse) en fonction du nombre de couches	32
Figure 25	Distribution de l'erreur quadratique moyenne (mse) en fonction du nombre de couches avec une taille de noyau égale à 9	33
Figure 26	Distribution de l'erreur quadratique moyenne (mse) en fonction de la taille du noyau pour un réseau à 2 couches	33

Figure 27	Boxplots des mse en fonction de la taille du noyau pour un réseau à 3 couches	33
Figure 28	Boxplots des mse en fonction de la taille du noyau pour un réseau à 4 couches	34

Liste des tableaux

Table 1	Description des différentes variables du jeu de données croissance_et_climat_decadaires	12
Table 2	Interprétation de l'indice de Martonne	12
Table 3	Tableau du temps moyen de prédiction de la croissance en fonction de chaque modèle .	22
Table 4	Série temporelle multidimensionnelle des variables climatiques de période décadaire pour l'année 2003 pour les cinq premières décades	29
Table 5	Tableau des erreurs quadratiques moyennes (mse) des modèles pour des nombres de filtres l de valeurs 4, 6 , 8 ou 10 et des tailles noyaux d valant 3 ou 5 pour les nombres de couches n du premier bloc de valeur 2 ou 3.	32

Résumé

Les prairies sont importantes car elles couvrent de larges surfaces et rendent de nombreux services écosystémiques. Prédire la croissance de l'herbe des prairies dans différents contextes climatiques est utile à la modélisation des systèmes d'élevage herbagers. Ce travail fait suite au stage d'un étudiant de l'ENSAI qui a utilisé des approches d'apprentissage automatique pour développer des modèles de prédiction de la croissance de l'herbe à partir de l'information sur le climat. Les objectifs de mon stage étaient d'offrir une visualisation graphique et chiffrée de la croissance de l'herbe prédite par ces modèles, et d'enrichir la partie modélisation déjà réalisée. Une interface a été développée sous Rshiny qui offre la visualisation et les fonctionnalités souhaitées.

Introduction

En 2020, les prairies s'étendaient sur 12,8 millions d'hectares en France soit à peu près 20% du territoire national. En Bretagne, les prairies constituent 674 000 hectares soit 25% du territoire régional. Par ailleurs, les prairies jouent un rôle primordial, notamment au travers des services écosystémiques qu'elles assurent tels que la production de fourrages, l'atténuation du changement climatique par le stockage de carbone et le maintien de la biodiversité.

La croissance de l'herbe désigne la production de biomasse par les espèces végétales composant la prairie pendant une certaine durée. Elle est habituellement exprimée en kg de matière sèche (MS) par hectare (ha) et par jour. Par accumulation, l'herbe qui croît produit une biomasse d'herbe qui peut être exploitée par les éleveurs pour nourrir les herbivores qu'ils élèvent. La croissance dépend des ressources en eau et en nutriments du sol, du climat local constitué du rayonnement global, de la température, des pluies qui alimentent la végétation et contribuent à recharger la réserve en eau du sol. La croissance de l'herbe dépend également de la gestion appliquée par l'éleveur en terme d'exploitation de l'herbe (fauche, pâturage), de fertilisation et des types de prairie (composition floristique, âge). Elle change peu d'une semaine ou décade (période de dix jours consécutifs) à l'autre, et est généralement mesurée sur le terrain à cette fréquence. La croissance de l'herbe à une décade donnée dépend aussi de la biomasse d'herbe présente à cette décade, via la surface d'interception de la lumière qu'elle offre pour réaliser la photosynthèse. La croissance de l'herbe à une décade donnée dépend donc aussi de la croissance de l'herbe des décades précédentes, qui a permis d'aboutir à la biomasse d'herbe présente à cette décade-là. La valorisation annuelle de l'herbe englobe l'herbe ingérée par les animaux au pâturage ainsi que l'herbe fauchée pour la production de fourrage (foin, ensilage).

L'an dernier, un étudiant de l'ENSAI (Laurent Spillemaecker) a effectué un stage à l'IRISA dont le but était la prédiction de la croissance de l'herbe à partir d'algorithmes d'apprentissage automatique en utilisant l'information, souvent connue, du climat passé et la valorisation annuelle de la prairie. Dix-huit modèles croisant trois types de régresseurs ou classifieurs (régression linéaire, machines à vecteurs de support et forêt aléatoire), deux types d'initialisation de la croissance des trois premières décades et trois types de post-traitement pour essayer de reconstruire les séries temporelles de la croissance brute ou transformée (dérivée, cumul) ont été développés et comparés. D'après les résultats obtenus, l'information sur le climat et la croissance des trois décades précédentes est suffisante pour une bonne prédiction de la croissance de l'herbe alors que l'information sur la valorisation annuelle apporte peu et peut-être exclue des prédicteurs. Aussi, l'utilisation de valeurs de croissance moyenne pour initialiser en hiver la croissance des trois premières décades n'introduit pas d'erreurs importantes. Parmi tous ces modèles, le plus performant est le modèle utilisant les Machines à Vecteur de Support, sans transformation de la croissance (série brute). Bien que performants, ces modèles présentent des limites dues à la technique d'auto-régression utilisée pour tenir compte de la dépendance temporelle des valeurs de croissance entre elles. Cette méthode nécessite de fixer les valeurs de la croissance des 3 premières décades de chaque série dans notre cas. Elle se traduit par une accumulation de l'erreur avec la décade et un temps de calculs important.

Mon stage s'inscrit dans la continuité du travail de Laurent avec deux objectifs principaux. Le premier est de développer une interface graphique permettant d'offrir une visualisation graphique et chiffrée de la croissance de l'herbe prédite par les modèles précédents, à partir du climat renseigné en entrée par l'utilisateur. Le deuxième est d'enrichir la partie modélisation déjà réalisée en développant des modèles basés sur des

réseaux de neurones convolutionnels, offrant un potentiel affranchissement des limites ci-dessus.

Pour la bonne réalisation de ce projet, le cahier des charges décrivant les objectifs et livrables attendus dans le cadre du développement de l'interface puis de la partie modélisation m'a été remis en début de stage. L'organisation de mon temps a été planifiée également via un diagramme de Gantt également remis en début de stage.

1 Environnement de stage

J'ai effectué mon stage à l'INRAE de Saint-Gilles (35590).

L'INRAE (Institut National de Recherche pour l'Agriculture, l'Alimentation et l'Environnement), né au premier janvier 2020 de la fusion entre l'INRA (Institut National de la Recherche Agronomique) et l'IRSTEA (Institut National de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture), est un organisme français de recherche en agronomie. Cet organisme est un Établissement Public à caractère Scientifique et Technologique (EPST) et il se trouve sous la double tutelle du ministère chargé de la Recherche, de l'Enseignement supérieur, de la Recherche et de l'Innovation, et du ministère chargé de l'Agriculture et de l'Alimentation. C'est le premier organisme de recherche spécialisé sur la vie, l'humain et la terre. L'INRAE mène des recherches dans le but de promouvoir une alimentation saine et de qualité, une agriculture durable, et un environnement préservé et valorisé. Il compte 18 centres de recherche à travers toute la France dont le centre Saint-Gilles au sein duquel j'ai effectué mon stage.

Ce stage a été réalisé pour le compte de l'unité mixte de recherche PEGASE (pour Physiologie, Environnement, et Génétique pour l'Animal et les Systèmes d'Elevage) et plus précisément de l'équipe Syslait (pour Systèmes laitiers). PEGASE est constituée de 54 scientifiques dont 10 enseignants-chercheurs de l'Institut Agro Rennes-Angers, 23 doctorants et post-doctorants. C'est plus de 165 personnes au quotidien. L'objectif scientifique principal de l'équipe Syslait est de produire des connaissances et de développer des innovations et des outils d'aide à la décision pour augmenter l'efficacité et la résilience des systèmes laitiers aux échelles de l'animal, du troupeau et du système tout en améliorant leur durabilité. Le défi pour cette équipe est donc de fournir de nouvelles approches et / ou des stratégies de gestion à court et long terme de ces systèmes et de proposer aux agriculteurs des solutions appropriées pour s'adapter aux changements (notamment climatiques) auxquels sont confrontées les exploitations laitières.

Mon sujet de stage s'inscrit dans la continuité d'un travail collaboratif de recherche entre Anne-Isabelle Graux (chercheuse à l'Inrae) et Thomas Guyet (chercheur chez Inria à Lyon depuis 2021) qui ont été mes encadrants de stage. Thomas Guyet effectue des recherches dans le domaine de l'analyse de séries temporelles et il a ainsi accompagné le développement de la partie modélisation du stage.

2 Présentation des données

La description des données effectuée est inspirée de celle réalisée par L. Spillemaecker, précédent stagiaire et ancien élève de l'Ensaï. L'UMR PEGASE de l'INRAE de Rennes a produit un jeu de données de simulation de taille conséquente sur la valorisation annuelle et la croissance journalière des prairies. Ces données ont été produites dans le cadre de l'étude « Les prairies françaises : Production, exportation d'azote et risques de lessivage » [4, 3]. Il s'agit de données simulées par le modèle STICS¹, qui est un modèle mécaniste et déterministe [1]. L'information utilisée en entrée du modèle sur le climat, le sol et les pratiques de gestion de l'herbe est connue. En particulier, l'information climatique (8 variables au pas de temps journalier) a été fournie par le système SAFRAN de Météo France. Différents types de prairies ont été considérés, correspondant à des prairies permanentes ou semées qui diffèrent par leur composition (graminées pures, en mélange

1. STICS : <https://www6.paca.inra.fr/stics/>

avec des légumineuses, ou légumineuses pures) et leur durée d'implantation. Les pratiques agricoles ont été résumées sous la forme de 30 modes d'exploitation définis par le type (fauche et/ou pâture) et le nombre d'exploitation de l'herbe dans l'année, ainsi que par la fertilisation azotée apportée à la prairie.

Les simulations ont été réalisées à l'échelle de la France à une haute résolution spatiale correspondant à des unités pédoclimatiques (UPC), issues du croisement de la résolution de l'information climatique (maille safran) et pédologique (unité cartographique de sol ou UCS), et pour lesquelles la surface de prairies est significative. Les sorties des simulations correspondent à des séries temporelles de 30 années (1984-2013) au pas de temps journalier.

A chaque UPC est associé un climat de 30 années (1984-2013), un à deux sols majoritaires, un à deux types de prairie majoritaires, et pour chacun des types de prairie, 1 à 18 modes d'exploitation. Au sein d'une même UPC, il peut donc y avoir plusieurs simulations (une dizaine en moyenne par UPC).

A l'instar du stage précédent, nous nous sommes limités aux données de la région Bretagne. Nous avons sélectionné uniquement les variables utiles au stage à savoir la croissance journalière des prairies ainsi que les données climatiques. Dans tous les jeux de données utilisés, la croissance correspond à des valeurs exprimées depuis la surface du sol. Seules les prairies de graminées pures ont été considérées dans le jeu de données breton.

Les données journalières de croissance et de climat ont été agrégées (moyenne ou cumul) à la décade, c'est-à-dire sur une période de dix jours consécutifs, période pendant laquelle la croissance varie peu et s'avère suffisante pour les besoins du stage. Chaque série annuelle comporte ainsi 37 valeurs.

Le jeu de données utilisé fut créé lors du précédent stage et regroupe l'information décadaire sur la croissance et le climat. Il est composé de près de 20 millions d'observations (477 439 séries de 37 décades = 17 665 243 d'observations) de 15 variables (voir Table 1). Ces données ont été utilisées dans le cadre de l'apprentissage des modèles développés.

L'indice de Martonne [2] est un indice d'aridité et l'intérêt de disposer de cet indice est qu'il peut être intégrateur de l'information sur la température et les pluies. La table 2 donne les interprétations des valeurs de l'indice.

La durée d'implantation d'une prairie étant fixe pendant une simulation, ces années d'installation reviennent régulièrement, tous les trois à cinq ans suivant la durée d'implantation de la prairie.

3 Présentation de l'interface développée

3.1 Objectifs de l'interface, intérêts et fonctionnalités attendues

Dans ce stage, il m'a été demandé de développer une interface capable de fournir une visualisation graphique et chiffrée de la prédiction de la croissance de l'herbe à partir de nouvelles données climatiques. Cette interface se destine à des chercheurs ou au conseil agricole. Elle doit être simple, intuitive, bien documentée, opérationnelle. Pour le moment, elle est uniquement en anglais.

Variable	Définition	Unité
<i>ucs</i>	numéro de l'unité cartographique de sol (ucs)	sans unité
<i>safran</i>	numéro de la maille safran qui correspond à la résolution climatique (maille carrée de 8x8 km)	sans unité
<i>sol</i>	numéro du sol	sans unité
<i>gestion</i>	numéro de la gestion de la prairie	sans unité
<i>ian</i>	année des observations	sans unité
<i>decade</i>	numéro de la décade (période de dix jours consécutifs)	sans unité
<i>Tmin</i>	moyenne des températures minimales journalières sur la décade	degré Celsius (°C)
<i>Tmax</i>	moyenne des températures maximales journalières sur la décade	degré Celsius (°C)
<i>Tmoy</i>	température moyenne (moyenne de Tmin et de Tmax)	degré Celsius (°C)
<i>Rain</i>	cumul des pluies sur la décade	mm
<i>RG</i>	cumul du rayonnement sur la décade	J/cm ²
<i>im</i>	indice de Martonne calculé à la décade et défini par : $I = 37 \times \frac{Rain}{Tmoy + 10}$	mm/C
<i>croissance</i>	croissance journalière nette moyenne sur la décade	kgMS/ha/jour, (MS signifiant matière sèche)

TABLE 1 – Description des différentes variables du jeu de données **croissance_et_climat_decadaires**

im	Classe d'aridité
> 35	très humide
]28; 35]	humide
]24; 28]	semi-humide
]20; 24]	méditerranéen
]10; 20]	semi-sec
[0; 10]	sec

TABLE 2 – Interprétation de l'indice de Martonne

Elle doit se présenter sous deux onglets. Le premier expliquant l'objectif de l'interface et sa méthode de construction ainsi que l'information climatique à fournir en entrée et l'information disponible en sortie. Le deuxième onglet sera le cœur de l'interface et contiendra toutes les fonctionnalités de celle-ci notamment l'importation du fichier de l'utilisateur, la sélection des données climatiques, la sélection des méthodes de prédiction, la visualisation graphique et enfin le téléchargement des graphiques et d'un tableau des données chiffrées. Ainsi, l'utilisateur pourra :

- Renseigner les données nécessaires à l'exécution des modèles, i.e. le climat d'une (ou plusieurs) année(s) d'intérêt
- Spécifier le pas de temps du climat (journalier ou déjà décadaire) et de recalculer les variables climatiques journalières sur la décade au besoin
- Gérer la correspondance entre les noms de variables climatiques fournies avec les noms attendus
- Sélectionner et comparer différents modèles de prédiction :
 - Forêt aléatoire (random Forest : RF)
 - Machine/régression à vecteur de support (Support Vector Machine : SVM)

- Réseau de neurones convolutionnels (Convolution Neural Network : CNN)
- Visualiser graphiquement la (ou les) courbe(s) de croissance de l’herbe prédite(s), suivant les modèles sélectionnés
- Télécharger les données et les graphiques de la croissance de l’herbe prédite, avec l’information sur les entrées et le modèle associé

3.2 Aperçu de l’interface développée

3.2.1 Onglet d’informations générales

Cet onglet (figure 1) résume toutes les informations générales de l’interface en quatre paragraphes correspondant à l’objectif de l’interface, les sources de données utilisées, l’information délivrée et utile pour faire tourner les modèles de prédiction de la croissance de l’herbe.

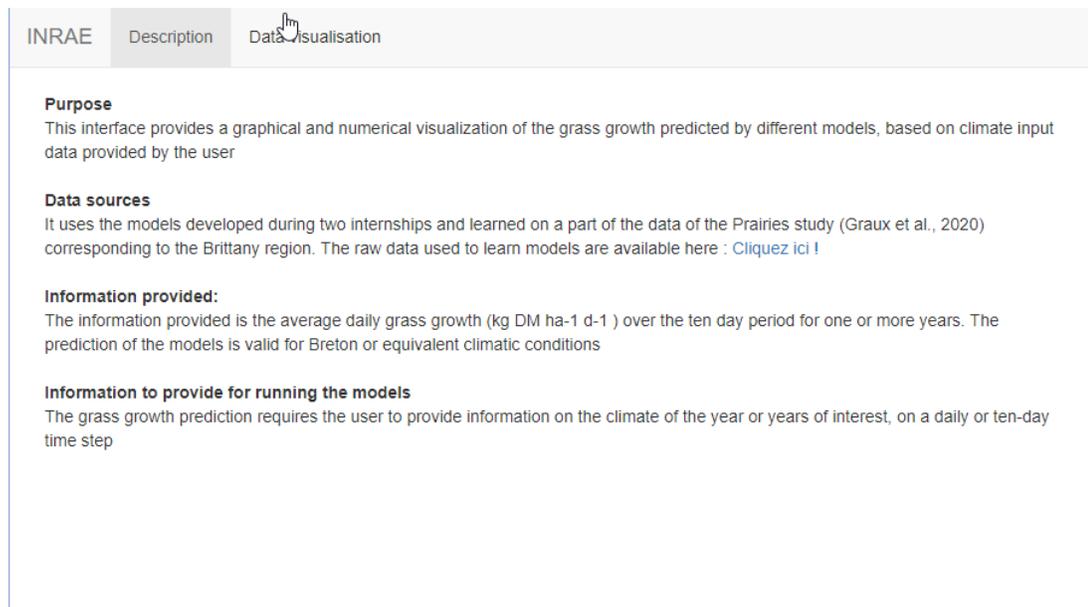


FIGURE 1 – Vue du premier onglet de l’interface

3.2.2 Onglet dédié à la prédiction et visualisation de la croissance de l’herbe

L’onglet est divisé en deux colonnes. Celle de gauche contient les objets de sélection appelés *widgets*. C’est la colonne des choix de l’utilisateur. On y trouve le widget d’importation des données, les widgets de sélection des variables, des années et des modèles de prédiction disponibles, ceux développés par L. Spillemaecker lors de son stage : forêt aléatoire (random forest : RF), machines à vecteurs de support (Support Vecteur Machine : SVM) et lors de mon stage : réseaux de neurones convolutifs (Convolution Neural Network : CNN). Enfin, des widgets de téléchargement des données et des graphiques générés. La colonne de droite contient deux panels : le premier sert à afficher les graphiques et le second sert à afficher les données associées comme illustré dans la figure 2.

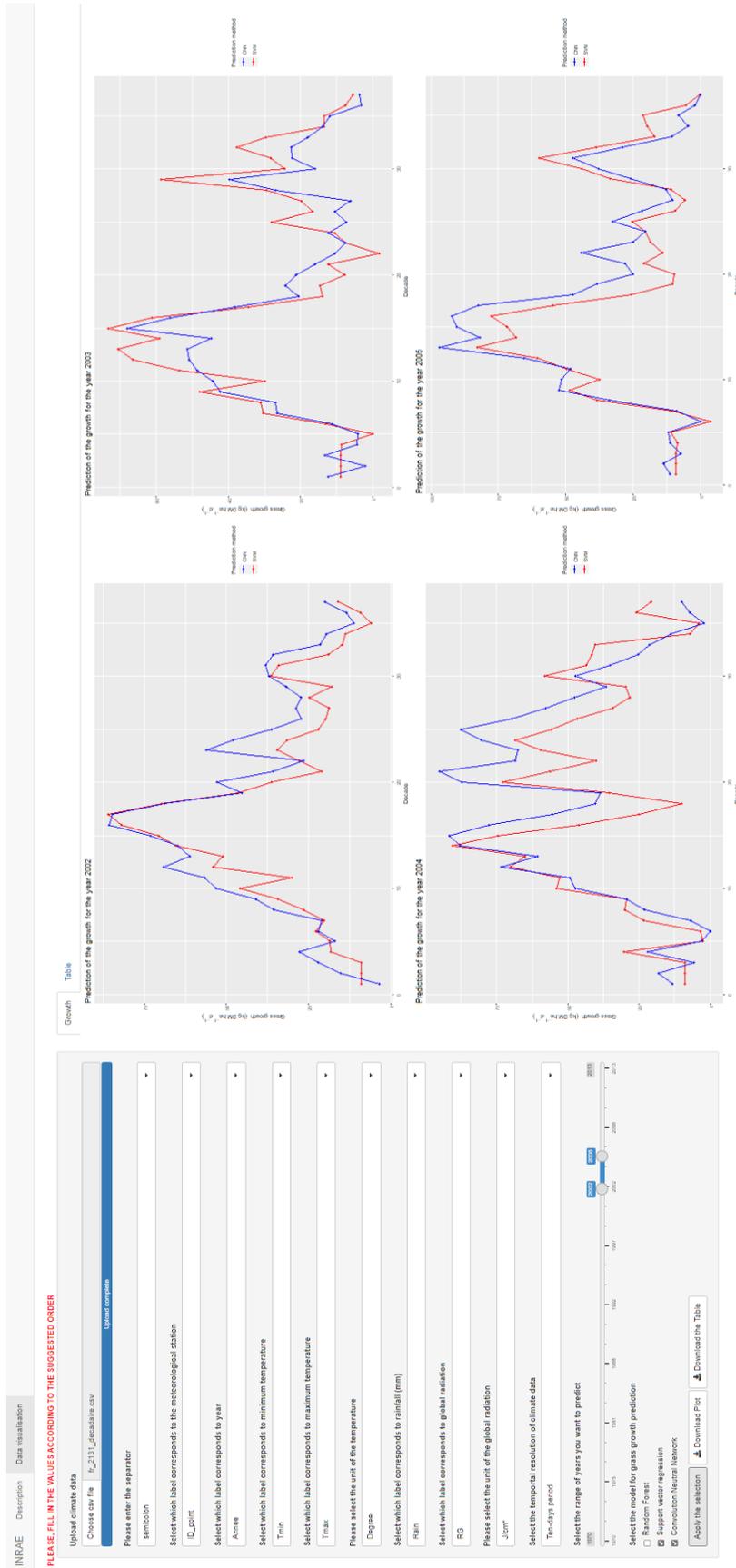


FIGURE 2 – Vue du deuxième onglet de l'interface

3.3 Guide d'utilisation

3.3.1 Les opérations obligatoires

Chaque utilisateur doit rentrer ses informations selon l'ordre des widgets de sélection qui sont disposés de haut en bas comme suit :

- **Le widget d'importation des données climatiques :** l'utilisateur importe ses nouvelles données climatiques sous la forme d'un tableau. Celui - ci doit contenir à minima les cinq variables climatiques suivantes : la température minimale, la température maximale, le rayonnement global, les pluies et la ou les années de prédiction. Le pas de temps peut être journalier ou décadaire.
- **Les widgets d'identification des variables de prédiction :** l'utilisateur doit faire la correspondance entre les libellés des variables climatiques de son jeu de données et les libellés utilisés dans les modèles de prédiction. Par exemple dans les modèles de prédiction, la variable température minimale a pour libellé "temperature_minimale_t".
- **Le widget de définition du pas de temps de l'information climatique :** l'utilisateur doit sélectionner le pas de temps de son jeu de données climatiques. Sachant que les modèles sont développés avec le pas de temps décadaire, lorsque l'information climatique est fournie au pas de temps journalier, il faut la recalculer au pas de temps décadaire.
- **Le widget de sélection des années de prédiction :** l'utilisateur peut choisir la ou les années utilisées pour la prédiction.
- **Le widget de sélection des modèles de prédiction :** l'utilisateur choisit les modèles qu'il souhaite utiliser parmi le Random Forest (RF) , les Support Vectors Regression (SVM) ou les Convolution Neural Network (CNN). On rappelle que les modèles développés dans ce stage s'appuient sur les réseaux de neurones.
- **le widget d'application de tous les choix effectués :** Il s'agit d'un bouton déclencheur de l'opération de prédiction. Il permet de commencer l'exécution du programme informatique.

3.3.2 Les opérations facultatives

Il s'agit des widgets de téléchargement des graphiques et du tableau de données chiffrées associées.

- Un widget permet de télécharger l'ensemble des graphiques de prédiction. Ils sont regroupés dans un fichier ZIP sous le format image JPEG. Chaque graphique représente la courbe de la croissance prédite selon le ou les modèles sélectionnés, pour une année.
- la table de sortie contient les données en entrée et en sortie des modèles, soit les variables climatiques et la croissance prédite.

3.4 Développement de l'interface

3.4.1 Présentation de R Shiny : l'outil de développement

R Shiny est une bibliothèque de R, qui permet la création de pages web interactives sur lesquelles il est possible de réaliser toutes les analyses ou actions disponibles sous R. Se distinguant parce qu'elle ne requiert aucune connaissance des langages HTML, CSS, Java-Script, elle donne la possibilité de développer une interface graphique répondant aux critères imposés.

Une application R Shiny est constituée de deux parties : une partie user (ui) et une partie server (server) comme présenté dans la figure 3.

```
# Exemple du développement d'application shiny

library(shiny)
library(datasets)

# importation d'une librairie
data("trees")

# Partie User
ui <- pagewithSidebar(
  titlePanel(" Ma première application R"),
  sidebarPanel(
    selectInput(inputId = "variable" ,
                label = "caractéristiques des arbres" ,
                choices = c("taille" = "Height" ,
                            "volume" = "volume" ,
                            "épaisseur" = "Girth") ,|
                selected = "Height")
  ),
  mainPanel(
    plotOutput("affichage")
  )
)

# Partie server
server <- function(input, output ){
  output$affichage <- renderPlot({hist(trees[, input$variable] ,
                                       main= "histogramme des variables des arbres" ,
                                       xlab = input$variable)})
  output$nom_variable <- renderTable(trees$input$variable)
}

shinyApp(ui = ui, server = server)
```

FIGURE 3 – Code R d'une application shiny basique

- La partie user, qui communique avec l'utilisateur, est le lieu de création des "inputs" qui sont les valeurs d'entrée du programme. C'est à travers les widgets que ces inputs sont générés. Par exemple, le widget de sélection d'entiers peut servir à choisir l'année de prédiction.
- La partie server, qui est la partie codage de l'interface, est le lieu où sont écrites toutes les fonctions du programme. Elle se charge du traitement des inputs et ressort des "outputs" qu'elle affichera par la suite. Comme illustration, si l'input 2016 est choisie comme année de prédiction, l'output est le tableau de sortie.

Considérons cette application R Shiny qui affiche l'histogramme d'une variable descriptive d'un arbre. L'input est le nom de la variable et l'output, l'histogramme. Dans la figure 4, la variable choisie est *volume*.

Ma première application R

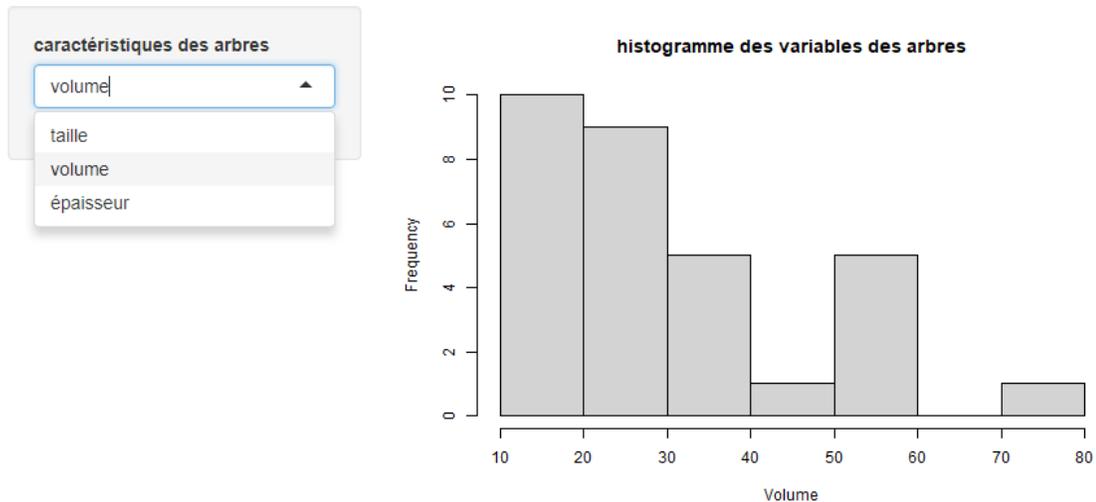


FIGURE 4 – Visualisation d’une application shiny basique

3.4.2 Gestion des entrées de l’utilisateur

Afin d’effectuer des prédictions pour les données climatiques de l’utilisateur, il est nécessaire que celles-ci correspondent aux noms et à l’unité des variables climatiques utilisées par les modèles de prédiction. Dans cette partie, nous expliquons le choix des widgets pour gérer les éventuels problèmes de compatibilité. D’abord, l’utilisateur importe sa table à travers le widget d’importation. Nous l’avons contraint d’accepter le format CSV. Tout autre format de données est automatiquement rejeté.

- **La correspondance entre les libellés du modèle et les libellés des colonnes de l’utilisateur**

Les modèles statistiques développés utilisent des libellés précis pour les variables de prédiction : `temperature_min` pour la température minimale, `indice_martonne` pour l’inde de Martonne etc ... Donc, la prédiction pour de nouvelles données climatiques requiert qu’elles aient les mêmes libellés. Nous avons élaboré une méthode pour faire cette correspondance.

D’abord, nous recueillons tous les noms des variables du fichier importé qui seront ensuite proposés à l’utilisateur afin qu’il sélectionne le libellé correspondant à chaque variable climatique. La figure 5 présente le diagramme d’activité UML.

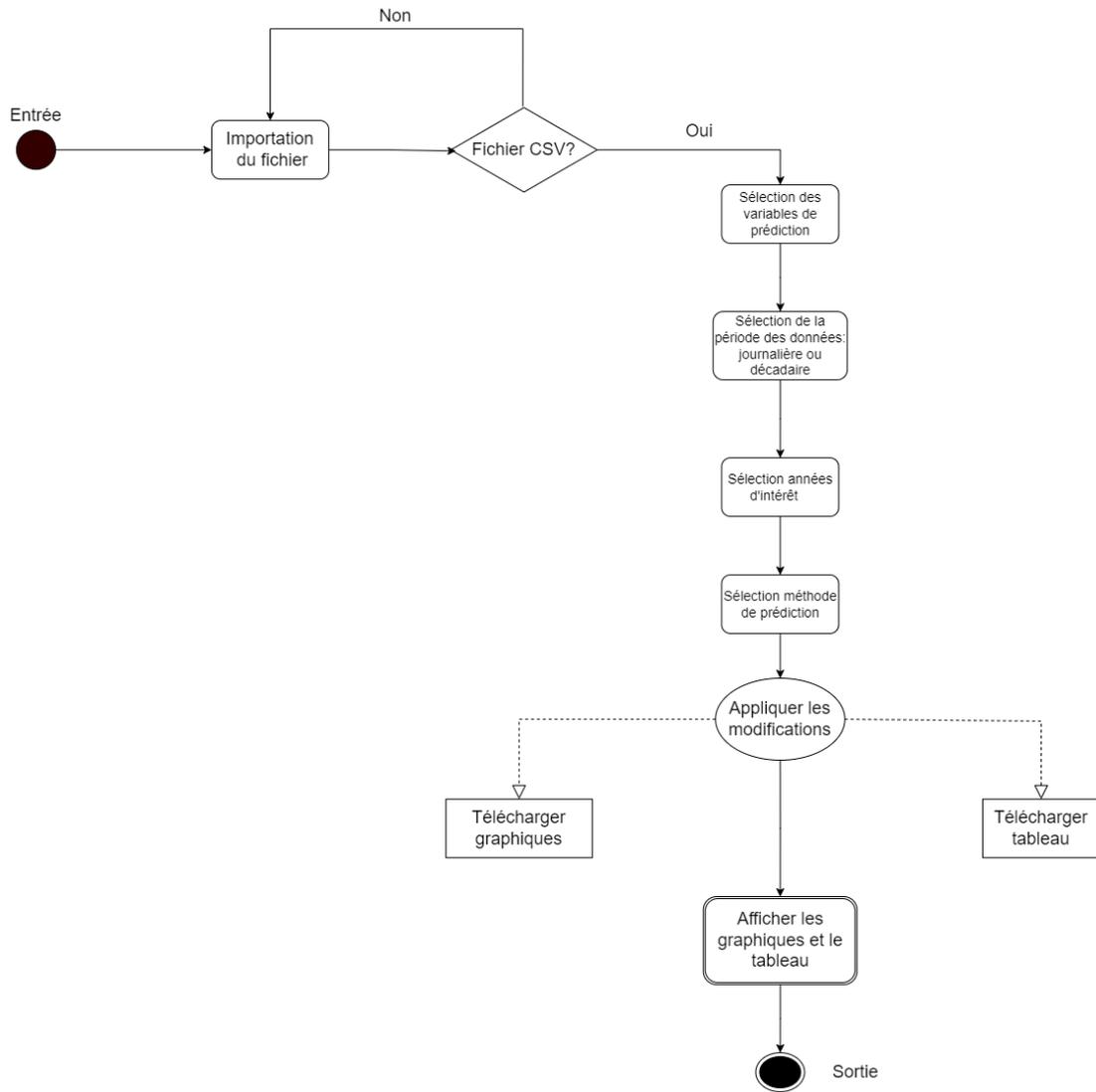


FIGURE 5 – Diagramme d'activité de l'interface

Le premier widget s'intitule **"Select which label corresponds to the meteorological station"** (figure 6) et propose de choisir entre tous les libellés importés celui correspondant à la station météorologique.

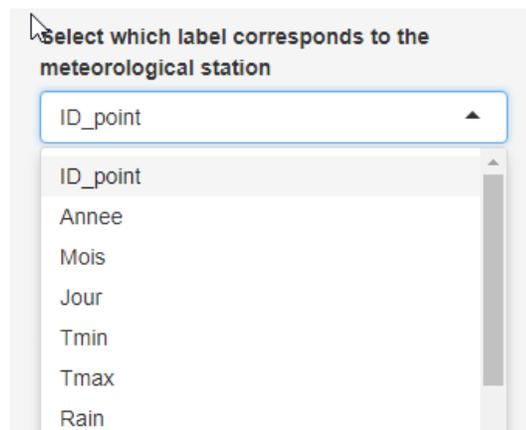


FIGURE 6 – Widget de sélection de la première variable

Le choix étant effectué, on arrive au widget suivant (figure 7) qui ne propose pas le libellé précédemment sélectionné. C'est l'idée la plus importante de la méthode : chaque libellé choisi dans un widget est automatiquement retiré dans la liste des choix du widget suivant. Ainsi le widget "Select which label corresponds to year" a la proposition de choix dans l'exemple utilisé pour l'illustration :

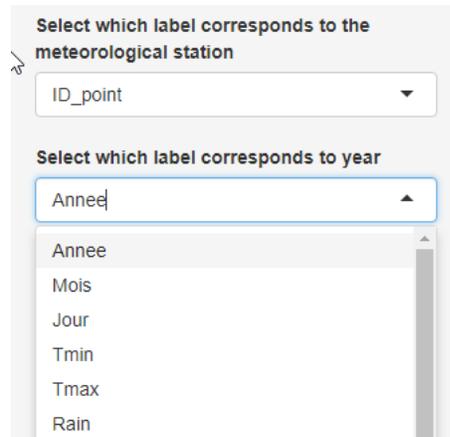


FIGURE 7 – Vue du deuxième widget de sélection

La même opération est effectuée pour toutes les entrées dans l'ordre, suivant (figure 8).

- ▷ station météorologique
- ▷ année
- ▷ température minimale
- ▷ température maximale
- ▷ précipitations
- ▷ rayonnement

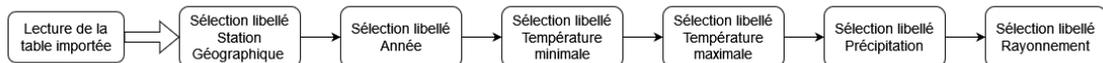


FIGURE 8 – Diagramme d'activité de la sélection des colonnes

• Correspondance entre les unités des variables

Pour une variable climatique, les unités utilisées dans les modèles peuvent être différentes de celles des données de l'utilisateur. Ainsi, pour la température et le rayonnement, on propose à l'utilisateur de préciser leurs unités parmi des choix conventionnels. Si celles-ci diffèrent des unités par défaut, nous effectuons la correspondance. Par exemple, la figure 9 montre les choix disponibles pour l'unité de la température : le degré, le Fahrenheit et le Kelvin.

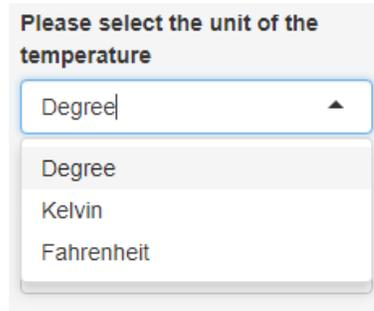


FIGURE 9 – Sélection de l'unité de la température

- **Période dans les données de l'utilisateur**

Sachant que les modèles ont été entraînés sur des données décennales, il faut convertir, si nécessaire, la nouvelle donnée journalière dans ce format de temps. Un widget de sélection présenté à la figure 10 permet à l'utilisateur de préciser le pas de temps. Si c'est journalier, nous effectuons la correspondance avec les opérations suivantes :

- ▷ Les observations journalières sont groupées par 10 pour former une décade
- ▷ La variable température minimale sera la moyenne des températures minimales
- ▷ La variable température maximale sera la moyenne des températures maximales
- ▷ La variable précipitation sera la somme des précipitations
- ▷ La variable rayonnement sera la somme des rayonnements

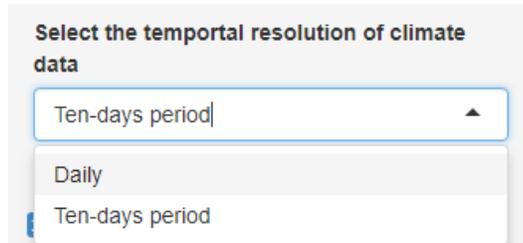


FIGURE 10 – Sélection de la résolution temporelle

- **Sélection des années d'intérêt**

L'utilisateur a la possibilité de sélectionner les années qui l'intéressent pour la prédiction. Même si l'idéal serait de sélectionner des années non consécutives de façon indépendante, nous avons fait le choix que l'utilisateur ne puisse choisir qu'un intervalle d'années consécutives en précisant la première année et la dernière année. Le widget de sélection se présente comme suit :

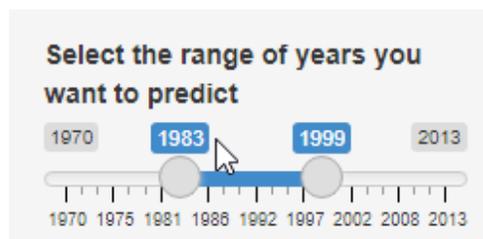


FIGURE 11 – Sélection de l'année ou des années de prédiction

L'utilisateur peut faire glisser les bornes de gauche et de droite avec la souris. La portion en bleu représentant l'intervalle de prédiction, pour choisir une seule année d'intérêt, il doit superposer ces deux bornes.

- **La sélection des modèles de prédiction**

Trois modèles de prédiction sont proposés :

- les forêts aléatoires (Random Forest en anglais, noté RF)
- les machines à vecteurs de support (Support Vecteur Machine, noté SVM)
- les réseaux de neurones convolutifs (Convolution Neural Network, noté CNN)

L'utilisateur peut en choisir un ou plusieurs. Par défaut, la méthode sélectionnée est le CNN car le temps d'exécution est le plus court. Le widget de sélection est présenté à la figure 12.

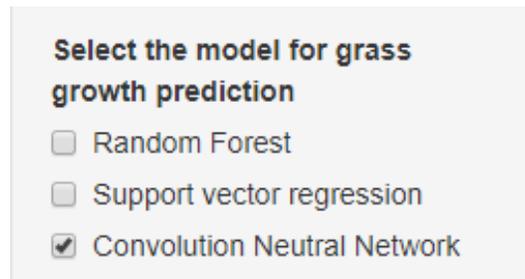


FIGURE 12 – Sélection du modèle de prédiction

L'utilisateur a la possibilité de télécharger les graphiques de prédiction ainsi qu'un tableau récapitulant toutes les prédictions effectuées.

- **Téléchargement des graphiques de croissance**

Un clique-bouton nommé "download table" situé en bas à gauche de l'onglet permet à l'utilisateur de télécharger, au format ZIP, un fichier contenant les graphiques de prédiction.

A chaque année d'intérêt correspond un graphe qui présente la croissance prédite selon chaque modèle distingué chacun par couleur. Les décades sont en abscisse et la croissance, en ordonnée.

- **Téléchargement du tableau récapitulatif**

L'utilisateur clique sur le bouton "download table" afin de télécharger un tableau, de format CSV, qui résume dans un fichier Excel, tous les résultats des prédictions effectuées à savoir les variables climatiques et les prédictions de chaque méthode sélectionnée.

5

3.5 Limites de l'interface

Une interface bien développée se caractérise par une interaction fluide avec l'utilisateur. En d'autres termes, un bref délai de réponse entre les deux acteurs. Dans notre cas, malheureusement, le Random Forest qui prédit la croissance décade après décade est impossible à paralléliser. Il prend donc beaucoup de temps pour prédire la croissance de l'herbe sur une année. On évalue à 5 secondes son temps moyen pour prédire la croissance d'une décade et donc à 3 minutes pour celle d'une année. L'ordinateur utilisé pour les calculs avait 8 Go de mémoire vive avec un processeur Intel Core i7.

Modèle de prédiction \ Période de prédiction	Décade	Année
RF	5 secondes	3 minutes
SVM	0,13 secondes	5 secondes
CNN	0,02 secondes	1 seconde

TABLE 3 – Tableau du temps moyen de prédiction de la croissance en fonction de chaque modèle

4 Ajout d'une méthode de prédiction plus performante

4.1 Apprentissage automatique

4.1.1 Principes de l'apprentissage automatique

L'apprentissage automatique est une branche de l'intelligence artificielle fondée sur des techniques mathématiques et statistiques donnant aux ordinateurs la capacité d'« apprendre » à partir de données. Ces techniques amélioreront les performances des machines à résoudre des tâches pour lesquelles elles ne sont pas spécialement programmées. Plus largement, l'apprentissage automatique inclut la conception, l'analyse, l'optimisation, le développement et l'implémentation de telles méthodes.

4.1.2 Descente du gradient

La descente du gradient, désigne un algorithme d'optimisation différentiable qui sert à minimiser une fonction réelle d'un espace Hilbertien ou euclidien. L'algorithme est itératif et s'améliore à chaque itération en effectuant un déplacement dans la direction opposée au gradient afin de décroître la fonction. Ainsi, il permet de retrouver les points stationnaires de la fonction, points de gradients nuls et donc des minimums globaux si la fonction est convexe.

Soient E un espace préhilbertien (produit scalaire $\langle \cdot, \cdot \rangle$ et norme associée notée $\| \cdot \|$) noté \mathbb{E} et de norme associée et $x \in \mathbb{E}, \mapsto f(x) \in \mathbb{R}$ une fonction différentiable. On note $df(x)$ la différentielle de f en x et $\nabla f(x)$ le gradient de f en x , si bien que pour tout direction $d \in \mathbb{E}$, $df(x)(d) = \langle \nabla f(x), d \rangle$

On se donne un point initial $x_0 \in \mathbb{E}$ et un seuil de tolérance $\varepsilon \geq 0$. L'algorithme du gradient définit une suite d'itérés $x_1, x_2, \dots \in \mathbb{E}$, jusqu'à ce qu'un test d'arrêt soit satisfait. Il passe de x_k à x_{k+1} par les étapes suivantes.

- Simulation : calcul de $\nabla f(x_k)$.
- Test d'arrêt : si $\|\nabla f(x_k)\| \leq \varepsilon$, arrêt.
- Calcul du pas $\alpha_k > 0$ par une règle de recherche linéaire sur f en x_k le long de la direction $-\nabla f(x_k)$
- Nouvel itéré : $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$.

ε est un nombre positif

4.2 Les réseaux de neurones

Un réseau de neurone artificiel est un système dont la conception est schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques. Généralement optimisé par des méthodes d'apprentissage de type probabiliste, en particulier bayésien, un réseau de neurones permet de créer des classifications ou d'effectuer une régression.

4.2.1 L'intuition du neurone biologique : origine du neurone artificiel

Le neurone biologique est constitué de trois parties principales :

- Le corps cellulaire, composé du noyau, il est le centre de contrôle de l'information qui est reçue par les dendrites
- Les dendrites sont les fils conducteurs par lesquels passe l'information qui vient de l'extérieur
- L'axone est le fil par lequel passe l'information de sortie traitée par le corps cellulaire

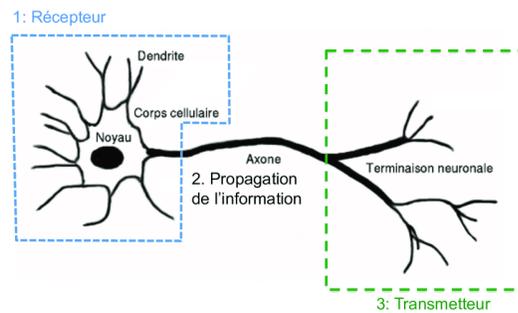


FIGURE 13 – Coupe du neurone biologique

Les synapses qui se trouvent dans la terminaison neuronale permettent de lier les neurones. Elles permettent donc de communiquer entre eux, et se chargent de la pondération de ces liaisons. Donc, dans un neurone biologique, le corps cellulaire fait la somme des informations recueillies par les dendrites, la traite et la transmet à travers l'axone aux autres neurones avec lesquels il est lié par les synapses.

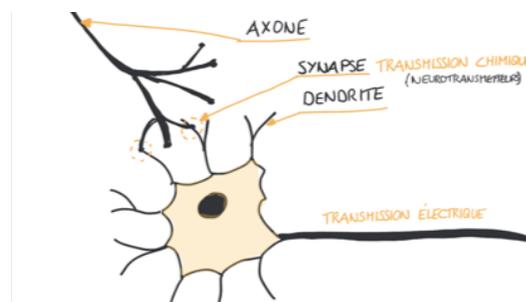


FIGURE 14 – Liaison synaptique entre deux neurones

4.2.2 Le perceptron ou le neurone artificiel : calque du neurone biologique

Le perceptron reprend les principes du neurone biologique. Par analogie, on a :

- les entrées sont des vecteurs notés x et qui représentent les informations reçues par les dendrites.

- Une sortie y qui représente l'information véhiculée par l'axone.
- Des paramètres notés w et b définissant le fonctionnement du neurone jouant le rôle du corps cellulaire comme illustré dans la figure 15

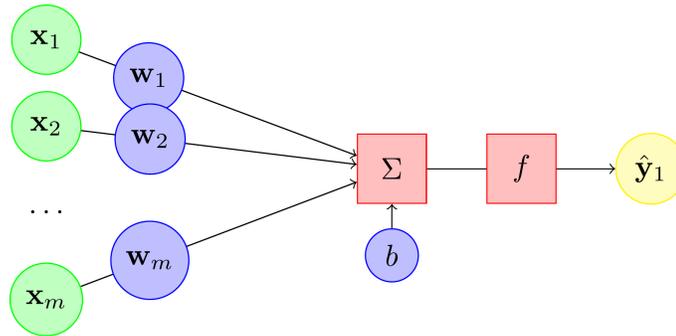


FIGURE 15 – Neurone artificiel

L'équation d'un réseau est :

$$\hat{y} = f(\langle W, x \rangle + b)$$

Toutes les entrées sont multipliées par un poids W . Les résultats sont alors sommés et additionnés à un biais b qui caractérise ce neurone. On y applique ensuite une fonction non linéaire appelée *fonction d'activation* qui produit la sortie souhaitée. Lorsque la fonction d'activation est l'identité, un neurone permet uniquement de faire de la régression linéaire et lorsque la fonction d'activation est de type sigmoïde, un neurone permet d'effectuer de la séparation linéaire. Apprendre un réseau de neurones, consiste donc à trouver les valeurs de W et b qui minimisent l'erreur de prédiction entre y et \hat{y} .

4.2.3 Des neurones montés en parallèle : réseau de neurones à une seule couche

Souvent, un seul neurone ne suffit pas pour résoudre des problèmes plus complexes. Ainsi, on regroupe plusieurs neurones en parallèle sous une forme de couche comme illustré dans la figure 16 suivant

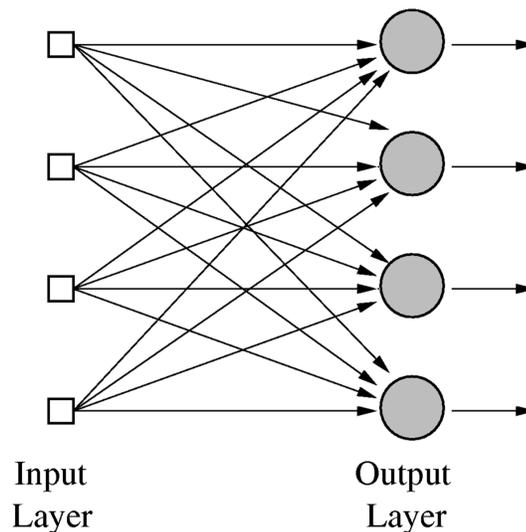


FIGURE 16 – Schéma d'une couche de quatre neurones artificiels montés en parallèle

Ce schéma présente quatre neurones formant une couche. Les entrées se trouvent dans la couche *input layer* et les neurones dans la couche *Output Layer*.

4.2.4 Réseaux de neurones multicouches

Il est possible de superposer plusieurs couches de neurones les unes à la suite de autres comme présenté sur la figure 17.

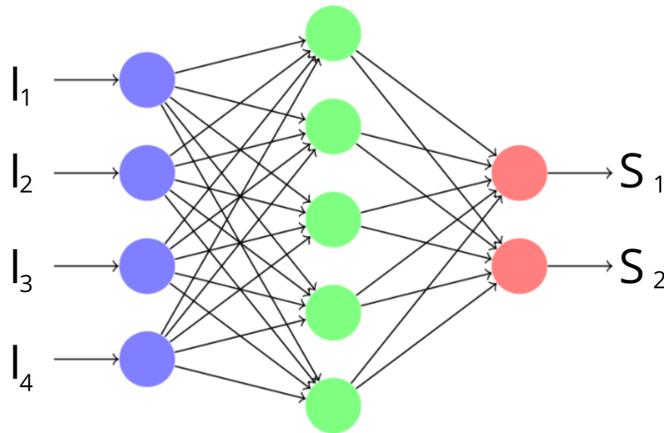


FIGURE 17 – Schéma d'un réseau de neurones multicouches :

Les sorties des neurones d'une couche constituent les entrées des neurones de la couche suivante. Cette disposition présente de nombreux avantages comme résoudre des problèmes non linéaires plus complexes, se comporter d'une meilleure manière en présence de données volumineuses et aussi minimiser le temps de prédiction après l'apprentissage.

Pour un neurone multicouches, le calcul des sorties est plus complexe que celui d'un perceptron. Considérons une couche l à deux neurones $S_1^{(l)}$ et $S_2^{(l)}$ d'un réseau de neurones multicouches.

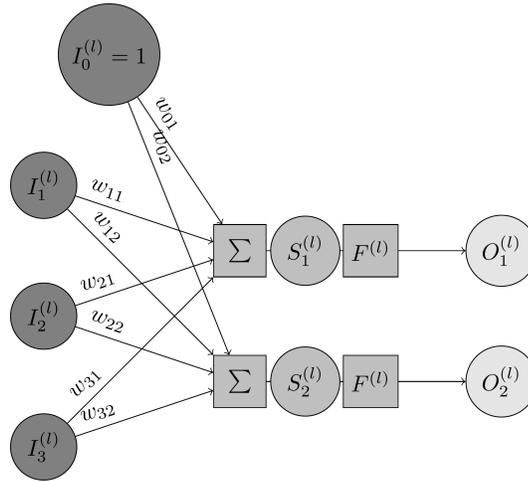


FIGURE 18 – Calcul sortie réseau de neurone multicouches

Ses entrées $I_i^{(l)}$ sont les sorties de la couche $l-1$ qui la précède. En notant W_{ij} avec $i \in 0, 1, 2, 3$ et $j \in 1, 2$ le poids de l'entrée $I_i^{(l)}$ du neurone $S_j^{(l)}$, les équations de sortie de cette couche sont :

$$S_j^{(l)} = \sum_i W_{ji}^{(l)} I_i^{(l)}$$

$$O_j^{(l)} = f^{(l)}(S_j^{(l)}) \rightarrow I^{(l+1)}$$

Les sorties $O_j^{(l)}$ sont obtenues en appliquant la fonction d'activation de la couche $F^{(l)}$ aux $S_j^{(l)}$

Pour résumer, les principaux éléments d'un réseau de neurones sont :

- des neurones répartis en plusieurs couches connectées entre elles
- chaque couche reçoit des données en entrée et les renvoie transformées. Pour cela, elle calcule une combinaison linéaire puis applique une fonction non-linéaire, appelée *fonction d'activation*. Les coefficients de la combinaison linéaire définissent les paramètres (ou poids) de la couche.
- les couches étant empilées les unes à la suite des autres, la sortie de l'une est l'entrée de l'autre.
- une dernière couche qui calcule le résultat final
- une fonction de perte (loss function) qui calcule l'erreur de prédiction ou de classification comme l'erreur quadratique moyenne (mean square error mse).
- apprendre un réseau de neurone, c'est apprendre les poids qui lient toutes les couches entre elles en minimisant l'erreur que fait le réseau sur des exemples.

4.3 Réseaux de neurones convolutionnels appliqués aux séries temporelles

4.3.1 Réseaux de neurones convolutionnels

Les réseaux de neurones convolutionnels désignent une sous-catégorie de réseaux de neurones spécialement conçus pour traiter des images en entrée afin de résoudre le problème de reconnaissance d'images par ordinateur.

- **Reconnaissance d'image par ordinateur**

Une image en informatique est désignée par un rectangle découpé suivant une grille régulière et dont, à chaque cellule appelée pixel, on attribue une valeur numérique quantifiant l'intensité lumineuse de ce pixel. Par exemple, pour une image en noir et blanc, cette valeur ne prend que les entiers compris entre 0 (noir) et 255 (blanc) et pour une image en couleur, un pixel est désigné par des triplets de niveaux de gris (rouge : (255,0,0), vert : (0,255,0) et bleu : (0,0,255)).

- **Features ou zones d'intérêt d'une image**

Le terme anglais *features* qui veut dire *caractéristiques* en français, désigne dans le domaine de la vision par ordinateur, des zones intéressantes d'une image numérique. Elles peuvent être des contours, des points ou des régions qui seront particuliers à l'image. Ainsi, le problème de reconnaissance d'images devient celui de *features matching* qui se résout en deux étapes :

⇒ Détecter les features dans une image et les décrire

⇒ Trouver les paires de features qui correspondent dans les deux images

Toute la performance d'un algorithme de reconnaissance d'images réside dans sa qualité à sélectionner les bonnes features. Une bonne feature a trois qualités principales. Elle doit être

- ▶ Répétable : on doit la retrouver dans des images représentant la même scène peu importe la transformation appliquée ; elle doit présenter des propriétés d'invariance à ces transformations.
- ▶ Distinctive : repérable facilement dans une image par son unicité et sans ambiguïté.
- ▶ Locale : suffisamment petite et uniquement décrite par son voisinage.

- **Extraction de features par réseaux de neurones convolutifs**

Le réseau de neurones convolutifs est constitué de deux blocs principaux. Le premier appelé en anglais Feature Extract Module fait la particularité de ces réseaux car il sert d'extracteur de *features* en réalisant un filtrage par convolution. La première couche de ce bloc filtre l'image avec plusieurs couches de convolution et retourne en sortie des *features maps* qui sont des cartes d'activation indiquant où se situent les features dans les images. On les renormalise avec la fonction d'activation ou redimensionne si nécessaire. Ce processus est réitéré plusieurs fois en fonction du nombre de couches de ce bloc. On concatène les valeurs des dernières features maps dans un vecteur qui est la sortie définitive du premier bloc et l'entrée du second.

Le second bloc appelé en anglais Classifier Module, n'est pas une caractéristique des réseaux de neurones convolutionnels car il se retrouve à la fin de tous les réseaux de neurones de classification. Il transforme les valeurs en entrée en un vecteur final qui représente les probabilités pour une image d'appartenir à une classe. Ce vecteur a autant d'éléments que de classes et la valeur à la position i représente la probabilité que l'image appartienne à la classe i . Par exemple, pour un réseau qui détecte les chats ou les insectes, la sortie de la couche finale est la possibilité que l'image d'entrée contienne l'un de ces animaux.

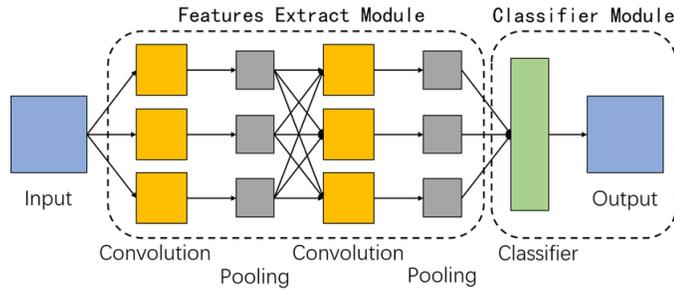


FIGURE 19 – Architecture d’un réseau de neurones convolutifs

- **Couche de convolution**

C’est la composante principale de ces réseaux qui représente toujours leur première couche. C’est elle qui effectue le *filtrage par convolution* afin de retrouver dans les images en entrée, un ensemble de features qui nous intéresse. Une feature est vue comme un filtre et c’est le résultat de la convolution entre ce filtre et l’image en entrée qui donne le feature map qui est une carte d’activation nous indiquant où se situent les features recherchés dans les images.

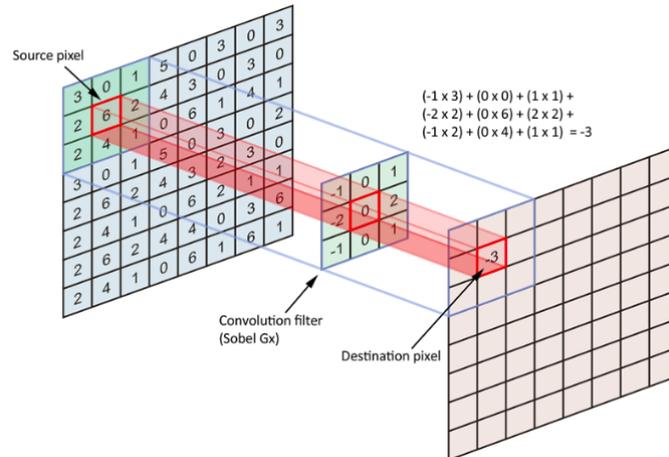


FIGURE 20 – Convolution de deux images

- **Choix des features d’une couche**

Les features particuliers d’un réseau de neurones convolutifs sont définis lors de l’entraînement de celui-ci par descente du gradient. La particularité des réseaux de neurones convolutifs est leur capacité à déterminer tous seuls les éléments distinctifs d’une image en s’adaptant au problème posé.

4.3.2 Application à la prédiction de séries temporelles

Le problème qui se présente dans le cadre de ce stage est la prédiction de séries de la croissance à partir de plusieurs autres séries des variables climatiques : la régression de séries temporelles. Ainsi, les données d’entrées ne sont plus des images, mais des séries temporelles multidimensionnelles. Une entrée est un tableau dont le nombre de lignes est la longueur de la série climatique et la largeur est le nombre de variables climatiques de la série.

Année	décade	Température min	Température max	Pluies	Rayonnement Global
2003	1	0.6	5.53	17.7	39.214
2003	2	0.98	6.99	26	47.726
2003	3	4.92	10.33	37	48.182
2003	4	2.75	7.42	40.1	49.599
2003	5	0.31	5.8	12	96.034

TABLE 4 – Série temporelle multidimensionnelle des variables climatiques de période décadaire pour l’année 2003 pour les cinq premières décades

La sortie est le vecteur de la croissance prédite dont la longueur est la même que celle de la série d’entrée. Le principe est le même pour la prédiction : une fenêtre, de même largeur que la table d’entrée se déplace de façon unidimensionnelle tout au long de celle-ci. Pour une couche de convolution, il y a autant de fenêtres que de filtres en sortie.

La longueur de la fenêtre de convolution représente le nombre de dates couvertes simultanément par la fenêtre de convolution. A la sortie de deuxième bloc, on obtient le vecteur final qui représente la série temporelle prédite. L’erreur est alors calculée en comparant la distance euclidienne entre la série exemple de croissance de l’herbe et la sortie du réseau. L’apprentissage du réseau se fait donc en déterminant les poids de chaque couche qui minimisent cette erreur.

4.4 Rappel des objectifs du modèle

On veut prédire la croissance de l’herbe moyenne journalière sur la décade au cours d’une année en utilisant comme entrées les variables climatiques et en apprenant sur les simulations de la croissance de l’herbe des prairies de la région Bretagne, prédite par le modèle STICS.

4.5 Architecture du réseau

4.5.1 Présentation de l’architecture

Le premier bloc du réseau est constitué d’une à trois couches de convolution présentant la même structure. Cette structure est composée de :

- Le nombre de couches du premier bloc n : chaque couche permet d’obtenir des informations plus précises. Cependant, un nombre de couches élevé peut s’avérer inutile car très peu d’informations restantes. Au contraire, ne disposer que d’une seule couche dans ce bloc ne permet pas d’obtenir assez d’informations en raison du sous-apprentissage.
- Le nombre de filtres l : Plus ce nombre est élevé, plus le temps de traitement croît et la quantité d’informations extraite est grande. Ainsi donc, un arbitrage est à effectuer entre le temps de calcul et la qualité d’informations obtenues. Toutefois, prendre un trop grand nombre de filtres conduit au sur-apprentissage et donc à une grande erreur sur la prédiction de nouvelles valeurs climatiques.
- La longueur d d’une fenêtre de convolution : Cette longueur a un impact sur la résolution temporelle des informations obtenues. Quand elle est petite, on obtient des informations sur les séries pour des dates proches et quand elle est élevée, des informations pour des dates éloignées.

Le deuxième bloc est constitué d'une seule couche de convolution. Celle-ci retourne la série temporelle finale qui est la croissance prédite.

4.5.2 Fonctionnement de la méthode

La couche de convolution permet d'obtenir les features des séries climatiques d'entrée à travers l'opération de convolution unidimensionnelle. Comment fonctionne t -elle? Une fenêtre de longueur choisie se déplace tout au long des séries d'entrée pour effectuer l'opération de convolution.

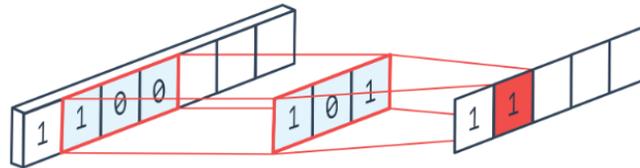


FIGURE 21 – Produit de convolution pour une série à une seule variable avec une taille de noyau de 3 et une longueur de vecteur égale à 7 et un seul vecteur de sortie

La convolution est effectuée sur la partie des variables d'entrée couvertes par la fenêtre. Les résultats sont additionnés et mis dans le vecteur de sortie. Dans notre cas, la fenêtre de convolution (1,0,1) couvre le vecteur (1,0,0). Le produit de convolution vaut $1 \times 1 + 0 \times 0 + 0 \times 1 = 1$. Généralement, la taille du features maps est inférieures à la longueur de la séries d'entrée. Souvent, il est nécessaire d'obtenir des features maps de même longueur que l'entrée. Pour cela, on utilise l'opération de *padding*. Il s'agit d'ajouter des pixels de valeurs 0 aux données d'entrée lors de l'opération de convolution.

$$\begin{array}{r}
 \begin{array}{|c|c|c|c|c|} \hline 1 & 5 & 1 & 4 & 2 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1 & -1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline -3 & 8 & -1 \\ \hline \end{array} \\
 \begin{array}{|c|c|c|c|c|c|} \hline 0 & 1 & 5 & 1 & 4 & 2 & 0 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1 & -1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline 4 & -3 & 8 & -1 & 2 \\ \hline \end{array}
 \end{array}$$

FIGURE 22 – Opération de *padding* permettant de conserver la même longueur du vecteur de sortie que la longueur du vecteur d'entrée sur un vecteur de longueur 5

Dans notre cas, cette opération est nécessaire à lors de chaque convolution car les vecteurs de sortie représentent la croissance sur une année donc 37 décades, même longueur que les séries d'entrée.

4.6 Expérimentations

4.6.1 Préparation de l'expérimentation

⊗ Paramètres du modèle

Des paramètres optimisés, permettent de minimiser les erreurs de prédiction. Malheureusement, il n'existe pas de méthodes algorithmique permettant de trouver les paramètres idéaux afin de minimiser l'erreur quadratique moyenne (mse). D'où l'usage de la méthode empirique qui consiste à évaluer l'erreur par validation croisée des modèles en faisant varier les paramètres n, l et d .

- n peut prendre les valeurs 1, 2 ou 3. Une seule couche conduit au sous-apprentissage, et au-delà de trois couches, les informations obtenues sont faibles.
- l varie de 4 à 32 par saut de 4. Au-delà de 32 filtres, le temps d'exécution devient très élevé alors que le gain sur la mse devient négligeable voire négatif.
- d varie de 3 à 11 par saut de 2. Cette longueur de fenêtre est adaptée à la longueur de la série décadaire qui est de 37.

Nous avons donc évalué 108 modèles en calculant les valeurs des mse associées.

⊗ Paramètres de l'algorithme

Faire tourner un réseau de neurones nécessite deux hyperparamètres importants : le nombre d' *epochs* et le nombre de *batches*. Un epoch correspond à un passage total de tout l'échantillon d'apprentissage. Les poids de chaque couche du réseau sont actualisés à la fin d'un epoch. Un *batch* est la taille de l'échantillon utilisé pour un passage dans le réseau. Il s'exprime en *batch_size* qui représente le nombre de séries ou images utilisées simultanément pour une itération dans un epoch. Pour couvrir la totalité de l'échantillon d'apprentissage, il faut K itérations. Par exemple si on a un échantillon d'apprentissage de 1000 séries ou images, une taille de batch de 500, alors le nombre d'itérations est 2.

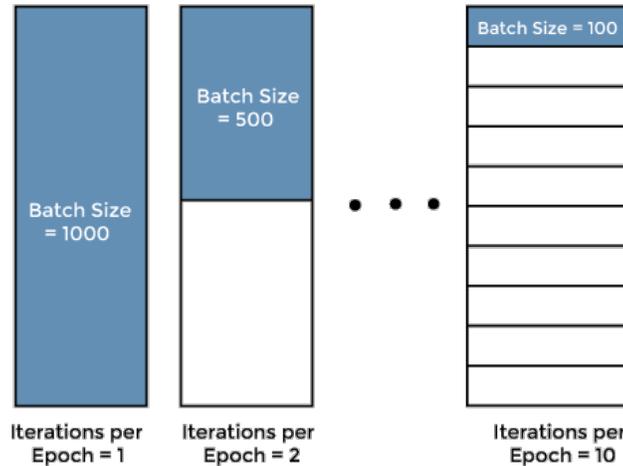


FIGURE 23 – Epochs et nombre de batches

Nous avons défini un epoch de 30 et une taille de batch de 64 pour tous les modèles. Au bout de 25 epochs, l'erreur de généralisation est stable pour la majorité des modèles, d'où le choix de la valeur de 30 pour une marge de confiance. Le nombre de batches a été choisi d'après la littérature. Généralement, on utilise une valeur de 16, 32, 64 ou 128. Avec 128 et les valeurs inférieures à 32, l'erreur de généralisation est très élevée. 64 est donc un bon compromis.

4.6.2 Résultats de l'expérimentation

Nous avons analysé les résultats en trois étapes en fixant un à un chacun des paramètres et en faisant varier les autres.

- Le nombre de couches a un impact sur l'erreur lorsque la taille du noyau est faible. On fixe la longueur du noyau et on fait varier pour chaque nombre de couches, le nombre de filtres. On présente les résultats

dans le tableau suivant :

n	d		3	5
	l			
2	4		282,07	237,73
2	6		289,81	243,89
2	8		272,76	225,72
2	10		263,05	226,05
3	4		244,69	221,80
3	6		236,99	204,85
3	8		226,75	194,32
3	10		226,38	187,84

TABLE 5 – Tableau des erreurs quadratiques moyennes (mse) des modèles pour des nombres de filtres l de valeurs 4, 6 , 8 ou 10 et des tailles noyaux d valant 3 ou 5 pour les nombres de couches n du premier bloc de valeur 2 ou 3.

Nous avons réalisé les boxplots des répartitions des erreurs quadratiques moyennes (mean square error : mse) selon chaque nombre de couches pour une taille de noyau fixé. L'erreur médiane de chaque distribution décroît avec le nombre de couches.

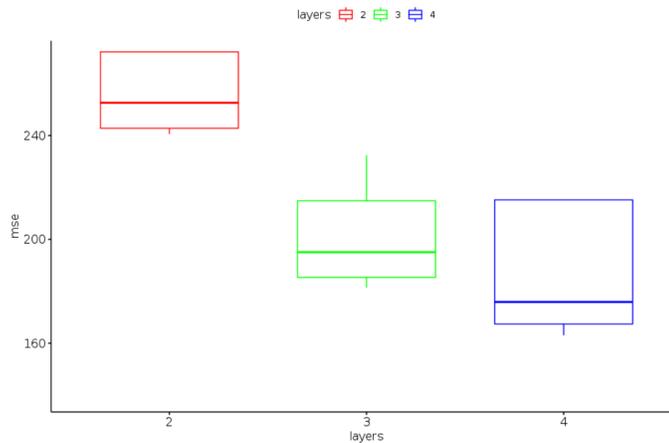


FIGURE 24 – Distribution de l'erreur quadratique moyenne (mse) en fonction du nombre de couches

Cependant, la différence entre les distributions des erreurs est d'autant plus marquée que la taille du noyau est petite. Pour une taille de noyau fixée, la mse diminue lorsque le nombre de couches augmente et cela, peu importe le nombre de filtres en sortie.

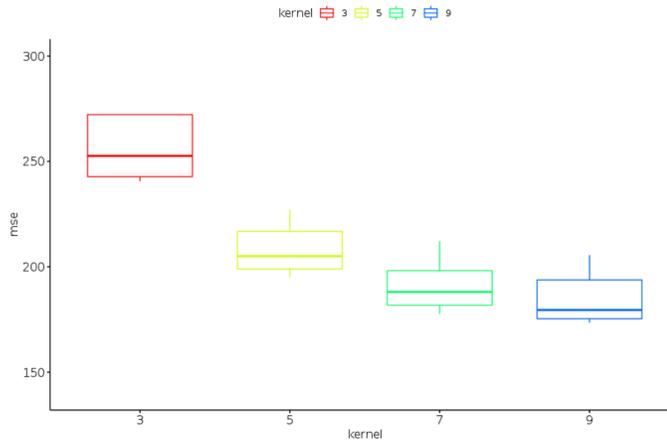


FIGURE 25 – Distribution de l’erreur quadratique moyenne (mse) en fonction du nombre de couches avec une taille de noyau égale à 9

- Pour un nombre de couches donné, la mse diminue lorsque la longueur du noyau augmente, indépendamment du nombre de filtres. Cette différence est d’autant plus marquée que le nombre de couches est élevé

Ces boxplots présentent les distributions des erreurs quadratiques moyennes (mse) obtenues avec un réseau à 2 couches en faisant varier l le nombre de filtres en sortie. Les boxplots obtenus ont des valeurs assez différentes.

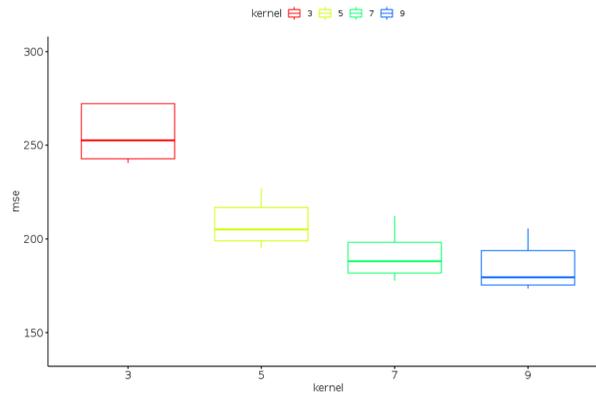


FIGURE 26 – Distribution de l’erreur quadratique moyenne (mse) en fonction de la taille du noyau pour un réseau à 2 couches

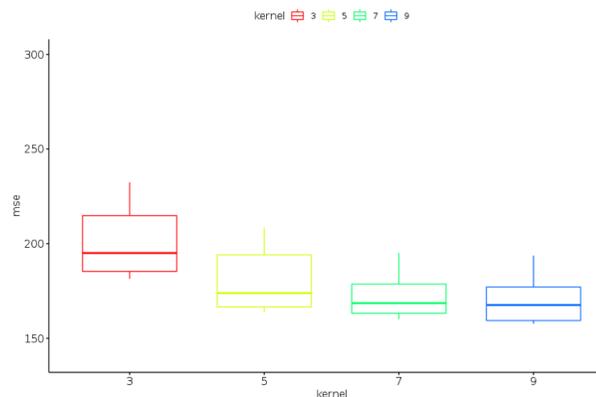


FIGURE 27 – Boxplots des mse en fonction de la taille du noyau pour un réseau à 3 couches

L'écart entre ces distributions se fait moins ressentir pour un réseau à trois couches.

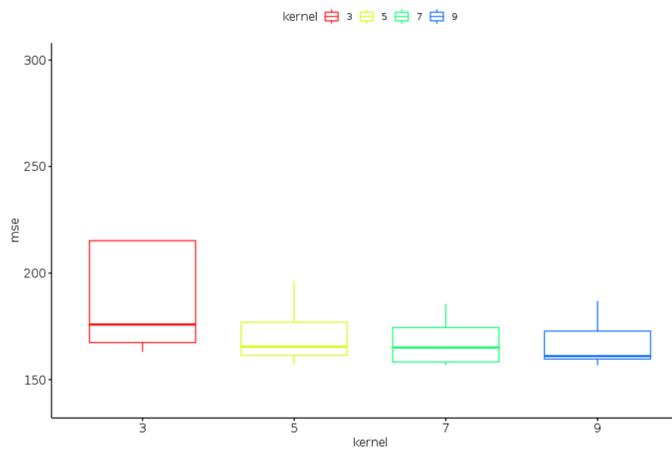


FIGURE 28 – Boxplots des mse en fonction de la taille du noyau pour un réseau à 4 couches

Pour un réseau à 4 couches, il devient difficile d'affirmer une différence. Cependant, on observe une tendance décroissante de la médiane en fonction de la taille du noyau.

Nous avons comparé les meilleurs modèles entre eux. Le meilleur correspond à un réseau ayant 4 couches, 7 comme longueur du noyau et ayant 32 vecteurs en sortie de chaque couche (couches du premier bloc). L'erreur en validation croisée est d'environ $154 \text{ (kg MS/ha/j)}^2$ et donc une RMSE (Root Mean Square Error : racine de l'erreur quadratique moyenne de $12,40 \text{ kg MS/ha/j}$)

4.6.3 Comparaison avec les autres méthodes précédemment développées (SVM, RF)

On compare les modèles développés dans le cadre de ce stage et ceux développés l'an dernier.

La meilleure performance obtenue l'an dernier était de 14 (kg MS/ha/j) comme RMSE (Root Mean Square Error : Racine de l'erreur moyenne quadratique) avec un random forest classique. L'erreur de prédiction de la croissance du modèle développé dans le cadre de ce stage et reposant sur l'utilisation des réseaux de neurones est moindre que celle des précédents modèles développés avec une RMSE de $12,40 \text{ kg MS/ha/j}$.

5 Conclusions et perspectives

Ce stage visait à, d'une part, développer une interface permettant à un utilisateur disposant de l'information climatique de prédire et visualiser la croissance de l'herbe prédite à partir de son climat par les modèles statistiques précédemment développés, et d'autre part, à enrichir le choix des modèles disponibles par un modèle utilisant les réseaux de neurones convolutionnels. Ces objectifs ont été atteints. Le modèle utilisant les réseaux de neurones convolutionnels s'avère plus performant que le modèle le plus performant développé dans le cadre du stage de Laurent Spillaemacker, avec une erreur sur la croissance prédite de 12,45 kg MS/ha/jour en moyenne sur la décade. Son temps d'exécution est plus court que celui des deux autres méthodes (machine à vecteur de support et forêt aléatoire). Cela est lié notamment au fait qu'il permet de s'affranchir de l'autocorrélation temporelle des valeurs de croissance entre décades. En revanche, il n'existe pas pour cette méthode de façon de déterminer les paramètres optimaux du modèle, si bien que la méthode empirique utilisée peut fournir des paramètres non optimaux. On pourrait imaginer de poursuivre ce travail en essayant d'identifier parmi les variables climatiques utilisées pour la prédiction de la croissance, celles qui sont le plus informatives et influentes, et permettent peut-être une performance équivalente avec moins de variables. Cela permettrait à l'utilisateur d'avoir peut-être moins d'informations climatiques à fournir et d'éviter le sur-apprentissage. Une autre piste serait d'essayer d'utiliser l'information disponible sur le type de sol et les pratiques de gestion de l'herbe pour développer des modèles qui tiennent compte de cette information et voir si la qualité de la prédiction de la croissance s'en voit améliorée.

Références

- [1] Nadine Brisson, Christian Gary, Eric Justes, Romain Roche, Bruno Mary, Dominique Ripoche, Daniel Zimmer, Jorge Sierra, Patrick Bertuzzi, Philippe Burger, et al. An overview of the crop model stics. *European Journal of agronomy*, 18(3-4) :309–332, 2003.
- [2] Emmanuelle de Martonne. Aréisme et indice d’aridité. *Comptesrendus de L’Academie des Sciences*, 182 :1395–1398, 1926.
- [3] A-I Graux, Rémi Resmond, Eric Casellas, Luc Delaby, Philippe Faverdin, Christine Le Bas, Dominique Ripoche, Françoise Ruget, Olivier Therond, Françoise Vertès, et al. High-resolution assessment of french grassland dry matter and nitrogen yields. *European Journal of Agronomy*, 112 :125952, 2020.
- [4] Anne-Isabelle Graux, Luc Delaby, Jean-Louis Peyraud, Eric Casellas, Philippe Faverdin, Christine Le Bas, Anne Meillet, Thomas Poméon, Helene Raynal, Rémi Resmond, Dominique Ripoche, Françoise Ruget, Olivier Therond, and Françoise Vertès. Les prairies françaises : production, exportation d’azote et risques de lessivage. Technical report, Ministère de l’Alimentation, l’Agriculture et de la Forêt, 2017.