République Algérienne Démocratique Et Populaire

Université de Saad Dahleb 1

Faculté des Sciences

Département d'Informatique

MEMOIRE DE MASTER

En Informatique

Spécialité : Ingénierie des Logiciels

THEME

Apprentissage par renforcement pour l'optimisation de la consommation d'énergie dans un contexte IoT

Réalisé par :

Kabir Selma

Proposé par :

Mr.Kameche

- 2019 / 2020 –

Remerciement

Je tiens tout d'abord à remercier ***ALLAH*** *le tout puissant, le tout miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.*

Je remercie sincèrement mes chers parents de tout mon cœur pour leur encouragement tout au long de mon parcours.

Je remercie également mon promoteur, Mr.Kameche qui m'a donné la chance d'explorer ce domaine, et qui m'a guidé tout au long de mon travail.

Je tiens aussi de remercier les membres de jury qui ont accepté d'évaluer mon travail.

Enfin, je remercie tous mes amis pour leurs encouragements et leurs soutiens.

Résumé

Dans ce modeste travail, nous explorons une approche informatique qui est l'apprentissage par renforcement. Nous explorons principalement les situations d'optimisation d'énergie pour l'internet des objets dans une maison intelligente et évaluons l'efficacité de différentes méthodes d'apprentissage par renforcement, en évaluant les conceptions par l'analyse mathématique ou par des expériences de calcul, et en exposant les techniques de préservation d'énergie existante.

Nous travaillerons sur le développement d'un modèle de consommation d'énergie au sein d'apprentissage par renforcement s'exécutant sur le nœud de détection.

Mots clés : Internet des Objets, Apprentissage par renforcement, Optimisation, consommation énergétique.

Abstract

In this modest work, we explore a computing approach, which is reinforcement learning. We mainly explore energy optimization situations for the Internet of Things in a smart house and evaluate the effectiveness of different reinforcement learning methods, evaluating the designs by mathematical analysis or computational experiments, and exposing existing energy conservation techniques.

We will work on the development of an energy consumption model within reinforcement learning running on the detection node.

Key words: Internet of Things, Reinforcement Learning, optimization, energy consumption.

نبذة مختصرة:

في هذا العمل المتواضع، نستكشف نهجًا حسابيًا وهو التعلم المعزز. نستكشف بشكل أساسي حالات تحسين الطاقة لإنترنت الأشياء في المنزل الذكي ونقيم فعالية طرق التعلم المعززة المختلفة، ونقيّم التصاميم عن طريق التحليل الرياضي أو عن طريق التجارب الحسابية، ومن خلال الكشف عن التقنيات الحالية للحفاظ على الطاقة. سنعمل على تطوير نموذج لاستهلاك الطاقة ضمن التعلم المعزز الذي يعمل على عقدة الاستشعار.

الكلمات الرئيسية: إنترنت الأشياء، التعلم المعزز، التحسين، استهلاك الطاقة.

Sommaire

Listes des figures :

Liste des tables

Liste des abréviations

IA : Intelligence Artificiel.

IoT: Internet of Things

IdO: Internet des Objets.

ML: Machine Learning.

RL: Reinforcement Learning.

MDP: Markov Decision Problem.

DP: Dynamic Programming.

MC: Monte Carlo.

MCTS: Monte Carlo Tree Search.

CNN: Convolutional Neural Network.

ANN: Artificial Neural Network.

RNN: Recurrent Neural Network.

DQN: Deep Q-Network.

DQL: Deep Q-Learning.

GPS: Global Positioning System.

TCC & R: Tracking and Control, Command and Routing.

M2M: Machine to Machine.

RFID: Radio Frequency Identification.

MCU : Multipoint Control Unit (microcontrôleur).

MPU : Réseau téléphonique commuté (microprocesseur).

ID: Identification.

CCD: Charge-Coupled Device.

WSN: Wireless Sensor Network.

PRRP : Position Responsive Routing Protocol.

RTC : Réseau téléphonique commuté

NFC: Near Field Communication.

QoS: Quality of Service:

TM: Transmission Media.

APRANET: Advanced Research Projects Agency Network.

MIT: Massachusetts Institute of Technologies.

DRL : Deep Reinforcement Learning.

Introduction générale

Introduction générale

En raison du développement rapide de différentes technologies telles que les semi-conducteurs, le sans fil et les capteurs, nous avons été témoins de l'infiltration d'appareils interconnectés dans notre vie quotidienne. Ces dispositifs embarqués avec capteurs et méthodes de communication sont des composantes clés de l'Internet des objets (IoT)[1][2]. L'Internet des objets fournit des services avancés de surveillance et de contrôle pour fournir de nouvelles applications ou améliorer l'efficacité des applications existantes.

L'apprentissage automatique (ML) est un moyenne clé qui permet l'inférence d'informations, le traitement des données et l'intelligence pour les dispositifs IdO. Du traitement des données sur le cloud à l'intelligence embarquée, la ML est une solution efficace et prometteuse dans différents domaines d'application de l'IdO. Plusieurs études sur le ML et l'IdO ont été présentées récemment, chacune d'entre elles couvre un aspect différent du ML dans l'IdO [3].

Problématique :

Certaines solutions existent pour améliorer la consommation énergétique, toutefois, beaucoup de scénarios ne peuvent être pris en considération, on peut en citer par exemple les pannes de courant, l'interaction hasardeuse de l'homme avec les équipements ou encore les changements climatiques. De plus, les solutions proposées actuellement sont applicables à une architecture bien spécifique et sont rarement réadaptables à une nouvelle architecture.

Objectifs du travail :

Pour remédier à cette problématique, nous nous intéressons à faire une étude comparative des différentes approches existantes, puis proposer une nouvelle solution d'optimisation de consommation énergétique en exploitant l'apprentissage par renforcement.

Organisation du mémoire :

Pour bien présenter notre travail, nous avons structuré notre mémoire autour des axes suivants :

- **Chapitre I :** Ce chapitre explique le concept d'apprentissage par renforcement et l'apprentissage par renforcement en profondeur et les différentes méthodes utilisées pendant ces processus.

- **Chapitre II :** Ce deuxième chapitre est consacré à la définition de la notion « Internet des objets », une présentation de quelques problématiques posées par l'IdO et à la fin il y est exposé quelques travaux existant sur l'économie d'énergie.
- **Chapitre III :** Le troisième chapitre est réservé à l'étude conceptuelle de notre application ainsi que le role de notre agent intelligent.
- **Chapitre IV :** Ce dernier chapitre est dédié à la présentation des outils du développement utilisés pour réaliser notre système, et l'implémentation de ce système.

Ce mémoire sera conclu par une conclusion générale qui va servir de récapitulatif des notions abordées précédemment.

Chapitre I : Apprentissage par renforcement

Introduction

Ce chapitre présente une méthode d'apprentissage qui a été largement développée dans le domaine de l'intelligence artificielle (IA). Parce qu'elle est issue de la théorie du renforcement, elle est appelée *apprentissage par renforcement*. L'apprentissage par renforcement de l'intelligence artificielle consiste en un ensemble de méthodes de calcul, bien que ces méthodes soient inspirées des principes de l'apprentissage animal, elles sont principalement motivées par leur capacité à résoudre des problèmes pratiques[4] .

On va consacrer ce chapitre pour la présentation des types d'apprentissage qui existe, ainsi pour la description de la technique d'apprentissage par renforcement et ses déférentes méthodes et de l'apprentissage par profond (Deep Q-Learning).

I.1. Apprentissage automatique :

L'apprentissage automatique est une branche de l'intelligence artificielle, et est définie par l'informaticien et créateur de l'apprentissage automatique Tom M. Mitchell comme suit : "L'apprentissage automatique est l'étude des algorithmes informatiques qui permettent aux programmes informatiques de s'améliorer automatiquement grâce à l'expérience".

L'apprentissage automatique fournit des méthodes automatiques pour détecter des modèles de données et les utiliser pour réaliser quelques tâches[5] [6].
Il y a trois types d'apprentissage automatique :

I.1.1.Apprentissage Supervisé :

L'apprentissage supervisé est la tâche d'inférer une classification ou régression à partir des données de formation étiquetées [7].

Dans sa forme la plus abstraite, l'apprentissage supervisé est une fonction f : X → Y qui prend comme entrée **x** ϵ **X** et donne comme sortie **y** ϵ **Y** (X et **Y** dépendent de l'application) : y = f(x).

L'apprentissage supervisé traite deux problèmes [8] :

a. La classification :

La classification consiste à regrouper les résultats dans une classe. Si l'algorithme tente de classer les éléments (l'entrée) en deux classes différentes, on parle de classification binaire. Le choix entre deux ou plusieurs catégories s'appelle la classification multi-catégories.

b. La régression :

La technique de régression utilise des données d'entraînement pour prédire une valeur de sortie unique. Par exemple, on peut utiliser la régression pour prédire les prix des maisons à partir des données de formation. Les variables d'entrée seront l'emplacement, la taille de la maison, etc.

I.1.2. Apprentissage non Supervisé :

L'apprentissage non supervisé est une branche d'apprentissage automatique qui apprend à partir des données qui n'ont pas d'étiquette[7]. Autrement, dans l'apprentissage non supervisé, on a un ensemble de données x (Dataset) sans variable y, et la machine apprend à reconnaître la structure dans les données x qui lui sont montrées.

a. Le regroupement (Clustering) :

Il s'agit principalement de trouver des structures ou des modèles dans une collection de données non classifiées. L'algorithme de clustering traitera nos données et recherchera des clusters naturels (groupes) s'ils existent dans les données. On peut également modifier le nombre de clusters que l'algorithme doit reconnaître [8].

I.1.3. Apprentissage par Renforcement (RL) :

L'apprentissage par renforcement est l'aire de l'apprentissage automatique qui traite une séquence de décision à faire [7].

Dans notre étude on s'intéresse de ce type d'apprentissage.

I.2. Apprentissage par Renforcement :

I.2.1. Le problème de l'apprentissage par renforcement :

On peut définir le problème de l'apprentissage par renforcement comme suit :

a. Le principe :

Le principe de RL est un principe qui permet à un agent d'apprendre de bonnes habitudes : c'est-à-dire qu'il modifie ou recueille des nouveaux habitudes et compétences .Une autre importance, l'agent ne prend pas le control total de l'environnement mais quand il interagir avec l'environnement il collecte de nouvelles informations [7].

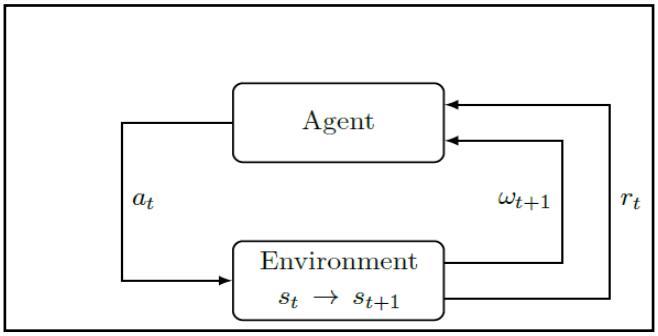
b. La formalisation :

Le problème général du RL est formalisé comme un processus de contrôle stochastique en temps discret. Dans ce processus, l'agent interagit avec son environnement comme suit :

l'agent commence par un état donné dans son environnement $s_0 \in S$ en collectant des observation $w_0 \in \Omega$. à chaque pas de temps t, l'agent doit prendre une action $a_t \in A$. Comme il est illustré dans la Figure 1, qui est composé de trois conséquences :

i. l'**agent** obtient une récompense $r_t \in R$.
ii. l'**état transition** à $s_{t+1} \in S$
iii. l'agent obtient une **observation** $w_{t+1} \in \Omega$.

Ce cadre de contrôle a été proposé pour la première fois par Bellman[9] et plus tard a été étendu à l'apprentissage par [10].



I.Figure 1.Agent-Environnement interaction en RL[7]

I.2.2. Le processus décisionnel de Marcov (MDP) :

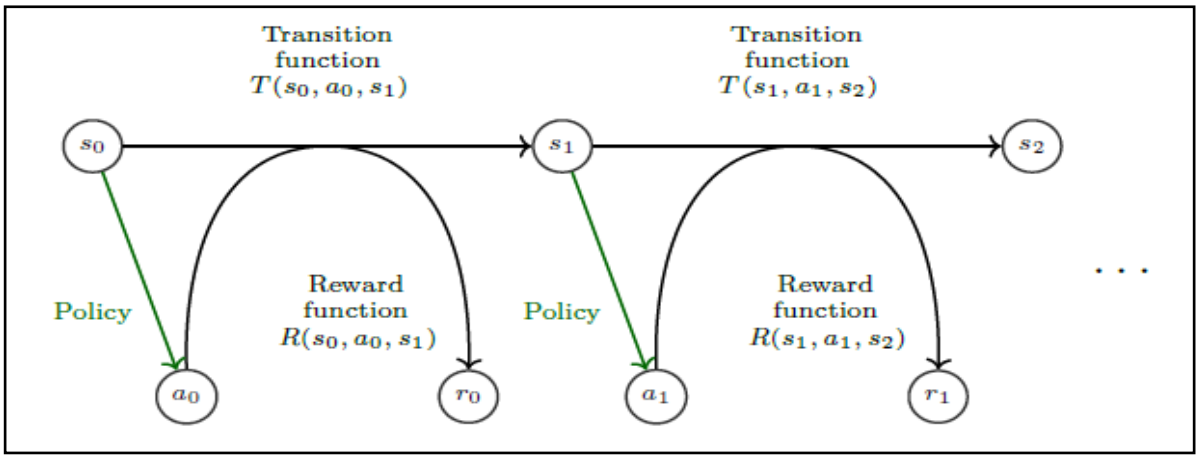
L'environnement est souvent modélisé comme un processus décisionnel de Marcov, où l'agent doit également avoir un objectif lié à l'état de l'environnement. L'apprentissage par renforcement utilise le cadre formel de ce processus pour définir l'interaction entre l'agent d'apprentissage et son environnement sous forme d'états, actions et de récompenses [11].

Le processus décisionnel de Markov [12] est un processus de contrôle stochastique en temps discret définit comme un 4-Tuple (S, A, T, R) où :

- **S** est l'ensemble d'états.
- **A** est l'ensemble d'actions.
- **T** : S×A×S' → [0,1] est une fonction de transition markovienne. Elle représente la probabilité d'aller de l'état « **s** » à l'état « **s`** » en effectuant l'action « **a** ».
- **R** : S×A×S' → R est une fonction de récompense. Elle représente la récompense obtenue en allant de l'état « **s** » à l'état « **s`** » en effectuant l'action « **a** ».

Le système est observable dans MDP, ce qui signifie que l'observation est la même que l'état de l'environnement ; $\mathbf{w_t} = \mathbf{s_t}$ (Figure 1). À chaque pas de temps **t**, la probabilité d'atteindre

l'état s_{t+1} est déterminé par la fonction de transition **T** (s_t , a_t, s_{t+1}), et la récompense est déterminé par la fonction de récompense **R** (s_t , a_t, s_{t+1}) (Figure 2) [7].



I.Figure 2.Illustration d'un MDP [7]

I.2.3. Le système d'apprentissage par renforcement :

En plus du sujet de l'agent, état et action ; on peut également identifier cinq sous-éléments principaux de système d'apprentissage par renforcement qui sont : la politique, la récompense, la fonction de valeur, le model et éventuellement l'environnement.

A. La politique :

La politique est le cœur d'un agent d'apprentissage par renforcement car elle sert à déterminer son comportement et la manière dont il choisit ses actions [11].

Les politiques peuvent être classés selon le critère de leur caractère qui peut être stationnaire ou non stationnaire :

a. Une politique non stationnaire :

Cette politique dépend du pas de temps et elle est utile dans le contexte d'horizon fini où les récompenses cumulées que l'agent cherche à optimiser sont limitées à un nombre fini de pas de temps futurs [13].

b. Une politique stationnaire :

La politique stationnaire est une politique qui ne change pas au fil de temps et elle peut être déterministe ou stochastique [7]:

- Dans le cas déterministe, la politique est décrite par : $\pi(s) : S \rightarrow A$. $\pi(s)$ indique l'action choisi en « s ».

- Dans le cas stochastique, la politique est décrite par : $\pi(s, a) : S \times A \rightarrow R$; où $\pi(s,a)$ indique la probabilité que l'action « **a** » puisse être choisie à l'état « **s** ».

B. Rendement :

La récompense définit l'objectif du problème d'apprentissage par renforcement et permet de définir les bons et les mauvais évènements de l'agent : à chaque pas de temps, l'environnement envoie un numéro unique à l'agent, appelé *récompense*. Le seul objectif de l'agent est de maximiser le rendement total obtenu à long terme.

En plus des récompenses actuelles, on doit également considérer les récompenses qu'on reçoit à l'avenir. La récompense totale d'un seul épisode peut être calculée comme suit :

$$\mathbf{RE = re_1 + re_2 + re_3 + ... + re_n}$$

Etant donnée un environnement stochastique, il n'est pas possible de garantir que les mêmes récompense seront obtenues en effectuant les mêmes actions. Les récompenses divergent à mesure que les systèmes progressent vers l'avenir. Pour cette raison, il est prévu d'utiliser une récompense future réduite à la place :

$$\mathbf{RE_t = re_t + \gamma\, re_{t+1} + \gamma^2\, re_{t+2} + ... + \gamma^{n-1}\, re_n}$$

Avec $\mathbf{0 \leq \gamma < 1}$. γ est un facteur de décompte pour les renforcements futurs

La récompense réduite dans le temps « **t** » peut être donnée en même terme que dans le temps « $\mathbf{t_{+1}}$ » :

$$\mathbf{RE_t = re_t + (re_{t+1} + (\gamma^2\, re_{t+2} + ...)) = re_t + RE_{t+1}}$$

Si le facteur réduit est égale à zéro, alors cette stratégie va dépendre seulement de la récompense actuelle. En séquence de fournir une balance entre l'actuel et la future récompense, il est nécessaire de mettre ce facteur à une valeur non nul comme la valeur 0,9. Si l'environnement est déterministe et les mêmes récompenses sont obtenues en effectuant les mêmes actions, alors le facteur réduit sera égale à 1. La meilleure stratégie est de choisir toujours l'action qui on donne une récompense future maximale [14].

C. Fonction de valeur (Value-function) :

Dans l'apprentissage par renforcement, l'agent a besoin d'informations sur la qualité d'un état « **s** » afin de trouver une politique optimal $\pi(s, a) \in \Pi$ [15].Une politique $\boldsymbol{\pi}$ est une cartographie entre l'état « **s** » et la probabilité de sélection de chaque action « **a** » possible.

Si l'agent suit la politique $\boldsymbol{\pi}$ au temps **t**, alors $\boldsymbol{\pi(s \mid a)}$ est la probabilité de $\mathbf{A_t = a}$ lorsque $\mathbf{S_t = s}$. Comme « **p** », $\boldsymbol{\pi}$ est une fonction ordinaire, le symbole " | " au milieu de $\boldsymbol{\pi(s \mid a)}$ définie une distribution de probabilités sur $\mathbf{a \in A(s)}$ pour chaque $\mathbf{s \in S}$ [11].

La fonction de valeur d'un état sous une politique, noté $\mathbf{V_\pi(s)}$, est l'espérance des récompenses attendu lorsqu'on commence dans l'état ($\mathbf{s_t=s}$) et suit une politique $\boldsymbol{\pi}$. Sur la base de MDP, la fonction *état-valeur* peut être formellement définie comme [15]:

$$\mathbf{V^\pi(s) = E_\pi \{R_t | s_t = s\} = E_\pi \{\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t=s\}}.$$

Avec $\mathbf{0 \leq \gamma < 1}$.

De même, la fonction *action-valeur* décrit l'espérance des récompenses attendues pour une action « **a** » dans l'état « **s** » et en suivant la politique $\boldsymbol{\pi}$:

$$\mathbf{Q^\pi(s, a) = E_\pi \{R_t | s_t = s, a_t=a\} = E_\pi \{\sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t=s, a_t=a\}}.$$

En résume :

La function *état-valeur* calcule la somme de toutes les actions possible multipliée par leur probabilité, la function *action-valeur* calcule la récompense pour une action explicite « a » au pas de temps « t » dans l'état « **s** ».Et le lien entre ces deux fonction est affichée par :

$$\mathbf{V^\pi(s) = \sum_a \pi(s, a)\, Q^\pi(s, a).}$$

Ici $\mathbf{V^\pi(s)}$ représente l'espérance moyenne des récompenses futurs pour toutes les actions possibles « **a** », pondérée par la distribution de probabilité $\boldsymbol{\pi}$**(s, a)** et $\mathbf{Q^\pi}$ représente une récompense réel pour un choix fixe d'actions.

D. Le modèle :

Le modèle est la représentation d'un environnement par l'agent. L'apprentissage peut être devisé en deux types [16]:

a. **L'apprentissage basé sur un modèle (Based-model learning) :** Dans ce type, l'agent exploite des informations préalablement apprises pour accomplir une tâche
b. **L'apprentissage sans modèle (Free-model learning) :** Dans ce type, l'agent se base simplement à une expérience d'essais et d'erreurs (l'exploration) pour effectuer la bonne action.

Par exemple, supposons que nous souhaitons arriver plus rapidement à notre bureau depuis notre domicile. Dans l'apprentissage basé sur un modèle, nous utilisons simplement une expérience précédemment acquise (carte) pour atteindre le bureau plus rapidement, tandis que dans l'apprentissage sans modèle, nous n'utiliserons pas d'expérience antérieure et nous essaierons tous les chemins différents et nous choisirons le plus rapide.

E. Environnement :

Tout ce avec quoi l'agent interagit est appelé **« *environnement* »**. L'environnement est le monde extérieur ; il comprend tout ce qui extérieur à l'agent.

On peut distinguer trois critères pour un environnement : le déterminisme, l'observation et la continuité[17].

1) Le déterminisme :

a. Environnement déterministe :

On dit qu'un environnement est déterministe lorsqu'on connaît le résultat en fonction de l'état actuel.

Par exemple, dans une partie d'échecs, on connaît le résultat exact du déplacement de n'importe quel joueur.

b. Environnement stochastique :

On dit qu'un environnement est stochastique lorsqu'on ne peut pas déterminer le résultat en fonction de l'état actuel.

Par exemple, on ne sait jamais quel chiffre apparaîtra au moment du lancement d'un dé.

2) L'observation :

a. Environnement entièrement observable :

Lorsqu' un agent peut déterminer l'état du système à tout moment, on dit que l'environnement est entièrement observable.

Par exemple, dans une partie d'échecs, l'état du système ; c'est-à-dire la position de tous les joueurs sur l'échiquier est disponible en permanence pour que le joueur puisse prendre une décision optimale.

b. Environnement partiellement observable :

Lorsqu' un agent ne peut pas déterminer l'état du système à tout moment, on dit que l'environnement est partiellement observable.

Par exemple, dans une partie de poker, on n'a aucune idée des cartes de l'adversaire.

3) La continuité :

a. Environnement discret :

Lorsqu'il n'y a qu'un état fini d'actions disponibles pour passer d'un état à l'autre, on parle de l'environnement discret.

Par exemple, dans une partie d'échecs, on n'a qu'un ensemble fini de mouvements.

b. Environnement continu :

Lorsqu'il existe une infinité d'actions permettent de passer d'un état à un autre, on parle d'un environnement continu.

Par exemple, on dispose de plusieurs chemins disponibles pour voyager de la source à la destination.

I.2.4. Méthodes de résolutions :

La résolution du problème d'apprentissage par renforcement est basée sur l'une des méthodes suivantes :

- Méthode de la **Programmation Dynamique** (**DP**).
- Méthode de **Monte Carlo** (**MC**).
- L'algorithme de **Q-learning**.
- L'**Apprentissage-Q approfondi (Deep Q-learning).**

A. Programmation Dynamique (DP) :

La programmation dynamique est une méthode permettant de résoudre des problèmes complexes en les décomposant en sous problèmes. Les solutions aux sous problèmes sont combinées pour résoudre le problème global. Comme elle peut aussi être utilisée pour résoudre les problèmes d'apprentissage par renforcement lorsque quelqu'un nous indique la structure du MDP (c'est-à-dire lorsque nous connaissons la structure de transition, la structure de récompense, etc.)[18].

a. Evaluation de la politique :

L'évaluation des politiques consiste à évaluer une politique donnée π et MDP (trouver à quel point une politique π est bonne) en utilisant une fonction itérative de Bellman [18],

$$\mathbf{V_{k+1}(s) = E_{\pi}[r_{t+1} + \gamma V_k(s_{t+1}) \mid s_t = s].}$$

Afin de générer chaque estimation successive V_{k+1} à partir de V_k, l'évaluation de la politique itérative applique la même opération à chaque état s. Il remplace l'ancienne valeur de « s » par la nouvelle valeur obtenue à partir de l'ancienne valeur de l'état successeur de « s », ainsi que la récompense immédiate attendue, et toutes les transitions en une étape possibles dans le cadre de la stratégie d'évaluation, jusqu'à ce qu'elle converge vers la valeur réelle de la stratégie donnée π fonction [19] .

$$V_0 \longrightarrow V_1 \longrightarrow V_2 \longrightarrow \cdots \longrightarrow V_\pi$$

Par exemple, commencez avec une valeur de 0, il n'y a donc pas de récompense. Ensuite, utilisez l'équation des attentes de Bellman pour calculer V_1 et répétez plusieurs fois, ce qui aboutira finalement à $V\pi$.

Dans l'approche ci-dessus, nous avons évalué une politique donnée mais n'avons pas trouvé la meilleure politique (actions à prendre) dans notre environnement.

b. Amélioration de la politique :

En utilisant l'évaluation des politiques, nous avons déterminé la fonction de valeur V pour une politique arbitraire π. Nous savons à quel point notre politique actuelle est bonne.

Maintenant, pour certains états « s », nous voulons comprendre quel est l'impact d'une action « a » qui ne concerne pas la politique « π ». Supposons que nous sélectionnons « a » dans « s », et que nous suivons ensuite la politique initiale « π ». La valeur de cette forme de comportement est représentée par [19] :

$$\mathbf{Q_\pi (s, a) = E [r_{t+1} + \gamma V_\pi (s_{t+1}) \mid s_t = s, a_t = a]}$$

$$\mathbf{= \Sigma_{s', r}\, p (s', r \mid s, a) [r + \gamma\, V_\pi (s')]}$$

S'il se trouve que cette valeur est supérieure à la fonction de valeur $V_\pi(s)$, cela implique que la nouvelle politique « π » serait meilleure à prendre. Nous procédons de manière itérative pour tous les États afin de trouver la meilleure politique.

c. Itération de la politique :

C'est de trouver la meilleure politique π* pour un MDP donné, en utilisant l'évaluation et l'amélioration des politiques.

On commence avec une politique donnée « $\boldsymbol{\pi_0}$ », on évalue cette politique avec l'évaluation de la politique (décrite ci-dessus).Après, on améliore la politique « $\boldsymbol{\pi_0}$ » par une *action greedy* sur $\mathbf{V_\pi}$, avec l'amélioration de la politique pour obtenir une nouvelle politique $\boldsymbol{\pi_1}$[18] .

On répète jusqu'à ce que la nouvelle politique π' converge vers la politique optimale π*.

$$\pi_0 \xrightarrow{E} V_{\pi 0} \xrightarrow{A} \pi_1 \xrightarrow{E} V_{\pi 1} \xrightarrow{A} \pi_2 \xrightarrow{E} \cdots \xrightarrow{A} \pi_* \xrightarrow{E} V_*$$

d. Itération de la valeur :

Nous pouvons également obtenir la politique optimale avec une seule étape d'évaluation de la politique suivie par la réactualisation de la fonction de valeur à plusieurs reprises (mais cette fois-ci avec les réactualisations dérivées de l'équation d'optimalité de Bellman) [19].

$$V_{k+1}(s) = \max_a E_\pi [r_{t+1} + \gamma V_k(s_{t+1}) \mid s_t = s, a_t = a].$$

Cette méthode est identique à la réactualisation du Bellman en ce qui concerne l'évaluation des politiques, la différence étant que nous prenons le maximum sur toutes les actions. Une fois que les updates sont suffisamment petites, nous pouvons prendre la fonction de valeur obtenue comme finale et estimer la politique optimale correspondant à cela.

B. Monte Carlo (MC) :

MCTS est une méthode pour trouver une décision optimale dans un domaine donnée en prélevant des échantillons aléatoires dans l'espace de décision et en construisant un arbre de recherche selon les résultats [20].

La base de l'algorithme MCTS est simple : un arbre de recherche est construit (nœud par nœud) selon les résultats de simulation. Les étapes de ce processus sont comme suit [21]:

- **Selection :** Selection de bons nœuds enfants, à partir du nœud racine **R**, qui représentent les états conduisant au meilleur résultat.
- **Expansion :** Si **L** (leaf node ou nœud feuille) n'est pas un nœud terminal (c.-à-d. qu'il ne termine pas le jeu), créez un ou plusieurs nœuds enfants et sélectionner un **C** (chosen node ou nœud choisi).
- **Simulation (Rollout) :** Une simulation est exécutée à partir du nœud choisi **C** jusqu'à ce qu'un résultat soit obtenu.

Mais qu'entend-on par **Rollout** ?

Jusqu'à ce que on atteint le nœud de feuille, on choisit aléatoirement (Rollout**)** une action à chaque étape et simuler cet action pour recevoir une récompense moyenne lorsque le jeu est terminé :

Loop Forever :

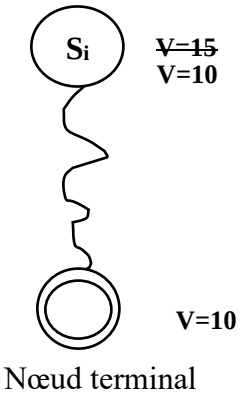
If S_i est un état terminal :

Return Value(S_i)

Else

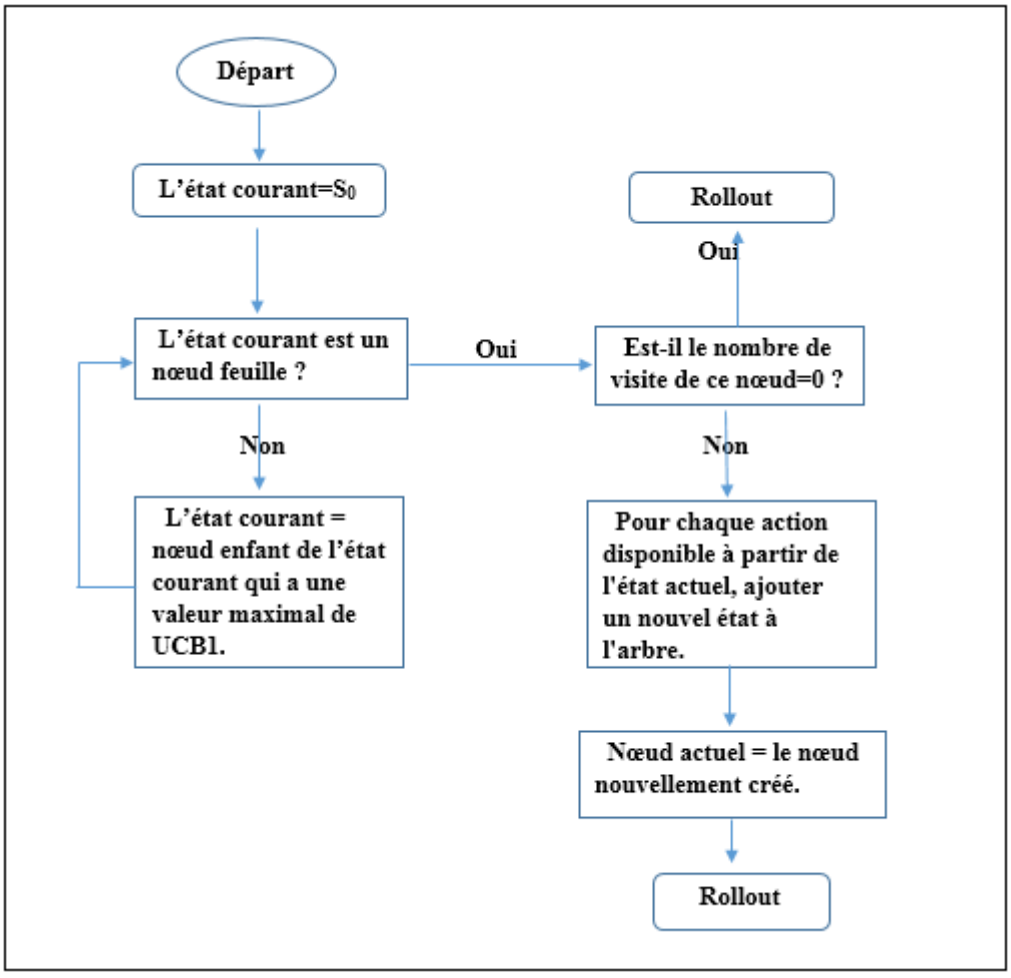
A_i = random (available_actions(S_i))

S_i = Simulate (S_i, A_i)



Cette boucle fonctionnera pour toujours jusqu'à ce qu'on atteigne un état terminal.

- **Backpropagation :** Le résultat de la simulation est "backed up" (c'est-à-dire rétropropagé) par les nœuds sélectionnés afin de mettre à jour leurs statistiques.



I.Figure 3.Diagramme de flux de MCT [21]

Explication du diagramme :

On commence avec $\mathbf{S_0}$, qui est l'état initial. Si le nœud courant n'est pas un nœud de feuille (leaf node), on calcule les valeurs pour UCB1[1] on choisit le nœud qui maximise la valeur UCB1. On continue de le faire jusqu'à ce qu'on atteigne le nœud feuille. Ensuite, on demande combien de fois ce nœud de feuille a été visité ($\mathbf{n_i}$). S'il n'a jamais été visité auparavant ($\mathbf{n_i=0}$), on fait simplement un *rollout*. Cependant, s'il a été visité avant ($\mathbf{n_i \neq 0}$), alors on ajoute un nouveau nœud (état) à l'arbre pour chaque action disponible (qu'on appelle ici *expansion)*. Notre nœud actuel est maintenant ce nœud nouvellement créé. On fait ensuite un *rollout* à partir de cette étape (Figure3).

Avec :

UCB1=V_i+2√ (ln N/n_i).

- $\mathbf{V_i}$: est la **récompense/le nombre de visite** de ce nœud.
- **N** : est le nombre de fois que le nœud parent a été visité.
- $\mathbf{n_i}$: est le nombre de fois que le nœud enfant **i** a été visité.

C. Q-Learning :

Q-learning est un algorithme simple pour créer un cheat-sheet (ou une fiche de renseignements) pour notre agent, cela aide l'agent de déterminer avec précision l'action à exécuter.

Pour utiliser *Q-learning*, on doit définir une fonction Q(s, a)=max RE_{t+1} qui représente une récompense future maximal quand une action « a » est exécuter dans un état « s » [14].

L'exécution d'un ensemble d'actions va générer une récompense totale maximale. Cette récompense totale est appelée aussi *Q-value* [22]:

$$\mathbf{Q\ (s_t, a_t) = r + \gamma\ max_{at+1}\ Q\ (s_{t+1}, a_{t+1})}$$

L'équation ci-dessus indique que *Q-value* résultant de l'état « s_t » et de l'action exécuté « a_t » est la récompense r (s_t, a_t) plus la *Q-value* la plus élevée de l'état « s_{t+1} ».

Q (s_{t+1}, a_{t+1}) dépend toujours de Q (s_{t+2}, a_{t+2}), ce qui on donnera le coefficient gamma carré. Par conséquent, la *Q-value* dépend de Q-value de l'état futur comme suit :

$$\mathbf{Q\ (s_t, a_t) \leftarrow \gamma\ Q\ (s_{t+1}, a_{t+1}) + \gamma^2\ Q\ (s_{t+2}, a_{t+2}) + \ldots}$$

Comme il s'agit d'une équation récursive, nous pouvons commencer par faire des hypothèses arbitraires pour toutes les valeurs de Q (Q-value). Avec l'expérience, elle

[1] Limite de confiance supérieure pour un nœud.

convergera vers la politique optimale. Dans les situations pratiques, elle est mise en œuvre sous forme de mise à jour (update-function) :

$$Q(s_t, a_t)_{new} = Q(s_t, a_t)_{old} + \alpha(r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)_{old}).$$

Avec :

- **$0 \leq \alpha < 1$**

Alpha est le taux d'apprentissage ou la taille du pas. Cela permet seulement de déterminer dans quelle mesure les informations nouvellement acquises sont meilleures que les anciennes.

- **$0 \leq \gamma < 1$**

La modification de la valeur du gamma diminuera ou augmentera la contribution des récompenses futures**.**

D. Apprentissage-Q approfondi (Deep Q-learning) :

Dans Q-learning, lorsque l'état et l'espace d'action sont discrets et que la dimension est faible, une table-Q peut être utilisée pour stocker la valeur-Q de chaque paire état-action. Cependant, lorsque l'espace d'état et d'action est de haute dimension et continu, une table-Q ne serait pas qualifiée. C'est pourquoi nous utilisons un réseau de neurones, pour traiter les tables-Q à haute dimension avec des états continus.

1) Les réseaux neuronaux profonds (DNN) :

Un réseau neuronal profond est un réseau neuronal artificiel (ANN**)** caractérisé par une succession de multiples couches de traitement. Chaque couche consiste en une transformation non linéaire et la séquence de ces transformations conduit à l'apprentissage de différents niveaux d'abstraction[23] [24].

Les deux réseaux de neurones les plus populaires sont :

a. Les réseaux de neurones convolutifs (CNN) :

Un réseau neuronal convolutionnel est un sous-ensemble du réseau de neurone artificiel (ANN) et d'apprentissage profond (Deep Learning) [25] . Il est généralement utilisé pour analyser des images visuelles dans le cas des jeux vidéo [26][27].

b. Les réseaux de neurones récurrents (RNN) :

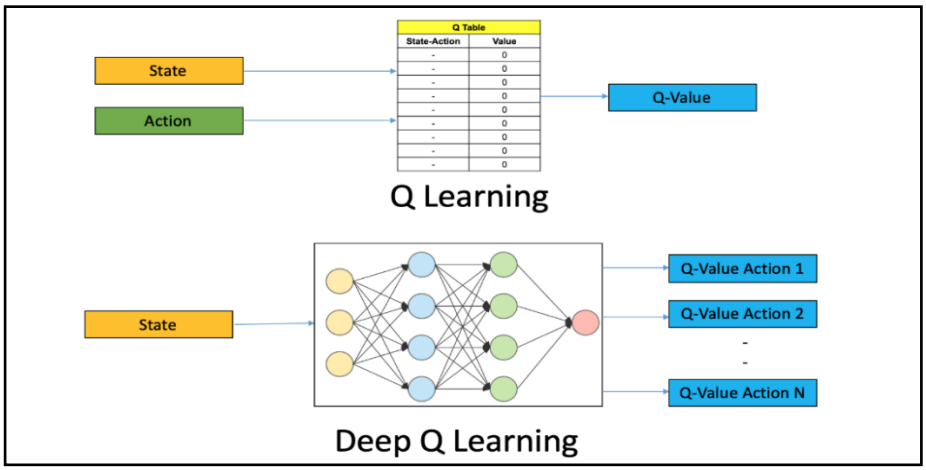
Les réseaux neuronaux récurrents sont une sorte de réseau neuronal spécialisé dans le traitement des séquences. Ils sont souvent utilisés dans les tâches de traitement du langage naturel (NLP) en raison de leur efficacité dans le traitement du texte [28].

Pour en savoir plus sur ces types de réseaux de neurones, vous pouvez consulter l'annexe.

2) Réseau-Q profond (DQN) :

DQN est l'un des nombreux algorithmes qui combinent l'apprentissage approfondi et l'apprentissage par renforcement pour apprendre des stratégies directement à partir de données brutes de haute dimension. En combinant le réseau neuronal convolutif (CNN) avec le Q-Learning, DQN a favorisé le développement du Reinforcement Learning et à élargi ses scénarios d'application.

Dans *deep Q-learning*, on utilise les réseaux de neurones (CNN) pour approximer la fonction *Q-value*. L'état est donné comme une entrée et la *Q-value* de toutes les actions possibles sont générés comme une sortie. La comparaison entre *Q-learning* et *deep Q-learning* est illustrée dans le schéma ci-dessous [22] :



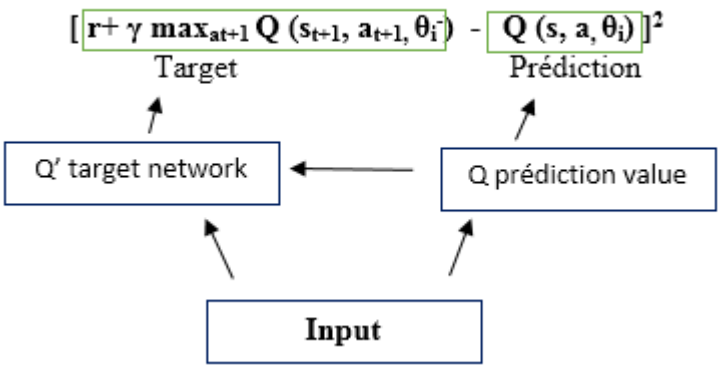
I.Figure 4.Reinforcement learning: Deep Q-Network [22]

L'algorithme de réseau-Q profond (DQN) introduit par [29] est en mesure d'obtenir une performance forte dans un cadre en ligne pour une variété de jeux ATARI[2], directement en apprenant à partir des pixels. Il utilise deux heuristiques pour limiter les instabilités [22] :

- **Target network** : Puisque le même réseau calcule la valeur prévue (Predicted value) et la valeur cible (Target value), il pourrait y avoir beaucoup de divergence entre ces deux éléments. Donc, au lieu d'utiliser un seul réseau neuronal pour l'apprentissage, on peut en utiliser deux : θ^- et θ.

  On utilise le premier pour récupérer les Q-value tandis que le deuxième inclut toutes les mises à jour dans l'entrainement.

[2] Atari est une entreprise française (situé à Paris) spécialisée dans le développement des jeux vidéo comme Pong, Breakout, Pac-Man, ...etc.



On pourrait utiliser un réseau distinct pour estimer le target. Ce réseau cible a la même architecture que la fonction d'approximation mais avec des paramètres gelés. Les paramètres du réseau de prédiction sont copiés sur le réseau cible. Cela conduit à une formation plus stable car il laisse la fonction cible fixe (pendant un certain temps).

- **Replay memory** : Dans un environnement en ligne, la mémoire de lecture [30] conserve toutes les informations pour les dernières N_{replay}, où l'expérience est recueillie en suivant une politique ε-greedy. Pour effectuer la relecture de l'expérience, on stocke les expériences de l'agent $\mathbf{e_t = (s_t, a_t, r_t, s_{t+1})}$.

Conclusion

Dans ce chapitre on a donné une brève définition de l'apprentissage automatique et ses différents types, après on s'est focalisé sur l'apprentissage par renforcement. On a consacré ce chapitre pour la description des techniques de l'apprentissage par renforcement et ses éléments essentiels ainsi que ses différents algorithmes de programmation.

De même, nous avons essayé de généraliser le problème d'apprentissage par renforcement basé sur le passage de l'état « s » à l'état « s' » en effectuant l'action « a » selon la stratégie π, et il est caractérisé par la fonction de valeur V et un signal de renforcement « r » juge l'action « a » effectué par l'agent qui doit maximiser ce signal.

Dans ce travail, nous nous intéressons au deep Q-learning, au Monte Carlo et au Q-learning.

Dans le prochain chapitre, on va définir l'internet des objets (IdO), son historique, les obstacles et les défit d'IdO.

Chapitre II : Internet des Objets

Introduction

De nos jours, nous vivons entourés de dispositifs électroniques, à la maison, au travail ou dans d'autres environnements, même dans le corps humain [31] ce qu'il nous faut des solutions pour optimiser l'utilisation de l'énergie afin d'assurer des transitions énergétiques durables de ces dispositifs.

Les technologies modernes telles que l'Internet des objets (IdO) fournissent un grand nombre d'applications dans le secteur de l'énergie, comme l'approvisionnement, le transport, la distribution, et la demande d'énergie. Alors on peut l'utiliser pour améliorer l'efficacité énergétique renouvelable et réduire la consommation d'énergie.

Dans ce chapitre, nous définirons ce qu'est l'internet des objets. Les cas d'utilisation d'IdO, les défis et les obstacles posés par IdO, et finalement les travaux existant pour la conservation énergétique. Ces travaux relèvent du développement des techniques intelligentes de gestion de l'énergie au niveau du système.

Dans ce chapitre, nous présenterons l'internet des objets. Tout d'abord, nous présenterons également l'histoire de l'IdO, puis nous définirons ce qu'est l'IdO ses cas d'utilisation ainsi les défis et les obstacles apportés par ce dernier. Enfin, nous effectuerons une étude de l'existant avec un tableau de comparaison pour ces travaux.

II.1. L'historique d'IdO :

La première idée de l'IdO est apparue il y a près de vingt ans, mais les technologies qui la sous-tendent existaient déjà et étaient en cours de développement depuis de nombreuses années [32].

Passons en revue l'histoire de l'évolution d'*Internet des Objets* et les technologies qui lui sont associées, par ordre chronologique :

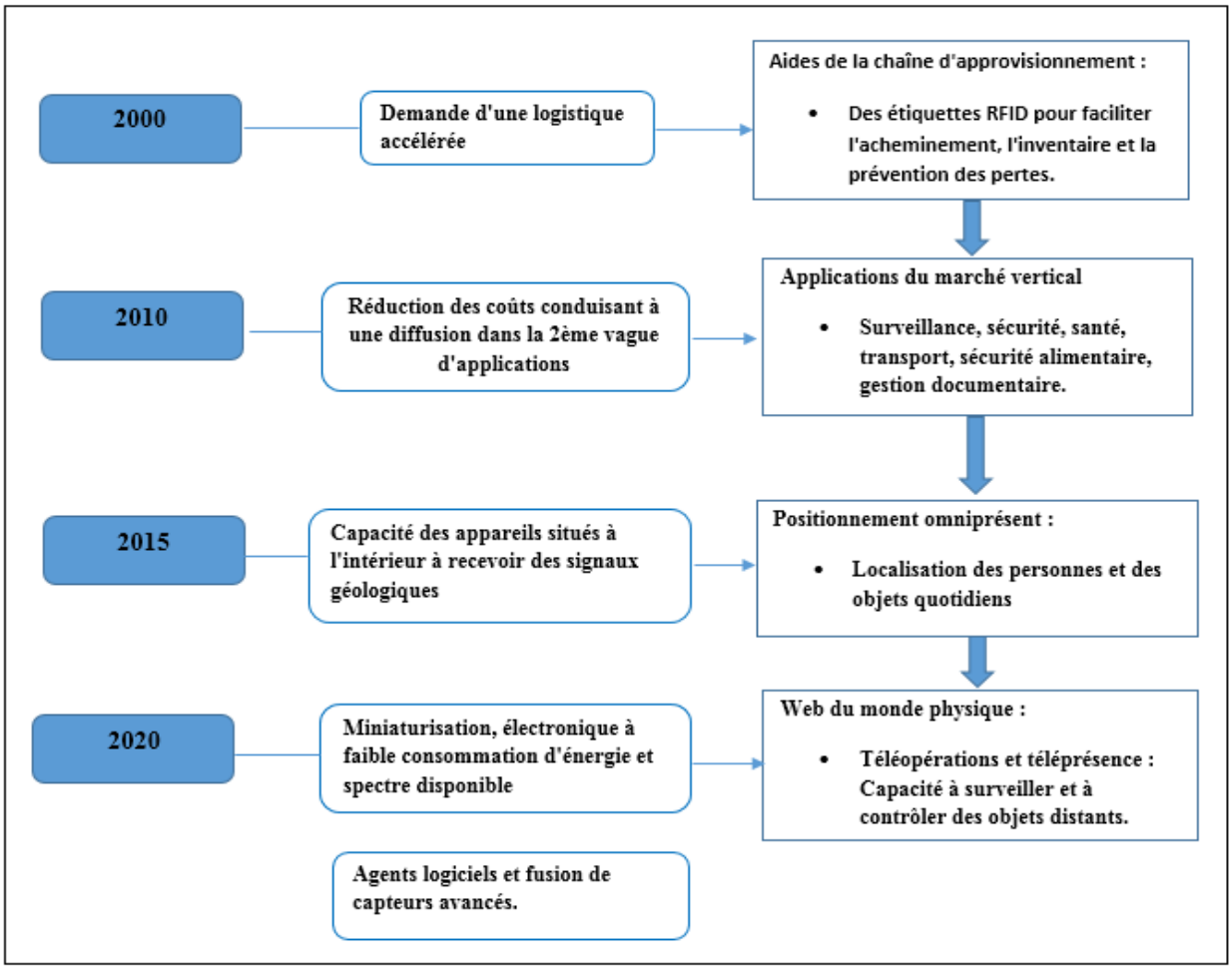
- **1969 :** L'internet est la principale technologie de l'IdO. Il se présente sous forme du réseau d'agences de projets de recherche avancée (APRANET), qui est principalement utilisé par les universités et les instituts de recherche pour partager les résultats de la recherche, développer de nouvelles technologies et relier les ordinateurs à de nombreux centres informatique multifonctionnels du département américain de la défense, ainsi que des secteurs public et privé [33].

- **1973** : Une autre technologie importante de l'IdO est la RFID[3]. Bien que les origines de la RFID remontent à la seconde guerre mondiale et qu'elle ait été continuellement développée tout au long des années 1950 et 1960, mais le premier brevet américain pour une étiquette RFID à mémoire réinscriptible a été reçu par Mario W.Cardullo[4] en 1973. Cependant, la même année, L'entrepreneur californien Charles Walton[5] a également reçu un brevet pour un transpondeur passif qui peut ouvrir la porte à distance [32].
- **1984 :** L'utilisation précoce de l'internet des objets sans faire baptiser. Le distributeur de coke a était connecté à l'internet pour signaler la disponibilité et la température du besoin (The "Only" Coke Machine on the Internet).
- **1990 :** La diffusion de l'internet sur les marchés et des consommateurs. Cependant, son utilisation reste limitée en raison de mauvaises performances de connectivité réseau.
- **1999 :** La communication appareil à appareil a été introduite par Bill Joy[6] dans sa taxonomie internet [34], et le terme *Internet des Objets* a été utilisé pour la première fois par Ashton [35].De plus, la création d'un centre d'identification au MIT a contribué au développement de la technologie RFID pour produire une puce peu couteuse qui peut stocker des informations et être utilisé pour connecter des objets à internet [36].
- **A partir de 2000 :** A la suite de la numérisation, la connectivité internet est devenue la norme pour de nombreuses applications, toutes les entreprises et tous les produits ont dû être en ligne et partager des informations en ligne. Cependant, ces appareils sont encore principalement des appareils sur internet qui nécessitent plus d'interaction humaine et de la surveillance via des applications et des interfaces dit que la figure 5 montre [32].

Le véritable potentiel de l'IdO vient juste de commencer à se matérialiser- lorsque la technologie invisible fonctionne dans les coulisses et répond de manière dynamique à nos attentes ou à notre besoin d'agir et de se comporter.

[3] RFID est une méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés « radio-étiquettes ».
[4] Mario Cardullo est un inventeur, il a déposé le premier brevet pour une balise d'identification radio passive, ancêtre du système RFID. Il est né en 1957, diplômé de l'école Tandon d'ingénierie de l'Université de New York
[5] Charles Walton est le premier titulaire de brevet pour le dispositif RFID. Il a obtenu dix brevets au total pour divers dispositifs liés à la RFID, y compris sa clé de 1973 pour un "identificateur portable émettant des radiofréquences".

[6] Bill Joy est un informaticien américain. Il est notamment connu pour son travail de développement du système d'exploitation Unix BSD et pour avoir été l'un des cofondateurs de Sun Microsystems en 1982.



II.Figure 5.Feuille de route de l'internet des objets à partir de 2000 [32]

II.2. Définition d'IdO :

Au cours des dix dernières années, il y a eu de nombreux développements dans la définition de l'internet des objets, sur la base des dernières technologies à l'époque et de la gamme d'applications fournie [37] . Différents chercheurs et scientifiques ont défini le terme *internet des objets* à leur manière, certains se concentrant sur les objets, les appareils, internet et les protocoles internet, tandis que d'autres se concentrent sur les processus de communication impliqué [32].

Cependant, il n'y a pas une définition universellement acceptée, c'est pourquoi nous fournissons les définitions les bien connues et acceptées suivantes :

- **L'Internet Architecture Board**[7] (IAB) definit l'IdO comme un service de communication. Selon eux, le terme *internet des objets* représente un ensemble de grands nombres d'appareils intégrés qui fournissent des services de communication basées sur des protocoles internet. Ces appareils sont souvent appelés *objets* ou *objets intelligents*, c'est-à-dire ces objets communiquent entre eux et nécessitent généralement d'invention humaine [38].
- **Le Magazine IEE Communications**[8] relie l'internet des objets au service de Cloud**.**[9] Ils définissent l'internet des objets comme une architecture dans laquelle chaque objet est identifié de manière unique sur internet. Plus précisément, l'internet des objets vise à réaliser l'interaction entre les objets et les applications dans le Cloud en utilisant la communication de machine à machine (M2M), fournissant ainsi déférentes applications et services pour éliminer le décalage entre le monde physique et le monde virtuel.
- La définition donnée par **les dictionnaires Oxford** est très précise et met l'accent sur l'utilisation de l'internet comme moyen de connexion entre les appareils. Ils définissent l'IdO comme « l'interconnexion via l'internet des systèmes informatiques intégrés dans des objets quotidiens, leur permettant d'envoyer et de recevoir des données »[39].
- **Forrester**[10] estime que l'internet des objets est un environnement intelligent qui peut fournir des services dans différents domaines tels que l'éducation, l'administration, la santé, les transports,...etc. à l'aide des technologies de l'information et de la communication [40].

Cependant, la définition distincte est que l'Internet des objets est un ensemble d'objets, de choses, d'équipements, de technologies et de protocoles qui vont changer l'ensemble du processus de communication. Cela peut être réalisé grâce par un cadre unifié qui comprend

[7] Un comité de chercheurs et de professionnels qui gère l'ingénierie et le développement technique liés à l'Internet.

[8]Un magazine mensuel publié par l'IEEE Communications Society qui traite de tous les domaines des communications.

[9] http://www.comsoc.org/commag/cfp/internet-thingsm2m-research-standards-next-steps.

[10] Une entreprise indépendante qui fournit à ses clients des études de marché sur l'impact des technologies dans le monde des affaires.

l'informatique universelle, le cloud computing, l'analyse de données et la représentation / visualisation des connaissances [41].

Ainsi, d'un point de vue technologique, l'IdO est défini comme des machines intelligentes qui interagissent et communiquent avec d'autres machines, objets, environnements et infrastructures, ce qui génère des volumes de données et transforme ces données en actions utiles qui peuvent *commander et contrôler* les choses et rendre la vie beaucoup plus facile pour les êtres humains.

II.3. Les cas d'utilisation d'IdO :

Lorsque les appareils peuvent détecter et communiquer sur l'internet, ils peuvent aller au-delà du traitement local intégré pour accéder et utiliser des nœuds de super information distante. Donc, ils vont être capables d'effectuer des analyses plus sophistiquées, de prendre des décisions et de répondre rapidement aux besoins locaux (souvent sans intervention humaine).

Voici quelques cas les plus courants d'IdO [42]:

II.3.1. Suivi/suivi à distance et commandement, contrôle et routage (TCC&R) :

Il s'agit de fonctions de *suivi/surveillance à distance* : réaliser des commandes, de contrôle pour des tâches et des processus qui sont aujourd'hui effectués manuellement. Par exemple, aujourd'hui, la plupart des foyers utilisent le processus manuel pour allumer et éteindre certaines lumières, machine à laver, télévision, etc. et régler la zone de température. À l'avenir, ces équipements et de nombreux autres types d'équipement autonomes deviendront *intelligents* grâce à une identification unique. Ces appareils intelligents peuvent être ensuite connectés par une communication filaire ou sans fil, ce qui permet à un utilisateur de surveiller sa maison ainsi que de modifier et contrôler les différentes tâches de ses équipements.

II.3.2. Suivi des actifs :

Une extension de ce type de services est le *suivi des actifs*, qui se fait aujourd'hui par code à barres et par diverses étapes manuelles, mais à l'avenir, elle s'appuiera sur des étiquettes intelligentes (NFC[11] et la RFID) pour suivre globalement toutes sortes d'objets, de manière interactive. Dans un scénario futur, un utilisateur pourrait utiliser Google Earth pour suivre n'importe quel objet muni d'une étiquette RFID. Par exemple, notre réfrigérateur peut garder

[11] La communication au champ proche est une technologie permettant d'échanger des données entre un lecteur et n'importe quel terminal mobile compatible ou entre les terminaux eux-mêmes

la trace de nos produits alimentaires étiquetés et indiqué à l'application de notre smart phone qu'on n'a pas assez de certains articles.

II.3.3. Contrôle et optimisation des processus :

C'est quand des différentes classes de capteurs sont utilisées pour la surveillance et pour fournir des données afin qu'un processus puisse être contrôlé à distance. Cela pourrait être aussi simple que l'utilisation de caméras (les nœuds de détection dans cet exemple) pour positionner des boites de différentes tailles sur une chaine de transport afin qu'une étiqueteuse puisse leur appliquer correctement des étiquettes. Cette tâche peut être effectuée en temps réel en envoyant les données à un ordinateur distant, en les analysants et en ramenant une commande sur la ligne afin que diverses actions de contrôle puissent être prises pour améliorer le processus (sans aucune intervention humaine).

II.3.4. Allocation et optimisation des ressources :

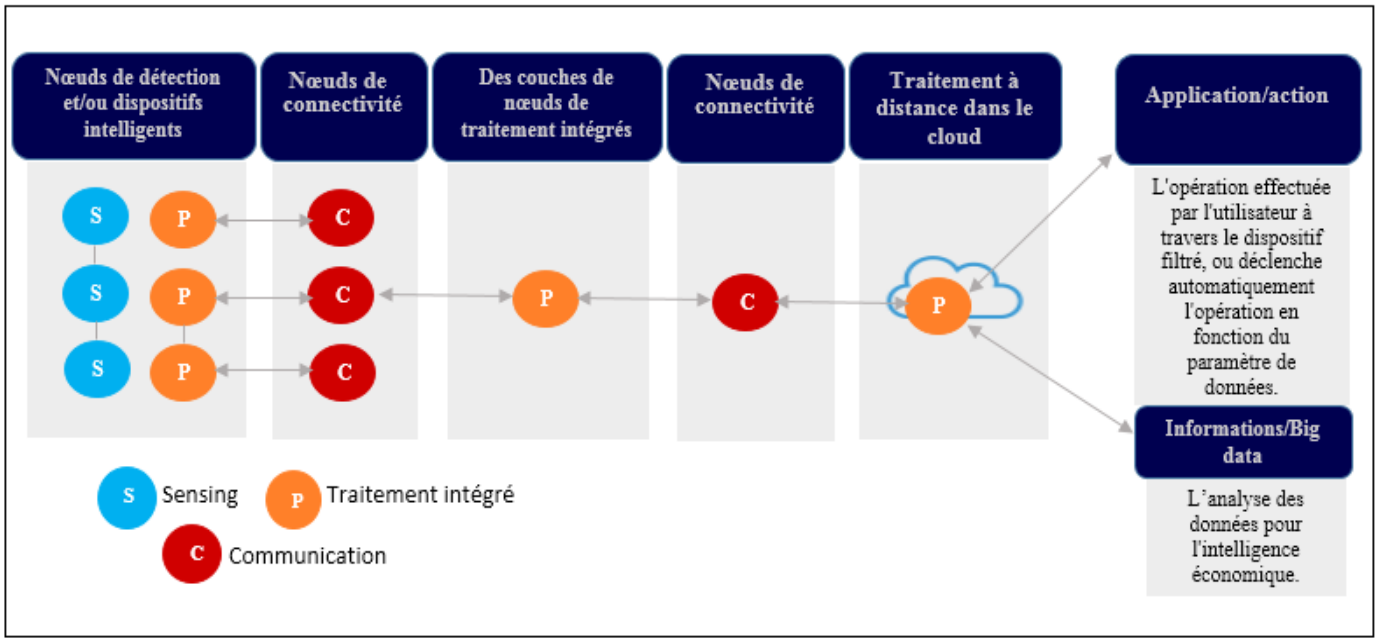
Le marché de l'énergie intelligente [43] fournit un exemple idéal de ce cas d'utilisation : le mot *énergie intelligente* désigne essentiellement l'accès aux informations sur la consommation d'énergie et la réaction à ses informations pour optimiser l'allocation des ressources (utilisation de l'énergie). Par exemple, dans le cas d'un ménage, une fois les résidents savent qu'ils ont utilisé leur machine à laver pendant les heures de pointe (lorsque le réseau est le plus contraint et le cout de l'électricité est élevé), ils peuvent adapter leur comportement et laver leur linge en dehors des heures de pointe, ce qui leur permet d'économiser de l'argent et d'aider la société de services publique à faire face à la demande de pointe.

II.3.5. Automatisation et optimisation des décisions en fonction du contexte :

Cette catégorie est la plus fascinante, car elle concerne le contrôle de facteurs inconnus (environnement, interaction entre machines,...etc.) et le fait de faire prendre aux machines des décisions aussi humaines que possibles. Il existe toute une série de nouvelles technologies disponibles aujourd'hui et en cours de développement qui pourrait permettre par exemple aux véhicules de communiquer entre eux ainsi qu'avec une unité de contrôle centrale. Ces véhicules intelligents pourraient également détecte la route et les panneaux de signalisation, et grâce au GPS et à une liaison de communication, ils vont être capables par exemple d'éviter de trafic entrant, ou des accidents dans un virage.

Parmi les exigences communes à tous les cas d'utilisation ci-dessus, on peut citer :

- Capacité de détection et de collection de données (nœuds de détection/sensing nodes).
- Capacité de traitement local intégré (nœuds de traitement local intégré/local embedded processing nodes).
- Capacité de communication filaire et/ou sans fil (nœuds de connectivité/ connectivity nodes).
- Logiciels pour automatiser les tâches et permettre de nouvelles classes de services.
- Capacité de traitement embarqué à distance par réseau/nuage (nœuds de traitement embarqué à distance/remote embedded processing nodes).
- Une sécurité totale sur le trajet du signal.



II.Figure 6.Vue fonctionnelle des technologies de l'Internet des objets [42]

Un exemple illustratif pour la figure ci-dessus :

On va prendre l'exemple de l'automatisation industrielle (application d'étiquettes sur des boites). Une caméra détecte des informations à l'aide d'un capteur à dispositif à couplage de charge (CCD) (nœud de détection), les données collectées sont ensuite communiquées à un processeur/contrôleur intégré (nœud de traitement intégré) à l'aide d'une technologie de communication câblé ou sans fil (nœud de connectivité), une décision est prise par le serveur distant(nœud de traitement intégré distant) et communiqué (nœud de connectivité), ce qui provoque une action mécanique qui corrige la situation [42].

Remarque :

Les nœuds de détections (sensing nodes) porteront tout un ID unique et pourront être contrôlés séparément via une topologie de commande et de contrôle à distant. Les types de nœuds de détection nécessaire pour l'IdO varient considérablement, en fonction des applications concernées. Ils peuvent comprendre un système de caméra pour la surveillance des images, des compteurs d'eau ou de gaz (pour l'énergie intelligente), ...etc.

Les nœuds de traitement intégrés sont au cœur de l'IdO. La capacité de traitement local est le plus souvent fournie par des microcontrôleurs/microprocesseurs hybrides (MCU/MPU) ou des dispositifs MCU intégrés, qui peuvent fournir le traitement embarqué "*en temps réel"* qui est une exigence clé de la plupart des demandes d'IdO.

II.4. Les défis et les obstacles posés par l'IdO :

Malgré la récente croissance de nombreuses applications IdO, il n'en est qu'à ses débuts. Par conséquent, il existe certains problèmes et des défis qui empêchent actuellement l'internet des objets d'atteindre ses objectifs de conception :

II.4.1. Systèmes d'adressage :

L'identification unique des objets est un enjeu clé pour le fonctionnement et le succès des applications IdO. Les applications IdO doivent classer, gérer et contrôler de manière unique des milliers d'appareils via internet. Les caractéristiques les plus importantes de la création d'une adresse unique sont la fiabilité, l'unicité, l'évolutivité et la durabilité [41].Ces appareils intelligents ont besoin d'une adresse unique et approprié pour pouvoir communiquer entre eux et faire partie d'internet. Le protocole de version 4 (IPv4) utilise des adresses de 32 bits et ne permet l'utilisation que de 4.3 milliards d'adresses IP, ce qui est presque insuffisant. La prochaine génération d'IPv6 utilise des adresses de 128 bits et dispose d'une grande quantité de ressources de $3.4*10^{38}$ (ou 340 billions de billions d'adresses) [44].

Les auteurs de [45] ont proposé un schéma d'identification et d'adressage pour les dispositifs IdO dans lequel ils utilisaient un algorithme d'attribution d'adresses distribuées pour mettre en œuvre l'identification automatique des nœuds IdO.

II.4.2. Données importantes (Big Data) :

L'internet des objets utilise une quantité massive de données collectées et agrégées via des objets intelligents et constitue l'une de ses caractéristiques les plus frappantes. Il est nécessaire de développer des techniques pour transformer ces données en connaissances

utilisables. Tous les deux ans, la taille des données doublera et devrait atteindre 44 ZettaOctets au cours des quatre prochaines années [46].

a. Les "5V" :

Les "5V" (valeur, vitesse, volume, variété et véracité) sont des problèmes importants dans les applications IdO [47]:

- **Vitesse :** Fait référence à la vitesse de collecte, de transmission et de traitement des données. La vitesse de traitement des données dépend du type de demande. Pour certaines applications, les données entrant peuvent être traitées en peu de temps, tandis que pour d'autres applications, un traitement en temps réel est nécessaire, comme un programme d'analyse.
- **Variété** : Fait référence à différents types de données collectées par des terminaux tels que des téléphones intelligents, des machines, des capteurs, etc. Le contenu des données n'est pas structuré et a différents types, tel que l'audio, la vidéo, l'image, le format XML, etc. Diverse données doivent être organisées et traitées de manière significative et cohérente.
- **Véracité :** Signifie s'assurer que les données collectées et stockées sont exactes. Cela inclure le filtrage des données indésirables ou corrompues pour améliorer la qualité de l'application.
- **Volume :** C'est la quantité de tous les types de données collectées, stockées, extraies et mises à jour à partir de différentes ressources. L'internet des objets a créé une énorme quantité de données qui connait une croissance exponentielle.
- **Valeur :** Une fois que les données volumineuses sont collectées avec précision, l'étape suivante consiste à en tirer la valeur. Par conséquent, divers algorithmes tels que l'extraction de caractéristiques, l'analyse des tendances à l'aide de l'IA peuvent prendre des décisions en connaissance de la cause dans les délais impartis, ce qui est un autre problème.

Les auteurs de [48] se concentrent sur les données en temps réel des appareils IdO dans les bâtiments intelligents. Ce cadre propose une nouvelle technologie d'analyse et de stockage des données à haut débit générées par les capteurs.

II.4.3. Consommation d’énergie :

L’internet des objets a créé des milliards d’appareils et des réseaux très diversifiés connectés à internet. L’énergie est considérée comme une ressource clé pour les appareils intelligents de l’IdO car la plupart des applications sont alimentées par des batteries ou utilisant une technologie de récupération d’énergie. Donc, il n’est pas prudent de gaspiller de l’énergie en transmettant des données et surchargeant les protocoles existants (tels que http, tcp, etc.). Par conséquent, la conception d’une architecture de réseau économe en énergie et d’un mécanisme de routage intelligent reste un défi majeur pour les réseaux IdO [49].

L'algorithme proposé par [50] a révélé l'efficacité et l'efficience en matière de consommation d'énergie et de temps de réponse du service.

II.4.4. Hétérogénéité des dispositifs/liens :

Une autre caractéristique importante de la vision de l'IdO est la variété des dispositifs et des liens puisqu'il fonctionnera sur différents ensembles de suites de protocoles, de formats de données, etc. Dans les WSN, la plupart des capteurs sont homogènes, c'est-à-dire qu'ils ont la même puissance, la même communication et la même capacité de calcul. L'IdO met en œuvre une grande variété de réseaux, de liens et de connectivité de dispositifs pour fournir différents services. Ainsi, la nature hétérogène des liens et des objets jouent un rôle essentiel dans l'interconnexion des dispositifs IdO et ajoute donc un défi unique à relever [47].

Dans [51], une architecture est présentée pour l'hétérogénéité des dispositifs et des réseaux basés sur les techniques SDN-Docker.

II.4.5. Sécurité :

Pendant de nombreuses années, les problèmes de sécurité ont été l'un des principaux problèmes du réseau. Ainsi, la sécurité, la confidentialité et la confiance sont également des facteurs essentiels pour les applications IdO. Lorsque les paquets de données sont acheminés vers le destinataire final sur Internet via différents liens et appareils, des mesures doivent être prises pour maintenir la confidentialité et l'intégrité des données. De plus, la plupart des appareils d’IdO sont des appareils à faible consommation d'énergie, de sorte que les solutions de chiffrement établies ne peuvent pas être directement appliquées au domaine d’IdO[47].

On peut distinguer quatre grands problèmes en matière de sécurité de l'IdO :

- **Confiance et intégrité des données :** Ceci permet de s'assurer que les données n'ont pas changé entre la détection des données et l'arrivée à la destination finale. Cela

implique également la vérification des données et la validation des certificats de vérification.

- **Des milliers de points de vulnérabilité :** Chaque appareil connecté à l'internet des objets présente des risques potentiels. Cela conduit à des questions : Dans quelle mesure l'organisation a-t-elle confiance dans l'intégrité des données collectées et envoyées ? Comment s'assurer que les données n'ont pas été modifiées ou détruites ?
- **Protection des données :** la loi devrait viser à protéger et à contrôler les données personnelles et organisationnelles collectées par des capteurs ou des applications et stockées dans le cadre du système d'archivage.
- **Confidentialité des données :** Il s'agit d'empêcher les données de fuir dans l'environnement IdO. Par exemple, toute entité logique ou physique peut se voir attribuer une adresse unique et communiquer automatiquement sur le réseau.

Un nouveau concept de protection dynamique pour la sécurité de l'IdO a été mis en place en [52].

II.4.6. Qualité de service(QoS) :

Dans de nombreuses applications, les données collectées doivent être livrées à la destination prévue dans un certain délai de temps, sinon leur valeur diminuera. Grâce à des services différenciés et à des retards de réseau, à la perte de paquets de données et à la gestion des paramètres de bande passante, les exigences de qualité de service peuvent être satisfaites. Ces exigences deviennent les secrets de la réussite des services de bout en bout. C'est pourquoi il est nécessaire de rechercher et de stabiliser la qualité de service pour la mise en œuvre, l'optimisation et la gestion[47].

Dans [53], les auteurs ont proposé un modèle général pour soutenir le déploiement d'une infrastructure cloud multi-composants pour l'IdO tout en tenant compte de la qualité de service. Le modèle proposé introduit une qualité du système d'exploitation appropriée pour le dispositif de brouillard.

II.4.7. Moyens de transmission (TM) :

Le moyen de transmission est le chemin physique qui établit la connexion et transmet les données de l’expéditeur au récepteur. Les réseaux IdO utilisent différents types de technologies

pour transmettre ou recevoir des données, tels que RFID, Bluetooth, LoRaWAN[12], Sigfox[13],...etc. L'internet des objets présente également des problèmes traditionnels liés au moyen de transmission (par exemple, taux d'erreur élevé, bande passante, etc.). Chaque moyen de transmission nécessite une énergie spécialisée, du matériel de réseau, et la largeur de bande passante doivent être compatible avec ce moyen. Par conséquent, l'optimisation de la TM est un défi pour soutenir et prolonger la durée de vie de réseau dans les applications IdO. De plus, actuellement, l'intégration des applications dans l'infrastructure du réseau se concentre uniquement sur la mise en œuvre des fonctions, plutôt que sur la prise en compte complète des exigences de sécurité lors de la conception des applications. Cela ouvre la porte aux attaques et aux tentatives de hackers. Les experts en cybersécurité préviennent que l'internet des objets est l'une des technologies les plus vulnérables et s'attendent à des attaques plus ciblées sur les infrastructures existantes et émergentes, telles que le vol de données, les blessures corporelles, les attaques DDoS, les logiciels de rançon utilisés pour les maisons et les voitures intelligentes, etc.[47].

Selon l'étude [54], les auteurs ont discuté et analysé la capacité et la couverture de LoRaWAN et Sigfox dans une zone à grande échelle.

II.5. Etude de l'existant :

II.5.1. L'informatique de pointe intelligente pour la gestion de l'énergie basée sur l'IdO dans les villes intelligentes :

L'efficacité énergétique à long terme est devenue un problème important lors de l'utilisation de structures de réseau basées sur l'IdO. Les auteurs de [55] ont exposé spécifiquement le problème de planification énergétique pour la réponse à la demande dans le scénario de réseau intelligent. Donc, ils ont présenté un système de gestion de l'énergie basé sur l'Internet des objets qui utilise l'informatique de pointe avec le DRL (Deep Reinforcement Learning) pour les villes intelligentes.

Ce système proposé, il permet non seulement de développer de nouveaux services à haute valeur énergétique, mais aussi de faciliter de manière intelligente l'intégration de diverses sources d'énergie et le contrôle automatique des opérations.

[12] LoRaWAN est un protocole de télécommunication permettant la communication à bas débit, par radio, d'objets à faible consommation électrique communiquant selon la technologie LoRa et connectés à l'Internet via des passerelles, participant ainsi à l'Internet des objets.
[13] Le premier fournisseur mondial de services pour l'Internet des objets (IoT).Un réseau 0G mondial pour connecter votre monde physique à l'univers numérique et à la transformation de l'industrie de l'énergie.

Pour que les auteurs atteint les résultats qu'ils veulent ; ils ont introduit un modèle logiciel du système proposé, la promotion de la technologie informatique de soutien avancée est introduite. Ensuite, afin de faire face à l'intermittence et à l'incertitude de l'offre et de la demande d'énergie urbaine, ils ont proposé un plan de planification énergétique basé sur la DRL pour atteindre des objectifs à long terme. Ils ont analysé l'efficacité du plan énergétique avec et sans serveurs de pointe. Grâce à leur explications, ils ont observé que le schéma qu'ils ont proposé permet de réduire les coûts énergétiques tout en entraînant moins de retard que le schéma traditionnel.

II.5.2. Optimisation de la consommation d'énergie dans un réseau de capteurs sans fil (WSN) à l'aide d'un protocole de routage sensible à la position (PRRP) :

Le WSN est un réseau sans fil composé de petits nœuds des capacités de détection, de calcul et de communication sans fil alimentés par de très petites piles qui ne sont pas rechargeables (la durée de vie est limitée)[56].

Les auteurs de [57] ont soulevé la question de la consommation énergétique élevé des protocoles de routages dans WSN. Alors, ils ont proposé un protocole de routage sensible à la position (PRRP) à l'aide de système de positionnement global (GPS).

Cette solution permet à tous les nouds de connaitre la localisation de leurs voisins sans envoyer des demandes d'envoi (RTC) pour savoir la position du nœud récepteur, et que les permet d'économiser une certaine quantité de ils énergie initiale.

Les auteurs, pour atteindre leurs objectifs, ils ont analysé l'implémentation de la technique du protocole de routage hiérarchique (clustering) dans WSN par rapport l'implémentation de leur approche PRRP. Ils ont trouvé comme résultat que leur approche de routage PRRP augmente la durée de vie du réseau de capteurs, ce qui peut améliorer l'optimisation et l'efficacité énergétique du réseau.

II.5.3. Modélisation et optimisation de la consommation d'énergie dans les systèmes multi-robots coopératifs :

Les auteurs de [58]ont posé le problème de la consommation énergétique des systèmes de fabrication robotisés .Alors, ils ont présenté une nouvelle méthode pour établir le programme qui minimise la consommation totale d'énergie des systèmes de fabrication robotisés.

Pour atteindre leurs objectifs, ils ont associe une loi de consommation d'énergie non linéaire à chaque opération du robot, paramétré par son temps d'exécution (cette loi est associée

de manière unique à une seule opération et au robot sélectionné, et est appelée *signature de consommation d'énergie*. Ensuite, ils ont utilisé des contraintes linéaires mixtes pour modéliser la consommation d'énergie tout au long du cycle de fabrication. Après ils ont mis en œuvre leur méthode proposé dans une application logicielle et ils l'ont démontré par une étude de cas.

Enfin, afin d'évaluer leur hypothèse de recherche, les auteurs ont mené des expériences sur un robot réel. Les résultats obtenus ont confirmé la possibilité de réduire la consommation énergétique du système robotique, mais ils ont souligné que différentes opérations ont des possibilités d'optimisation très différentes.

II.5.4. Un processeur hétérogène Dual-Core à faible consommation énergétique pour l'internet des objets :

Le travail effectué dans [59] propose un nouveau processeur dual-core hétérogène à faible consommation d'énergie, qui comprend à la fois un CoreL à très faible consommation d'énergie proche du seuil et un CoreH rapide pour répondre à ces nouvelles exigences.

En outre, un cadre optimal est proposé pour réaliser une cartographie et une planification des tâches efficaces sur le plan énergétique. Le processeur est fabriqué et sa consommation d'énergie en mode de faible puissance est aussi faible que 7,7pJ/cycle et surpasse les travaux connexes.

Une analyse détaillée de plusieurs applications réelles montre qu'il est possible d'améliorer l'efficacité énergétique jusqu'à 2,62 fois sans dépassement de délai par rapport à l'architecture à noyau unique à hautes performances.

II.5.5.Algorithme d'ordonnancement efficace sur le plan énergétique pour le protocole S-MAC dans un réseau de capteurs sans fil :

L'étude de [60] a introduit un nouvel algorithme d'ordonnancement pour résoudre le problème d'ordonnancement diversifié des nœuds frontaliers dans le S-MAC et évalué les performances par simulation.

Le protocole SMAC visait les nœuds frontaliers qui consomment une grande quantité d'énergie et adoptait différents horaires de sommeil et d'écoute. Ces nœuds passent en mode d'écoute souvent en raison d'une programmation diversifiée.

Le travail s'est concentré sur la minimisation de la consommation d'énergie des nœuds frontaliers et a donc prolongé la durée de vie des WSN.

Les résultats de la simulation ont montré que les nœuds frontaliers ont consommé moins d'énergie dans le cas des grands réseaux comme celui des petits.

II.6.Tableu comparatif des travaux :

Travail	Architecture ciblé	Solution proposé	Avantages	Inconvénients
[61]	Internet des objets.	Un système de gestion de l'énergie basé sur l'Internet des objets qui utilise l'informatique de pointe avec le DRL.	Atteindre un faible coût énergétique tout en provoquant un retard moins marqué par rapport aux systèmes traditionnels.	-Traiter les demandes en tenant compte les maisons plus prioritaires. -Un retard d'exécution car le serveur e peut pas être en position de traiter les taches de nombreux appareils en temps réel.
[57]	Réseaux de capteurs sans fil.	Un protocole de routage sensible à la position (PRRP) à l'aide de système de positionnement global (GPS).	Améliorer considérablement la durée de vie du réseau et de rendre le réseau plus économique sur le plan énergétique.	Les exigences de sécurité n'était pas prises en considération.
[58]	Réseaux de capteurs filaires.	Une méthode pour établir le programme qui minimise la consommation totale d'énergie des systèmes de fabrication robotisés.	Réduire la consommation énergétique des opérations des systèmes robotique même sans changer la séquence ou la durée du cycle.	-Les différentes opérations ont des possibilités d'optimisation très différentes. - L'application de cette méthode sur un échantillon du système robotique.
[59]	Internet des objets.	Un processeur double cœur hétérogène à faible consommation énergétique.	Atteindre une efficacité énergétique maximale et économiser beaucoup d'énergie.	-Les nœuds sont jetés une fois la durée de vie est terminé. -Le processeur purement proche du seuil manquera la date limite lorsque la charge de travail est élevée et ne peut pas être utilisé dans de nombreuses applications IdO.
[60]	Réseaux de capteurs sans fil.	un nouvel algorithme d'ordonnancement pour résoudre le problème d'ordonnancement diversifié des nœuds frontaliers dans le S-MAC.	-Cet algorithme d'ordonnancement permet d'économiser 4,70 fois plus d'énergie que les autres méthodes de programmation. -Augmenter la durée de vie des nœuds de détection. - Possibilité de transfère les	- Tous les nœuds fonctionneront selon un horaire unique et transmettront des données en même temps (Sécurité des données : un risque de perdre certain données).

			données entre plusieurs clusters virtuel.	

II.Tableau 1.un tableau comparatif des travaux.

Conclusion

Dans ce chapitre on a présenté une étude détaillée sur l'internet des objets, sa définition et ses défis. Ce chapitre concerne principalement les *concepts de base*, l'*hétérogénéité* et l'*influence du monde physique* sur l'IdO.

Dans la dernière partie de ce chapitre nous avons parlé sur les différents domaines d'application de l'IdO et l'apprentissage par renforcement pour l'optimisation d'énergie.

Dans le chapitre suivant, nous allons présenter notre solution proposée. Notre conception sera expliquée en détail à travers une transposition de notre problème, une architecture de notre système, ainsi la modélisation de notre agent intelligent.

Chapitre III : Conception

Introduction

Dans le chapitre précédent, nous avons définir l'internet des objets, son historique, les cas d'utilisation d'internet des objets et ses obstacles. À la fin on a présenté quelques travaux existant pour l'optimisation énergétique.

Dans ce chapitre, et dans un premier lieu, nous présenterons la transposition de notre problème ainsi l'architecture de notre système. Après, nous expliquerons le rôle de notre agent intelligent et les algorithmes qu'il utilise pour réaliser ses tâches.

III.1. Transposition de notre problème :

Dans notre problème on a catégorisé nos appareils selon leur consommation énergétique et par leur priorité. Alors on a distingué trois catégories :

- Des appareils à faible consommation énergétique qui ne sont pas indispensable, comme les lampes.
- Des appareils à moyenne consommation énergétique qui sont un peu indispensable, comme le TV.
- Des appareils à forte consommation énergétique qui sont indispensable, tel que la machine à laver et le réfrigérateur.

Chaque appareil consomme de l'énergie dans un intervalle de temps, à un instant donné t, la somme totale de toute l'énergie ne doit pas dépasser un certain seuil fixé par l'utilisateur (la valeur de seuil utilisé=10000 watts) :

- Chaque lampe consomme une valeur d'énergie comprise entre (15,100 watts).
- Les TVs consomment une valeur entre (200,400 watts).
- Les réfrigérateurs consomment une valeur énergétique entre (1000,2000 watts).

À chaque étape de notre environnement, l'utilisateur humain allumera et éteindra les appareils de façon aléatoire. Le rôle de notre agent intelligent est donc d'essayer de maintenir les appareils allumés ou éteints le moins possible afin de ne pas dépasser le seuil.

On a présenté notre problème tant qu'un environnement discret de façon stochastique, et observables. La politique que nous avons suivie nous indique que les appareils indispensables (le TV, le réfrigérateur) sont prioritaires pour être allumé, et pour les autres appareils non indispensables (les lampes) ne sont pas prioritaires et indispensables pour l'allumage. Donc on

commence à éteindre les appareils non indispensables en tenant compte des premières lampes qui ont être allumé.

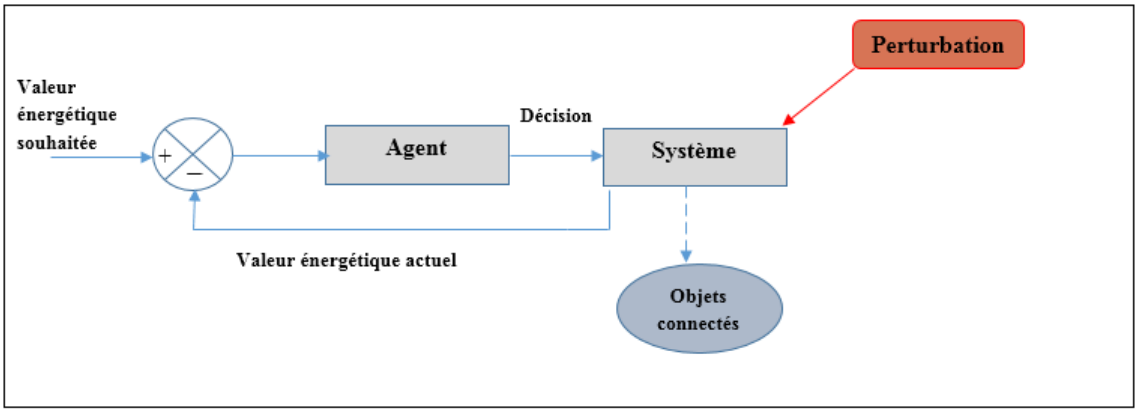
III.2. Comportement de la solution proposée :

Une maison intelligente ou un Smart Home est une résidence qui se compose d'un ensemble d'appareils et dispositifs intelligents connectés à internet, ces objets intelligents peuvent interagir et communiquer entre eux.

Ces objets sont équipés d'un capteur (sensor) qui les permit de collecter et générer des données depuis l'extérieur. Ces données sont envoyé à un agent afin qu'il analyse ses données et extraire les connaissances nécessaires. Ensuite, il va prendre les décisions. Ces décisions vont être envoyées aux actionneurs qui vont les exécuter.

Notre objectif est d'utiliser l'apprentissage par renforcement pour créer un système permettant de minimiser la consommation d'énergie pour les objets connectés.

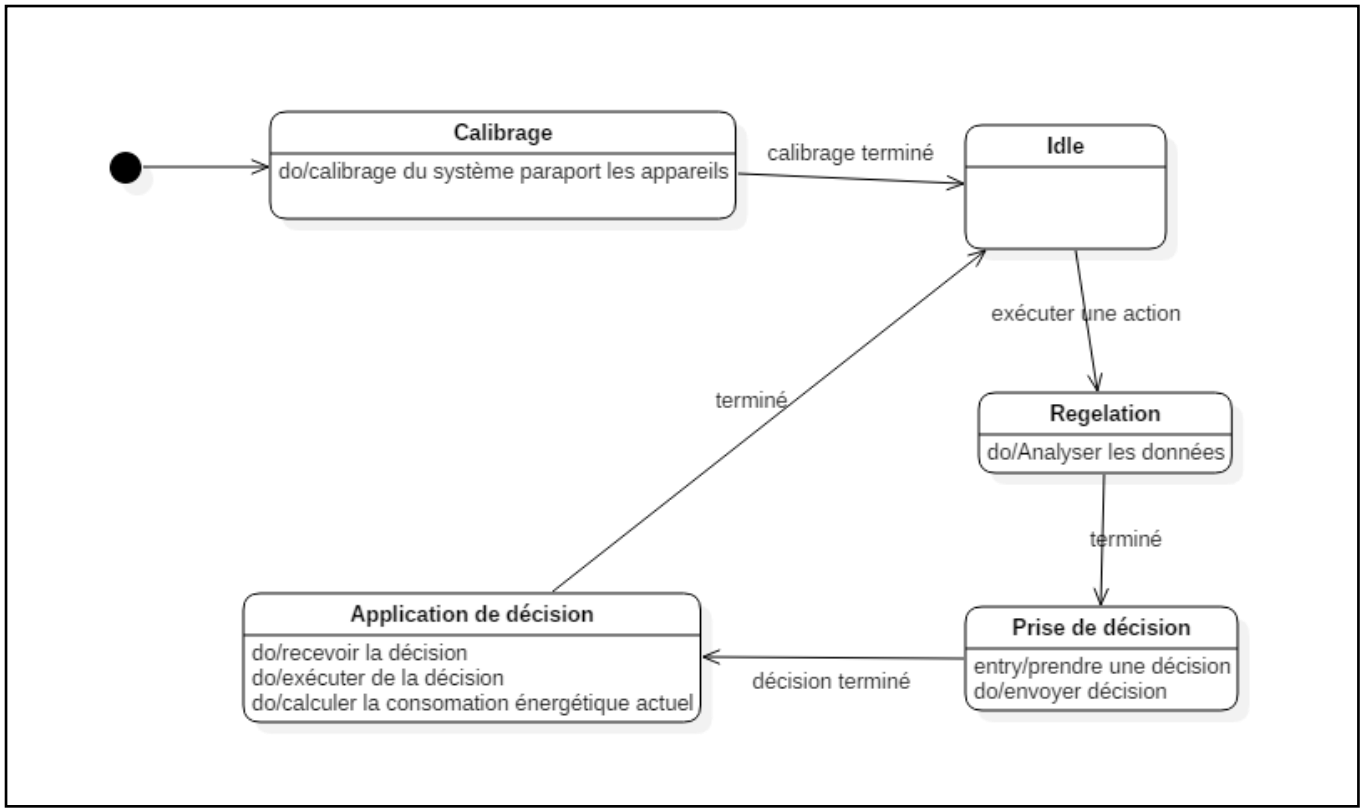
La figure suivante représente l'architecture de notre système :



III.Figure 7.Architecture du système.

III.2.1. Modélisation UML de l'évolution de l'état du système :

La figure suivante représente le diagramme d'état transition de notre solution :



III.Figure 8.Diagramme d'état transition.

Afin de résoudre notre problème, nous allons le transposer au problème de blackjack. D'un côté, nous disposons de l'utilisateur des appareils qui représentera le joueur du blackjack. De l'autre côté, nous aurons notre système qui jouera le rôle du croupier au blackjack.

Le concept du black jack est expliqué ci-après.

III.3. Le concept de Blackjack :

Le blackjack est un jeu de cartes dont le but est d'obtenir des cartes dont la somme est la plus proche possible de 21 sans dépasser ce chiffre. Ils jouent contre un croupier fixe.

Les *cartes de face* (valet, dame, roi) ont une valeur de 10 points. Les *as* peuvent compter pour 11 ou 1, et sont appelés *utilisables* à 11. Le jeu commence avec chaque joueur et croupier ayant une carte face visible et une carte face cachée. Le joueur peut demander des cartes supplémentaires (hit=1) jusqu'à ce qu'il décide d'arrêter (stick=0) ou de dépasser 21 (bust).

Une fois que le joueur a collé (stick), le croupier révèle sa carte face cachée et tire jusqu'à ce que sa somme soit de 17 ou plus. Si le croupier fait faillite, le joueur gagne. Si ni le joueur

ni le croupier ne font faillite, le résultat est déterminé par la somme qui se rapproche le plus de 21. La récompense pour la victoire est de +1, le tirage au sort est de 0 et la perte est de -1.[14]

III.4. Analogie avec le concept de Blackjack :

Dans notre système la consommation énergétique de tous les appareils ne doivent pas dépasser une valeur donnée (seuil=10000 watts), et la consommation énergétique de chaque appareil ne doit pas dépasser 2000 watts.

En allumant les appareils (on = 1) ou les éteindre (off = 0) peuvent augmenter la consommation énergétique et peut être dépassé notre seuil dépasser le seuil demandé.

Pour chaque épisode la somme énergétique des appareils allumés est calculée (sum_hand()), vérifié si elle est dépassé le seuil donné ou non (is_bust()). Ainsi les récompenses obtenues pour chaque épisode (if is_bust () reward= -1 else reward=1).

III.5. Modélisation de l'agent intelligent :

En intelligence artificielle, un agent intelligent est une entité autonome capable de percevoir son environnement grâce à des capteurs et aussi d'agir sur celui-ci via des actionneurs afin de réaliser des buts [62]. Un agent intelligent peut également apprendre ou utiliser des connaissances pour pouvoir réaliser ses objectifs. Par exemple, le *thermostat* est considéré comme étant un agent intelligent.

Notre agent intelligent ici va suivre l'un des politiques (une politique sample, politique prioritaire), en exécutant l'un des algorithmes d'apprentissage par renforcement (Monte Carlo, Q-learning et DQN) pour exécuter ses taches et prendre des décisions afin de résoudre le problème d'optimisation énergétique.

III.5.1. Les politiques :

a. La politique sample :

Dans le cadre cette politique, l'expérience de l'agent est ajoutée à un tampon de données (également appelé tampon de relecture) D, et chaque nouvelle politique « π_k » collecte des données supplémentaires, de sorte que D est composé d'échantillons provenant de $\pi_0, \pi_1, \ldots \pi_k$, et toutes ces données sont utilisées pour former une nouvelle politique mise à jour « π_{k+1} ».

[14] https://github.com/dennybritz/reinforcement-learning/blob/master/lib/envs/blackjack.py

b. La politique prioritaire :

Généralement, dans cette politique, les expériences sont recueillies en utilisant la dernière politique apprise, puis en utilisant cette expérience pour améliorer la politique. Il s'agit d'une sorte d'interaction en ligne.

Dans notre cas, à chaque itération, on prend en considération les derniers appareils allumés. Ces derniers, ont moins de chance d'être atteints dans l'itération suivante. Par contre , les appareils qui sont allumés depuis longtemps , ont plus de chance d'être éteints dans les prochaines itérations.

III.5.2. Les Algorithmes :

Les algorithmes que notre agent intelligent peut les utiliser durant l'exécution de ses tâches sont :

a. Monte Carlo :

L'algorithme de prédiction de Monte Carlo calcule la fonction de valeur pour une politique donnée en utilisant l'échantillonnage.

Les arguments :

- politique : Une fonction qui met en correspondance une observation avec des probabilités d'action.
- env : OpenAI gym.
- num_episodes : Nombre d'épisodes à échantillonner.
- discount_factor : Facteur de réduction gamma.

Le retour :

- Un dictionnaire qui établit une carte à partir de l'**état → valeur**.
- L'état est un tuple et la valeur est un flottant.

b. Q-learning :

L'algorithme Q-learning trouve la politique optimale tout en suivant une politique epsilon-greedy.

Les arguments :

- env : OpenAI.

- num_episodes : nombre d'épisodes à exécuter.
- discount_factor : Facteur d'actualisation gamma (gamma=1).
- alpha : le taux d'apprentissage (alpha=0,5).
- epsilon : Chance d'échantillonner une action aléatoire. Flotte entre 0 et 1.

Le retour :

- Un tuple (Q, durée de l'épisode).

c. Apprentissage-Q approfondi (DQL) :

L'algorithme Q-learning trouve la politique optimale tout en suivant une politique epsilon-greedy et en utilisant les réseaux de neurones.

Les arguments :

- sess : objet de la session de TensorFlow.
- env : OpenAI.
- q_estimator : l'objet estimateur utilisé pour les valeurs q.
- target_estimator : l'objet estimateur utilisé pour les cibles (targets).
- State_processor : l'objet d'un État processeur.
- num_episodes : nombre d'épisodes à exécuter.
- experiment_dir : annuaire pour sauvegarder les résumés de TensorFlow dans « replay memory ».
- replay_memory_size : taille de la mémoire de lecture
- replay_memory_init_size : nombre d'expériences aléatoires à traiter lors de l'initialisation de la mémoire de lecture.
- update_target_estimator_every : copier les paramètres de l'estimateur Q vers l'estimateur cible toutes les N étapes.
- discount_factor : facteur d'actualisation (gamma=0,99).
- epsilon_start : chance de sélectionner une action au hasard lors d'une action. Epsilon se dégrade au fil du temps et ceci est la valeur de départ (ici epsilon=1)
- epsilon_end : la valeur minimale finale de l'epsilon après la décroissance est effectuée (ici epsilon=0,1).
- epsilon_decay_steps : nombre d'étapes pour décomposer epsilon
- batch_size : taille des lots à extraire de la mémoire de lecture

- record_video_every : enregistrer une vidéo tous les N épisodes

Le retour :

Un objet Episode_Stats avec deux tableaux numériques pour les durées d'épisode et les récompenses d'épisode.

Conclusion

Dans ce chapitre, tout d'abord nous avons présenté la transposition de notre problème et on a donné l'architecture du système. Apres nous on a parlé de blackjack et l'analogie de ce concept avec notre système proposé. A la fin, nous avons parlé de notre agent intelligent et comme on détaille les différents algorithmes et les politique que notre agent a utilisé pour réaliser ces taches.

Dans le prochain chapitre, nous allons parler des déférents outils utilisé pour réaliser notre système ainsi un aperçu de quelque résultat de notre test et réalisation.

Chapitre IV : Test et Réalisation

Introduction

Dans ce chapitre, nous entamons la mise en œuvre de l'application. Après la phase de conception prédéfinie, nous entrerons dans cette phase en présentant et en définissant les outils liés à la réalisation des applications, puis nous donnerons un aperçu des différents tests réalisés.

IV.1. Description de l'environnement :

IV.1.1. Langage de programmation :

Pour développer notre application, on a choisis le langage de programmation python.

a. Python :

Python est un langage de programmation interprété, orienté objet, de haut niveau et doté d'une sémantique dynamique. Ses structures de données intégrées de haut niveau, combinées à un typage dynamique et à une liaison dynamique, le rendent très attrayant pour le développement rapide d'applications, ainsi que pour une utilisation en tant que langage de script ou de colle pour relier des composants existants entre eux. La syntaxe de Python, simple et facile à apprendre, met l'accent sur la lisibilité et réduit donc le coût de la maintenance du programme. Python prend en charge les modules et les paquets, ce qui encourage la modularité des programmes et la réutilisation du code. L'interpréteur Python et la vaste bibliothèque standard sont disponibles sous forme source ou binaire sans frais pour toutes les principales plates-formes et peuvent être distribués librement.[15]

[15] python.org

IV.1.2. Outils de développements :

Pour les outils de développement, on a travaillé avec Jupyter Notebook et Spyder

a. Jupyter Notebook :

Jupyter est une application web utilisée pour programmer dans plus de 40 langages de programmation, dont Python, Julia, Ruby, R, ou encore Scala. Jupyter est une évolution du projet IPython.

Le bloc-notes (Jupyter notebook) étend l'approche de l'informatique interactive basée sur les consoles dans une direction qualitativement nouvelle, en fournissant une application web adaptée à la saisie de l'ensemble du processus de calcul : développement, documentation et exécution du code, ainsi que communication des résultats Plateforme de test (la solution domotique).

b. Spyder :

Spyder est un environnement scientifique libre et gratuit écrit en Python, pour Python, et réalisé par et pour des scientifiques, des ingénieurs et des analystes de données. Il présente une combinaison unique de fonctionnalités avancées d'édition, d'analyse, de débogage et de profilage d'un outil de développement complet avec les capacités d'exploration de données, d'exécution interactive, d'inspection approfondie et de visualisation d'un ensemble scientifique[16].

IV.2. Plateforme de test (solution domotique) :

La domotique, appelée également parfois *smart home,* ou encore maison connectée**,** ou maison intelligente. La domotique est l'ensemble des techniques de l'électronique, de

[16] www.spyder-ide.org

physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans les maisons, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques, etc**.).**

La domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut retrouver dans les maisons, les hôtels, les lieux publics, etc.

IV.3. Présentation de l'application :

On a utilisé les bibliothèques :

- **OpenAI gym** pour créer notre environnement,
- **TensorFlow** pour les réseaux de neurones.
- **Matplotlib** pour utiliser le plotting et dessiner les graphiques.
- **NumPy** pour la création des tableaux.
- **Random** pour générer des nombres aléatoires.

IV.3.1. OpenAI gym :

Gym est une boîte à outils pour développer et comparer des algorithmes d'apprentissage par renforcement. Il ne fait aucune hypothèse sur la structure de l'agent et est compatible avec n'importe quelle bibliothèque de calcul numérique (telle que TensorFlow).La bibliothèque Gym est une collection de problèmes de tests (environnements) qui peut être utilisée pour créer des algorithmes d'apprentissage par renforcement. Ces environnements ont des interfaces partagées qui vous permettent d'écrire des algorithmes conventionnels.[17]

[17] gym.openai.com

IV.3.2. TensorFlow :

TensorFlow est la principale bibliothèque Open Source pour le développement et l'entraînement de modèles de machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine de la machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie.[18]

IV.3.3. Matplotlib :

Matplotlib est une bibliothèque Python capable de produire des graphes de qualité, comme elle peut être utilisée dans des scripts Python, le notebook Jupyter, des serveurs d'applications web.[19]

Matplotlib essai de rendre les tâches simples et de rendre possibles les choses compliquées. Avec cette bibliothèque, on peut générer des graphes, histogrammes, des graphiques à barres, des graphiques d'erreur,...etc. en quelques lignes de code.

IV.3.4. NumPy :

NumPy est le package fondamental pour le calcul scientifique en Python. C'est une bibliothèque Python qui fournit un objet de tableau multidimensionnel, plusieurs objets dérivés (tels que des tableaux et des matrices masqués), et un ensemble de fonctions pour des opérations

[18] www.tensorflow.org

[19] matplotlib.org

rapides sur des tableaux, y compris des opérations mathématiques, logiques, de tri, de sélection, d'entrées/sorties, d'opérations statistiques de base, de simulation aléatoire et bien plus encore.[20]

IV.3.5. Random :

Random est une bibliothèque Python regroupant plusieurs fonctions permettant de travailler avec des valeurs aléatoires. La distribution des nombres aléatoires est réalisée par le générateur de nombres pseudo-aléatoires Mersenne Twister, l'un des générateurs les plus testés et utilisés dans le monde informatique.[63]

IV.4. Test et réalisation :

La validation du travail s'est faite sur un environnement domotique. L'exécution doit être dans une classe main qui est la classe SmartHomeEnv.

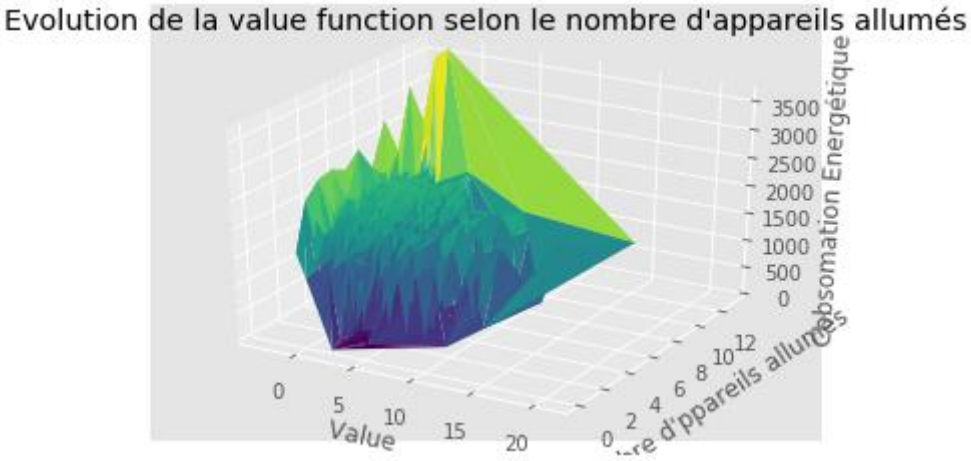
```
class SmartHomeEnv(BlackjackEnv):


    def __init__(self, lampe, tv, fridge):
        self.lampe=lampe
        self.tv=tv
        self.fridge=fridge
        self.action_space = spaces.Discrete(2)
        self.listTv=[]
        self.listFridge=[]
        self.listLampe=[]
        self.TFL()
        self._seed()
```

IV.Figure 9.la classe de notre environnement SmartHomeEnv

On va calculer la consommation énergétique de trois types d'appareils (lampe, télévision, réfrigérateur) et tester si elle dépasse le seuil ou non.

Le graphe suivant affiche le nombre d'appareils allumés, la fonction de valeur des récompenses obtenues et l'énergie totale consommée.

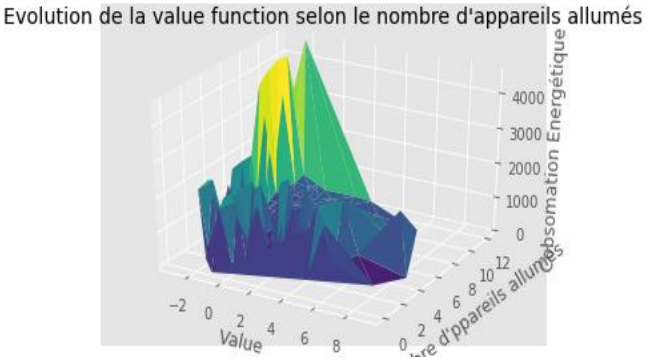
[20] numpy.org



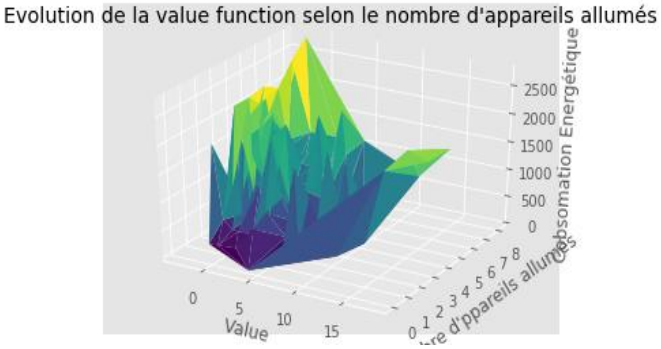
IV.Figure 10.graphe de la consommation énergétique avec 1000 épisodes

4.1. Impact du nombre d'épisodes sur l'approche Monte-Carlo :

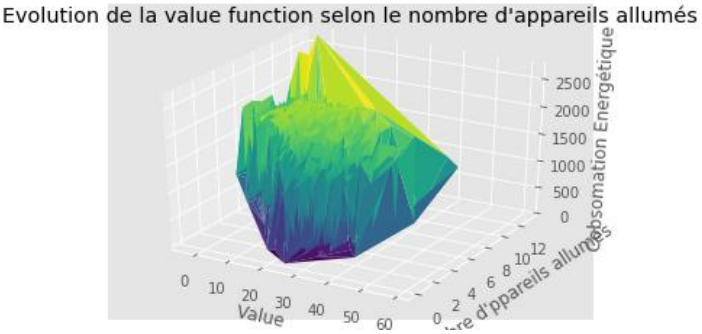
Les graphes de la figure suivante (figure 11) présentent l'évolution de la consommation énergétique avec l'augmentation du nombre d'épisodes dans la solution Monte-Carlo. Ici, le nombre d'appareils connectés est compris entre 8 et 12. On considère ici la politique prioritaire uniquement.



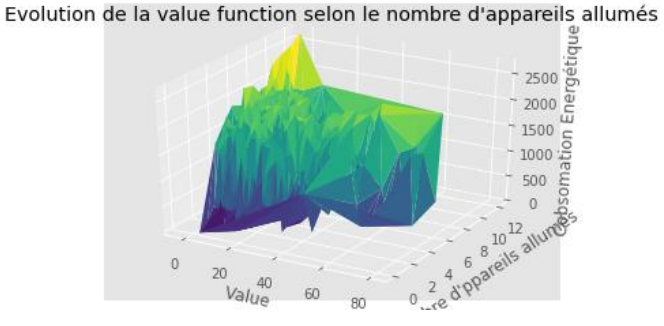
a- 500 épisodes



b- 1000 épisodes



c- 10000 épisodes



d- 50000 épisodes

IV. Figure 11.Evolution de la consommation d'énergie avec le changement du nombre d'épisodes

Discussion :

Les graphes de la figure 11 , allant de a à d , montrent l'évolution de la value function et de la consommation énergétique selon un nombre d'épisodes allant de 500 (graphe a) jusqu'à 50000 (graphe d).

Avec un nombre d'appareils fixé entre 8 et 12 (tous types d'appareils confondus), on constate que la consommation énergétique dépasse rarement le seuil fixé (2000 watts) , de plus la function value s'améliore avec l'augmentation du nombre d'épisodes. La stabilité de la consommation énergétique est constatée avec un nombre d'épisodes égal à 10000 (graphe c) et est confirmée avec un nombre d'épisodes égal à 50000 (graphe d).

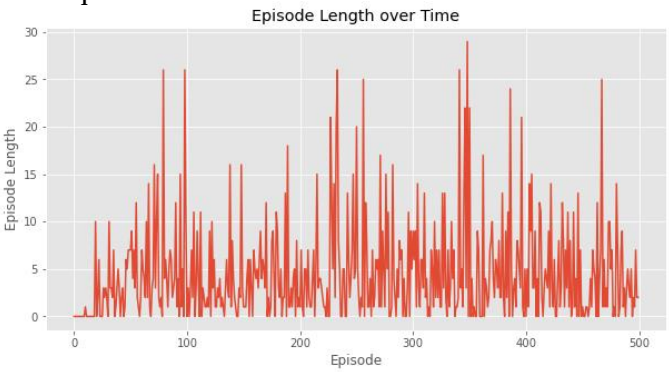
4.2. Impact du nombre d'épisodes sur l'approche Q-Learning :

Afin d'étudier l'impact du nombre d'épisodes sur l'approche Q-learning , on s'intéressera à l'étude de l'évolution du nombre d'étapes par épisode.

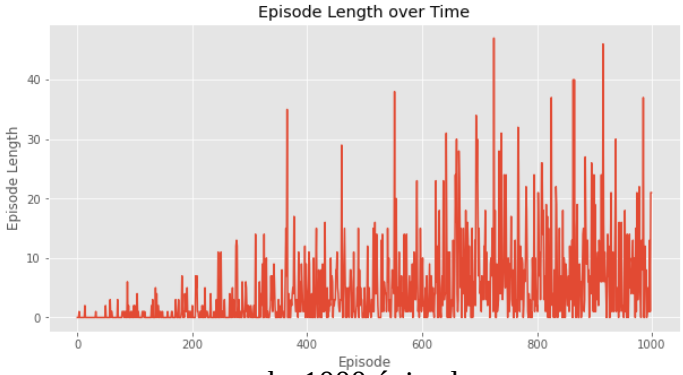
On considérera qu'à chaque fois, si la valeur consigne (seuil) de la valeur énergétique est dépassée par le système, cela veut dire que le système a perdu, et de ce fait le jeu s'interrompt. Plus le nombre d'étapes par épisode est important, plus le système est stable.

Toutefois, on limitera le nombre d'étapes à 200 par épisode. Au bout de ce nombre, on considérera que le système a gagné. (A réussi à stabiliser la consommation énergétique).

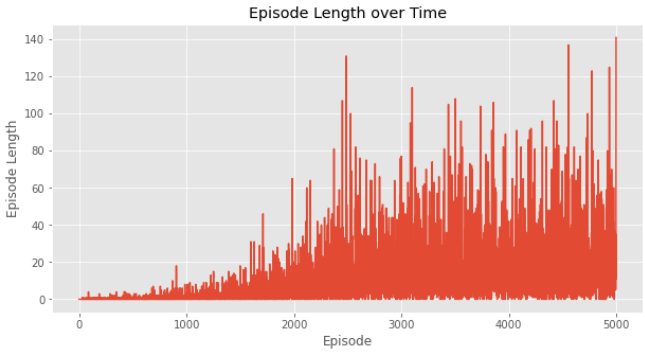
Les graphes de la figure suivante (figure 12) présentent l'évolution de la consommation énergétique avec l'augmentation du nombre d'épisodes dans la solution Q-Learning. Ici, le nombre d'appareils connectés est compris entre 8 et 12. On considère ici la politique prioritaire uniquement.



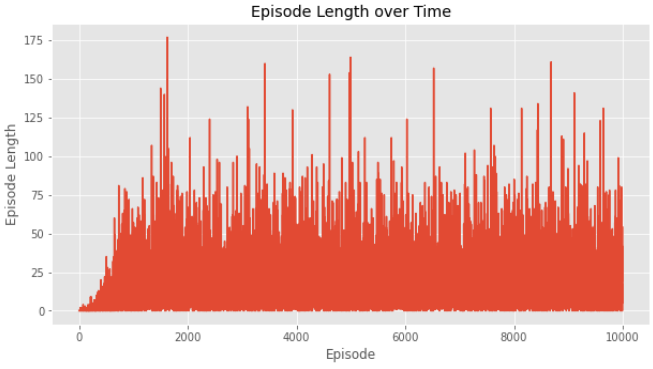
a- 500 épisodes



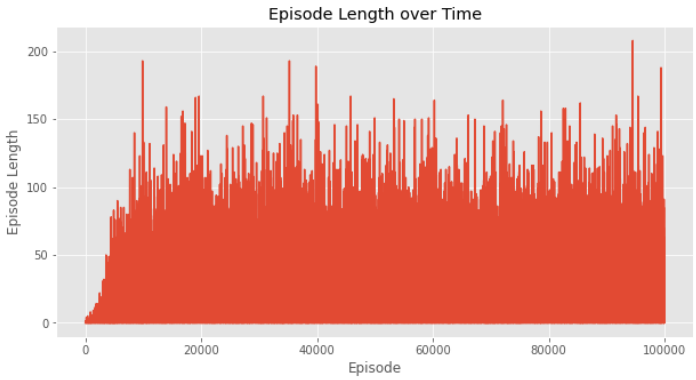
b- 1000 épisodes



c- 5000 épisodes



d- 10000 épisodes



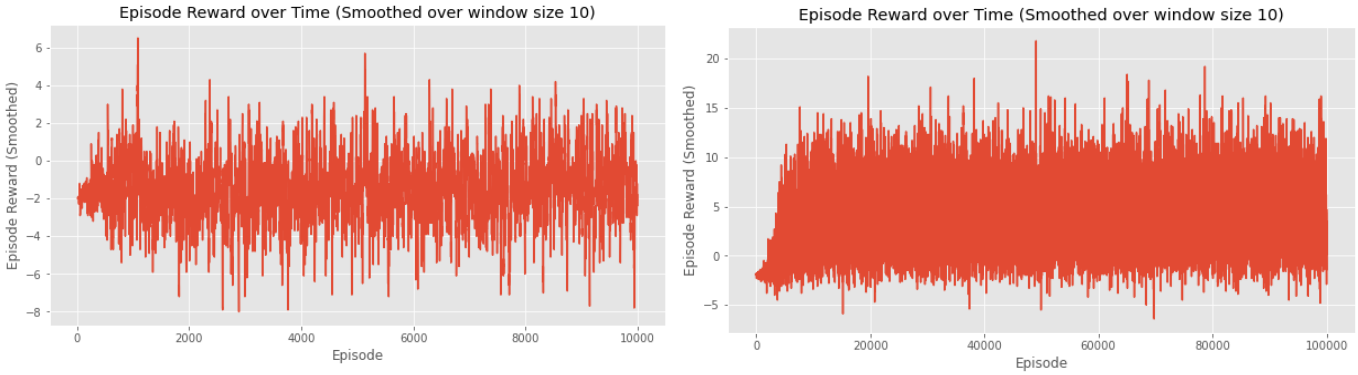
e- 100000 épisodes

IV. Figure 12.Evolution du nombre d'étapes avec le changement du nombre d'épisodes

Discussion :

Les graphes de la figure 12 , allant de a à e , montrent l'évolution du nombre d'étapes selon un nombre d'épisodes allant de 500 (graphe a) jusqu'à 100000 (graphe e).

Avec un nombre d'appareils fixé entre 8 et 12 (tous types d'appareils confondus), on constate que le nombre d'étapes augmente au fur et à mesure que le nombre d'épisode augmente. Le seuil de 100 étapes par épisode est toujours atteint au bout de 20000 épisodes (graphe e). On atteint souvent le nombre d'étapes max à partir d'un nombre d'épisode égal à 40000.Cette tendance de stabilité est confirmée avec l'évolution de la value function par rapport à l'augmentation du nombre d'épisodes. La récompense est souvent négative quand le nombres d'épisodes est en dessous de 10000 (figure 13.a) mais , une fois la barre de 20000 épisode , souvent , la valeur de récompense est positive avec des valeurs élevées (figure 13.b)



a- 10000 épisodes b- 100000 épisodes

IV. Figure 13.Evolution de la récompense avec le changement du nombre d'épisodes

Conclusion

Dans ce chapitre, nous avons présenté notre application développée et les résultats de test obtenus. Au début, nous avons présenté notre environnement de développement, les différents langages de programmation utilisée. Comme nous avons parlé de la plateforme de teste (la solution domotique. À la fin nous avons terminé avec une série de tests pour valider notre travail.

Conclusion générale

Conclusion générale

L'Internet des objets (IdO) est une technologie émergente et la consommation d'énergie est l'un des enjeux importants. Plusieurs appareils et des objets quotidiens fonctionnants sur la batterie sont connectés à l'internet et elles sont soumises à une contrainte énergétique. Motivés par ce défi, et après l'une étude des différentes approches qui était faites, nous avons proposé une nouvelle solution d'optimisation de consommation énergétique en exploitant l'apprentissage profond par renforcement.

Lors de réalisation de ce modeste travail, tout d'abord, nous avons donné une description générale à l'apprentissage par renforcement et on parler de ces déférents algorithmes. Après, nous avons parlé de l'IdO, comme nous avons fait une étude comparative de quelques travaux comparative sur le problème de la consommation énergétique.

Ensuite, nous avons donné une description détaillée à notre problème de consommation énergétique dans les maisons intelligentes (smart home), comme nous avons parlé de notre solution proposée, qu'elle était une analogie avec le concept de Blackjack. À la fin, nous avons effectué plusieurs tests pour évaluer nos approches.

On constate qu'avec l'approche Q-Learning, la value function est toujours positive et élevée au bout de 20000 épisodes. Avec l'approche Monte-Carlo, seulement 10000 épisodes sont nécessaires pour avoir plus ou moins un système stable.

Toutefois, d'autres expérimentations peuvent être effectuées. Pour mieux étudier ce problème, on propose comme perspectives :

- L'étude de l'impact du nombre d'appareils sur le comportement du système (scalabilité)
- L'utilisation d'autres politique à part la politique prioritaire
- L'utilisation d'autres approches d'apprentissage par renforcement, notamment le Deep Q-Learning.
- L'amélioration de la simulation de l'environnement pour le rendre plus réaliste.

References Bibliographies

[1] F. Samie, L. Bauer, and J. Henkel, "IoT technologies for embedded computing: A survey," in *2016 International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS 2016*, 2016, pp. 1–10, doi: 10.1145/2968456.2974004.

[2] W. Yu *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[3] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in internet of things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4921–4934, 2019, doi: 10.1109/JIOT.2019.2893866.

[4] A. G. Barto and R. S. Sutton, "Chapter 19 Reinforcement learning in artificial intelligence," in *Advances in Psychology*, vol. 121, no. C, Elsevier, 1997, pp. 358–386.

[5] N. Abramson, D. Braverman, and G. Sebestyen, *Pattern recognition and machine learning*, vol. 9, no. 4. springer, 1963.

[6] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[7] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *Found. Trends Mach. Learn.*, vol. 11, no. 3–4, pp. 219–354, 2018, doi: 10.1561/2200000071.

[8] GURU 99, "Supervised vs Unsupervised Learning: Key Differences," *2020*, 2020. [Online]. Available: https://www.guru99.com/supervised-vs-unsupervised-learning.html. [Accessed: 26-Nov-2020].

[9] R. Bellman, "Dynamic Programming," 1957.

[10] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-13, no. 5, pp. 834–846, 1983, doi: 10.1109/TSMC.1983.6313077.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[12] R. Bellman, "A Markovian Decision Process," *Indiana Univ. Math. J.*, vol. 6, no. 4, pp. 679–684, 1957, doi: 10.1512/iumj.1957.6.56038.

[13] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1, no. 2. Athena scientific Belmont, MA, 1995.

[14] V. Gokul, P. Kannan, S. Kumar, and S. G. Jacob, “Deep q-learning for home automation,” *Int. J. Comput. Appl.*, vol. 152, no. 6, pp. 1–5, 2016.

[15] F. Kunz, “An Introduction to Temporal Difference Learning,” in *Citesee*, 2000, pp. 21–22.

[16] S. Ravichandiran, “Elements of RL - Hands-On Reinforcement Learning with Python,” 2018. [Online]. Available: https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788836524/1/ch01lvl1sec13/elements-of-rl. [Accessed: 27-Nov-2020].

[17] S. Ravichandiran, “Types of RL environment - Hands-On Reinforcement Learning with Python,” 2018. [Online]. Available: https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788836524/1/ch01lvl1sec15/types-of-rl-environment. [Accessed: 25-Oct-2020].

[18] R. W, “Planning by Dynamic Programming: Reinforcement Learning,” 25-Nov-2019. [Online]. Available: https://towardsdatascience.com/planning-by-dynamic-programming-reinforcement-learning-ed4924bbaa4c. [Accessed: 07-Nov-2020].

[19] A. CHOUDHARY, “Dynamic Programming In Reinforcement Learning,” 18-Sep-2018. [Online]. Available: https://www.analyticsvidhya.com/blog/2018/09/reinforcement-learning-model-based-planning-dynamic-programming/. [Accessed: 09-Nov-2020].

[20] C. B. Browne *et al.*, “A survey of Monte Carlo tree search methods,” *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, 2012, doi: 10.1109/TCIAIG.2012.2186810.

[21] A. Choudhary, “Monte Carlo Tree Search Tutorial | DeepMind AlphaGo,” 24-Jan-2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/01/monte-carlo-tree-search-introduction-algorithm-deepmind-alphago/. [Accessed: 30-Oct-2020].

[22] A. Choudhary, “A Hands-On Introduction to Deep Q-Learning using OpenAI Gym in Python,” *Blog*, 2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-q-learning-python/. [Accessed: 30-Oct-2020].

[23] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, “Visualizing higher-layer features of a deep network,” *Bernoulli*, vol. 1341, no. 1341, pp. 1–13, 2009.

[24] C. Olah, A. Mordvintsev, and L. Schubert, “Feature Visualization,” *Distill*, vol. 2, no. 11, p. e7, 2017, doi: 10.23915/distill.00007.

[25] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv Prepr. arXiv1511.08458*, 2015.

[26] C. Clark and A. Storkey, “Training deep convolutional neural networks to play go,” in *32nd International Conference on Machine Learning, ICML 2015*, 2015, vol. 3, pp. 1766–1774.

[27] X. Guo, S. Singh, H. Lee, R. Lewis, and X. Wang, “Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning,” in *Advances in Neural Information Processing Systems*, 2014, vol. 4, no. January, pp. 3338–3346.

[28] V. Zhou, “An Introduction to Recurrent Neural Networks for Beginners,” 25-Jul-2019. [Online]. Available: https://towardsdatascience.com/an-introduction-to-recurrent-neural-networks-for-beginners-664d717adbd. [Accessed: 01-Nov-2020].

[29] V. Mnih *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.

[30] L. J. Lin, “Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching,” *Mach. Learn.*, vol. 8, no. 3, pp. 293–321, 1992, doi: 10.1023/A:1022628806385.

[31] A. Pedroza, D. Welkener, S. Lima, F. S. Freitas, and G. Mendes, “A Motivational Study Regarding IoT and Middleware for Health Systems A Comparison of Relevant Articles,” *Iaria*, pp. 102–106, 2016.

[32] N. Sharma, M. Shamkuwar, and I. Singh, “The history, present and future with iot,” in *Intelligent Systems Reference Library*, vol. 154, Springer, 2019, pp. 27–51.

[33] B. Beranek, “A history of the ARPANET: the first decade,” *Tech. Rep.*, 1983.

[34] J. Pontin, “ETC: Bill Joy’s Six Webs - MIT Technology Review,” *MIT Technology*. MIT Technology Review, 2005.

[35] K. Ashton, “That ‘internet of things’ thing,” *RFID J.*, vol. 22, no. 7, pp. 97–114, 2009.

[36] M. Roberti, “The History of RFID Technology,” *RFID J.*, vol. 16, 2005.

[37] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and Challenges for Realising the Internet of Things. European Commission," *Inf. Soc. Media DG*, vol. 3, no. 3, pp. 34–36, 2010.

[38] H. Tschofenig, J. Arkko, D. Thaler, and D. McPherson, "Architectural Considerations in Smart Object Networking," *Rfc 7452*, vol. 1, pp. 1–24, 2015.

[39] K. Rose, S. Eldridge, and L. Chapin, "The internet of things: An overview," *Internet Soc.*, vol. 80, pp. 1–50, 2015.

[40] J. Bélissent, "Getting clever about smart cities: new opportunities require new business models," *Forrester Res. inc*, vol. 193, p. 33, 2010.

[41] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[42] D. A. K. Karimi, "What the Internet of Things (IoT) Needs to Become a Reality," *Free. White Pap.*, p. 16, 2013.

[43] P. Sorknæs *et al.*, "Smart Energy Markets - Future electricity, gas and heating markets," *Renew. Sustain. Energy Rev.*, vol. 119, p. 109655, 2020, doi: 10.1016/j.rser.2019.109655.

[44] M. R. Palattella *et al.*, "Standardized protocol stack for the internet of (important) things," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013, doi: 10.1109/SURV.2012.111412.00158.

[45] R. Ma, Y. Liu, C. Shan, X. L. Zhao, and X. A. Wang, "Research on Identification and Addressing of the Internet of Things," in *Proceedings - 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2015*, 2015, pp. 810–814, doi: 10.1109/3PGCIC.2015.40.

[46] L. Farhan, S. T. Shukur, A. E. Alissa, M. Alrweg, U. Raza, and R. Kharel, "A survey on the challenges and opportunities of the Internet of Things (IoT)," in *Proceedings of the International Conference on Sensing Technology, ICST*, 2017, vol. 2017-Decem, pp. 1–5, doi: 10.1109/ICSensT.2017.8304465.

[47] L. Farhan, R. Kharel, O. Kaiwartya, M. Quiroz-Castellanos, A. Alissa, and M. Abdulsalam, "A Concise Review on Internet of Things (IoT)-Problems, Challenges and

Opportunities," in *2018 11th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2018*, 2018, pp. 1–6, doi: 10.1109/CSNDSP.2018.8471762.

[48] M. R. Bashir and A. Q. Gill, "Towards an IoT big data analytics framework: Smart buildings systems," in *Proceedings - 18th IEEE International Conference on High Performance Computing and Communications, 14th IEEE International Conference on Smart City and 2nd IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2016*, 2017, pp. 1325–1332, doi: 10.1109/HPCC-SmartCity-DSS.2016.0188.

[49] L. Farhan, R. Kharel, O. Kaiwartya, M. Hammoudeh, and B. Adebisi, "Towards green computing for Internet of things: Energy oriented path and message scheduling approach," *Sustain. Cities Soc.*, vol. 38, pp. 195–204, 2018, doi: 10.1016/j.scs.2017.12.018.

[50] S. Abdullah and K. Yang, "An Energy Efficient Message Scheduling Algorithm Considering Node Failure in IoT Environment," *Wirel. Pers. Commun.*, vol. 79, no. 3, pp. 1815–1835, 2014, doi: 10.1007/s11277-014-1960-3.

[51] I. Bedhief, M. Kassar, and T. Aguili, "SDN-based architecture challenging the IoT heterogeneity," in *2016 3rd Smart Cloud Networks and Systems, SCNS 2016*, 2017, pp. 1–3, doi: 10.1109/SCNS.2016.7870558.

[52] C. Liu, Y. Zhang, and H. Zhang, "A novel approach to IoT security based on immunology," in *Proceedings - 9th International Conference on Computational Intelligence and Security, CIS 2013*, 2013, pp. 771–775, doi: 10.1109/CIS.2013.168.

[53] A. Brogi and S. Forti, "QoS-aware deployment of IoT applications through the fog," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1185–1192, 2017, doi: 10.1109/JIOT.2017.2701408.

[54] B. Vejlgaard, M. Lauridsen, H. Nguyen, I. Z. Kovacs, P. Mogensen, and M. Sørensen, "Interference impact on coverage and capacity for low power wide area IoT networks," in *IEEE Wireless Communications and Networking Conference, WCNC*, 2017, pp. 1–6, doi: 10.1109/WCNC.2017.7925510.

[55] Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, "Intelligent Edge Computing for IoT-

Based Energy Management in Smart Cities," *IEEE Netw.*, vol. 33, no. 2, pp. 111–117, 2019, doi: 10.1109/MNET.2019.1800254.

[56] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: A survey," *IEEE Wirel. Commun.*, vol. 11, no. 6, pp. 6–27, 2004, doi: 10.1109/MWC.2004.1368893.

[57] N. Zaman, A. B. Abdullah, and L. T. Jung, "Optimization of energy usage in Wireless Sensor Network using Position Responsive Routing Protocol (PRRP)," in *ISCI 2011 - 2011 IEEE Symposium on Computers and Informatics*, 2011, pp. 51–55, doi: 10.1109/ISCI.2011.5958882.

[58] A. Vergnano *et al.*, "Modeling and optimization of energy consumption in cooperative multi-robot systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 423–428, 2012.

[59] Z. Wang, Y. Liu, Y. Sun, Y. Li, D. Zhang, and H. Yang, "An energy-efficient heterogeneous dual-core processor for Internet of Things," in *Proceedings - IEEE International Symposium on Circuits and Systems*, 2015, vol. 2015-July, pp. 2301–2304, doi: 10.1109/ISCAS.2015.7169143.

[60] D. Saha, M. R. Yousuf, and M. A. Matin, "Energy Efficient Scheduling Algorithm for S-MAC Protocol in Wireless Sensor Network," *Int. J. Wirel. Mob. Networks*, vol. 3, no. 6, pp. 129–140, 2011, doi: 10.5121/ijwmn.2011.3610.

[61] H. Li, K. Ota, and M. Dong, "Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Netw.*, vol. 32, no. 1, pp. 96–101, 2018, doi: 10.1109/MNET.2018.1700202.

[62] S. Russel and P. Norvig, *Intelligence artificielle 3e édition : Avec plus de 500 exercices*. Pearson Education France, 2010.

[63] "random — Documentation Bibliothèques Python 1.0.0." [Online]. Available: https://he-arc.github.io/livre-python/random/index.html. [Accessed: 11-Nov-2020].

[64] A. Karpathy, "The Unreasonable Effectiveness of Recurrent Neural Networks," *Web Page*, 21-May-2015. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/. [Accessed: 18-Nov-2020].

[65] V. Zhou, "An Introduction to Recurrent Neural Networks for Beginners," 25-Jul-2019. [Online]. Available: https://towardsdatascience.com/an-introduction-to-recurrent-

neural-networks-for-beginners-664d717adbd. [Accessed: 18-Nov-2020].

Annexe

1. Réseau neuronal artificiel (ANN)

ANN est caractérisée par une succession de multiples couches de traitement. Chaque couche consiste en une transformation non linéaire et la séquence de ces transformations conduit à l'apprentissage de différents niveaux d'abstraction[23] [24].

Tout d'abord, décrivons un réseau neuronal très simple avec une seule couche caché entièrement connecté Figure3. La première couche reçoit une valeur d'entrée (c.-à-d. les caractéristiques d'entrée) **x** sous la forme d'un vecteur de taille de colonne $\mathbf{n_x}$ ($\mathbf{n_x} \in \mathbf{N}$).Les valeurs de la couche cachée (hidden layer) suivante sont la conversion de ces valeurs par une fonction paramétrique non linéaire, qui est une matrice multiplication par W_1 de taille $\mathbf{n_h} \times \mathbf{n_x}$ ($\mathbf{n_h} \in \mathbf{N}$), plus un terme de biais $\mathbf{b_1}$ de taille $\mathbf{n_h}$, suivie d'une transformation non linéaire :

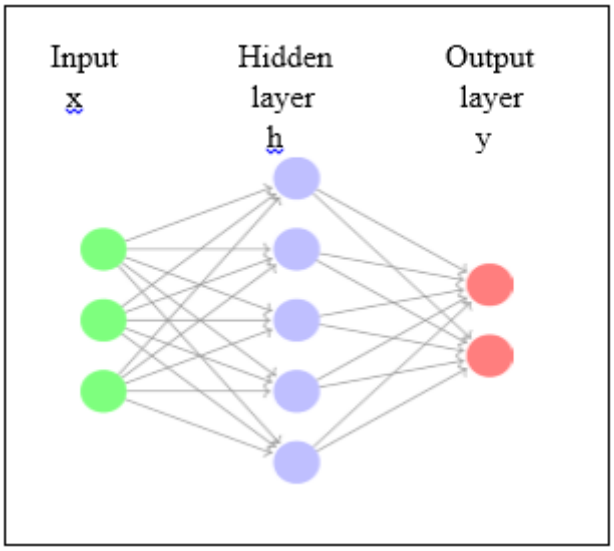
$$\mathbf{h = A\,(W1 \cdot x + b_1),}$$

A est une fonction d'activation. Cette fonction d'activation non linéaire est ce qui rend la transformation à chaque couche non linéaire, ce qui fournit l'expressivité du réseau neuronal.

La couche cachée **h** de taille $\mathbf{n_h}$ peut à son tour être transformée en d'autres ensembles de valeurs jusqu'à la dernière transformation qui fournit les valeurs de sortie **y**. Dans ce cas :

$$\mathbf{y = (W2 \cdot h + b_2),}$$

W2 est de taille $\mathbf{n_y} \times \mathbf{n_h}$ et $\mathbf{b_2}$ est de taille $\mathbf{n_y}$ ($\mathbf{n_y} \in \mathbf{N}$).

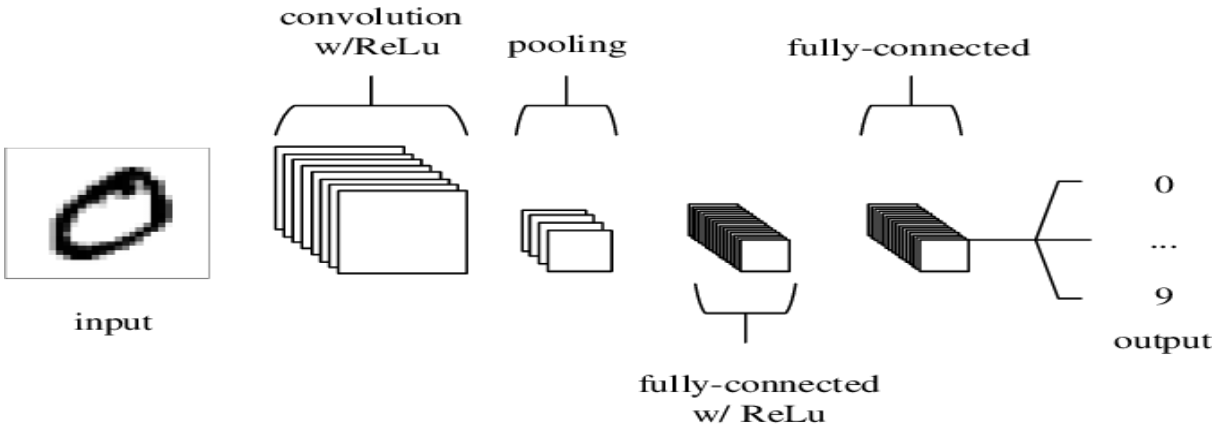


A.Figure 14.Exemple d'un réseau de neurone avec une couche caché[7]

2. Réseau neuronal convolutionnel (CNN) :

Un réseau neuronal convolutionnel (CNN) est un sous-ensemble du réseau de neurone artificiel(ANN) et d'apprentissage profond (Deep Learning), il est généralement utilisé pour analyser des images visuelles[25] .

Les CNNs sont composés de trois types de couches : des couches convolutives (convolutional layers), des couches de regroupement (pooling layers) et des couches entièrement connectées (fully-connected layers). Lorsque ces couches sont regroupées, une architecture CNN est formée (Figure 9).



A.Figure 15.une simple architecture d'un CNN [7].

La fonctionnalité de base de l'exemple CNN ci-dessus peur être composée en quatre couches [25]:

- Couche d'entrée (input layer) : elle contient les valeurs des pixels de l'image (une matrice des pixels).
- Couche convolutionnelle (convolutional layer) : elle déterminera la sorties des neurones connectés à la zone locale d'entrée en calculant le produit scalaire entre leurs poids(les filtres) et la zone connectée au volume d'entrée. L'unité rectifiée (fonction réelle non-linéaire par ReLU=max (0, x).) vise à appliquer une fonction d'activation par élément (remplace toutes les valeurs négatives reçues en entrée par des zéros) telle que le sigmoïde à la sortie de l'activation produite par la couche précédente.
- Couche de regroupement (pooling-layer) : Ce type de de couche est souvent placé entre deux couche de convolution : elle reçoit en entrée plusieurs features maps, et applique à chaque d'entre d'elles l'opération de pooling (réduire la taille des images, tout en préservant leurs caractéristiques importantes).

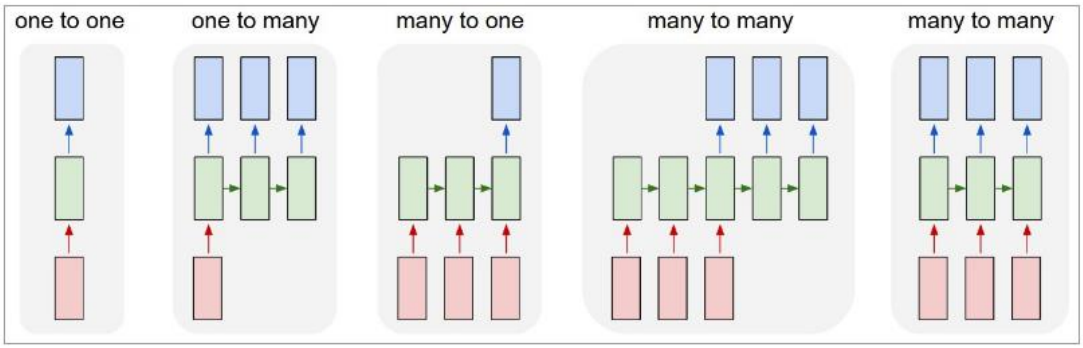
- Couche entièrement connecté (fully-connected layer) : elle est similaires aux choix des réseaux neurones traditionnels(ANN) (Figure 7).

Au lieu de traiter les entrées comme une matrice de pixel, ils sont traitées en tant que liste unique, elles prennent l'image filtré de haut niveau et les traduisent en vote.

3. RNN

Les réseaux neuronaux récurrents sont une sorte de réseau de neurones spécialisés dans le traitement des séquences. Ils sont souvent utilisés dans les tâches de traitement du langage naturel (NLP) en raison de leur efficacité dans le traitement du texte[64].

Un problème avec les réseaux de neurones vanille (et les CNN) est qu'ils ne peuvent fonctionner qu'à une taille prédéterminée : ils prennent une entrée de taille fixe et produisent une sortie de taille fixe. Les RNN sont utiles car ils nous permettent d'entrer et de sortir des séquences de différentes longueurs. Voici quelques exemples de RNN :



A.Figure 16.architecture des RNN[64]

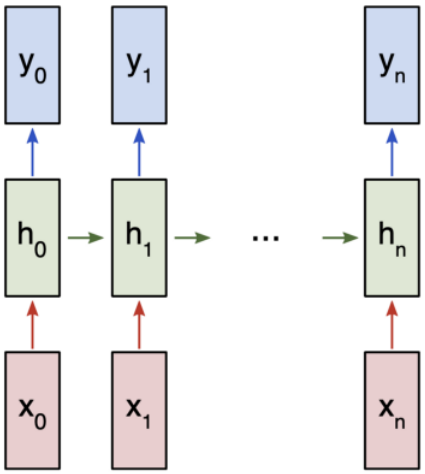
Cette capacité à traiter des séquences rend les RNN très utiles. Par exemple :

- La traduction automatique (par exemple Google Translate) se fait avec des RNN "many to many". La séquence de texte originale est introduite dans un RNN, qui produit ensuite le texte traduit comme sortie.
- L'analyse du sentiment (par exemple, s'agit-il d'un examen positif ou négatif ?) est souvent effectuée avec des RNN "plusieurs à un". Le texte à analyser est introduit dans un RNN, qui produit alors une seule classification de sortie (par exemple, il s'agit d'une révision positive).

Prenons un RNN "many to many" avec des entrées x_0, x_1, ... x_n qui veut produire des sorties y_0, y_1, ... y_n. Ces x_i et y_i sont des vecteurs et peuvent avoir des dimensions arbitraires.

Les RNN fonctionnent en mettant à jour de manière itérative un état caché h, qui est un vecteur qui peut également avoir une dimension arbitraire. À toute étape donnée t [65]:

1) L'état caché suivant h_t est calculé en utilisant l'état caché précédent h_{t-1} et l'entrée suivante x_t.
2) La sortie suivante y_t est calculée à l'aide de h_t.



A.Figure 17.architecture d'un RNN "plusieurs à plusieurs"[65]

Voici ce qui rend un RNN récurrent : il utilise les mêmes poids pour chaque étape. Plus précisément, un RNN vanille typique n'utilise que 3 séries de poids pour effectuer ses calculs [65]:

- Wxh, utilisé pour tous les liens x_t → h_t.
- Whh, utilisé pour tous les liens h_{t-1} → h_t.
- Why, utilisé pour tous les liens h_t → y_t.

Nous utiliserons également deux biais pour notre RNN :

- b_h, ajouté lors du calcul de h_t.
- b_y, ajouté lors du calcul de y_t.

Nous représenterons les poids sous forme de matrices et les biais sous forme de vecteurs. Ces trois poids et 2 biais constituent l'ensemble du RNN.

Voici les équations qui mettent tous ensemble :

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

$$y_t = W_{hy}h_t + b_y$$

Tous les poids sont appliqués en utilisant la multiplication matricielle, et les biais sont ajoutés aux produits résultants. Nous utilisons ensuite « tanh » comme fonction d'activation pour la première équation (mais d'autres activations comme sigmoïde peuvent également être utilisées).