



Chapitre II
Systemes Multi-Agents



II.1 Introduction

Longtemps, l'intelligence artificielle (IA) en particulier a considéré les programmes comme des entités individualisées capables de rivaliser avec l'être humain dans des domaines précis.

Ainsi, l'informatique devient de plus en plus diffuse et distribuée dans de multiples objets et fonctionnalités qui sont amenés à coopérer. La décentralisation est donc la règle et une organisation coopérative entre modules logiciels est un besoin. De plus, la taille, la complexité et l'évolutivité croissantes de ces nouvelles applications informatiques font qu'une vision centralisée, rigide et passive (contrôlée explicitement par le programmeur) atteint ses limites.

Cette complexité croissante des problèmes qui sont posés oblige à concevoir, non pas un programme, mais plusieurs entités couplées en interaction, chacune définie localement sans vision d'ensemble d'un système qui la dépasse. [11]

Le thème des systèmes multi agents (SMA), s'il n'est pas récent, est actuellement un champ de recherche très actif. Cette discipline est à la connexion de plusieurs domaines en particulier de l'intelligence artificielle, des systèmes informatiques distribués et du génie logiciel. C'est une discipline qui s'intéresse aux comportements collectifs produits par les interactions de plusieurs entités autonomes et flexibles appelées agents, que ces interactions tournent autour de la coopération, de la concurrence ou de la coexistence entre ces agents[12].

II.2 Intelligence Artificielle Distribuée (IAD)

II.2.1 Définitions

Généralement le domaine de recherche de l'IAD est défini comme l'étude et de la conception d'organisations d'agents artificiels pour obtenir des systèmes intelligents » [13]

II.2.2 Historique de l'intelligence artificielle distribuée

Le terme de l'IA apparaît dans les années 50. L'IAD apparaît quant à elle, dans les années 70 avec la distribution des unités de calcul et la parallélisations des algorithmes et, au même moment, avec la notion de modularité. C'est pendant les années 80



que la métaphore d'experts coopérants devient populaire et donne naissance à des approches telles que le modèle acteur [20] et l'architecture du tableau noir de Hearsay [15].

La démarche première de l'IAD est de mettre à profit le parallélisme et les avantages de la modularité au service des problèmes issus de l'IA classique.

Au cours des années 80, la réutilisabilité devient un axe majeur en informatique en général et en IAD en particulier. Ceci, associé à l'apparition des systèmes ouverts -dans lesquels les entités sont la possibilité d'entrer et de sortir du système à n'importe quel moment-, amène la mise en place de mécanismes décentralisés de coordination des activités. Cette nouvelle voie diverge des approches traditionnelles de la résolution distribuée de problèmes, pour lesquels les agents doivent principalement faire ce qu'on leur dit [14].

Au vu de la complexité de ces nouveaux systèmes, un nouveau domaine de l'IAD se constitue afin de les étudier : l'étude des SMA est née.

II.3 Notion d'agents

II.3.1 Définition

Plusieurs définitions ont été proposées par différents acteurs pour clarifier le concept .nous citerons deux :

- un agent est défini comme étant une entité physique ou virtuelle évoluant dans un environnement dont il n'a qu'une représentation partielle et sur lequel il peut agir Il est capable de communiquer avec d'autres agents et est doté d'un comportement autonome. [10]
- un agent fait référence à tout artefact, logiciel aussi bien que matériel, capable d'exhiber une certaine autonomie donc opérationnellement et informationnellement clos par rapport à l'environnement dans lequel il est plongé [16]

II.3.2 Caractéristiques d'un agent

Autonomie : Un agent autonome est un agent dont l'existence ne se justifie pas par l'existence des autres agents et de l'opérateur humain. Pour être autonome, un agent doit avoir ses propres buts et être capable de prendre des décisions, et ainsi résoudre des conflits internes [20].



La réactivité :

Des agents sont dits réactifs s'ils peuvent réagir à un changement qui s'est effectué dans leur environnement.

La Proactivité :

Des agents sont dits proactifs s'ils peuvent agir sans même que leur environnement ait changé.

Planification :

C'est le processus qui permet la construction d'un plan d'actions à réaliser pour atteindre un certain but,

Intelligence

Un agent intelligent est celui qui possède des capacités de raisonnement et d'apprentissage. L'agent doit non seulement planifier ses propres actions, mais aussi tenir compte de celles des autres agents.

Sociabilité :

Des agents sont dits sociables s'ils peuvent communiquer les uns avec les autres, et possiblement avec des humains, grâce un langage de communication pour réaliser les tâches pour lesquelles ils ont été conçus.

Apprentissage :

Les agents doivent évoluer et améliorer leurs connaissances, ainsi, adapter ou changer leurs comportements face à des situations similaires et cela en fonction de leurs expériences passées. Cette caractéristique est principalement réservée aux agents qualifiés d'intelligents

Prise de décision

C'est le choix auquel est confronté l'agent pour sélectionner le but à satisfaire en premier, et pour chaque but, l'action qui permet de l'atteindre.

II.3.3 L'environnement de l'agent

L'environnement dans lequel agit un agent est crucial dans le choix de son architecture. Les propriétés de l'environnement affectent bien entendu la conception de l'agent en le rendant plus ou moins complexe selon la situation. `

Un environnement peut être [18] :

- Accessible si un agent peut, à l'aide des primitives de perception, déterminer l'état de l'environnement et ainsi procéder, par exemple, à une action.
- Déterministe, ou non, selon que l'état futur de l'environnement ne soit, ou non, fixé que par son état courant et les actions de l'agent.



- Episodique si le prochain état de l'environnement ne dépend pas des actions réalisées par les agents.
- Statique si l'état de l'environnement est stable (ne change pas) pendant que l'agent réfléchit. Dans le cas contraire, il sera qualifié de dynamique.
- Discret si le nombre des actions faisables et des états de l'environnement est fini.

II.3.4 Type des agents

Les caractéristiques principales des agents sont : autonomie, de coopération et d'adaptation.

Il existe deux grandes tendances de typologie des agents qui sont [19] :

II.3.4.1 les agents réactifs :

L'école réactive considère qu'il n'est pas nécessaire que les agents soient intelligents Individuellement pour que le système ait un comportement global intelligent, elle suppose que les agents sont très simples et que l'intelligence émerge de l'interaction entre ces agents, ils se caractérisent par des agents qui ont la capacité de réagir rapidement à des problèmes simples, ils ne nécessitant pas un haut niveau de raisonnement

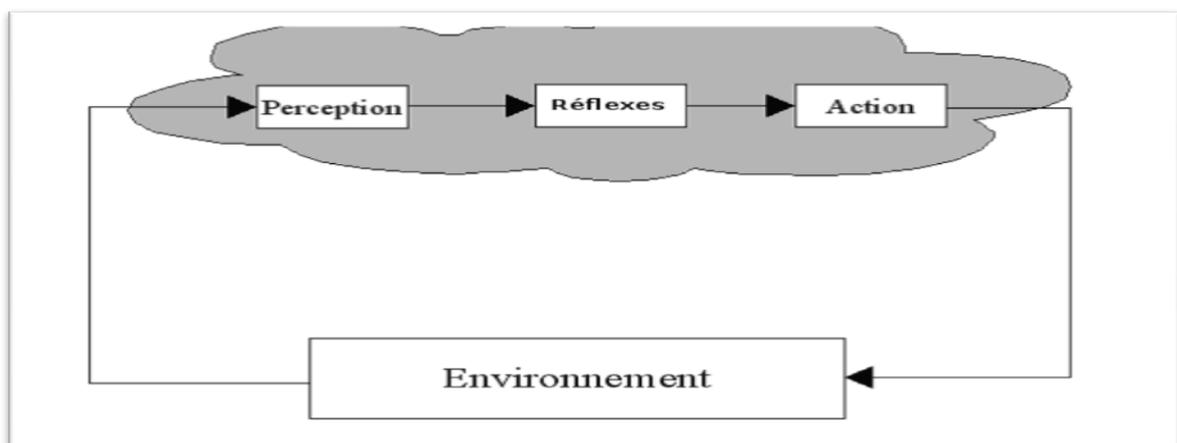


Figure II. 3 - Agent réactif [19]

II.3.4.2 les agents cognitifs

Les systèmes multi-agents cognitifs sont fondés sur la coopération d'agents capables, à eux seuls, d'effectuer des opérations complexes [17] [21] [22]. Chaque agent dispose d'une capacité de raisonnement, d'une aptitude à traiter des informations diverses liées au domaine d'application, et d'informations relatives à la gestion des interactions avec d'autres Agents et l'environnement.

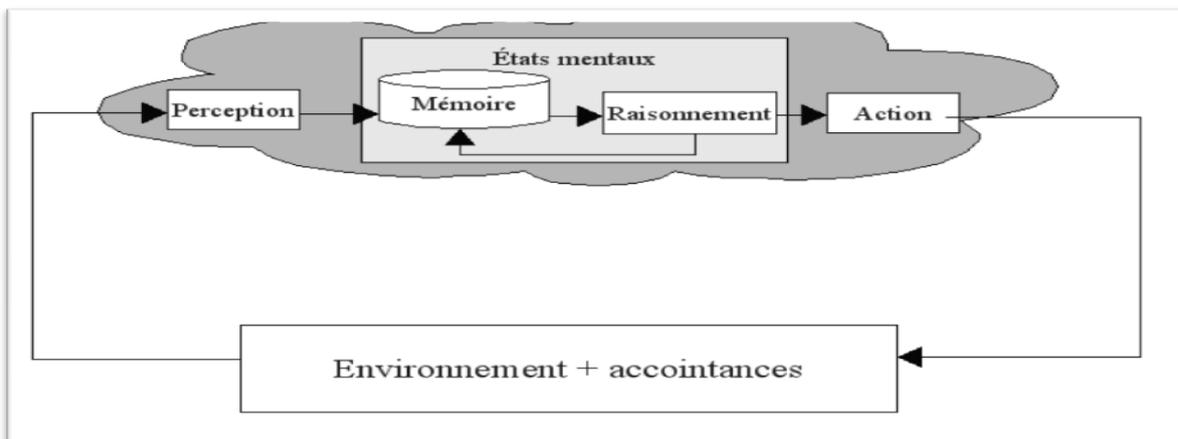


Figure II.4 - Agent cognitif [19]

II.4 Système multi agents (S.M.A.)

Il est plutôt rare que les concepteurs d'agents n'aient besoin que d'un seul agent dans l'environnement qu'ils construisent. Lorsque plusieurs agents se retrouvent dans un même environnement et que ces agents ont besoin d'interagir entre eux, on parle alors de système multi agent

II.4.1 Définition

Un système multi-agent peut se définir comme un macro-système composé d'agents autonomes qui interagissent dans un environnement commun afin de réaliser une activité collective cohérente [23].

Le résultat de l'organisation de ces agents, les liens les reliant, définit l'identité du système multi-agent.

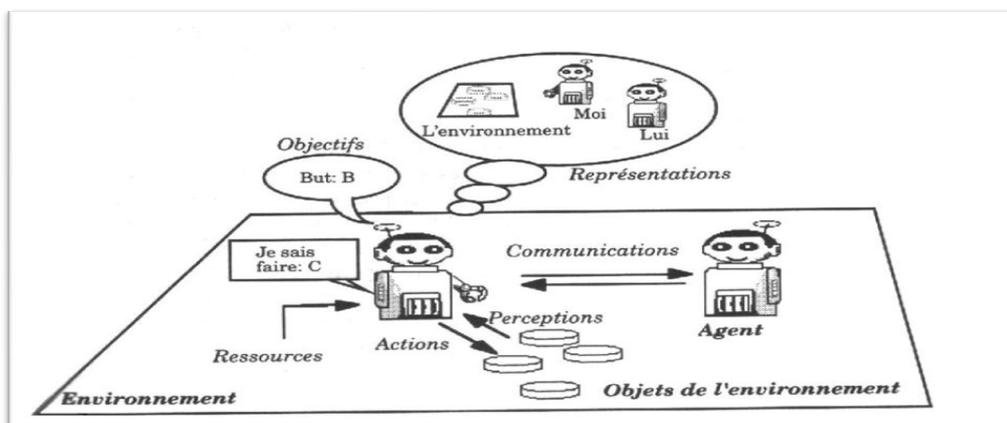


Figure II.5 Représentation imagée d'un agent en interaction avec son environnement et les autres agents [23]



II.4.2 Intérêts des SMA

Les apports les plus importants des systèmes multi-agents sont résumés comme suit [24] :

- L'automatisation et l'amélioration des processus de prise de décisions.
- La décentralisation d'un système en sous-systèmes coopératifs.
- La réutilisation par la création de nouveaux systèmes en interagissant avec ceux déjà existants.
- La représentation des connaissances d'une manière distribuée.
- La simulation des fonctionnements des organisations, comme ils sont utilisés pour simuler de nombreux mécanismes afin de vérifier les hypothèses des nombreuses recherches

II.4.3 Propriétés des SMA

Outre la notion d'environnement, qui comme on l'a expliqué, est fondamentale dans la définition des SMA, deux autres concepts caractérisent ces derniers. Le concept d'interaction qui permet aux agents d'échanger et le concept d'organisation, qui les structure.

II.4.4 Principe des SMA

Un SMA peut être ouvert (les agents y entrent et en sortent librement) ou fermé, homogène (les agents sont issus d'un même modèle) ou hétérogène. Au niveau de sa conception, un SMA impose une vision locale et décentralisée.

II.4.5 Interaction entre agents

Le concept d'interaction est fondamental pour les SMA. En effet, sans interactions aucune métaphore sociale n'est possible. Dans le cadre des SMA, une interaction peut être définie comme la mise en relation dynamique de deux ou plusieurs éléments du système par le biais d'un ensemble d'actions réciproque.

Ces éléments peuvent être des agents ou des éléments de l'environnement ou encore l'environnement lui-même.

Les interactions sont basées sur :

- La coopération
- La coordination
- La négociation :

I.4.5.1 La coordination

Est un Ensemble des activités supplémentaires qu'il est nécessaire d'accomplir dans un environnement comprenant plusieurs agents et qu'un agent seul poursuivant les mêmes buts n'accomplirait pas.[25]



II.4.5.2 La coopération

La coopération est une attitude adoptée par les agents qui décident de travailler ensemble. Elle permet à un agent de [15] :

- Mettre à jour les connaissances globales du système.
- Intégrer des informations venant d'autres agents.
- Interrompre son plan d'exécution pour aider les autres agents.
- Déléguer la tâche qu'il ne sait pas résoudre à un autre agent dont il connaît les compétences.

II.4.5.3 La négociation

Processus visant à améliorer les accords(en réduisant les inconsistances et l'incertitude) sur des points de vue communs ou des plans d'action grâce à l'échange structurée d'informations pertinentes.

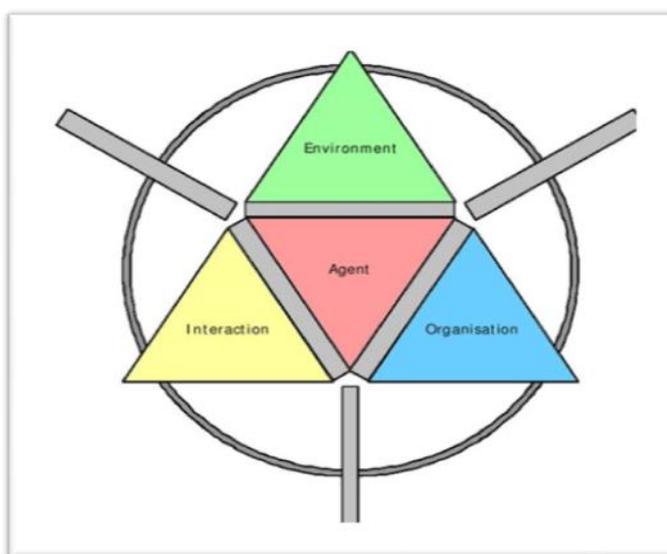


Figure II.6 Représentation symbolique d'un système multi-agents [15]

II.5 Langage de communication dans un SMA

Les agents peuvent interagir soit en accomplissant des actions linguistiques (en communiquant entre eux), soit en accomplissant des actions non-linguistiques qui modifient leur environnement. En communiquant, les agents peuvent échanger des informations et coordonner leurs activités.

Dans les SMA deux stratégies principales ont été utilisées pour supporter la communication entre agents: les agents peuvent échanger des messages directement ou ils peuvent accéder à une base de données partagées (appelée tableau noir ou "blackboard") dans laquelle les informations sont postées. Les communications sont à la base des interactions et de l'organisation sociale d'un SMA.



Il existe plusieurs types de protocoles de communication, les plus connus sont

II. 5-1 Protocole KQML (Knowledge query and manipulation language)

Le langage KQML [28] a été proposé pour supporter la communication inter-agents. Ce langage définit un ensemble de types de messages (appelés abusivement “performatifs”) et des règles qui définissent les comportements suggérés pour les agents qui reçoivent ces messages.

II.5-2 Protocole FIPA-ACL (Foundation for Intelligent Physical Agents)

Ces dernières années, KQML semble perdre du terrain au profit d’un autre langage plus riche sémantiquement ACL (pour Agent Communication Language). Un langage mis de l’avant par la FIPA qui s’occupe de standardiser les communications entre agents. ACL est basé également sur la théorie du langage et a bénéficié grandement des résultats de recherche de KQML.

II.6 Domaines d’application des systèmes multi agents

De nos jours, la technologie multi agent a trouvé sa place dans plusieurs domaines tel que

- ✓les systèmes médicaux
- ✓les systèmes manufacturiers,
- ✓Les systèmes financiers,
- ✓les loisirs,
- ✓les télécommunications,
- ✓le contrôle-commande,
- ✓les systèmes embarqués, et pas mal d’autres applications

II.7 Les plateformes de développement

II.7.1 plateforme JADE

Est une plate-forme Java pour les systèmes multi-agents respectant le standard FIPA. JADE a été développée par l’université de Parme et C-SELT – centre de recherche télécom italien.

Le but de JADE est pour simplifier le développement des systèmes multi-agents

- en assurant la conformité des standards par un ensemble complet de services et agents.
- en se conformant aux standards FIPA : service de nom, service de pages jaunes, messages transportés et service d’analyse, et une bibliothèque de protocole d’interactions de FIPA à utiliser.

Jade possède trois modules principaux nécessaires aux normes FIPA. Ils sont lancés à chaque démarrage de la plate-forme : [26]



- DF « Directory Facilitator » fournit un service de « pages jaunes » à la plate-forme.
- ACC « Agent Communication Channel » gère la communication entre les agents.
- AMS « Agent Management System » supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système

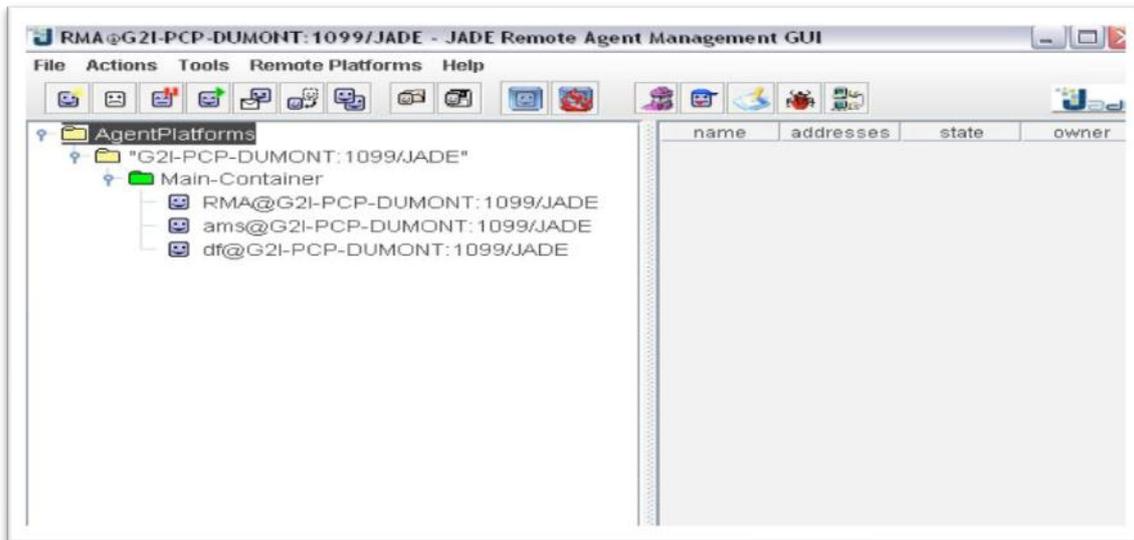


Figure II.7 - Interface graphique de l'agent RMA de JADE

II.7.2 La plate forme Jack

Jack est décrit comme étant un environnement pour construire, exécuter et intégrer des systèmes multi-agents commerciaux, écrite en Java et utilisant une approche orientée composants. Elle est développée par la société australienne Agent Oriented Software Pty. Ltd. Les agents sont basés sur le modèle BDI (Belief, Desire, Intention) développés à l' Australian Artificial Intelligence Institute (AAIL)

II.8 Conclusion

Le principe qui fait la force des SMA est de partager les tâches à réaliser entre plusieurs entités appelées 'agents' qui travaillent en collaboration pour atteindre un objectif commun. La solution globale est la coordination et le regroupement des solutions partielles des agents, Les SMA considèrent que l'action et l'interaction sont les éléments moteurs de la structuration d'un système dans son ensemble. Le système est alors fiable : la panne d'un sous-système n'entrave pas le fonctionnement de l'ensemble, de plus le système est extensible : l'adjonction d'un nouveau sous-système se fait de façon naturelle.

Ainsi la technologie multi agent semble être la solution pour le développement des logiciels de demain [27]



Chapitre III

Techniques De Classification



PREMIERE PARTIE

RESEAUX DE NEURONES

III.1.1 Introduction

Les réseaux de neurones, fabriquées de structures cellulaires artificielles, constituent une approche permettant d'aborder sous des angles nouveaux les problèmes de perception, de mémoire, d'apprentissage et de raisonnement au même titre que les algorithmes génétiques. Ils s'avèrent aussi des alternatives très prometteuses pour contourner certaines des limitations des méthodes numériques classiques.

Grâce à leur traitement parallèle de l'information et à leurs mécanismes inspirés des cellules nerveuses (neurones), ils infèrent des propriétés émergentes permettant de solutionner des problèmes complexes.

III.1.2 Principe des réseaux de neurones

Les réseaux de neurones formels ou neuromimétiques sont des réseaux qui en simulant le fonctionnement des neurones biologiques du cerveau humain, permettent aux différents systèmes d'apprendre à effectuer certaines tâches spécifiques.

Ils ne fonctionnent pas de façon programmée mais de façon intelligente, qui va extraire les traits caractéristiques des objets présentés lors de l'apprentissage. Cela leur permet de fonctionner même avec des informations partielles ou brouillées.

Les réseaux neuronaux sont composés souvent de plusieurs couches de cellules reliées entre elles et formant une vaste toile.

Ce réseau est capable d'ajuster la force de ses propres interconnexions (poids synaptiques) exactement comme cela se passe pour celles du cerveau de l'être vivant, lors de l'apprentissage.

III.1.3 Apprentissage des réseaux de neurones

III.3.1 : principe de l'apprentissage

Le point crucial du développement d'un réseau de neurones est son apprentissage. Il s'agit d'une procédure adaptative par laquelle les poids synaptiques sont ajustées face à une base d'exemple ([29]; [30]).



Les réseaux de neurones artificiels sont souvent liés à des algorithmes d'apprentissage.

Les poids sont initialisés avec des valeurs aléatoires. Puis des exemples d'apprentissage. sont présentés au réseau de neurones.

Une méthode d'optimisation modifie les poids au fur et à mesure des itérations pendant lesquelles on présente la totalité des exemples, afin de minimiser une fonction d'erreur.

Les poids retenus sont ceux pour lesquels l'erreur obtenue sur la base de test est la plus faible.

En effet, si les poids sont optimisés sur tous les exemples de l'apprentissage, on obtient une précision très satisfaisante sur ces exemples mais on risque de ne pas pouvoir généraliser le modèle à des données nouvelles. A partir d'un certain nombre d'itérations, le réseau ne cherche plus l'allure générale de la relation entre les entrées et les sorties du système, mais s'approche trop près des points et « apprend » le bruit (sur-apprentissage).

pour éviter le problème de sur-apprentissage ; nous divisons la base d'exemple en trois parties :

- base d'apprentissage
- base de test
- base de validation

Nous arrêtons l'apprentissage lorsque l'erreur de test commence à diverger (valeur minimal)

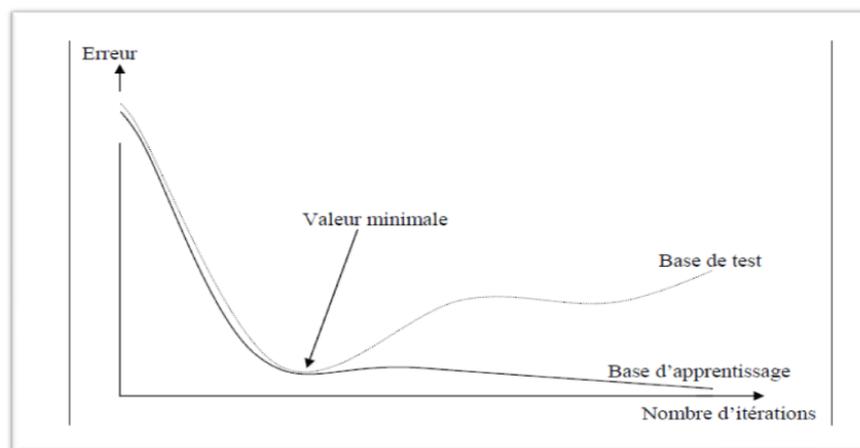


Figure III.8 Erreur moyenne sur la base d'apprentissage en fonction du nombre d'itérations

III.3.2 : types d'apprentissage

Il existe trois types d'apprentissages principaux. Ce sont l'apprentissage supervisé, l'apprentissage non-supervisé et l'apprentissage semi supervisé [31].

✓ *Apprentissage supervisé* : On parle d'apprentissage supervisé quand le réseau est alimenté avec la bonne réponse pour les exemples d'entrées donnés.



✓ *Apprentissage non supervisé* : Dans le cas de l'apprentissage non-supervisé le réseau décide lui-même quelles sont les bonnes sorties. [32].

✓ *Apprentissage semi-supervisé* : est un apprentissage de type essai-erreur où le réseau donne une solution et est seulement alimenté avec une information indiquant si la réponse était correcte ou si elle était au moins meilleure que la dernière fois.

III.1.4 Différents types des réseaux de neurones

Plusieurs types de réseaux de neurones ont été développés qui ont des domaines d'application souvent très variés. Notamment trois types de réseaux sont bien connus:

- le réseau de Hopfield (et sa version incluant l'apprentissage, la machine de Boltzmann),
- les cartes auto-organisatrices de Kohonen.
- les réseaux multicouches de type rétro propagation.

Les réseaux multicouches de type rétro propagation sont les réseaux les plus puissants des réseaux de neurones .ils utilisent l'apprentissage de type supervisé.

III.14.1 Les réseaux multicouches de type rétro propagation

Un réseau de type rétro propagation se compose d'une couche d'entrée, une couche de sortie et zéro ou plusieurs couches cachées [30]. Les connections sont permises seulement d'une couche inférieure (plus proche de la couche d'entrée) vers une couche supérieure (plus proche de la couche de sortie). Il est aussi interdit d'avoir des connections entre des neurones de la même couche.

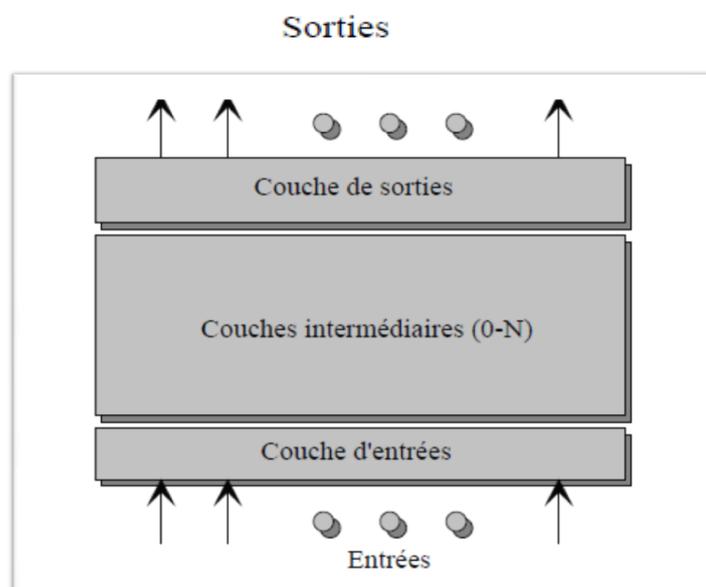


Figure III.9 -Réseau de rétro propagation [30]



La couche d'entrée sert à distribuer les valeurs d'entrée aux neurones des couches supérieures, éventuellement multipliées ou modifiées d'une façon ou d'une autre.

La couche de sortie se compose normalement des neurones linéaires qui calculent seulement une somme pondérée de toutes ses entrées.

Les couches cachées contiennent des neurones avec des fonctions d'activation non-linéaires, normalement la fonction sigmoïde.

Il a été prouvé [31] qu'il existe toujours un réseau de neurones de ce type avec trois couches seulement (couche d'entrée, couche de sortie et une couche cachée) qui peut approximer une fonction $f : [0,1]^n \Rightarrow \mathbb{R}^m$ avec n'importe quelle précision $\varepsilon > 0$ désirée. Un problème consiste à trouver combien de neurones cachés sont nécessaires pour obtenir cette précision. Un autre problème est de s'assurer a priori qu'il est possible d'apprendre cette fonction.

III.1.5 La mise en œuvre des réseaux de neurones multicouches

III.1.5.1 : Les entrées pertinentes

La détermination des entrées pertinentes peut être beaucoup plus délicate pour le processus naturel complexe (écologique ou bien social ou financier. . .). Ainsi, si l'on veut prédire une propriété spécifique, le choix des descripteurs pertinents n'est pas évident.

III.1.5.2 : La normalisation des données

Les données utilisées dans un réseau de neurone doivent être numériques et leurs modalités comprises dans l'intervalle $[0,1]$, ce qui implique quand ce n'est pas le cas, une normalisation des données est exigée.

La normalisation des données est considérée comme le prétraitement le plus important lorsqu'on utilise les réseaux de neurones.

III.1.5.3 : nombre de couche cachée

Mis à part les couches d'entrée et de sortie, il faut décider du nombre de couches intermédiaires ou cachées. Sans couche cachée, le réseau n'offre que de faibles possibilités d'adaptation. Néanmoins, il a été démontré qu'un Perceptron Multicouches avec une seule couche cachée pourvue d'un nombre suffisant de neurones, peut approximer n'importe quelle fonction avec la précision souhaitée [33].



III.1.5.4: Nombre de neurones cachés

Chaque neurone peut prendre en compte des profils spécifiques de neurones d'entrée.

Un nombre plus important permet donc de mieux "coller" aux données présentées mais diminue la capacité de généralisation du réseau. Il faut alors trouver le nombre adéquat de neurones cachés nécessaire pour obtenir une approximation satisfaisante.

III.1.5.5 : L'apprentissage des perceptrons multicouches

L'apprentissage consiste en un entraînement du réseau. On présente au réseau des entrées et on lui demande de modifier sa pondération de telle sorte que l'on retrouve la sortie correspondante.

L'algorithme consiste dans un premier temps à propager vers l'avant les entrées jusqu'à obtenir une entrée calculée par le réseau. La seconde étape compare la sortie calculée à la sortie réelle connue. L'algorithme modifie alors les poids de telle sorte qu'à la prochaine itération, l'erreur commise entre la sortie calculée et connue soit minimisée.

III.1.5.6 : l'algorithme de rétro propagation

L'algorithme de rétro propagation (ARP) ou de propagation arrière « backpropagation » est l'exemple d'apprentissage supervisé le plus utilisé à cause de l'écho médiatique de certaines applications spectaculaires telles que la démonstration de Sejnowski et Rosenberg (1987) dans laquelle l'ARP est utilisé dans un système qui apprend à lire un texte.

➤ L'algorithme par descente de gradient

Cette technique d'apprentissage classique a été mise au point par Rumelhart et est

Cette technique calcule l'erreur entre la sortie théorique du réseau et la valeur de sortie du réseau de neurones. La principale caractéristique de cette méthode d'apprentissage est que la valeur de l'erreur est ensuite directement utilisée pour modifier les différents poids intervenant dans le réseau de neurones. Cette modification est effectuée de la couche de sortie vers la couche d'entrées. L'objectif de cette rétro propagation est de minimiser l'erreur globale du réseau.

III.1.6 : Avantages et Inconvénients des réseaux de neurones

Les RNS présentent une capacité de généralisation et une facilité d'apprentissage ; par contre ils ne sont pas interprétables (boîtes noires)