

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed Seddik Ben Yahia - Jijel
Faculté des Sciences et de la Technologie



د الصديق بن يحيى - جيجل

ب

ية الع

Département D'Électronique

Mémoire

Présenté en vue de l'obtention du diplôme

Master en Électronique

Option : Électronique des Systèmes Embarqués

Thème

**Application des réseaux de neurones profonds pour la prédiction
de la puissance produite par un micro-réseau, dédié au
chargement des véhicules électriques**

Réalisé par :

Mr. AYYOUB LEFZA

Encadré par :

Pr . ADEL MELLIT

Promotion : **Octobre 2020**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

REMERCIEMENTS

Tout d'abord, louange à **dieu** Tout-Puissant qui était avec moi dans tous mes pas et chemin, je le remercie pour mon succès.

Je remercie mon encadreur, le professeur **#Adel_Mellit**, pour toute l'aide qu'il m'a apportée, pour ses précieux conseils et pour sa confiance en moi.

Notre vifs remerciements également aux membres **jury** pour son intérêt pour mon recherches et pour son acceptation d'examiner mon travail.

Notre remerciements vont tous les **enseignants** qui m'ont soutenu dans ma carrière universitaire.

Merci à toute personne ayant contribué de près ou de loin à la concrétisation de ce projet.

DEDICACE

Avec l'expression de ma reconnaissance, je dédie ce modeste travail à ceux qui, quels que soient les termes embrassés, je n'arriverais jamais à leur exprimer mon amour sincère.

À la femme qui a souffert sans me laisser souffrir, et qui m'a appris la patience et à ne pas abandonner dans l'adversité, et je n'oublie pas que c'est elle qui m'a encouragé à étudier à l'université: mon adorable mère **Karima**.

À l'homme, mon précieux don du dieu, Qui doit ma vie, mon succès et tout mon respect, à l'homme qui a sacrifié tout son savoir, son argent et ses efforts pour moi, mon cher père **Hocine**.

À l'homme, dont j'ai appris beaucoup de choses qui m'ont aidé dans ma vie, mon cher grand-père, **Mohammed Al-Taher Mezhoud**.

À la seule princesse de notre famille, ma chère sœur, **Chaima**.

À mes chers frères **Ahmed, Taha**, et à la plus jeune personne de notre famille, Mon cher frère **Lokman**. Et je n'oublie pas mon cher ami **Fares Lokrai**, que je considère comme mon frère.

À tous mes **amis proches**, à tous mes **collègues**.

À toute la troisième promotion d'Électronique des Systèmes Embarqués (promotion 2019/2020).

Et enfin, À l'âme de l'une des meilleures personnes que j'ai connues au cours des cinq dernières années en résidence universitaire, mon cher collègue, L'humble étudiant **Soufiane Mezziche** qui est mort quelques mois avant d'avoir obtenu son diplôme de master, et je prie Dieu de lui pardonner et de le faire entrer au ciel.

ملخص

تعد التنبؤات الدقيقة قصيرة المدى للطاقة التي تتجها الخلايا الكهروضوئية (PV) ضرورة للتحكم في أنظمة إدارة الطاقة الذكية وتصميمها للشبكات الصغيرة. في هذا العمل ، تم تطوير ومقارنة أنواع مختلفة من الشبكات العصبية للتعمق العميق (DLNNS) للتنبؤ بفترة الإخراج الكهروضوئية قصيرة المدى: الذاكرة طويلة المدى (LSTM) ، الوحدة المتكررة المغلقة (GRU) ، والشبكة العصبية التلافيفية أحادية البعد (CNN1D).

تُستخدم قاعدة بيانات للطاقة الكهروضوئية التي تتجها شبكة صغيرة مثبتة في جامعة تريست (إيطاليا) لتدريب الشبكات العصبية ونياها. تم تقييم الأداء على مدى أربعة آفاق زمنية مختلفة (1 دقيقة ، 5 دقائق ، 30 دقيقة و 60 دقيقة) ، لخطوة واحدة أو أكثر مقدما. أظهرت النتائج أن شبكات DLNN التي تمت دراستها تقدم دقة جيدة جداً ، لا سيما في حالة الأفق الزمني الذي يبلغ دقيقة واحدة مع طول التقدم (معامل الارتباط قريب من 1).

الكلمات المفتاحية : الشبكات العصبية العميقة ، التنبؤ ، الشبكة الدقيقة ، الخلايا الكهروضوئية.

Abstract

Accurate short-term forecasting of photovoltaic (PV) power is indispensable for controlling and designing smart energy management systems for micro grids. In this work, different kinds of deep learning neural networks (DLNN) for short-term output PV power forecasting have been developed and compared: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional and One-Dimension Convolutional Neural Network (CNN_{1D}).

A database of the PV power produced by the micro grid installed at the University of Trieste (Italy) is used to train and comparatively test the neural networks. The performance has been evaluated over four different time horizons (1 min, 5 min, 30 min and 60 min), for one-step and multi-step ahead.

The results show that the investigated DLNNs provide very good accuracy, particularly in the case of 1 min time horizon with one-step ahead (correlation coefficient is close to 1).

Keywords : Deep Neural Network, prediction, Micro grid, photovoltaic.

Résumé

Des prévisions précises à court-terme de la puissance produite par le photovoltaïque (PV) sont indispensables pour contrôler et concevoir des systèmes intelligents de gestion de l'énergie pour les micro-réseaux. Dans ce travail, différents types de réseaux de neurones d'apprentissage en profondeur (DLNN) pour la prévision de la puissance PV de sortie à court terme ont été développés et comparés: mémoire à long-court terme (LSTM), unité récurrente fermée (GRU), et convolutif à une dimension Réseau neuronal (CNN_{1D}).

Une base de données de la puissance PV produite par un micro-réseau installé à l'Université de Trieste (Italie) est utilisée pour former et tester comparativement les réseaux de neurones. La performance a été évaluée sur quatre horizons temporels différents (1 min, 5 min, 30 min et 60 min), pour une et plusieurs étapes en avance.

Les résultats montrent que les DLNN étudiés offrent une très bonne précision, en particulier dans le cas d'un horizon temporel de 1 min avec une longueur d'avance (le coefficient de corrélation est proche de 1).

Mots-clés : Réseaux de neurones profonds, prédiction, micro-réseau, photovoltaïque.

Table des matières

Introduction générale	1
Chapitre 1 : Généralités sur le photovoltaïque	
I.1. Introduction	3
I.2. Energie solaire	3
I.3. Les caractéristiques du rayonnement solaire sur Terre	3
I.4. L'énergie solaire photovoltaïque (PV)	4
I.5. L'effet photovoltaïque	5
I.5.1. La cellule photovoltaïque	5
I.5.2. Principe de fonctionnement d'une cellule photovoltaïque	6
I.5.3. Technologies de cellules solaires	7
I.5.3.1. La cellule au silicium amorphe	7
I.5.3.2. La cellule au silicium monocristallin	7
I.5.3.3. La cellule au silicium multi cristallin	7
I.5.4. Les autres technologies	7
I.5.4.1. Arséniure de gallium (GaAs)	8
I.5.4.2. Tellurure de cadmium (CdTe)	8
I.5.4.3. Diséléniure de cuivre et d'indium	8
I.5.4.4. Cellules multi jonctions (matériaux III-V)	8
I.5.4.5. Cellules organiques	8
I.5.4.6. Cellules à particules quantiques	8
I.5.4.7. Cellules Cuivre Zinc Étain Soufre et Sélénium (CZTS)	9
I.5.4.8. Cellules à colorants	9
I.5.4.9. Cellules pérovskite	9
I.5.5. Caractéristiques électriques d'une cellule photovoltaïque	9
I.5.5.1. Modèle mathématique d'un module solaire	11
I.5.6. Rendement et puissance crête des cellules photovoltaïques	11

I.5.7. Les performances du générateur photovoltaïque	12
I.5.7.1. Caractéristique Courant-Tension	12
I.5.7.2. Influence de l'éclairement	13
I.5.7.3. Influence de la température	14
I.5.7.4. Limite de Shockley-Queisser pour le silicium	14
I.5.8. Le module photovoltaïque	15
I.5.8.1. Association en série	15
I.5.8.2. Association en parallèle	15
I.5.9. La puissance caractéristique d'une cellule PV	15
I.5.9.1. La puissance maximale	15
I.5.9.2. Le facteur de forme	16
I.6. Conclusion	16
 Chapitre 2 : Réseaux de neurones profonds	
II.1. Introduction	17
II.2. Réseaux de neurones	18
II.2.1. définition	18
II.2.2. Bref Historique	18
II.2.3. Fonctionnement	19
II.2.4. Neurone biologique	19
II.2.5. Neurone formel	20
II.3. Réseaux de neurones récurrents	21
II.3.1. Problématique	25
II.4. RÉSEAU DE NEURONES PROFONDS	26
II.4.1. Long Short-Term Memory (LSTM)	26
II.4.1.1. La cellule LSTM pas à pas	27
II.4.2. Gated Recurrent Units (GRU)	32
II.4.2.1. Définition	32
II.4.2.2. Pourquoi est-ce utile?	33

II.4.2.3. Applications d'une unité récurrente fermée	33
II.4.3. Réseau de neurone convolutif (CNN)	33
II.5. Conclusion	34
Chapitre 3 : Tensorflow, Keras et Anaconda	
III.1. Introduction	35
III.2. Ressources matérielles	35
III.3. Ressources logicielles	35
III.3.1. Système d'exploitation	35
III.3.2. Langage de programmation	35
III.3.3. Anaconda	37
III.3.3.1. TensorFlow	38
III.3.3.2. Keras	39
III.4. Conclusion	40
Chapitre 4 : Simulation et résultats	
IV.1. Introduction	41
IV.2. MICROGRID	41
IV.2.1. Description	41
IV.2.2. Base de données	43
IV.4. Methodologie	44
IV.5. Résultats et Discussion	45
IV.6. Conclusion	52
Conclusion générale	53
Références	54
Annexe	57

Liste des figures

- Figure 1.1:** Modélisation de l'inégale répartition de l'énergie solaire dans le monde : ici la «GHI» ou « Global Horizontal Irradiation ».
- Figure 1.2:** Constitution d'une cellule photovoltaïque (Connaissance des Énergies, d'après CEA).
- Figure 1.3:** Circuit équivalent d'une cellule solaire.
- Figure 1.4:** Caractéristique $I=f(V)$ d'un module photovoltaïque.
- Figure 1.5:** Influence de l'éclairement.
- Figure 1.6:** Influence de température.
- Figure 1.9:** Puissances maximale idéale et pratique.
- Figure 2.1:** La relation entre l'intelligence artificielle, l'apprentissage automatique et L'apprentissage en profondeur.
- Figure 2.2:** Structure d'un neurone biologique.
- Figure 2.3:** Schématisation d'un neurone formel.
- Figure 2.4:** Représentation compacte des RNN. Toutes les flèches représentent des connexions complètes. La flèche en pointillée représente les connexions ayant un décalage temporel ($t - 1$).
- Figure 2.5:** Représentation dépliée des RNN.
- Figure 2.6:** Représentation dépliée hiérarchisée des RNN.
- Figure 2.7:** Une chaîne de cellules LSTM.
- Figure 2.8:** Entrées d'une cellule LSTM : input, hidden state, cell state.
- Figure 2.9:** Filtre de la forget gate.
- Figure 2.10:** Cell state filtré par la forget gate.
- Figure 2.11:** Filtre de l'input gate.
- Figure 2.12:** Vecteur candidat à la mise à jour du cell state.
- Figure 2.13:** Vecteur candidat filtré par l'input gate.
- Figure 2.14:** Mise à jour du cell state par addition du vecteur candidat filtré.
- Figure 2.15:** Filtre de l'output gate.
- Figure 2.16:** Nouveau hidden state.
- Figure 2.17:** Schéma d'un réseau GRU à une unité.

Liste des figures

- Figure 2.18:** Architecture standard d'un réseau à convolutions.
- Figure 3.1:** Logo Python.
- Figure 3.2:** Interface Anaconda.
- Figure 3.3:** Logo TensorFlow.
- Figure 3.4:** Logo Keras.
- Figure 4.1:** Le schéma fonctionnel de la MG disponible à l'université de Trieste, Italie.
- Figure 4.2:** La station de chargement des véhicules électriques à base d'un MG a l'Université de Trieste, Italie.
- Figure 4.3:** Un exemple des données surveillées du 01/01/2020 au 30/03/2020: (a) la puissance PV produite, (b) la puissance consommée et le (c) l'état de charge (SoC).
- Figure 4.4:** Comparaison entre la puissance de sortie PV mesurée et prédit (cas de Bi-LSTM)
a) Superposition plot entre mesuré et prédit, et b) Scatter plot.
- Figure 4.5:** Comparaison entre la puissance de sortie PV mesurée et prédit (cas de GRU)
a) Superposition plot entre mesuré et prédit, et b) Scatter plot.
- Figure 4.6:** Comparaison entre la puissance de sortie PV mesurée et prédit (cas de GRU)
a) Superposition plot entre mesuré et prédit, et b) Scatter plot.
- Figure 4.7:** Comparaison entre la puissance de sortie PV mesurée et prédit (cas de Stacked)
a) Superposition plot entre mesuré et prédit, et b) Scatter plot.
- Figure 4.8:** a) L'erreur absolue pour les différentes méthodes étudiées à 1 min, 5 min, 30 min et 1h. (la couleur rouge indique les valeurs aberrantes, qui sont plus grandes que la valeur maximale, le centre de la boîte est la valeur moyenne). b) Évolution du rendement par rapport à la taille de la base de données (cas de Bi-LSTM).

Liste des tableaux

Tableau 2.1: Types de fonctions de transfert pour R.N.F.

Tableau 3.1: Paramètre de comparaison entre Anaconda et Python.

Tableau 4.1: Les caractéristiques de panneau solaire PV utilisée dans la recherche.

Tableau 4.2: Erreurs métriques : R, RMSE et MAPE entre la puissance de sortie PV mesurée et prédit (Cas 1 && 5 min à l'avance).

Tableau 4.3: Erreurs métriques : R, RMSE et MAPE entre la puissance de sortie PV mesurée et prédit(Cas 30 min && 1 h à l'avance).

Liste des sigles et abréviations

- T : Augmentation de température par rapport aux conditions STC (25°C).
- a : Coefficient de température Voc [mV/K], valeur fournie par le fabricant.
- $P(W)$: Puissance mesurée aux bornes de la cellule PV.
- $V(V)$: Tension mesurée aux bornes de la cellule PV.
- $I(A)$: Intensité mesurée aux bornes de la cellule PV.
- $P_{Max\ idéalé}(W)$: La puissance mesurée aux bornes de la cellule PV.
- $V_{co}(V)$: La tension de circuit ouvert mesurée aux bornes de la cellule PV.
- $I_{cc}(A)$: L'intensité de court-circuit mesurée aux bornes de la cellule PV.
- f : Facteur de forme.
- $P_{Max}(W)$: Puissance maximale aux bornes de la cellule PV.
- $V_{co}(V)$: Tension de circuit ouvert.
- $I_{cc}(A)$: Intensité de court-circuit.

Introduction générale

L'énergie solaire est la source la plus puissante car elle fait appelle à la source d'énergie la plus partagée qui est le rayonnement solaire. Cette énergie se caractérise par une absence de pollution et une disponibilité exceptionnelle. D'autre part, les systèmes d'exploitations qui utilisent cette forme d'énergie demandent une maintenance légère et présente une bonne fiabilité de fonctionnement, une autonomie de plus en plus accrue, une résistance extrême aux conditions naturelles (température, humidité, vent, corrosion, ...etc.) et donc une grande longévité.

L'utilisation de l'énergie photovoltaïque (PV) à grande échelle passe dans les pays industrialisés par la réalisation de systèmes connectés au réseau électrique. Ces systèmes énergétiques à sources renouvelables d'énergie photovoltaïque connaissent, à l'heure actuelle, un essor important. La simplicité de leur mise en œuvre, l'évolution favorable des prix et la politique environnementale menée par de nombreux gouvernements jouent un rôle prépondérant dans leur développement. [1]

Beaucoup de recherches se concentrent sur la fourniture d'un outil de prévision afin de prévoir la production d'énergie photovoltaïque (exemple micro-réseau) avec une bonne précision [2,3]. Récemment, des études faites dans ce domaine montrent certaine lacune des réseaux de neurones classiques dans la prédiction des séries temporelles [4]. Les réseaux de neurones profonds se sont révélés être des outils très puissants dans la prévision de la puissance produite par des installations photovoltaïques. [5]

Ce travail a pour but d'appliquer certain types de réseaux de neurones profonds pour la prédiction de la puissance produite par une installation photovoltaïque utilisé pour le chargement des véhicules électriques. Pour cela une base de données de puissance (PV) collectée durant une période de 4 mois sera utilisée afin d'implémenter différent modèles de prédiction.

Ce mémoire est composé de quatre chapitres :

Le premier chapitre est consacré à une introduction générale sur les installations photovoltaïques, ainsi que la conversion photovoltaïque.

Introduction générale

Le deuxième chapitre porte sur la présentation générale des réseaux de neurones et les réseaux de neurones de type récurrent profonds (LSTM, GRU and CNN).

Le troisième chapitre décrit les différentes fonctions de python utilisées pour le développement de modèles de prédiction à base de réseaux de neurones profonds.

Les résultats et discussion seront présentés dans le quatrième chapitre.

Chapitre 1

Généralités sur le photovoltaïque

I.1. Introduction

Dans le monde, il existe différentes sources d'énergies utilisables afin de pouvoir répondre aux nombreuses demandes de la population mondiale, qui elle, augmente de jours en jours. Cette utilisation de l'énergie a permis une amélioration du niveau de vie des populations des pays développés.

L'être humain a toujours utilisé le (charbon, pétrole, gaz naturel...) pour son bien-être. Le coût et la diversité de leur utilisation a fait la principale et dominante source d'énergie, jusqu'à ces dernières années où l'environnement commence à se dégrader et l'épuisement des énergies fossiles ont pris place aux yeux du monde, d'où la nécessité de chercher des alternatives. Comme solution, on a des tendances à se diriger vers les énergies renouvelables. L'une des plus importantes de ces énergies renouvelables est l'énergie solaire.

I.2. Energie solaire

L'énergie solaire est une source d'énergie qui dépend du soleil. Cette énergie permet de fabriquer de l'électricité à partir de panneaux photovoltaïques ou des centrales solaires thermiques, grâce à la lumière du soleil captée par des panneaux solaires.

L'énergie solaire est propre, n'émet aucun gaz à effet de serre et sa matière première, le soleil bien que distant de plus de 150 millions de kilomètres de nous, est gratuite, inépuisable et disponible partout dans le monde. [6]

Actuellement, il existe deux voies principales d'exploitation de l'énergie solaire :

-) le solaire photovoltaïque qui transforme directement le rayonnement solaire en électricité.
-) le solaire thermique qui transforme directement le rayonnement en chaleur. Le solaire dit « thermodynamique » est une variante du solaire thermique. Cette technique se différencie en cela qu'elle utilise l'énergie thermique du soleil afin de la transformer dans un second temps en électricité.

De nombreux programmes de recherche sont en cours afin d'améliorer les rendements des nouvelles technologies d'exploitation de l'énergie solaire. [7]

I.3. Les caractéristiques du rayonnement solaire sur terre

L'énergie solaire reçue en un point du globe dépend de :

-) l'énergie solaire envoyée par le Soleil, qui fluctue avec son activité de fusion.

- J) la nébulosité (nuages, brouillards, etc.), qui est par exemple importante à l'équateur et plus faible en milieu intertropical.
- J) la latitude, la saison et l'heure, qui influent sur la hauteur du soleil dans le ciel et donc l'énergie reçue par unité de surface au sol. (Figure 1.1) [7]

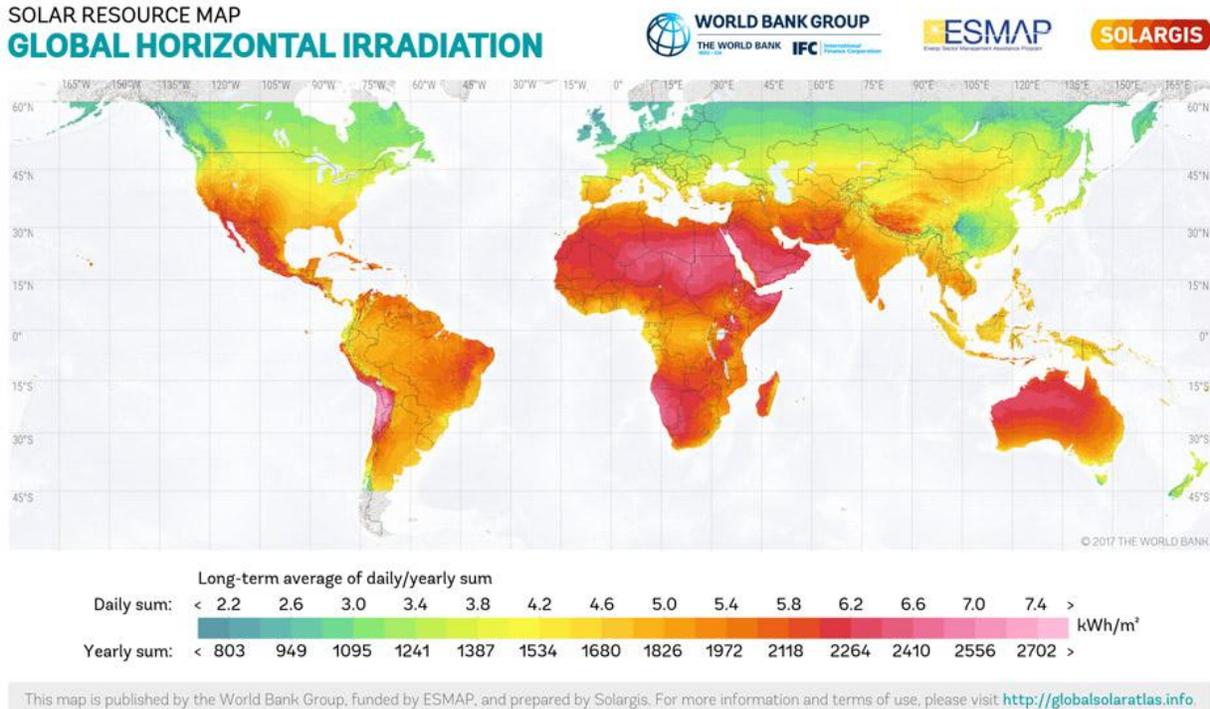


Figure 1.1: Modélisation de l'inégale répartition de l'énergie solaire dans le monde : ici la « GHI » ou « Global Horizontal Irradiation ».

I.4. L'énergie solaire photovoltaïque (PV)

Le terme « photovoltaïque », souvent abrégé par les lettres PV, a été formé à partir des mots « photo », un mot grec signifiant lumière, et « Volta », le nom du physicien italien Alessandro Volta, qui a inventé la pile électrochimique en 1800. L'effet photovoltaïque, c'est la conversion directe de l'énergie solaire en électricité. Cette technologie ne produit pas de chaleur, contrairement aux chauffe-eau et aux chauffe-piscines solaires résidentiels. Elle diffère également de la technologie utilisée dans les centrales thermodynamiques solaires où l'énergie solaire concentrée sert à produire de la vapeur qui active une turbine raccordée à un groupe électrogène. Les systèmes photovoltaïques ne contiennent aucune pièce mobile. [7]

Ils sont fiables, requièrent peu d'entretien, sont silencieux et ne produisent aucune émission de polluants. Ce sont des systèmes modulaires : les éléments de base (modules de cellules solaires)

sont vendus dans un vaste éventail de capacités d'alimentation électrique qui vont d'une fraction de watt (par exemple les montres et les calculettes à pile solaire) à plus de 300 watts. L'interconnexion des modules permet d'obtenir la puissance qu'exige votre application. Certaines centrales PV de démonstration produisent une puissance allant jusqu'à plusieurs mégawatts, bien que la plupart des systèmes installés soient de dimension beaucoup plus modeste. [8]

I.5. L'effet photovoltaïque

Lorsqu'un matériau est exposé à la lumière du soleil, les atomes exposés au rayonnement sont " bombardés " par les photons constituant la lumière; sous l'action de ce bombardement, les électrons des couches électroniques supérieures (appelés électrons des couches de valence) ont tendance à être " arrachés ".

Si l'électron revient à son état initial, l'agitation de l'électron se traduit par un échauffement du matériau. L'énergie cinétique du photon est transformée en énergie thermique. Par contre, dans les cellules photovoltaïques, une partie des électrons ne revient pas à son état initial. Les électrons " arrachés " créent une tension électrique continue faible. Une partie de l'énergie cinétique des photons est ainsi directement transformée en énergie électrique: c'est l'effet photovoltaïque.

L'effet photovoltaïque constitue la conversion directe de l'énergie du rayonnement solaire en énergie électrique au moyen de cellules généralement à base de silicium. Pour obtenir une puissance suffisante, les cellules sont reliées entre elles et constituent le module solaire. L'effet photovoltaïque, c'est-à-dire la production d'électricité directement de la lumière, fut observée la première fois, en 1839, par le physicien français Edmond Becquerel. Toutefois, ce n'est qu'au cours des années 1950 que les chercheurs de la compagnie Bell Téléphone, aux Etats-Unis, parvinrent à fabriquer la première photopile, l'élément primaire d'un système photovoltaïque.

I.5.1. La cellule photovoltaïque

Une cellule photovoltaïque est un composant électronique qui, exposé aux photons de la lumière, produit de l'électricité. Cet effet photovoltaïque fut découvert en 1839 par le physicien français Antoine Becquerel. Il faut attendre les années 1960 et les satellites spatiaux pour que les cellules trouvent de réelles applications. Les panneaux solaires, formés par des assemblages de cellules, commencent à arriver sur les toits à la fin des années 1980. L'essor du photovoltaïque, avec de grands parcs de production électrique, n'a plus cessé depuis le début du XXI^e siècle. [9]

I.5.2. Principe de fonctionnement d'une cellule photovoltaïque

Les cellules photovoltaïques exploitent l'effet photoélectrique pour produire du courant continu par absorption du rayonnement solaire. Cet effet permet aux cellules de convertir directement l'énergie lumineuse des photons en électricité par le biais d'un matériau semi-conducteur transportant les charges électriques.

Une cellule photovoltaïque est composée de deux types de matériaux semi-conducteurs, l'une présentant un excès d'électrons et l'autre un déficit d'électrons. Ces deux parties sont respectivement dites « dopées » de type n et de type p. Le dopage des cristaux de silicium consiste à leur ajouter d'autres atomes pour améliorer la conductivité du matériau. Un atome de silicium compte 4 électrons périphériques. L'une des couches de la cellule est dopée avec des atomes de phosphore qui, eux, comptent 5 électrons (soit 1 de plus que le silicium). On parle de dopage de type n comme négatif, car les électrons (de charge négative) sont excédentaires. L'autre couche est dopée avec des atomes de bore qui ont 3 électrons (1 de moins que le silicium). On parle de dopage de type p comme positif en raison du déficit d'électrons ainsi créé. Lorsque la première est mise en contact avec la seconde, les électrons en excès dans le matériau n diffusent dans le matériau p (Figure 1.2), En traversant la cellule photovoltaïque, les photons arrachent des électrons aux atomes de silicium des deux couches n et p. Les électrons libérés se déplacent alors dans toutes les directions. Après avoir quitté la couche p, les électrons empruntent ensuite un circuit pour retourner à la couche n. Ce déplacement d'électrons n'est autre que de l'électricité. [10]

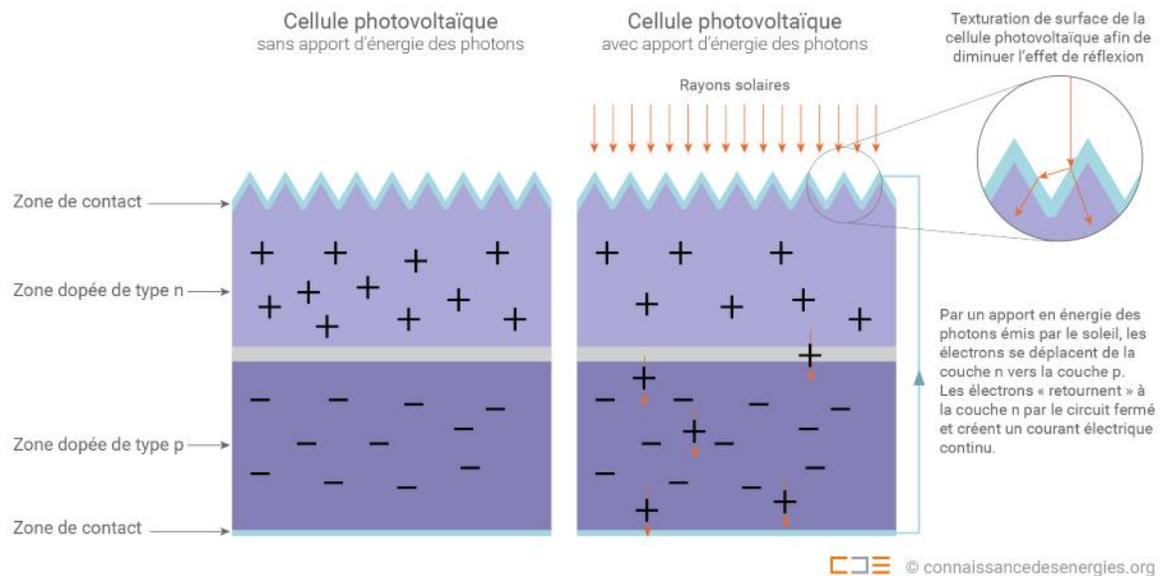


Figure 1.2: Constitution d'une cellule photovoltaïque (Connaissance des Énergies, d'après CEA).

I.5.3. Technologies de cellules solaires

L'énergie photovoltaïque est obtenue à l'aide de capteurs appelés cellules solaires ou photopiles. Il existe plusieurs technologies de cellules solaires, mais toutes sont composées de matériaux semi-conducteurs. Le principal matériau semi-conducteur utilisé est le silicium. Trois technologies utilisent ce matériau:

- la cellule au silicium amorphe.
- la cellule au silicium monocristallin.
- la cellule au silicium polycristallin.

De façon plus marginale, certaines cellules sont composées d'un autre matériau semi-conducteur comme le tellure de cadmium (Cd Te) ou le diséléniure de cuivre et d'indium. [11]

I.5.3.1. La cellule au silicium amorphe

La cellule au silicium amorphe est constituée d'un silicium ayant une structure atomique désordonnée, c'est à dire non-cristallisé. Ce matériau absorbe la lumière beaucoup plus qu'un silicium cristallin. Il fait partie de la technologie "couche mince" qui consiste à déposer sous vide sur un substrat (verre, métal, plastique, ...) une fine couche uniforme composée d'un et plus souvent de plusieurs matériaux réduits en poudre. Les cellules sont gris foncé. [11]

I.5.3.2. La cellule au silicium monocristallin

La cellule au silicium monocristallin est constituée d'un silicium formé d'un seul cristal ordonné. Ce matériau est obtenu directement d'un germe ou recristallisé à haute température sous forme d'un lingot. Il est ensuite découpé en fines tranches qui donneront les cellules. Ces cellules sont en général d'un bleu uniforme. [11]

I.5.3.3. La cellule au silicium multi cristallin

La cellule au silicium multi cristallin (ou poly cristallin) est constituée de cristaux de 1mm à environ 2cm assemblés, appelé "grains". La multitude de cristaux se forme pendant le refroidissement du silicium dans la lingotière. La cellule photovoltaïque est d'aspect bleuté, mais pas uniforme, on distingue des motifs créés par les différents cristaux.

I.5.4. Les autres technologies [11]

Les cellules sont faites aussi avec d'autres matériaux comme l'arséniure de gallium (GaAs), le tellure de cadmium (Cd Te) ou le diséléniure de cuivre et d'indium (CIS) qui ont l'avantage

d'absorber fortement la lumière, mais leur manipulation est difficile et certains matériaux comme le cadmium pose des problèmes de toxicité.

I.5.4.1. Arséniure de gallium (GaAs)

Ce matériau possède un très haut rendement, mais reste très onéreux du fait des éléments utilisés. Il est utilisé dans le domaine spatial ou sur des systèmes à concentration.

I.5.4.2. Tellurure de cadmium (CdTe)

Ce matériau absorbe très bien la lumière et peut être déposé en couches minces. Cependant son rendement reste encore assez faible et la toxicité du cadmium pose un problème une fois la cellule en fin de vie.

I.5.4.3. Diséléniure de cuivre et d'indium

Ce matériau est déposé sur la cellule en couche mince par des méthodes chimiques ou par un dépôt sous vide, sur une couche de sulfure de cadmium (Cd S). Cependant la toxicité du cadmium pose un problème une fois la cellule en fin de vie.

I.5.4.4. Cellules multi jonctions (matériaux III-V)

Les cellules multi jonctions sont composées d'un grand nombre de semi-conducteurs (GaAs, Ge, GaInP2...), absorbant chacun dans un spectre limité. Cette technologie permet de dépasser le plafond théorique du rendement d'une cellule photovoltaïque au-dessus de 33,5%, appelé « limite de Shockley-Queisser ».

I.5.4.5. Cellules organiques

Les cellules organiques sont composées de molécules organiques pour transformer la lumière en électricité.

Il en existe principalement de trois types:

-) Les cellules photovoltaïques organiques moléculaires.
-) Les cellules photovoltaïques organiques à base de polymères.
-) Les cellules photovoltaïques hybrides.

I.5.4.6. Cellules à particules quantiques

Les cellules à particules (ou points) quantiques sont des cellules composés d'un nanocrystal semi-conducteur. Les particules quantiques ont l'avantage d'avoir une largeur de bande qui peut

être réglée en changeant simplement la taille des nanoparticules, de sorte qu'ils peuvent facilement être conçus pour absorber les différentes parties du spectre solaire.

I.5.4.7. Cellules Cuivre Zinc Étain Soufre et Sélénium (CZTS)

La cellule CZTS est composée d'un absorbant fait de cuivre, de zinc, d'étain, de soufre et de sélénium, de formule $Cu_2ZnSn(S, Se)_4$. L'ajout d'un autre semi-conducteur dopé n'est requis pour former une hétérojonction, comme pour les cellules CIGS. Cette technologie a l'avantage de n'utiliser que d'éléments courants et non toxiques.

Les couches d'une cellule CZTS sont les suivantes:

-) un verre de protection.
-) une couche conductrice transparente d'oxyde de zinc intrinsèque (elle est parfois précédée d'une couche de ZnO dopée à l'aluminium, de 400 nm d'épaisseur).
-) une couche de sulfure de cadmium (CdS) ou d'un composé mixte comprenant du zinc, de l'oxygène, du soufre et des hydroxydes $Zn(S, O, OH)$. D'une épaisseur de 70 nm, ils sont naturellement dopés n.
-) la couche de CZTS (épaisseur d'environ 1,5 μm).
-) un conducteur inférieur, généralement du molybdène (épaisseur de 300 nm).

I.5.4.8. Cellules à colorants

Les [Cellule à colorant cellules à colorants], dites cellules de Grätzel sont des cellules mixtes organiques-inorganiques, constituées d'une matrice poreuse de TiO_2 imprégnée de colorant et d'un électrolyte liquide.

I.5.4.9. Cellules pérovskite

La cellule pérovskite est un type de cellule solaire qui comprend un composé à structure de pérovskite ($CaTiO_3$ - titanate de calcium).

I.5.5. Caractéristiques électriques d'une cellule photovoltaïque

Une cellule photovoltaïque peut être schématisée par un circuit électrique à une ou deux diodes (voir figure.1.3).

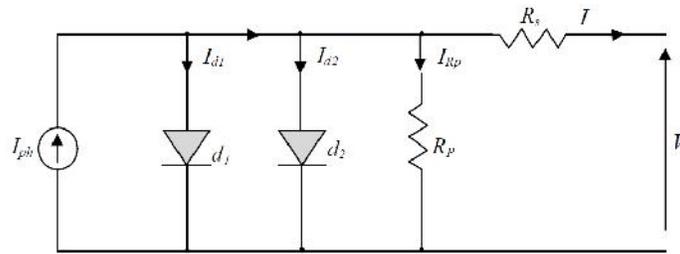


Figure 1.3: Circuit équivalent d'une cellule solaire.

Comme la montre de la figure 1.3, une cellule solaire comporte en réalité une résistance série R_s et une résistance en dérivation ou shunt R_p . Ces résistances ont une certaine influence sur la caractéristique $I = f(V)$ de la cellule:

-) La résistance série est la résistance interne de la cellule, elle dépend principalement de la résistance du semi-conducteur utilisé, de la résistance de contact des grilles collectrices et de la résistivité de ces grilles.
-) La résistance shunt est due à un courant de fuite au niveau de la jonction, elle dépend de la façon dont celle-ci a été réalisée.

D'après la figure 1.3 le modèle mathématique pour la caractéristique courant-tension est donné par:

$$I = I_{ph} - I_{d1} \left[e^{\frac{q(V + I R_s)}{n_1 k T}} - 1 \right] - I_{d2} \left[e^{\frac{q(V + I R_s)}{n_2 k T}} - 1 \right] - \frac{V + I R_s}{R_p} \quad (I.1)$$

I_{d1} et I_{d2} sont les courants de saturation des diodes; n_1 et n_2 les facteurs de pureté de la diode, R_s et R_p sont respectivement la résistance série et la résistance parallèle, et T est la température absolue en Kelvin. L'équation contient également la charge élémentaire constante q ($1,602 \cdot 10^{-19}$.C) et la constante de Boltzmann k ($1,380 \cdot 10^{-23}$ J/K) .Le photo courant $I_{ph,max}$ est atteint à une insolation maximum, souvent nous avons ($I_{ph} = S \cdot I_{ph,max}$) avec S : pourcentage d'insolation. [31]

Il est évident, de l'équation (I.1), que la caractéristique courant-tension dépend fortement de l'insolation et de la température. La dépendance, vis-à-vis de la température, est encore amplifiée par les propriétés du photo-courant I_{ph} et les courants de saturation inverse des diodes qui sont donnés par:

$$I_{ph}(T) = I_{ph}(T_{298K}) \left[\frac{T}{298K} \right]^{\alpha} \quad (I.2)$$

$$I_{d1} = K_1 T^3 e^{-\frac{E_g}{kT}} \tag{I.3}$$

$$I_{d2} = K_2 T^{\frac{5}{2}} e^{-\frac{E_g}{kT}} \tag{I.4}$$

$$K_1 = 1,2 A / cm^2 . K^3 \tag{I.5}$$

$$K_2 = 2,9 . 10^5 A / cm^2 . K^{5/2} \tag{I.6}$$

I.5.5.1. Modèle mathématique d'un module solaire

L'équation suivante montre le modèle mathématique d'un module photovoltaïque, avec N_r cellules photovoltaïques raccordées en série.

$$I = I_{ph} - I_{d1} e^{\frac{q(V - I N_r R_s)}{N_r n_1 k T}} - I_{d2} e^{\frac{q(V - I N_r R_s)}{N_r n_2 k T}} - \frac{V - I N_r R_s}{N_r R_p} \tag{I.7}$$

Ces modules peuvent être alors arrangés en série ou en parallèle pour obtenir la tension et les valeurs de courant désirées pour le système. [31]

I.5.6. Rendement et puissance crête des cellules photovoltaïques

Le rendement d'une cellule ou d'un module photovoltaïque est le rapport entre l'énergie électrique produite par cette cellule ou module et l'énergie lumineuse reçue sur la surface correspondante :

$$\eta [\%] = P_{produite} [kW] / P_{incidente} [kW]$$

Le rendement réel varie donc continuellement, en fonction notamment de l'énergie solaire incidente. Pour permettre une comparaison de l'efficacité de différentes cellules, on définit ces caractéristiques dans des conditions de test bien précises (STC = Standard Test Conditions). Ces conditions sont : émission lumineuse de 1 000 W/m², température de 25 °C, conditions spectrales Air Mass 1.5 (composition du spectre identique au spectre solaire lorsqu'il traverse une épaisseur et demie d'atmosphère, ce qui correspond à un angle d'incidence de 41.8° par rapport à l'horizontale).

La puissance crête est définie comme la puissance électrique produite par la cellule (ou un panneau) lorsqu'elle est soumise aux conditions STC. Cette valeur est utilisée comme référence pour comparer les panneaux photovoltaïques entre eux. [12]

On obtient alors le rendement par la formule suivante :

$$\eta_{\text{STC}} [\%] = \text{Puissance crête} [\text{W/m}^2] / \text{Puissance du spectre STC} [\text{W/m}^2]$$

Avec :

$$J) \text{ Puissance du spectre STC} = 1\,000 [\text{W/m}^2].$$

I.5.7. Les performances du générateur photovoltaïque

Un module photovoltaïque est constitué d'un ensemble de cellules photovoltaïques élémentaires montées en série et/ou en parallèle afin d'obtenir des caractéristiques électriques désirées tels que: la puissance, le courant de court-circuit I_{cc} ou la tension en circuit ouvert V_{co} . Un générateur photovoltaïque est constitué d'un ou plusieurs modules PV en série et/ou en parallèle pour obtenir une puissance, un I_{cc} et un V_{co} désirés. [12]

I.5.7.1. Caractéristique Courant-Tension

La figure 1.4 représente la courbe $I = f(V)$ d'un module photovoltaïque typique dans des conditions constantes d'irradiation et de température. L'irradiation standard adoptée pour mesurer la réponse des modules photovoltaïques est une intensité rayonnante de 1000 W/m^2 et une température de 25°C .

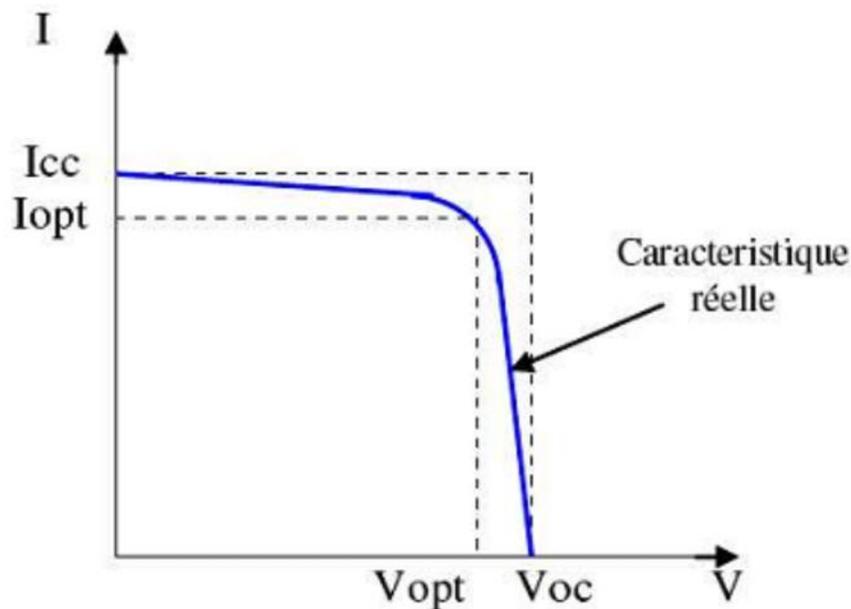


Figure 1.4: Caractéristique $I = f(V)$ d'un module photovoltaïque.

Il est difficile de donner un caractère source de courant ou de tension à un module photovoltaïque sur toute l'étendue de la caractéristique courant-tension. Par conséquent, le module photovoltaïque est considéré comme une source de puissance avec un point P_m où la puissance se trouve être maximale. Il est donc intéressant de se placer sur ce point pour tirer le maximum d'énergie et ainsi exploiter au mieux la puissance crête installée. Il est important de noter que certains régulateurs solaires réalisent une adaptation d'impédance afin qu'à chaque instant on se trouve proche de ce point P_m . [12]

I.5.7.2. Influence de l'éclairement

La luminosité influence considérablement les performances des cellules.

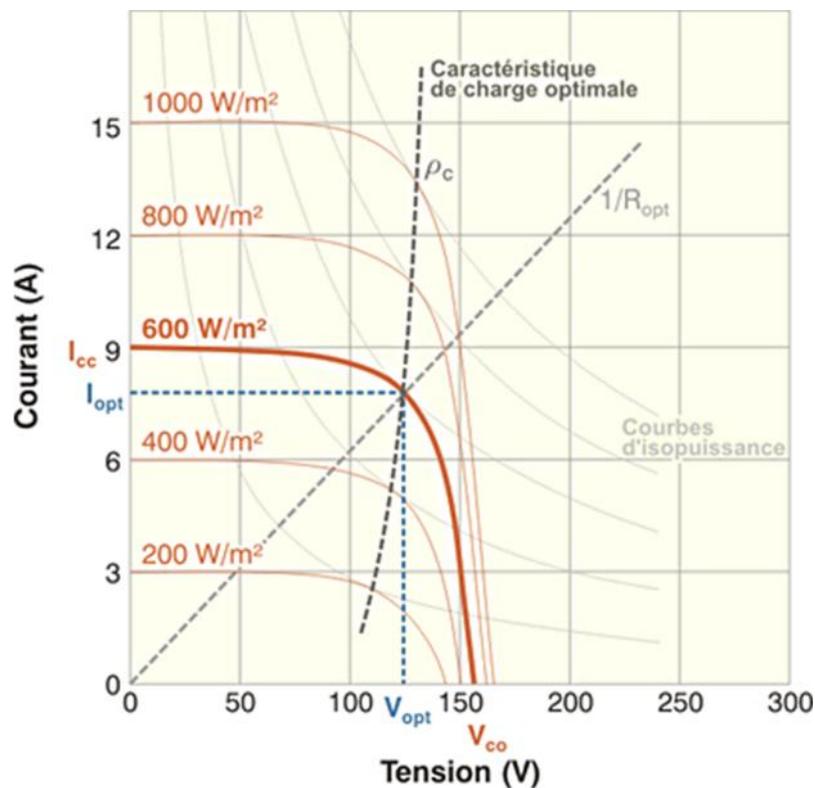


Figure 1.5: Influence de l'éclairement.

Comme le montre ce graphique, le courant de court-circuit (I_{cc}) croît proportionnellement avec l'éclairement, alors que la tension à vide (V_{co}) varie très peu (environ 0,5 V). Ainsi, au plus la couverture nuageuse est importante, au plus l'intensité du courant généré est faible. [12]

I.5.7.3. Influence de la température

La température a une influence considérable sur le comportement de la cellule et donc sur son rendement. Cette influence se traduit principalement par une diminution de la tension générée (et une très légère augmentation du courant). Suivant les modèles, ce comportement induit, par degré, une perte de 0.5 % du rendement par rapport au rendement maximum de la cellule. On comprendra donc tout l'intérêt d'une ventilation correcte à l'arrière des panneaux.[12]

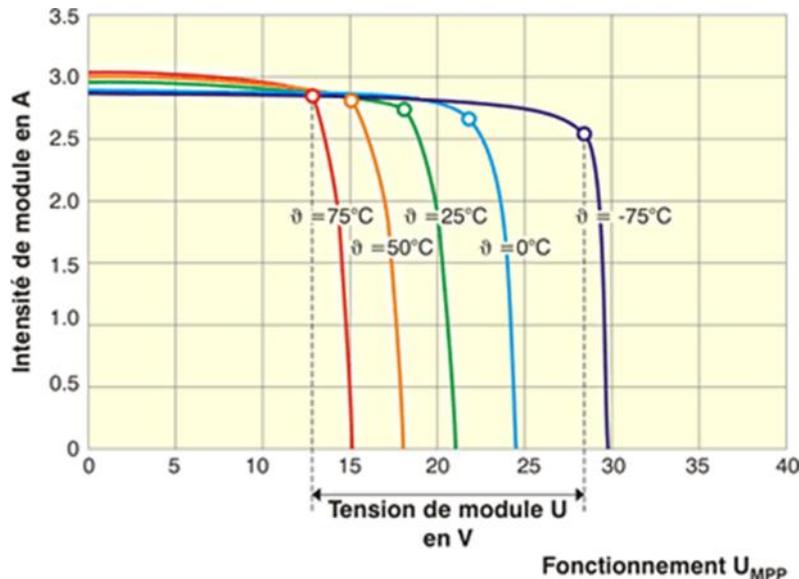


Figure 1.6: Influence de température.

La perte de tension d'un module ou d'une cellule peut être estimée par la formule suivante :

$$U(T^{\circ}) = U(25) + (T^{\circ} * a) \quad (I.8)$$

I.5.7.4. Limite de Shockley-Queisser pour le silicium

La limite de Shockley-Queisser est une limite physique bornant à 30 % le rendement atteignable par les panneaux solaires photovoltaïques courants (composés de cellules en Silicium à jonction simple). Ce plafond de verre provient de deux phénomènes :

-)] Le panneau photovoltaïque ne valorise pas l'entièreté du spectre d'émission solaire (une partie du rayonnement est systématiquement perdue par transformation en chaleur).
-)] La seconde loi de la thermodynamique qui limite le rendement des « machines thermiques » en fonction des températures de la source chaude (soleil) et de la source froide (la cellule).

Actuellement, en fonction des installations et des technologies utilisées, le rendement de cellules se situe en moyenne entre 10 % et 20 % (avec un record en Labo à 26,6 %).

Pour s'affranchir de cette limite, des cellules multi-jonctions voient le jour permettant d'atteindre des plafonds théoriques de l'ordre de 80 % de rendement. Ce type de cellule, en condition laboratoire a déjà dépassé le cap des 40 % de rendement.[12]

I.5.8. Le module photovoltaïque

Afin d'augmenter la tension d'utilisation, les cellules photovoltaïques sont connectées en série. La tension nominale du module est habituellement adaptée à la charge de 12 volts et les modules auront donc généralement 36 cellules. Elles sont généralement encapsulées sous verre ou sous composé plastique. Le tout est appelé un module photovoltaïque.[13]

I.5.8.1 Association en série

En additionnant des cellules ou des modules identiques en série, le courant de la branche reste le même mais la tension augmente proportionnellement au nombre de cellules (modules) en série.

I.5.8.2 Association en parallèle

En additionnant des modules identiques en parallèle, la tension de la branche est égale à la tension de chaque module et l'intensité augmente proportionnellement au nombre de modules en parallèle dans la branche.

I.5.9. La puissance caractéristique d'une cellule PV

Dans des conditions ambiantes de fonctionnement fixes (éclairage, température, vitesse de circulation de l'air ambiant, etc.), [14] la puissance électrique $P(W)$ disponible aux bornes d'une cellule PV:

$$P = V.I \quad (I.9)$$

I.5.9.1. La puissance maximale

Pour une cellule solaire idéale, la puissance maximale $P_{max,idéale}$ correspondrait donc à la tension de circuit ouvert V_{CO} multipliée par le courant de court-circuit I_{CC} :

$$P_{max,idéale} = V_{CO} \cdot I_{CC} \quad (I.10)$$

En pratique, la courbe caractéristique d'une cellule PV est plus "arrondie" (figure 1.9), et la tension au point de puissance maximale V_{PMax} est inférieure à la tension de circuit ouvert V_{CO} , de même que le courant fourni I_{PMax} est inférieur, pour cette même tension, au courant de court-circuit I_{CC} . L'expression de la puissance en ce point est donnée par :

$$P_{Max} = V_{pMax} \cdot I_{pMax} \quad (I.11)$$

I.5.9.2. Le facteur de forme

On appelle facteur de forme f le rapport entre la puissance maximale fournie par la cellule P_{Max} , dans des conditions d'éclairement, de température et de vitesse de circulation de l'air ambiant donnés, et le produit du courant de court-circuit I_{cc} par la tension de circuit ouvert V_{co} (c'est à dire la puissance maximale d'une cellule idéale):

$$f = \frac{P_{Max}}{V_{co} \cdot I_{co}} \quad (I.12)$$

Le facteur de forme f est de l'ordre de 70 % pour une cellule de fabrication industrielle.

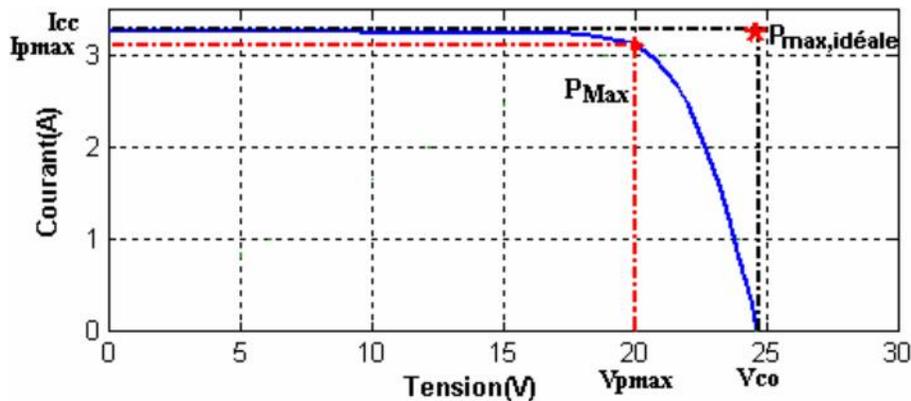


Figure 1.9: Puissances maximale idéale et pratique.

I.6. Conclusion

Dans ce chapitre nous avons présenté le principe de conversion de l'énergie solaire en énergie électrique par le biais des cellules photovoltaïques, ainsi que les différentes configurations des systèmes photovoltaïques. Dans le prochain chapitre, nous allons voir le principe de l'apprentissage profond (Deep learning).

Chapitre 2

Réseaux de neurones apprentissage profonds (DLNN)

II.1. Introduction

L'intelligence artificielle est une discipline scientifique recherchant des méthodes de solution de problèmes à forte complexité logique ou algorithmique. L'apprentissage automatique champ d'étude de l'intelligence artificielle. Par conséquent, L'apprentissage profond (en anglais deep learning, deep structured learning, hierarchical learning) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Dans ce chapitre nous allons présenter tout d'abord les notions en relation avec l'apprentissage profond.

L'apprentissage en profondeur est un nouveau domaine de recherche du l'apprentissage automatique (ML), qui a été introduit dans le but de rapprocher le ML de son objectif principal : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

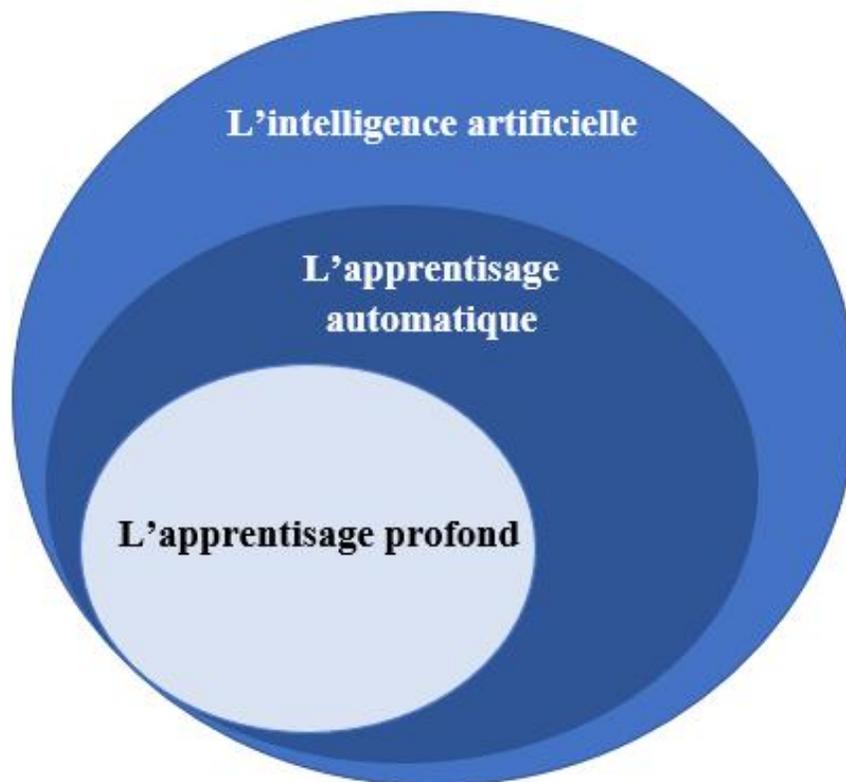


Figure 2.1: La relation entre l'intelligence artificielle, l'apprentissage automatique et L'apprentissage en profondeur.

II.2. Réseaux de neurones

II.2.1. définition

Les réseaux de neurones, fabriqués de structures cellulaires artificielles, constituent une approche permettant d'aborder sous des angles nouveaux les problèmes de perception, de mémoire, d'apprentissage et de raisonnement (en d'autres termes... d'intelligence artificielle ou abrégée "I.A.") au même titre que les algorithmes génétiques. Ils s'avèrent aussi des alternatives très prometteuses pour contourner certaines des limitations des méthodes numériques classiques. Grâce à leur traitement parallèle de l'information et à leurs mécanismes inspirés des cellules nerveuses (neurones), ils infèrent des propriétés émergentes permettant de solutionner des problèmes jadis qualifiés de complexes. [15]

II.2.2. Bref Historique

-) Le réseau de neurones est apparu dans les années 1940 sous le nom (perceptron monocouche) par les travaux de Warren McCulloch et Walter Pitts qui ont montré que l'on pouvait approximer n'importe quelle fonction arithmétique ou logique, avec des perceptrons multicouche).
-) 1958-1969 : les recherches sur les perceptrons multicouches se sont multipliées avec l'apparition des premiers ordinateurs qui ont permis de faire plusieurs calculs.
-) 1969 : les perceptrons ne peuvent faire que des classifications binaires et par contre ils rencontrent des difficultés avec la classification de la porte logique XOR.
-) 1980 : achèvement de l'algorithme de rétro-propagation par David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams.
-) Yann LeCun a popularisé les réseaux de neurones avec LeNet le premier réseau de neurones convolutif pour la reconnaissance des caractères en 1998. [16]

II.2.3. Fonctionnement

Au point de vue fonctionnel, il faut considérer le neurone comme une entité polarisée, c'est-à-dire que l'information ne se transmet que dans un seul sens : des dendrites vers l'axone.

Le neurone va donc recevoir des informations, venant d'autres neurones, grâce à ses dendrites. Il va ensuite y avoir sommation, au niveau du corps cellulaire, de toutes ces informations et via un potentiel d'action (un signal électrique) le résultat de l'analyse va transiter le long de l'axone jusqu'aux terminaisons synaptiques. A cet endroit, lors de l'arrivée du signal, des vésicules synaptiques vont venir fusionner avec la membrane cellulaire, ce qui va permettre la libération des neurotransmetteurs (médiateurs chimiques) dans la fente synaptique. Le signal électrique ne pouvant pas passer la synapse (dans le cas d'une synapse chimique), les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre. [17]

II.2.4. Neurone biologique

Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone. [18]

Bref un neurone biologique est une cellule qui se caractérise par (Figure 2.2) :

- Des synapses, les points de connexion avec les autres neurones, Fibres Nerveuses ou musculaires.
- Des dendrites ou entrées des neurones.
- Les axones, ou sorties du neurone vers d'autres neurones ou fibres musculaires.

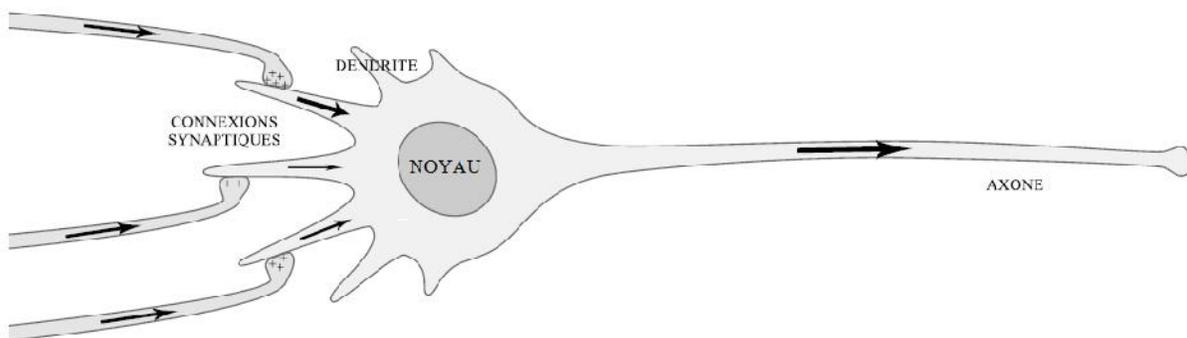


Figure 2.2: Structure d'un neurone biologique.

II.2.5. Neurone formel

Un neurone formel est un modèle mathématique. Il est utilisé dans le cadre des recherches sur l'intelligence artificielle, principalement en association avec d'autres neurones formels pour former un réseau de neurones. [19]

Il contient plusieurs formules mathématiques afin de reproduire le plus fidèlement possible le fonctionnement d'un neurone avec ses différentes entrées (dendrites), leur importance (coefficient de pondération) et sa sortie (axone) et ainsi comprendre son potentiel d'interaction avec les autres neurones (Figure 2.3).

Il est caractérisé par :

- Des entrées x_i et leur poids w_i synaptique.
- Une fonction f d'activation (ou de transfert).
- Une sortie y .

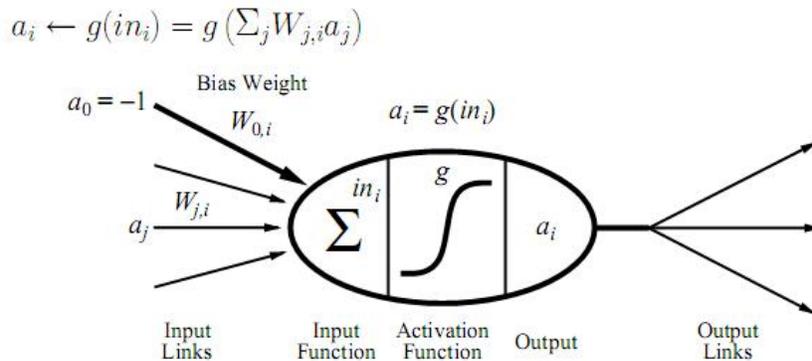


Figure 2.3 : Schématisation d'un neurone formel.

- Les entrées a_j du neurone i proviennent soit d'autres neurones soit de l'environnement.
- Les poids W_{ji} associés à chaque neurone j déterminent l'influence de chaque entrée.
- Le seuil (biais) W_0 permet de contrôler l'entrée de la fonction d'activation g du neurone i .
- La fonction de combinaison permet de calculer une combinaison linéaire des entrées a_j pondérées par les coefficients W_{ji} .
- La fonction d'activation qui permet le passage de la somme pondérée dans la fonction d'activation g et calcule la valeur de sortie du neurone qui sera transmise aux neurones suivants. Ainsi, chaque signal entrant est associé un poids car les synapses n'ont pas toutes la même valeur.

Plusieurs fonctions d'activation g peuvent être utilisées. Toutefois dans la pratique, les deux fonctions de transfert les plus utilisées sont la fonction de Heaviside et la fonction sigmoïde. [19]

Nom de la fonction	Relation d'entrée/sortie	Icône
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
linéaire	$a = n$	
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
sigmoïde	$a = \frac{1}{1+\exp^{-n}}$	
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
compétitive	$a = 1$ si n maximum $a = 0$ autrement	

Tableau 2.1: Types de fonctions de transfert pour R.N.F.

II.3. Réseaux de neurones récurrents [20,21]

La principale caractéristique des Réseaux de Neurones Récurrents, (en anglais Recurrent Neural Networks (RNN)), est l'existence de cycles dans le graphe orienté d'interconnexions des neurones qui représente l'architecture du réseau. La propagation du signal n'est plus unidirectionnelle (de l'entrée vers la sortie), contrairement aux réseaux à propagation avant. La conséquence immédiate est l'introduction du facteur temps dans le fonctionnement du réseau, notamment dans l'algorithme d'apprentissage associé. En effet, le passage du signal d'un neurone à un autre prend un pas de temps pour les connexions simples. L'avantage de ce type d'architectures est que le réseau réalise une mémoire interne qui peut remplacer la fenêtre temporelle utilisée dans les FFNN [20]. Nous passerons rapidement des exemples d'architecture de réseaux récurrents (Figure 2.4).

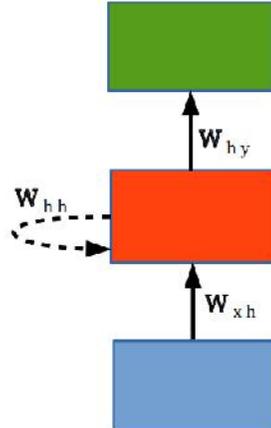


Figure 2.4: Représentation compacte des RNN. Toutes les flèches représentent des connexions complètes. La déclenche en pointillée représente les connexions ayant un décalage temporel ($t - 1$).

Contrairement aux MLP, les réseaux de neurones récurrents présentés dans la figure 2.5, comportent des cycles au sein du graphe de neurones (Elman, 1990). La motivation principale derrière ce type d'architectures est de pouvoir manipuler des séquences de vecteurs d'entrée, représentant chacun un événement temporel, et non pas seulement des données isolées n'ayant pas de signification temporelle. En déroulant, par rapport au temps, la modélisation compacte d'un RNN (figure 2.5), ce type de réseaux peut ainsi être considéré comme une suite temporelle de réseaux MLP reliés entre eux à travers leurs couches cachées respectives. Cette liaison permet aux RNN d'encoder des dépendances latentes entre les événements d'une séquence de vecteurs d'entrée. Suivant cette modélisation, un RNN prend en entrée une séquence d'événements $x = (x_1, x_2, \dots, x_T)$ et définit la séquence d'états cachés $h = (h_1, h_2, \dots, h_T)$ pour produire la séquence de vecteurs de sortie $y = (y_1, y_2, \dots, y_T)$ en itérant de $t = 1$ à T :

$$h_t = H(\mathbf{w}_{xh}x_t + \mathbf{w}_h h_{t-1} + b_h) \quad (\text{II.1})$$

$$y_t = \mathbf{w}_{hy}h_t + b_y \quad (\text{II.2})$$

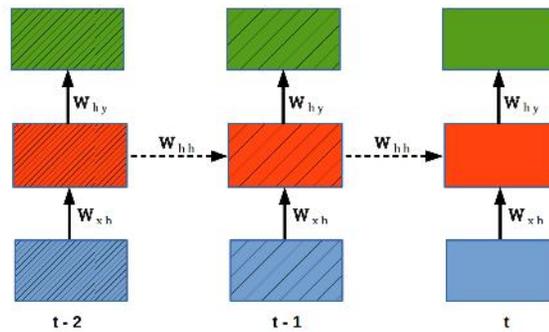


Figure 2.5: Représentation dépliée des RNN.

où T est le nombre total de vecteurs d'entrée, w_{α} est la matrice de poids entre les couches α et β , et b_{β} est le vecteur de biais de la couche β . La fonction H utilisée dans le cas des RNN est généralement la tangente hyperbolique (\tanh). Étant conçu pour les réseaux non bouclés, l'algorithme de rétropropagation du gradient n'est pas suffisant pour la prise en compte des liaisons temporelles exprimées à travers les formules (II.1) et (II.2). Une solution à ce problème consiste à considérer une représentation dépliée « hiérarchisée » des RNN. Dans la schématisation offerte dans la figure 2.6, l'échelle temporelle, représentée à travers les arcs diagonaux, porte maintenant une signification hiérarchique dans le sens où les couches cibles sont de plus hauts niveaux. C'est à travers cette représentation hiérarchisée que nous pouvons utiliser l'algorithme de rétropropagation du gradient généralisé aux réseaux de neurones non bouclés. Cette version est appelée rétropropagation du gradient à travers le temps (Backpropagation Through Time ou BPTT). La particularité de cette représentation par rapport à un réseau de neurones non bouclé « classique » est l'existence de plusieurs paramètres partagés. Par exemple, une matrice de poids commune w_h transmet l'information à travers les arcs diagonaux. En outre, les matrices de poids w_{xh} et w_{hy} , représentées par les arcs verticaux, sont partagées respectivement à travers le temps.

En se basant sur ce qui précède, l'algorithme d'apprentissage basé sur la BPTT comporte lui aussi 3 étapes :

- 1) **Propagation vers l'avant :** L'information circule comme dans un réseau non bouclé normal, du bas vers le haut. À chaque instant t (variant de 1 à T), la valeur de l'état caché à l'instant précédent h_{t-1} ainsi que le vecteur d'entrée de l'instant t (x_t) servent à déterminer le nouveau état caché h_t (voir la formule (II.1)). À partir

de ce dernier est calculé le vecteur de sortie y_t (voir la formule (II.2)).

J) **Propagation vers l'arrière :** Pour chaque instant $t \in [1..T]$, le terme $\Delta_i^L(t)$ du nœud de sortie i est déterminé comme suit :

$$\Delta_i^L(t) = (r f_i(t) - a_i^L(t)) \frac{\partial (a_i^L(t))}{\partial I_i^L(t)} \tag{II.3}$$

où $r f_i(t)$ et $a_i^L(t)$ sont les sorties, respectivement attendues et produites, du nœud i de la couche de sortie de l'instant t .

Ensuite, pour chaque instant t (variant de T à 1), soit Δ_j le terme d'erreur correspondant au nœud j du niveau supérieur (selon la hiérarchie de la figure 2.6),

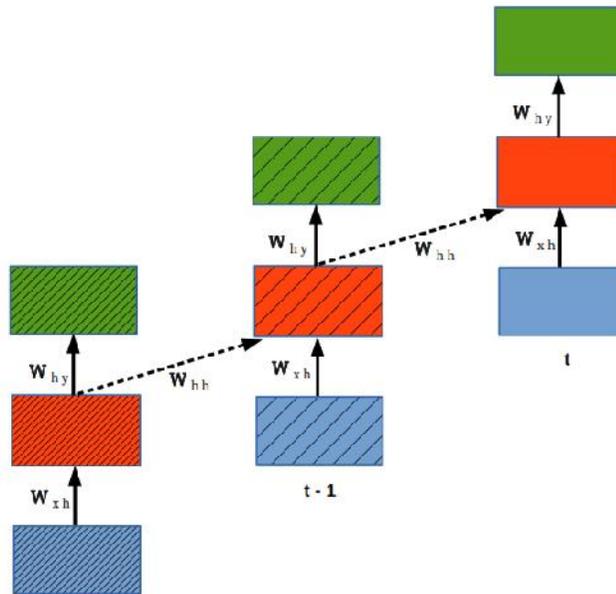


Figure 2.6: Représentation dépliée hiérarchisée des RNN.

C'est-à-dire, appartenant à la couche de sortie de l'instant t ou à la couche cachée de l'instant $t+1$. Chaque élément w_i de la matrice w^h ou de la matrice w^y représente le poids de la connexion partant du neurone i de la couche cachée de l'instant t vers le neurone j . Le terme $\Delta_i^h(t)$ du neurone i est calculé comme suit :

$$\Delta_i^h(t) = \left(\sum_j \Delta_j w_{ij} \right) \frac{\partial (a_i^h(t))}{\partial I_i^h(t)} \tag{II.4}$$

- J) **Adaptation des paramètres.** Pour chaque neurone j dans une couche cachée ou de sortie l qui reçoit des connexions venant des neurones i de la couche k , mettre à jour le biais et les poids comme suit :

$$b_i^l * = b_i^l + \alpha \sum_{\tau=0}^n \Delta_i^k(\tau) \quad (\text{II.5})$$

$$w_i^l * = w_i^l + \alpha \sum_{\tau=0}^t \Delta_i^k(\tau) a_j^l(\tau) \quad (\text{II.6})$$

où α est le taux d'apprentissage, $\Delta_i^k(\tau)$ est l'erreur du nœud i de la couche k de l'instant τ et $a_j^l(\tau)$ est l'activation de l'unité j de la couche l de l'instant τ .

II.3.1. Problématique

Les réseaux récurrents RNN sont des réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches. En cela, ils sont plus proches du vrai fonctionnement du système nerveux, qui n'est pas à sens unique. Ces réseaux possèdent des connexions récurrentes au sens où elles conservent des informations en mémoire : ils peuvent prendre en compte à un instant t un certain nombre d'états passés. Pour cette raison, les RNNs sont particulièrement adaptés aux applications faisant intervenir le contexte, et plus particulièrement au traitement des séquences temporelles comme l'apprentissage et la génération de signaux, c'est à dire quand les données forment une suite et ne sont pas indépendantes les unes des autres. Néanmoins, pour les applications faisant intervenir de longs écarts temporels (typiquement la classification de séquences vidéo), cette « mémoire à court-terme » n'est pas suffisante. En effet, les RNNs « classiques » (réseaux de neurones récurrents simples ou Vanilla RNNs) ne sont capables de mémoriser que le passé dit proche, et commencent à « oublier » au bout d'une cinquantaine d'itérations environ.

Ce transfert d'information à double sens rend leur entraînement beaucoup plus compliqué, et ce n'est que récemment que des méthodes efficaces ont été mises au point comme les LSTM (Long Short Term Memory). Ces réseaux à large « mémoire court-terme » ont notamment révolutionné la reconnaissance de la voix par les machines (Speech Recognition) ou la compréhension et la génération de texte (Natural Language Processing). D'un point de vue théorique, les RNNs ont un potentiel bien plus grand que les réseaux de neurones classiques : des recherches ont montré qu'ils sont « Turing-complet » (ou Turing-complète), c'est à dire qu'ils permettent théoriquement de

simuler n'importe quel algorithme. Cela ne donne néanmoins aucune piste pour savoir comment les construire pour cela dans la pratique. [22]

II.4. RÉSEAU DE NEURONES PROFONDS

Ces réseaux de neurones multicouches peuvent comporter des millions de neurones, répartis en plusieurs dizaines de couches. Ils sont utilisés en apprentissage profond pour concevoir des mécanismes d'apprentissage supervisés et non supervisés.

Dans ces architectures mathématiques, chaque neurone effectue des calculs simples mais les données d'entrées passent à travers plusieurs couches de calcul avant de produire une sortie. Les résultats de la première couche de neurones servent d'entrée au calcul de la couche suivante et ainsi de suite. Il est possible de jouer sur les différents paramètres de l'architecture du réseau : le nombre de couches, le type de chaque couche, le nombre de neurones qui composent chaque couche...

Le perceptron multicouche, capable de traiter des phénomènes non linéaires, est un exemple de ce type de réseau. Concrètement, les premières couches permettront d'extraire des caractéristiques simples (comme des contours) que les couches suivantes combineront pour former des concepts de plus en plus complexes et abstraits : assemblages de contours en motifs, de motifs en parties d'objets, de parties d'objets en objets, etc.

L'espoir est que plus on augmente le nombre de couches, plus les réseaux de neurones apprennent des choses compliquées, abstraites, correspondant de plus en plus à la manière dont un humain raisonne. Il est néanmoins difficile de mettre au point des mécanismes d'apprentissage efficaces pour chacune des couches intermédiaires (appelées couches profondes ou couches cachées). La solution consiste à allier apprentissage supervisé (comme cela est fait pour un réseau à deux couches) et un apprentissage non-supervisé, utilisant des exemples dont la valeur de sortie est inconnue.(inspiré d'un article d'Interstices).

II.4.1. Long Short-Term Memory (LSTM)

Les réseaux de mémoire à long terme à court terme généralement appelés simplement LSTM, sont un type spécial de RNN. Ils ont été introduits par Hochreiter Schmidhuber (1997). Les Réseaux neuronaux récurrents présentés dans la section précédente sont capables d'apprendre des règles de mise à jour de séquence arbitraire en théorie. Dans la pratique, cependant, ces modèles oublient généralement rapidement le passé. C'est ce qu'on appelle le problème de la disparition de gradient et c'est pourquoi ils ont inventé le LSTM. La cellule LSTM est une adaptation de la couche

récurrente qui permet aux signaux plus anciens des couches profondes de se déplacer vers la cellule du présent (figure.2.7). [23]

La figure suivante représente une chaîne de trois cellules LSTM :

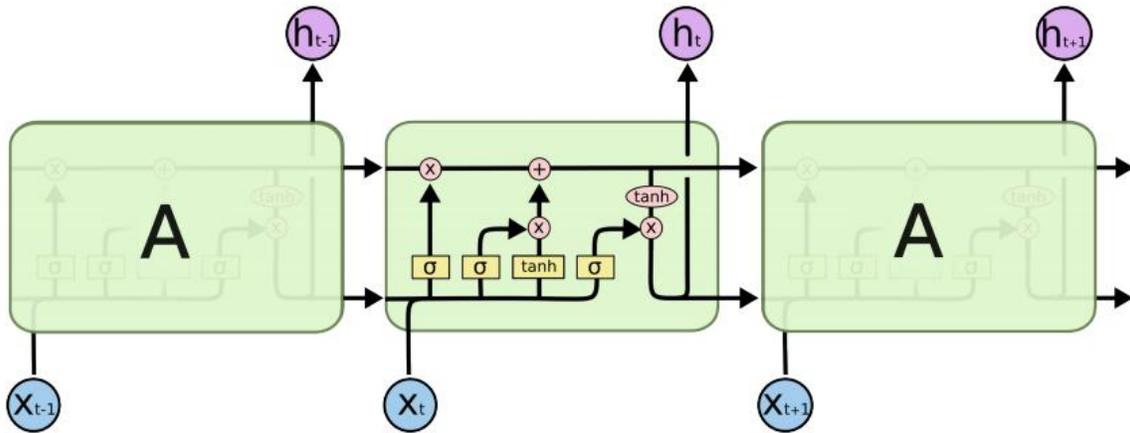


Figure 2.7: Une chaîne de cellules LSTM.

II.4.1.1. La cellule LSTM pas à pas

Dans le paragraphe suivant, nous allons essayer de comprendre pas à pas les opérations réalisées par une cellule LSTM. [24]

1 – La $t^{\text{ème}}$ cellule LSTM prend 3 vecteurs en entrée :

-)] L'élément courant de la séquence \mathbf{x}_t : représentation vectorielle du $t^{\text{ème}}$ caractère (vecteur de taille \mathbf{M}).
-)] Le hidden state de la cellule précédente \mathbf{h}_{t-1} (vecteur de taille \mathbf{R}).
-)] Le cell state de la cellule précédente \mathbf{c}_{t-1} (vecteur de taille \mathbf{R}).

C'est ce dernier vecteur que nous allons suivre particulièrement. Il s'agit d'une route privilégiée de transmission d'information sur la séquence. Pour éviter le problème du vanishing gradient, le cell state est en effet mis à jour de façon **additive** à chaque étape, sans passer par une activation.

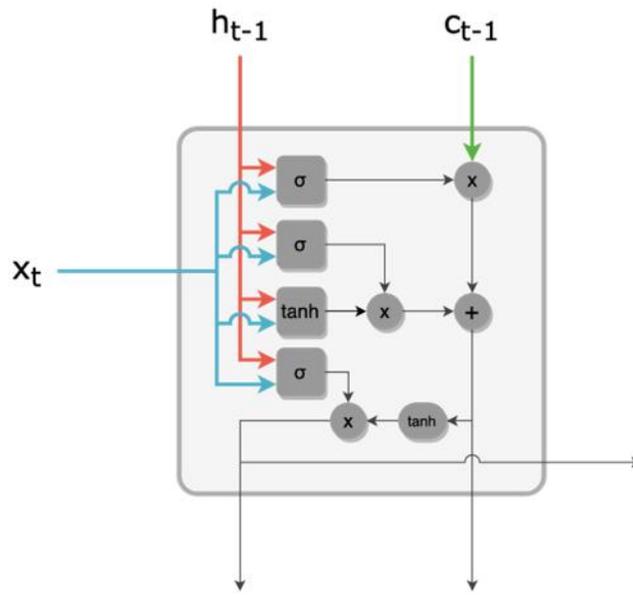


Figure 2.8: Entrées d’une cellule LSTM : input, hidden state, cell state.

2 – La **forget gate** est une couche dense de taille \mathbf{R} avec une activation **sigmoïde**. À partir de h_{t-1} et x_t , cette forget gate produit un vecteur f_t de taille \mathbf{R} , dont les valeurs sont comprises entre 0 et 1.

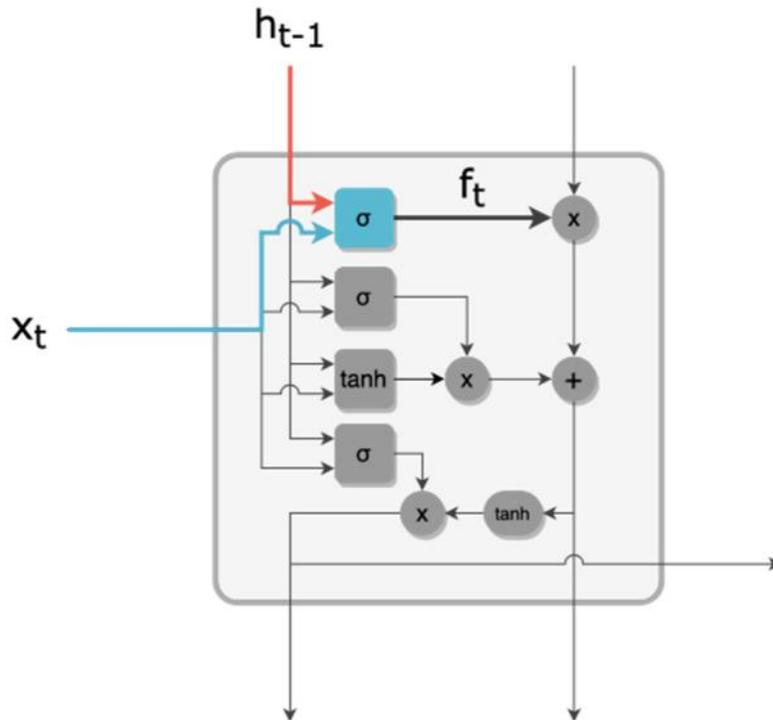


Figure 2.9: Filtre de la forget gate.

3 – La forget gate agit comme un filtre pour « oublier » certaines informations du **cell state**. En effet, on effectue une multiplication terme à terme entre f_t et c_{t-1} , ce qui a tendance à annuler les composantes de c_{t-1} dont les homologues côté f_t sont proches de 0. On obtient alors un cell state *filtré*, toujours de taille R .

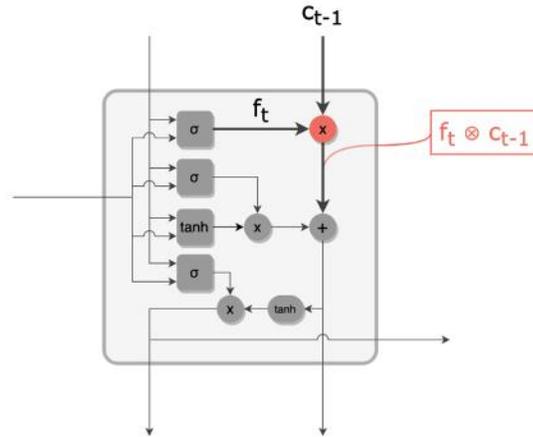


Figure 2.10: Cell state filtré par la forget gate.

4 – L'**input gate** produit un filtre i_t de valeurs comprises entre 0 et 1 et de taille R , à partir de h_{t-1} et x_t , de façon similaire à la forget gate.

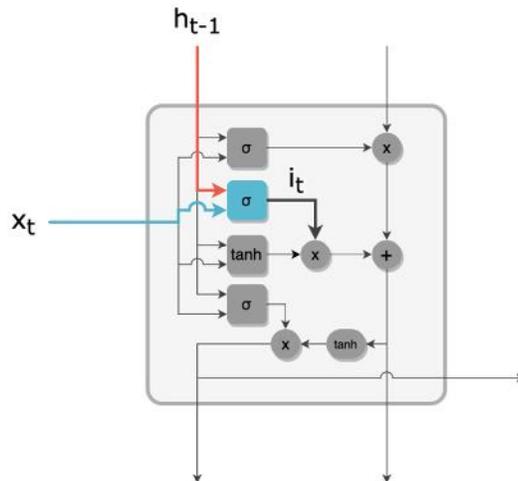


Figure 2.11: Filtre de l'input gate.

5 – En parallèle, un vecteur k_t de taille R est créé par une couche tanh. k_t est le vecteur candidat pour **mettre à jour** le cell state.

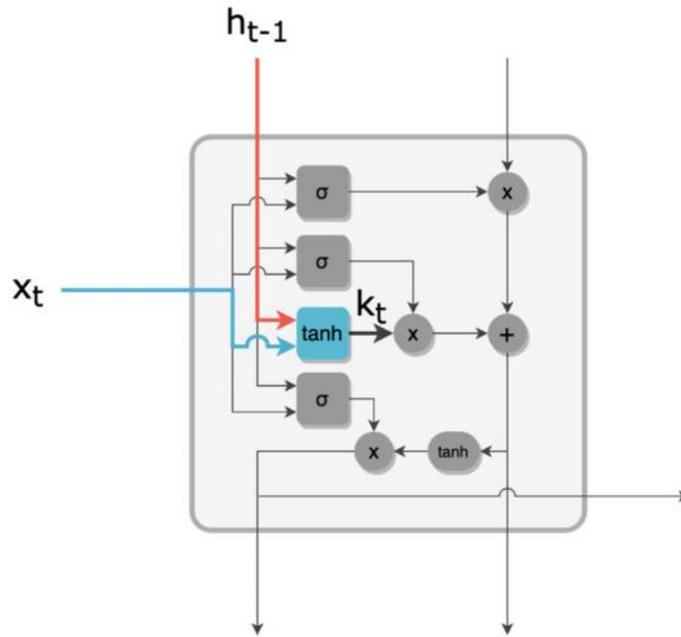


Figure 2.12: Vecteur candidat à la mise à jour du cell state.

6 – Le candidat k_t est filtré par l’input gate i_t via une multiplication terme à terme. On obtient le vecteur de mise à jour du cell state.

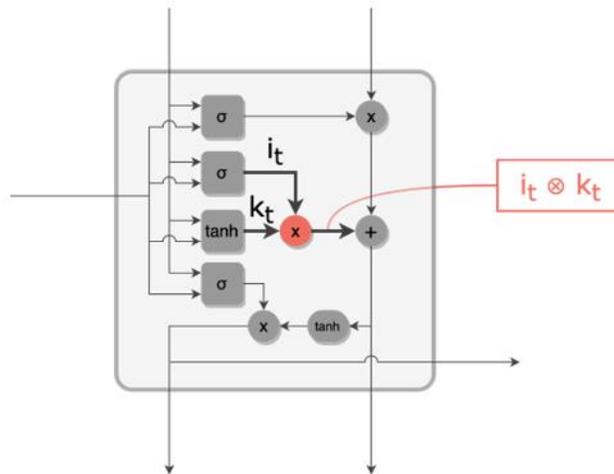


Figure 2.13: Vecteur candidat filtré par l’input gate.

7 – Le cell state filtré (obtenu à l’étape 3) est mis à jour grâce au vecteur candidat filtré (obtenu à l’étape 6). La mise à jour est une simple addition terme à terme de ces deux vecteurs. On obtient alors le **nouveau cell state** c_t .

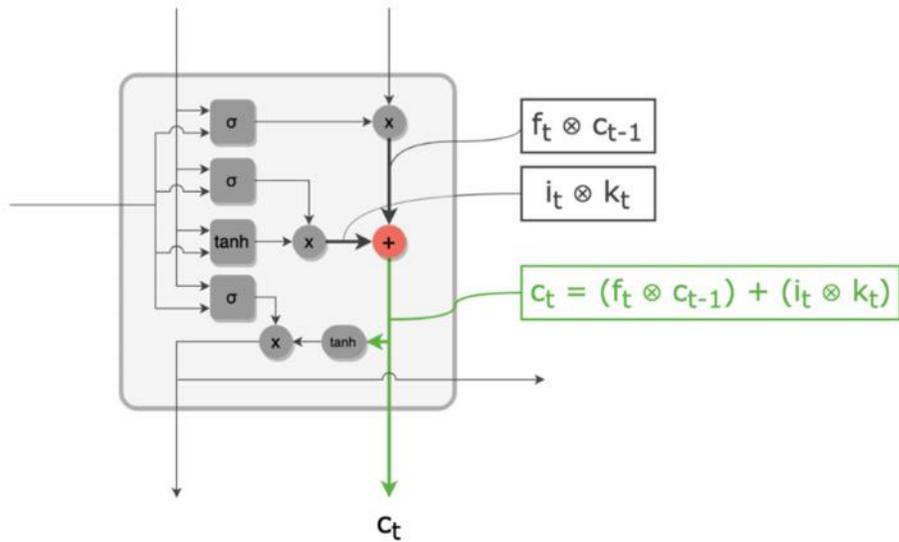


Figure 2.14: Mise à jour du cell state par addition du vecteur candidat filtré.

8 – De façon analogue à f_t et i_t , l’output gate produit un filtre o_t de valeurs entre 0 et 1, et de taille R .

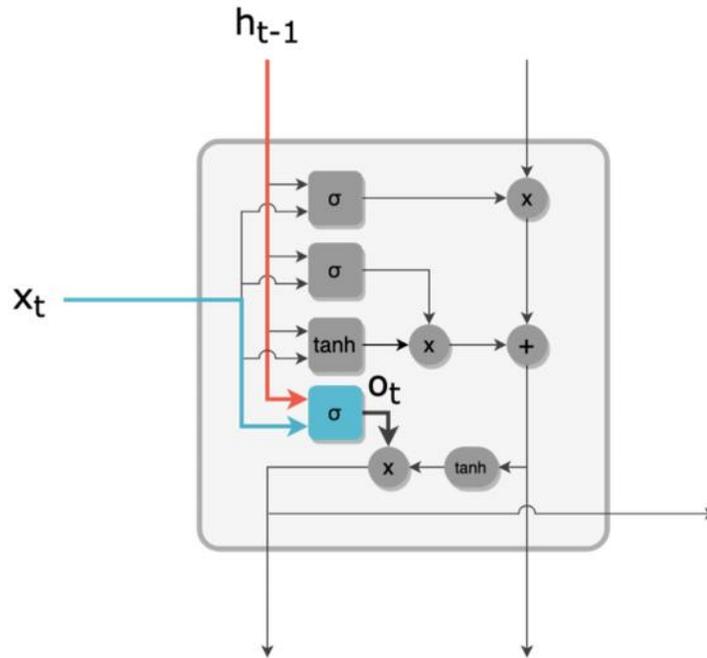


Figure 2.15: Filtre de l’output gate.

9 – Les valeurs du **nouveau cell state** c_t sont ramenées à l’intervalle $]-1, 1[$ par une activation \tanh . Un filtrage par l’output gate o_t est ensuite effectué pour enfin obtenir la sortie h_t .

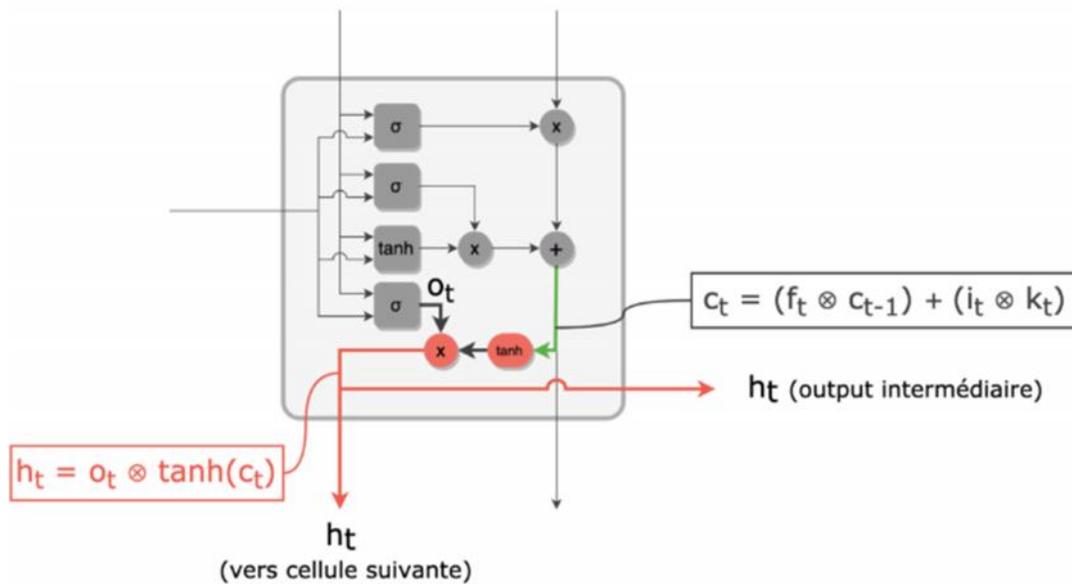


Figure 2.16: Nouveau hidden state.

II.4.2. Gated Recurrent Units (GRU)

II.4.2.1. Définition

Les cellules GRU (Gated Recurrent Unit) permettent au réseau récurrent de sauvegarder plus d'informations historiques pour une meilleure prédiction. Il introduit un update gate qui détermine la quantité d'informations à garder du passé ainsi qu'un reset gate qui définit la quantité d'informations à oublier. Le graphique ci-dessous schématise la cellule GRU. [25]

Équations : Nous définissons les équations dans la cellule GRU comme suit:

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \quad (\text{II.7})$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \quad (\text{II.8})$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * \phi_h(W_h x_t + U_z(r_t * h_{t-1}) + b_z) \quad (\text{II.9})$$

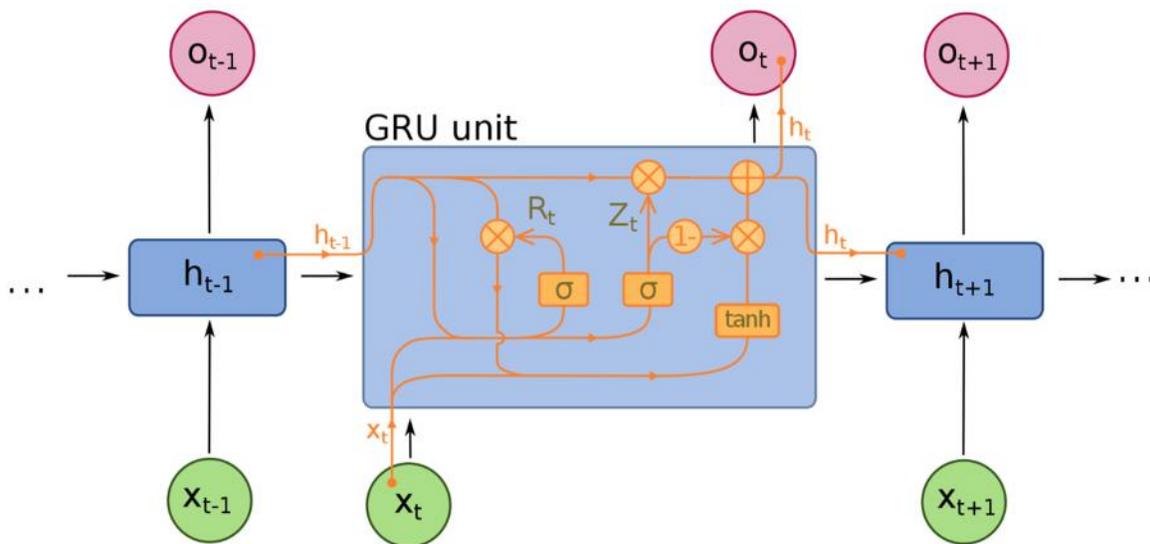


Figure 2.17: Schéma d'un réseau GRU à une unité.

II.4.2.2. Pourquoi est-ce utile?

Un GRU est un mécanisme très utile pour résoudre le problème du gradient de disparition dans les réseaux de neurones récurrents. Le problème du gradient de disparition se produit dans l'apprentissage automatique lorsque le gradient devient extrêmement petit, ce qui empêche le poids de changer sa valeur. Ils ont également de meilleures performances que LSTM lorsqu'il s'agit d'ensembles de données plus petits. [25]

II.4.2.3. Applications d'une unité récurrente fermée

-) Modélisation de la musique polyphonique.
-) Modélisation du signal vocal.
-) Reconnaissance de l'écriture manuscrite.

II.4.3. Réseau de neurone convolutif (CNN)

Un réseau de neurone convolutif est un type particulier de réseau de neurone qui se base sur l'opération de convolution. Les réseaux à convolution sont dérivés des architectures de type perceptron multi-couches (Multi Layer Perceptron : MLP), cependant ils utilisent des poids partagés, liés à la fenêtre de convolution, qui leur permettent une extraction implicite de caractéristiques locales. Les CNN sont particulièrement adaptés à la reconnaissance d'images.

Un réseau est dit convolutif quand chaque neurone reçoit ses informations non pas de toute la couche précédente, mais seulement des neurones situés dans son champ réceptif.[26]

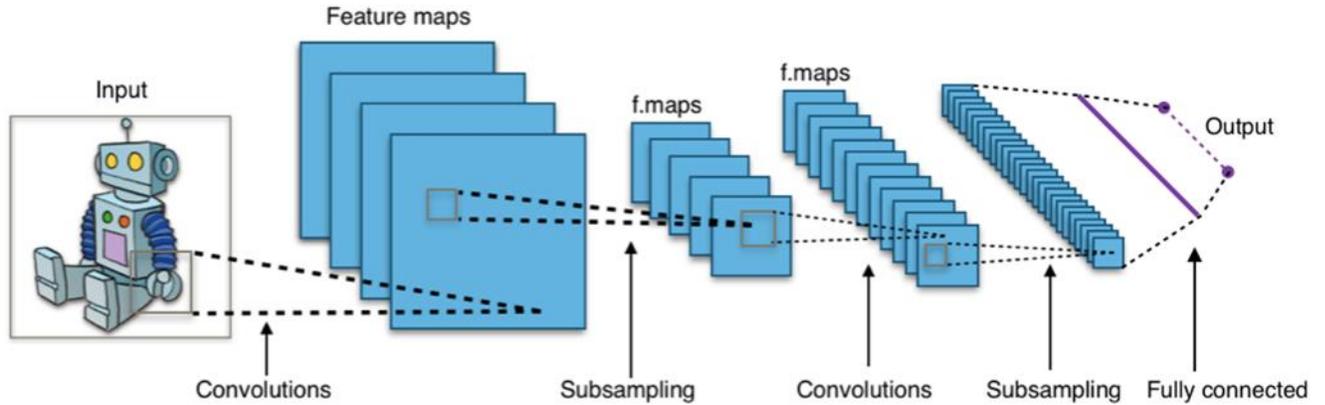


Figure 2.18: Architecture standard d'un réseau à convolutions.

II.5. Conclusion

Dans ce chapitre, nous avons présenté un aperçu générale sur réseaux neuronaux artificiels, Aussi les concepts importants du l'apprentissage profond (définition, structures, etc.) avec un détail sur les différent type de DLNNs choisi dans notre travail. Le chapitre suivant traite quelques détails les outils qui seront utilisés pour réaliser notre travail, tel que la présentation de la plateforme TensorFlow, bibliothèque Keras et l'environnement Anaconda.

Chapitre 3

TensorFlow, Keras et Anaconda

III.1. Introduction

Dans ce chapitre, nous allons présenter les ressources matérielles et logicielles nécessaires afin d'achever notre travail. Une description de langage de programmation Python, la plateforme Tensorflow, suivi par une présentation également des bibliothèques utilisées dans ce travail (exemple Keras pour la machine automatique et l'apprentissage profonds).

III.2. Ressources matérielles

Notre système est développé dans un ordinateur dont les caractéristiques techniques, sont les suivantes :

-) Fabricant du système : TOSHIBA.
-) Modèle de système : TECRA Z40-A.
-) Processeur : Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz (4 CPUs), ~2.7GHz.
-) RAM : 8 Go « DDR3».

III.3. Ressources logicielles

III.3.1. Système d'exploitation

-) Linux Ubuntu version 20.04.

III.3.2. Langage de programmation

Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes. Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement. [27]

On distingue deux versions de Python : **Python 2** et **Python 3**. Les différences entre ces deux versions sont multiples. **Python 2.x** est l'ancienne version, qui continuera d'être supportée et donc de recevoir des mises à jour officielles jusqu'en 2020. Après cette date, elle continuera d'ailleurs sans doute de subsister de façon non officielle. **Python 3.x** est la version actuelle du langage. Elle apporte **de nombreuses fonctionnalités nouvelles et très utiles**, telles qu'un meilleur contrôle de concurrence et un interpréteur plus efficace. Cependant, l'adoption de Python

3 a été longtemps ralentie par le manque de bibliothèques tierces prises en charge. Un grand nombre D'entre elles n'étaient compatibles qu'avec Python 2, ce qui rendait la transition compliquée. Cependant, ce problème est aujourd'hui pratiquement résolu et il reste peu de raisons valables de continuer à utiliser Python 2. [27]

Les principales utilisations de Python par les développeurs sont:

-) la programmation d'applications.
-) la création de services web.
-) la génération de code.
-) le méta programmation.

Techniquement, ce langage servira surtout pour le scripting et l'automatisation (interaction avec les navigateurs web).

Dans le domaine de l'intelligence artificielle, la plupart des développeurs dans le monde ont tendance à utiliser Python par rapport aux autres langages de programmation, et la raison en est que ce dernier présente de nombreux avantages, notamment l'abondance de bibliothèques de programmation que vous pouvez utiliser dans ce domaine comme la bibliothèque Numpy, qui vous aide à résoudre de nombreux comptes, En outre, le peu de temps consacré au développement des technologies d'IA par rapport à d'autres langages tels que Java, par exemple, est bon pour le processus de test d'algorithmes sans avoir à les implémenter.

Dans notre cas, nous utilisons la version du **langage Python 3.7** .



Figure 3.1: Logo Python.

III.3.3. Anaconda

Est une distribution libre et open source des langages de programmation Python et R appliqué au développement d'applications dédiées à la science des données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique), qui vise à simplifier la gestion des paquets et de déploiement. Les versions de paquetages sont gérées par le système de gestion de paquets conda. La distribution Anaconda est utilisée par plus de 20 millions d'utilisateurs, plus de 7500 packages de science des données et d'apprentissage automatique adaptés pour Windows, Linux et MacOS.[28]

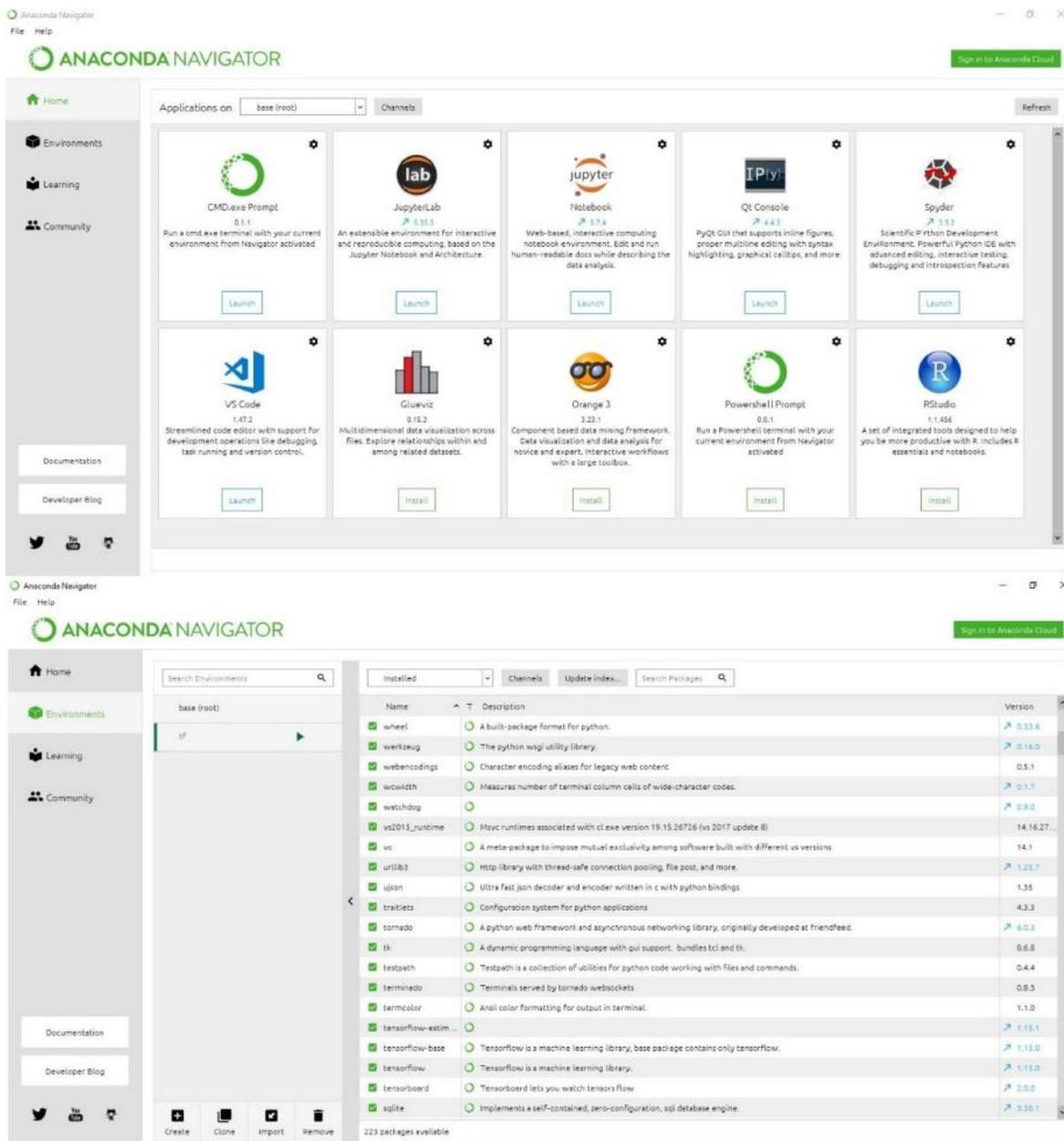


Figure 3.2: Interface Anaconda.

III.3.3.1. TensorFlow

C'est une bibliothèque open-source de logiciels pour le flux de données et la programmation différentielle pour diverses tâches. De même, TensorFlow est utilisé dans la machine Learning par les réseaux de neurones. Développé par Google en 2011 sous le nom de DistBelief, TensorFlow a été officiellement lancé en 2017 gratuitement. La bibliothèque est capable de fonctionner sur plusieurs CPUs et GPUs et est disponible sur plusieurs plates-formes, y compris mobiles. Le nome vient de tableaux multidimensionnels appelés tenseurs, couramment utilisés dans les réseaux de neurones.

Pour la plupart, les applications Tensorflow sont des entreprises d'IA avancées et à grande échelle dans les domaines du machine Learning et du deep Learning. En alimentant le système d'apprentissage automatique Rankbrain de Google, Tensorflow a été employé pour améliorer les capacités de recherche d'information du moteur de recherche phare de l'entreprise. Google a également utilisé le cadre pour les applications qui comprennent la génération automatique de réponse par courriel, la classification des images et la reconnaissance optique des caractères, ainsi qu'une application de découverte de médicaments sur laquelle l'entreprise a travaillé avec des chercheurs de l'Université de Stanford.

D'autres sociétés inscrites sur le site Web Tensorflow en tant qu'utilisateurs du cadre comprennent Airbnb, Coca-Cola, eBay, Intel, Qualcomm, SAP, Twitter, Uber et Snapchat développeur Snap Inc. Un autre utilisateur est STATS LLC, une société de conseil sportif qui gère Tensorflow-basé des modèles d'apprentissage profond pour analyser des choses telles que les mouvements des joueurs pendant les jeux sportifs professionnels. [29]



Figure 3.3: Logo TensorFlow.

III.3.3.2. Keras

C'est une API (application programming interface) de réseaux de neurones de haut niveau, écrite en Python et capable de fonctionner sur TensorFlow ou Theano. Il a été développé en mettant l'accent sur l'expérimentation rapide. Être capable d'aller de l'idée à un résultat avec le moins de délai possible est la clé pour faire de bonnes recherches. Il a été développé dans le cadre de l'effort de recherche du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), et son principal auteur et mainteneur est François Chollet, un ingénieur Google.

En 2017, l'équipe TensorFlow de Google a décidé de soutenir Keras dans la bibliothèque principale de TensorFlow. Chollet a expliqué que Keras a été conçue comme une interface plutôt que comme un cadre d'apprentissage (end to end). Il présente un ensemble d'abstractions de niveau supérieur et plus intuitif et qui facilitent la configuration des RNs indépendamment de la bibliothèque informatique de backend. Microsoft travaille également à ajouter un backend CNTK à Keras aussi.

Keras est le cadre de DL le plus utilisé parmi les 5 meilleures équipes gagnantes sur Kaggle. Parce que Keras facilite l'exécution de nouvelles expériences, il vous permet d'essayer plus d'idées que votre concurrence, plus rapidement. Et c'est ainsi que vous gagnez.

Keras est utilisé par le CERN, la NASA, le NIH et de nombreuses autres organisations scientifiques dans le monde (et oui, Keras est utilisé au LHC). Keras a la flexibilité de bas niveau pour mettre en œuvre des idées arbitraires de recherche tout en offrant des fonctionnalités de haut niveau de commodité en option pour accélérer les cycles d'expérimentation.[30]



Figure 3.4: Logo Keras.

Paramètre de comparaison	Anaconda	Python
Définition	Anaconda est la plate-forme de science des données d'entreprise qui distribue R et Python pour l'apprentissage automatique (Machine Learning) et la science des données (data science).	Python est un langage de programmation général de haut niveau utilisé pour l'apprentissage automatique (Machine Learning) et la science des données (data science).
Catégorie	Anaconda appartient à Data Science Tools.	Python appartient à Computer Languages.
Gestion des paquets	Anaconda a conda a son gestionnaire de paquets . conda permet d'installer Python ainsi que les dépendances de bibliothèque Non-python.	Python a pip comme gestionnaire de paquets. pip permet d'installer toutes les dépendances Python .
Applications utilisateur	Anaconda est principalement conçu pour soutenir les tâches de science des données et d'apprentissage automatique.	Python est non seulement utilisé dans la science des données et l'apprentissage automatique, mais aussi une variété d'applications.

Tableau 3.1: Paramètre de comparaison entre Anaconda et Python.

III.4. Conclusion

Dans ce chapitre, nous avons fourni l'environnement nécessaire pour le fonctionnement des prévisions grâce à l'utilisation de l'environnement basé sur Anaconda Python. Dans le prochain chapitre, nous allons décrire le système de chargement des véhicules électriques ainsi que la présentation de la base de données et les résultats de simulation.

Chapitre 4

Simulation et résultats

IV.1. Introduction

Dans ce chapitre, nous allons faire la mise en œuvre de notre méthode en utilisant le langage de programmation Python. On commençant tout d'abord par une présentation du notre système qui nous fournira les informations nécessaires dans le processus de la prédiction. Ensuite, nous montrerons les résultats et les commentaires.

IV.2. MICROGRID

IV.2.1. Description

La figure 4.1 montre le diagramme de Micro-réseau (MG) installée et la figure 4.2 montre la photo réelle de la MG considérée. Il se compose d'un réseau PV de 4kWp, onduleur hybride de 4,6kVA avec capacité de stockage (batterie li-ion) de 10 kWh, station de charge électrique et autres liens de communication. Les spécifications du réseau PV sont indiquées dans le tableau 4.1.

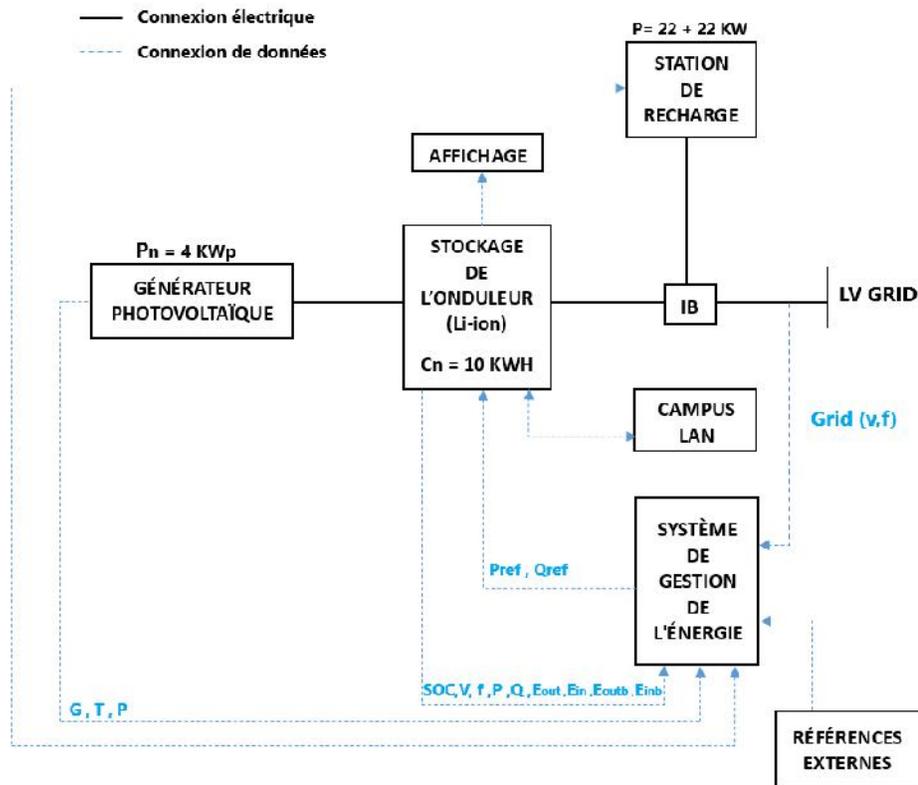


Figure 4.1: Le schéma fonctionnel de la MG disponible à l'université de Trieste, Italie.



Figure 4.2: La station de chargement des véhicules électriques à base d'un MG a l'Université de Trieste, Italie.

La technologie	m-Si
Efficacité minimale	16 %
Puissance nominale	4 kWp
Zone	25 m ²
Angle azimutal	+30°
Angle d'inclinaison	20°
Rendement annuel	4,400 kWh
Nombre de cordes	2
Facteur de puissance	[-0,9 - +0,9]
Efficacité maximale	95 %

Tableau 4.1: Les caractéristiques de panneau solaire PV utilisé.

IV.2.2. Base de données

Les données utilisées sont collectées par un enregistreur de données 'Sonnen' pour une étape de temps min (127.000 échantillons). Un exemple de données mesurées est représenté à la Figure 4.3, y compris la puissance produite par PV, la puissance consommée et l'état de charge.

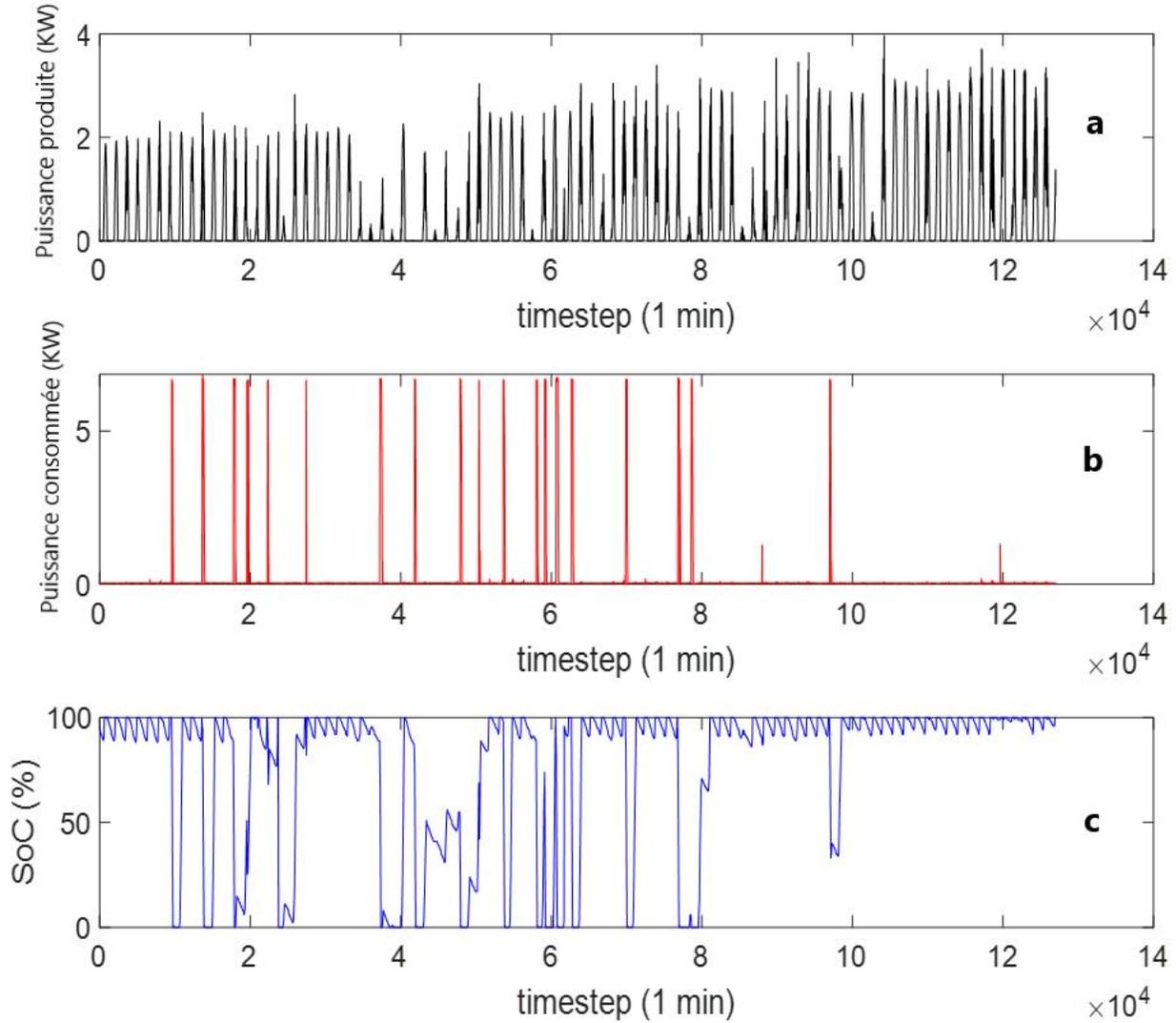


Figure 4.3: Un exemple des données enregistrées du 01/01/2020 au 30/03/2020: (a) la puissance PV produite, (b) la puissance consommée et le (c) l'état de charge (SoC).

IV.3. Méthodologie

Les réseaux de neurones de types LSTM, GRU, Bi-LSTM, Stacked-LSTM et CNN sont utilisées pour développer différents modèles de prédiction de puissance, en se basant sur la stratégie suivante :

$$\hat{x}_{t+1} = f(x_t, x_{t-1}, \dots, x_{t-k+1})$$

Où $t \in \{k, \dots, N - 1\}$, $\{x_t, x_{t-1}, \dots, x_{t-k+1}\}$ sont les valeur réels et précédents de la série temporelle, \hat{x}_{t+1} est la valeur prévoit, f est le modèle, et N est la taille de la base de données.

Par exemple, dans le cas d'un pas en avant avec trois entrées, la relation entre les entrées et les sorties (matrice d'entraînement) peut être formulée comme suit:

$$\begin{bmatrix} \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \\ \hat{x}_7 \\ \vdots \end{bmatrix} = f \begin{bmatrix} x_1 & x_2 & x_3 \\ x_2 & x_3 & x_4 \\ x_3 & x_4 & x_5 \\ x_4 & x_5 & x_6 \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (\text{IV.1})$$

L'ensemble de données présenté ci-dessus ont été divisé en parties entraînement (training) et d'essai (test) de 70 % et de 30 % respectivement. Les données ont été normalisées, sur la base de certaines fonctions disponibles dans la bibliothèque scikit-learn en Python. Diverses simulations ont été construites en variant la taille du patch, les unités, les itérations et le nombre des entrées.

Les mesures d'évaluation courantes, y compris le coefficient de corrélation (R), les erreurs quadratiques moyennes (RMSE) et les erreurs absolues moyennes en pourcentage (MAPE), sont utilisées pour évaluer la précision de ces modèles.

$$R = \frac{\sum_{i=1}^n ((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}} \quad (\text{IV.2})$$

$$R = \sqrt{\frac{1}{n} (\sum_{i=1}^n (x_i - y_i)^2)} \quad (\text{IV.3})$$

$$M = \frac{1\%}{n} \sum_{i=1}^n \left| \frac{x_i - y_i}{x_i} \right| \quad (\text{IV.4})$$

Où x_i et y_i sont les valeurs mesurées et prévoit, respectivement, \bar{x} et \bar{y} sont leur valeurs moyennes.

Les modèles utilisés sont résumés en annexe. Cinq expériences ont été réalisées, en variant les horizons temporels : 1 min, 5 min, 30 min et 1h.

IV.5. Résultats et discussion

Après plusieurs expériences, les structures optimales pour les cinq cas (1 min, 5 min, 30 min et 1 h) sont indiquées dans les tableaux (4.2 et 4.3). Les figures (4.4.a, 4.5.a, 4.6.a et 4.7.a) montrent une superposition entre les puissances mesurées et prévues, tandis que les figures (4.4.b, 4.5.b, 4.6.b et 4.7.b) présentent des graphiques de dispersion (en Anglais: Scatter plots) qui indiquent la corrélation entre les puissances PV mesurées et prévues.

Pour tester variation entre les données mesures et ceux estimés, quelques tests statistiques comme R, RMSE et MAPE ont été présenté dans les Tableaux (4.2 et 4.3). La figure 4.8 montre l'erreur absolue calculée (AE) pour les méthodes étudiées à différents horizons temporels (1 min, 5 min, 30 min et 1 h).

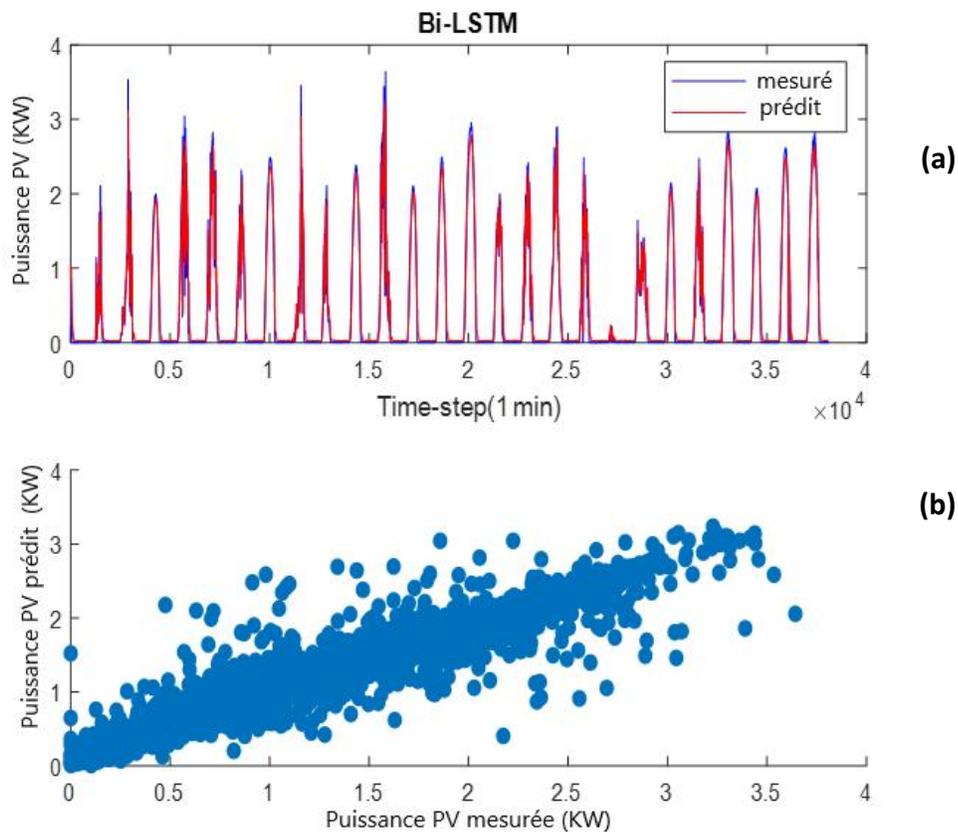


Figure 4.4: Comparaison entre la puissance de sortie PV mesurée et prédit (cas de Bi-LSTM)

a)Superposition plot entre mesuré et prédit, et b)Scatter plot.

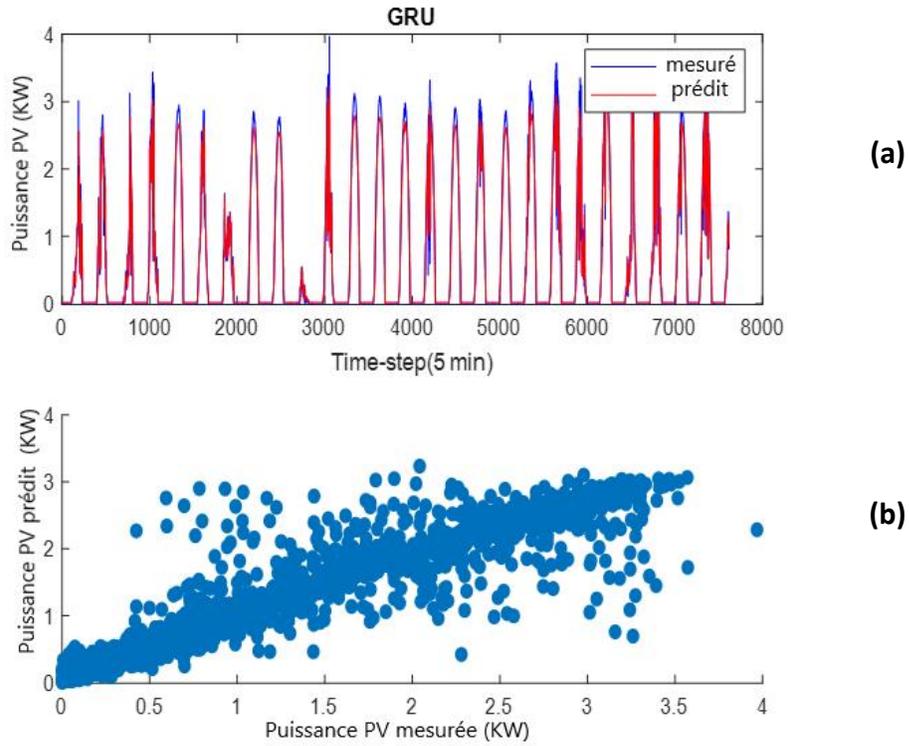


Figure 4.5: Comparaison entre la puissance de sortie PV mesurée et prédit (cas de GRU)
 a) Superposition plot entre mesuré et prédit, et b) Scatter plot.

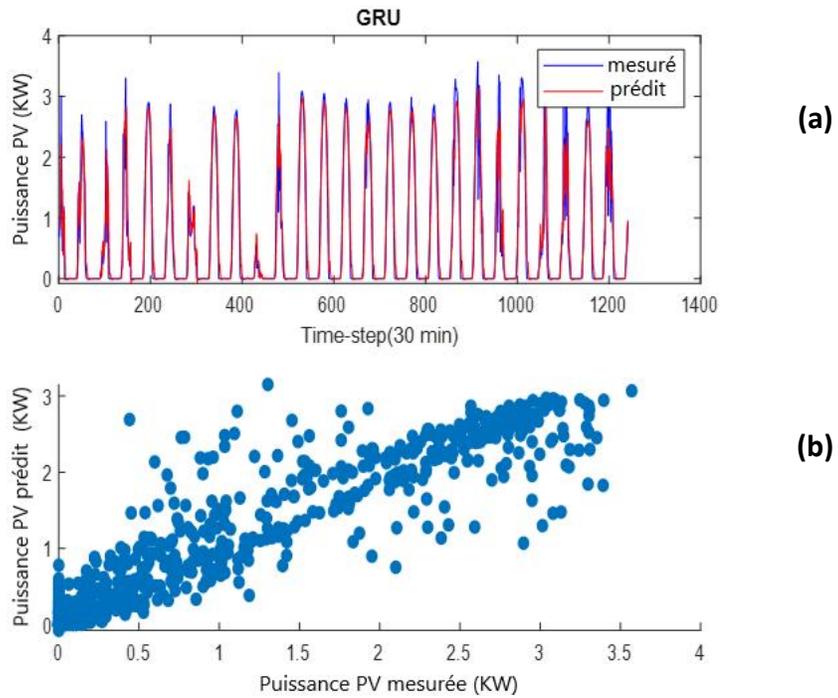


Figure 4.6: Comparaison entre la puissance de sortie PV mesurée et prédit (cas de GRU)
 a) Superposition plot entre mesuré et prédit, et b) Scatter plot.

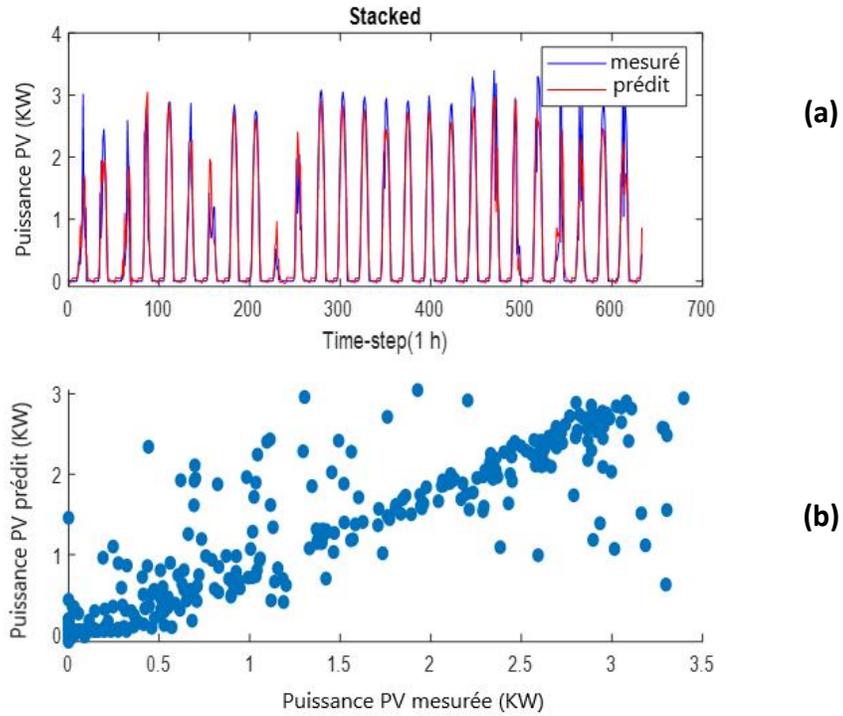


Figure 4.7: Comparaison entre la puissance de sortie PV mesurée et prédit (cas de Stacked)

a) Superposition plot entre mesuré et prédit, et b) Scatter plot.

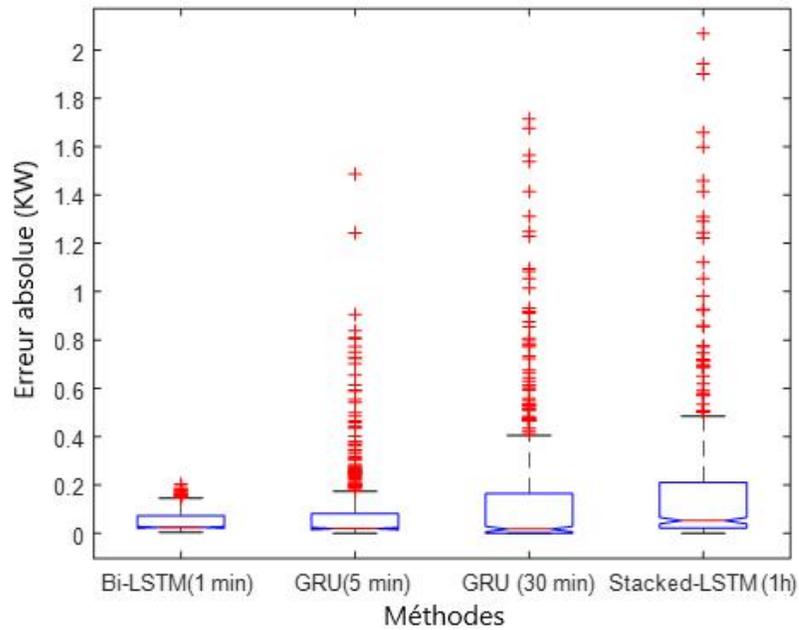


Figure 4.8: L'erreur absolue pour les différentes méthodes étudiées à 1 min, 5 min, 30 min et 1 h. (la couleur rouge indique les valeurs aberrantes, qui sont plus grandes que la valeur maximale, le centre de la boîte est la valeur moyenne).

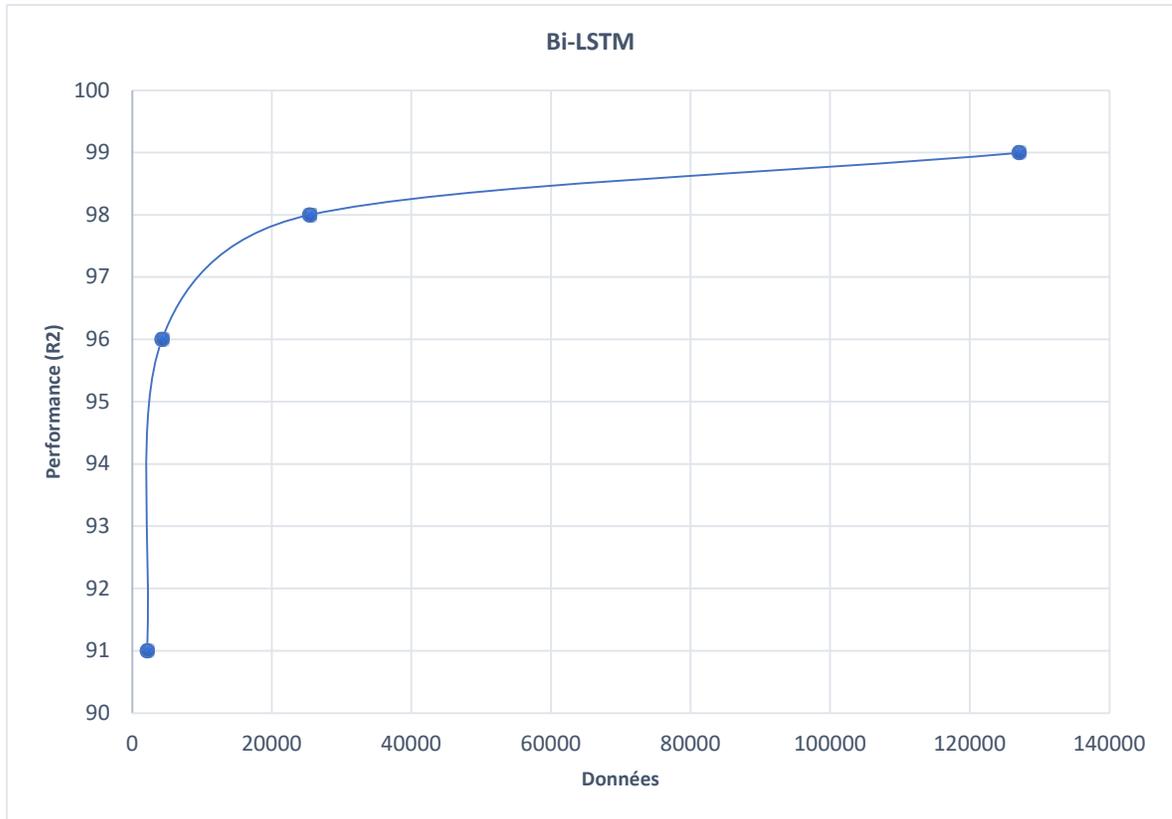


Figure 4.9: Évolution du rendement par rapport à la taille de la base de données (cas de Bi-LSTM).

Considérant que notre travail basé sur quatre types de base des données : 1 min , 5 min , 30 min et 1 h .et après l'application des réseaux de neurones de types LSTM, GRU, Bi-LSTM, Stacked-LSTM et CNN sur les données mentionnées précédemment . La figure 4.8 montre L'erreur absolue pour les différentes méthodes étudiées à les quatres cas de base des données qui utilisé dans notre recherche .

Par exemple dans le cas de 1 min , Nous avons remarqué que L'erreur absolue lors de notre utilisation de **Bi-LSTM** Mieux que les autres modèles . par contre , dans le cas de 5 min et 30 min , l'erreur absolue lors de l'utilisation de **GRU** afficher de meilleurs résultats que les autres modèles. Et enfin , Et enfin, dans la dernier cas (1 h) , l'erreur absolue lors de l'utilisation de **Stacked-LSTM** a donné de meilleurs résultats que ceux enregistrés par les models (LSTM, GRU, Bi-LSTM et CNN).

Models	R ()	RMSE (KW)	MAPE (%)	TEMPS DE FONCTIONNEMENT (min)
Cas 1: 1 min à l'avance				
Premier ensemble de données: Totale =127017, Training (70%)= 88912, Testing (30%)= 38105.				
Structure #1 (Single cell LSTM): Batch size=64, Time steps=10, Epoch=10, units=50.	0.99	0.07	8.48	~07 min
Structure #2 (Stacked LSTM): Batch size=64, Time steps=30, Epoch=10, units=50, 50.	0.99	0.07	6.67	~30 min
Structure #3 (Bi-LSTM): Batch size=64, Time steps=100, Epoch=10, units=16.	0.99	0.09	2.51	~24 min
Structure #4 (GRU): Batch size=64, Time steps=30, Epoch=10, units= 8, 4.	0.99	0.08	11.79	~4 min
Structure #5 (CNN): Batch size=32, Filters=64, Poolsize=2, Time steps=50, Epoch=10, Dense=50, 1.	0.99	0.13	2.60	~5 min
Cas 2: 5 min à l'avance				
Deuxième ensemble de données: Totale =25404, Training (70%)= 17783 , Test (30%)= 7621,2.				
Structure #1 (Single cell LSTM): Batch size=64, Time steps=10, Epoch=30, units=50.	0.98	0.41	15.56	~4 min
Structure #2 (Stacked LSTM): Batch size=64, Time steps=10, Epoch=10, units=50.	0.98	0.20	3.24	~6 min
Structure #3 (Bi-LSTM): Batch size=64, Time steps=100, Epoch=10, units=16.	0.98	0.23	12.81	~6 min
Structure #4 (GRU): Batch size=64, Time steps=30, Epoch=10, units= 8, 4.	0.98	0.20	2.79	~3 min
Structure 5# (CNN): Batch size=32, Filters=64, Poolsize=2, Time steps=100, Epoch =30, Dense=50, 1.	0.97	0.3	18.51	~3 min

Tableau 4.2: Erreurs métriques : R, RMSE et MAPE entre la puissance de sortie PV mesurée et

prédit (Cas 1 min et 5 min à l'avance).

Modèles	R ()	RMSE (kW)	MAPE (%)	TEMPS (min)
Cas 3: 30 min à l'avance				
Troisième ensemble de données: Totale =4234, Training (70%)=2963,8 , Test (30%)=1270,2.				
Structure #1 (Single cell LSTM): Batch size=32, Time steps=10, Epoch=30, Units=50.	0.96	0.45	16.48	~3 min
Structure #2 (Stacked LSTM): Batch size=32, Time steps=50, Epoch=30, Units=50, 50.	0.95	0.33	15.78	~20 min
Structure #3 (Bi-LSTM): Batch size=16, Time steps=100, Epoch=30, Units=128.	0.95	0.32	11.61	~15 min
Structure #4 (GRU): Batch size=32, Time steps=100, Epoch=30, Units= 128, 50.	0.95	0.32	1.60	~16 min
Structure 5# (CNN): Batch size=32, Filters=64, Poolsize=2; Time steps=10, Epoch =10, Dense=50, 1.	0.98	0.23	9.31	~30 s
Cas 4 : 1 h à l'avance				
Quatrième ensemble de données: Totale =2117, Training (70%)= 1481,9 , Test (30%)= 635,1.				
Structure #1 (Single cell LSTM): Batch size=32, Time steps=10, Epoch=30, units=16.	0.91	0.48	10.11	~1 min
Structure #2 (Stacked LSTM): Batch size=32, Time steps=10, Epoch=50, Units=128, 128.	0.93	0.37	6.73	~3 min
Structure #3 (Bi-LSTM): Batch size=32, Time step=100, Epoch=30, Units=50.	0.91	0.42	9.43	~4 min
Structure #4 (GRU): Batch size=32, Time steps=100, Epoch=30, units= 128, 50.	0.92	0.40	4.69	~6 min
Structure 5# (CNN): Batch size=32, Filters=64, Poolsize=2, Time steps=50, Epoch =10, Dense=50, 1.	0.91	0.45	15.87	~15 s

Tableau 4.3: Erreurs métriques : R, RMSE et MAPE entre la puissance de sortie PV mesurée et

prédit (Cas 30 min et 1 h).

À partir des tableaux (4.2 et 4.3) et les images (4.8 et 4.9), l'observation suivante peut être mise en évidence :

- ✓ Pour les deux cas étudiés (1 min et 5 min), le R est proche de 1, ce qui indique une bonne corrélation entre les puissances mesurées et prédit. Alors que dans les cas de 30 min et 1 h le R est compris entre 0,93 et 0,96. Il a également été confirmé que la précision est augmentée avec la taille de l'ensemble de données (Figure 4.9).
- ✓ La variation de l'échelle de temps d'entrée change d'un modèle à un autre, mais en général l'augmentation des étapes de temps peut améliorer relativement la précision. Le RMSE est compris entre 0,09 kW et 0,37 kW, tandis que la valeur inférieure de MAPE est obtenue par le réseau Bi-LSTM dans le cas des prévisions à 1 minute d'avance.
- ✓ Les meilleurs résultats sont obtenus respectivement par Bi-LSTM, GRU, Stacked-LSTM et GRU pendant 1 min, 5 min, 30 min et 1 h (Figure 4.8). Cependant, dans le cas d'un horizon temporel de 30 minutes, CNN a montré un bon coefficient de corrélation par rapport à d'autres structures, il peut être utilisé comme candidat dans ce cas.
- ✓ Augmenter les itérations, à un peu d'impact sur la précision dans le cas de 1 min et 5 min, signifie base de données relativement grande, donc avec 10 époques pour les deux cas 1 min et 5 min de bons résultats peuvent être atteints. Cependant, l'augmentation de l'itération dans les cas de 30 min et 1 h à l'avance avec une base de données limitée peut améliorer la précision. Par exemple, dans le cas des prévisions horaires de 50 à 100 itérations, nous pouvons obtenir des résultats acceptables, mais la meilleure valeur de R ne dépasse pas 0,93.
- ✓ La variation de la taille du patch (**patch size**) d'architecture n'affecte pas bien la précision, en particulier avec une base de données relativement grande (1 min et 5 min). Les valeurs appropriées sont 32 et 64.
- ✓ Pour une implémentation en temps réel, une structure à faible complexité comme celle de LSTM est la plus adaptée, même avec moins de précision que Bi-LSTM et GRU.
- ✓ L'augmentation de la valeur du **dropout** à 0,45 peut améliorer légèrement la précision de tous les cas étudiés.
- ✓ À propos du temps de fonctionnement, CNN présente un temps de fonctionnement

inférieur dans la plupart des horizons temporels.

IV.4. Conclusion

Dans le présent chapitre, seule la puissance produite par le champ PV (qui fait partie de la centrale MG) est considérée. Les résultats de la simulation indiquent que toutes les structures (LSTM, GRU, Bi-LSTM, CNN et Stacked-LSTM) étudiées peuvent donner de bons résultats. Par exemple, les réseaux Bi-LSTM et GRU sont les plus adaptés pour très court-terme (1 min) avec un ensemble de données relativement important (environ 127 000 points), tandis que Stacked-LSTM est le plus approprié pour court-terme (1 h) avec une base de données limitée (environ 4 000 points). Le GRU peut fournir de bon résultats pour les deux horizons de temps 5 min et 30 min. CNN peut également présenter des résultats acceptables dans le cas d'un horizon de temps de 30 min. Ce n'est pas une règle générale, car les résultats dépendent principalement de la quantité de la base de données utilisée ainsi que de la qualité des données.

Pour une application en temps réel, des structures DLNN moins compliquées sont les plus appropriées. Un ordinateur puissant est nécessaire pour réduire le temps de fonctionnement, en outre, il a été observé que le système d'exploitation Ubuntu fonctionne mieux que Windows en termes de temps de fonctionnement, qui peut atteindre 30 minutes avec la première base de données (127 000 échantillons).

Conclusion générale

Ce travail porte sur l'application de réseaux de neurones profonds pour la prédiction de puissance produite par un micro-réseau, destiné au chargement des véhicules électriques. Nous avons commencé tout d'abord, dans le premier chapitre, par une présentation détaillée sur les systèmes photovoltaïques connectés au réseau électrique et enfin nous avons présenté les facteurs qui peuvent influencer la productivité des systèmes PV connectés au réseau électrique.

Dans le deuxième chapitre, nous avons introduit certaines définitions des réseaux de neurones classiques et profonds. Dans le troisième chapitre, nous avons présenté les ressources matérielles et logicielles nécessaires à notre travail.

Les résultats que nous avons obtenus sont restitués dans le dernier chapitre. Différents DLNNs ont été développés pour la prévision de la puissance d'un champ photovoltaïque, pour différents horizons temporels (1 min, 5 min, 30 min et 60 min).

Il a été démontré qu'une structure simple (telle que LSTM ou GRU) peut fournir de très bonne précision de prévision ($R = 99\%$).

Il convient de souligner que les paramètres (tels que la taille du Batch, le nombre d'unités et le nombre de couches cachées, la taille des filtres, etc.) doivent également être soigneusement sélectionnés. Cependant, la valeur de ces paramètres diffère d'un modèle à l'autre. Il a été vérifié que la disponibilité d'une grande base de données est nécessaire pour obtenir de bons résultats. Il est à noter que, même avec les nouveaux algorithmes DRNN avancés, une précision acceptable en cas de jours nuageux est obtenue, mais le problème n'est pas résolu définitivement. Ainsi, des efforts supplémentaires devraient être déployés (par exemple, en combinant les données de prévisions météorologiques, les images du ciel, l'indice nuageux, etc).

La prévision à long-terme de la puissance PV reste également un problème difficile qui doit être soigneusement traité, en testant d'autres types DLNN (par exemple, l'apprentissage Seq2Seq) ou en développant des algorithmes plus avancés.

Références

- [1] M. Bouzerdoum, “ *Contribution à la modélisation et l’analyse de la performance des systèmes photovoltaïques connectés au réseau électrique* “, thèse de Doctorat, Université de Jijel, 2016.
- [2] A. Gensler, J. Henze B. Sick, N. Raabe, “ Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks “, *IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 2858-2865, 2016.
- [3] D. Su, E. Batzelis, B. Pal, “ Machine learning algorithms in forecasting of photovoltaic power generation “, *International Conference on Smart Energy Systems and Technologies (SEST)*, pp. 1-6, 2019.
- [4] B. Jason, *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python* , Machine Learning Mastery, 2018.
- [5] A. Mellit, A. Massi Pavan, E. Ogliairi, S. Leva and V. Lughi., “ Advanced Methods for Photovoltaic Output Power Forecasting: A Review “, *Applied Sciences*, Vol. 10, No. 2, pp.487, 2020.
- [6] Tout sur l’énergie, produit de l’électricité <https://www.edf.fr/groupe-edf/espaces-dedies/l-energie-de-a-a-z/tout-sur-l-energie/produire-de-l-electricite/qu-est-ce-que-l-energie-solaire>.
- [7] Les caractéristiques du rayonnement solaire sur Terre <https://www.connaissancedesenergies.org/fiche-pedagogique/energie-solaire-exploitation>.
- [8] Les systèmes photovoltaïques: Guide de l’acheteur <https://www.batteriesquebec.com/uploads/OrtBIsGx/Guidephotovoltaquedelacheteur.pdf> .
- [9] La cellule photovoltaïque : comment ça marche ? <https://www.planete-energies.com /fr/medias/decryptages/la-cellule-photovoltaïque-comment-ca-marche> (consulté mars 2020).
- [10] Solaire photovoltaïque <https://www.connaissancedesenergies.org/fiche-pedagogique/solaire-photovoltaïque>.
- [11] Technologies de cellules solaires [http://fr.solarpedia.net/wiki/index.php?title=Technologies de cellules solaires](http://fr.solarpedia.net/wiki/index.php?title=Technologies_de_cellules_solaires).
- [12] Rendement et puissance crête des cellules photovoltaïques <https://energieplus-lesite.be /theories/photovoltaïque6/rendement-et-puissance-crete-des-cellules-photovoltaïques/>.

Références bibliographiques

- [13] D.Thomas, R. Jimmy, S. Eric , S.Bocar, “Le pompage photovoltaïque: Manuel de cours à l’intention des ingénieurs et des techniciens “, *Institut de l’énergie des pays ayant en commun l’usage du français*, 1998.
- [14] M.I. Salim, “ *Commande d’une poursuite du point de puissance maximum (MPPT) par les Réseaux de Neurones* “, mémoire de Master, Université ABDERAHMANE MIRA – Béjaia, 2006.
- [15] Cours d’analyse numérique : réseaux de neurones Formels (<http://informatique.coursgratuits.net/methodes-numeriques/reseaux-de-neurones-formels.php>).
- [16] *les reseaux de neurones artificiels, introduction au connexionnisme* Claude Touzet formels(https://halamu.archivesouvertes.fr/hal01338010/file/Les_reseaux_de_neurones_artificiels.pdf).
- [17] <https://www.peoi.org/Courses/Coursesfr/neural/neural2.html> (consulté septembre 2020).
- [18] F. Blayo , M. Verleysen, “ Les réseaux de neurones artificiels “, Presses Universitaires de France(PUF), Que Sais-je, 1996.
- [19] N.Malki, “*Classification automatique des textes par Les réseaux de neurones à convolution* “, Mémoire de master, Université Larbi ben M’hidi- Oum El Bouaghi, 2019.
- [20] A. CHERIF,“ *Réseaux de neurones, SVM et approches locales pour la prévision des séries temporelles* “ , Thèse de doctorat , Université François Rabelais de Tours, 2013.
- [21] M. BOUAZIZ, “ *Réseaux de neurones récurrents pour la classification de séquences dans des flux audiovisuels parallèles* “, Thèse de doctorat, Université d’Avignon et des Pays de Vaucluse, 2017.
- [22] H.T. Siegelmann, “Computation beyond the Turing limit “, *Neural Networks and Analog Computation*, pp. 153-164, 1997.
- [23] Understanding LSTM Networks formel (<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>).
- [24] Les réseaux de neurones récurrents : des RNN simples aux LSTM formel(<https://blog.octo.com/les-reseaux-de-neurones-recurrents-des-rnn-simples-aux-lstm/>).
- [25] Réseaux de neurones récurrents,formel (<http://ichi.pro/fr/post/201698563206245>).
- [26] M. Marty, G. Wenzek , E. Schmitt, J. Coulmance , “ Analyse d’opinions de tweets par réseaux de neurones convolutionnels “, *Traitement Automatique des Langues Naturelles-Caen*, pp. 22, 2015.

Références bibliographiques

- [27] Le site officiel de python <https://www.python.org/>.
- [28] Le site officiel d'anaconda <https://www.anaconda.com/distribution/>.
- [29] Le site officiel de tensorflow <https://www.tensorflow.org/>.
- [30] Le site officiel de keras <https://keras.io>.
- [31] F. CHEKIRED, “*Etude et implémentation d’une commande MPPT neuro-floue sur FPGA*,” “Mémoire de Magister en Electronique, Université de Jijel, 2008.

Annexes

❖ Les structures utilisées :

➤ LSTM

```
model = Sequential()
model.add(LSTM(units = 50, activation='relu',input_shape = (X_train.shape[1], 1)))
model.add(Dense(units = 1))
```

➤ Stacked LSTM :

```
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences=True,
input_shape=(X_train.shape[1], 1)))
model.add(LSTM(50, activation='relu'))
model.add(Dense(1))
```

➤ Bidirectional LSTM :

```
model = keras.Sequential()
model.add(keras.layers.Bidirectional(keras.layers.LSTM( units=50,
input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(keras.layers.Dropout(rate=0.45))
model.add(keras.layers.Dense(units=1))
```

➤ GRU :

```
model = keras.models.Sequential()
model.add(keras.layers.GRU(8, return_sequences=True, input_shape=(train_X.shape[1],
train_X.shape[2]) ))
model.add(keras.layers.GRU(4))
model.add(keras.layers.Dense(1))
```

➤ CNN :

```
Model.add (conv1D(filters=32,kernel_size=3, activation='relu',input_shape=(21,11)))
Model.add(maxPooling1D(pool_size=2))
Model.add(dropout(0.2))
Model.add(Flatten())
Model.add(Dense(8,activation='relu'))
Model.add(dense(1))
```

