



DES RESEAUX DE NEURONES ARTIFICIELS A LA ROBOTIQUE COOPERATIVE

Claude Touzet

► To cite this version:

Claude Touzet. DES RESEAUX DE NEURONES ARTIFICIELS A LA ROBOTIQUE COOPERATIVE. Intelligence artificielle [cs.AI]. Aix Marseille Université, 1998. tel-01338022

HAL Id: tel-01338022

<https://amu.hal.science/tel-01338022>

Submitted on 27 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université de Droit, d'Economie et des Sciences d'Aix-Marseille III

Faculté des Sciences et Techniques de Saint-Jérôme

Mémoire

présenté par

Claude TOUZET

pour l'**Habilitation à Diriger des Recherches**

DES

RESEAUX DE NEURONES ARTIFICIELS

A

LA ROBOTIQUE COOPERATIVE

Soutenance le 28 Septembre 1998 devant le Jury composé de :

Jacob	BARHEN	(rapporteur)
Jean-Claude	BERTRAND	
Eugène	CHOURAQUI	
Norbert	GIAMBIASI	
Christian	JUTTEN	(rapporteur)
Jean-Arcady	MEYER	(rapporteur)
Claude-Charles	THIRION	

Des réseaux de neurones artificiels à la robotique coopérative

Claude TOUZET

Résumé de l'Habilitation à Diriger des Recherches

Les travaux décrits dans ce mémoire rapportent une trajectoire scientifique d'une dizaine d'années constamment guidée par le désir d'étudier et de développer des modèles de réseaux de neurones artificiels en prise directe avec le monde réel. La première partie de nos recherches s'est intéressée à l'apprentissage au sein de systèmes connexionnistes multi-réseaux. En droite ligne depuis le modèle de la Machine Séquentielle Connexionniste (MSC, développée durant la thèse de doctorat), qui met en jeu 2 réseaux multicouches, 6 MSCs sont mises en oeuvre qui permettent l'acquisition et le contrôle de la marche chez un robot hexapode. Le paradigme utilisé pour la distribution des informations nécessaires à chacun des modules connexionnistes est l'apprentissage par pénalité-récompense. Un robot hexapode a été construit qui valide les résultats préalablement obtenus en simulation. L'apprentissage par pénalité-récompense appartient à la classe des apprentissage par renforcement.

La seconde partie de nos recherches s'est intéressée à étudier les interactions entre les réseaux de neurones artificiels et l'apprentissage par renforcement. Une implantation sur réseaux multicouches, puis sur cartes auto-organisatrices du Q-learning a été proposée. Nous obtenons ainsi des réductions de la taille mémoire requise et du nombre d'itérations d'apprentissage nécessaires qui autorisent une utilisation pratique. Nous avons ensuite développé des mécanismes de distribution de l'apprentissage par renforcement, soit au sein d'un seul robot doté de plusieurs comportements, soit au sein d'un groupe de robots dans une tâche impliquant la coopération. A la différence des courants de recherches actuels, qui prouvent l'utilisation d'*a priori* face à la combinatoire élevée de l'espace de recherche, nous proposons l'emploi d'*a posteriori*, l'utilisation du "lazy learning" pour construire un modèle non explicite et le développement d'outils et méthodes d'aide à la conception des fonctions de renforcement.

A moyen terme, l'objectif de nos recherches est d'automatiser la décomposition d'un comportement robotique complexe en une succession de comportements élémentaires. L'utilisation de marqueurs temporel et spatial est envisagée pour permettre le séquençage des cartes auto-organisatrices implantant les comportements élémentaires. Dans ce cas, la simple définition de l'objectif à atteindre suffirait alors à générer le comportement solution.

Mots clefs : Réseaux de neurones artificiels, Connexionnisme, Systèmes multi-réseaux, Apprentissage par renforcement, Q-learning, "Behavior based" robotique, Robotique coopérative.

SOMMAIRE

État Civil	i
Formation et situations successives	ii
1. ACTIVITÉS SCIENTIFIQUES	1
1.1 Réseaux de neurones séquentiels	4
1.2 Apprentissage par renforcement sur structure neuronale	10
1.3 Synthèse neuronale de comportement adaptatifs	15
2. PROGRAMME DE RECHERCHE	21
2.1 Sujet proposé	22
3. DIFFUSION DES CONNAISSANCES	24
3.1 Thèse	25
3.2 Ouvrages scientifiques	25
3.3 Brevet	26
3.4 Revues scientifiques avec comité de lecture	26
3.5 Conférences avec actes et comité de lecture	26
3.6 Colloques nationaux et internationaux avec actes à diffusion restreinte	29
3.7 Rapports de contrats	30
3.8 Tutoriaux	31
4. ENCADREMENT DE TRAVAUX DE RECHERCHE	33
4.1 Co-Encadrement de thèses	34
4.2 Encadrement de DEA	35
4.3 Participation à des contrats d'études	36
4.4 Participation à des comités de programme et d'organisation	38
5. ACTIVITÉS D'ENSEIGNEMENTS	39
6. ANNEXES JOINTES	44
A - Exemples et résumés de quelques travaux contractuels	44
B- Publications	81

1 - C. TOUZET, N. GIAMBIASI, "The Connectionist Sequential Machine: a General Model of Sequential Networks", *Australian Conf. on Neural Networks*, in Canberra, P. Leong & M. Jabri Eds. Sydney University Electrical Engineering, NSW 2006, Australia, February 1992.

2 - C. TOUZET, N. GIAMBIASI, "Application of Connectionist Models to Fuzzy Inference Systems", in *Parallelization in Inference Systems*, Lectures Notes in Artificial Intelligence 590, B. Fronhöfer & G. Wrightson Eds., Springer Verlag, April 1992.

3 - C. TOUZET, N. KIEFFER and M. LE GOC, "Artificial Neural Networks Forecasting and Monitoring Scaffold and Scaffolding Phenomena in Blast Furnaces," *IEEE International Conference on SMC*, Vancouver, Canada, October 1995.

4 - C. TOUZET, "Neural Reinforcement Learning for Behaviour Synthesis," *Robotics and Autonomous Systems*, Special issue on Learning Robot: the new wave, N. Sharkey Guest Editor, Vol. 22, Nb. 3-4, pp. 251-281, December 1997.

ÉTAT CIVIL

Nom : **TOUZET**

Prénoms : **Claude** Frédéric Maria

Date et lieu de naissance : 16 novembre 1963, Neuilly sur Seine, France.

Nationalité : Française.

Situation de famille : marié, deux enfants

Situation militaire : exempté

Adresse personnelle : 2 Allée Montvert, 13013 Marseille, Tél. : 04 91 66 30 59

Adresse professionnelle : Oak Ridge National Laboratory

Computer Science and Mathematical Division

Center for Engineering Science Advanced Research

P.O.Box 2008

Oak Ridge, TN 37831-6355, USA

Tél.: + 1 423 241 2985, Fax: + 1 423 241 0381

Email: touzetc@mars.epm.ornl.gov

<http://avalon.epm.ornl.gov/~touzetc>

FORMATIONS ET SITUATIONS SUCCESSIVES

Lorsqu'en provenance d'un DEUG A (Physique-Chimie), je suis admis en 1983 à suivre les enseignements de la Licence, puis de la Maîtrise de Neurosciences du Comportement, je bénéficie des avantages de la (récente) politique d'ouverture vers la multidisciplinarité. Et, lorsque j'achève ce cycle de formation, c'est au même état d'esprit que je dois de pouvoir suivre les cours du DESS Informatique et puis de m'inscrire l'année suivante en DEA XIAO, avec un sujet en Robotique.

1987 est une année importante pour les réseaux de neurones artificiels¹ et aussi pour le Laboratoire d'Étude et Recherche en Informatique (LERI) qui ouvre ses portes à Nîmes. Le LERI accueille l'équipe marseillaise du Professeur Norbert Giambiasi, qui s'accroît pour l'occasion d'un nouveau sujet de recherche: je viens de faire mes débuts en recherche (doctorale) sur thème des réseaux de neurones séquentiels. Du fait des liens conservés avec Jean-Claude Gilhodes (Maître de Conférences, intervenant en Licence et Maîtrise de Neurosciences), nous établissons quelques contacts avec le TIRF, un laboratoire ayant une longue histoire et une importante notoriété en RNA (réseaux de neurones artificiels). Grâce à l'appui scientifique de son directeur, le Professeur Jeanny Hérault, le congrès NEURO-NIMES (les réseaux neuro-mimétiques et leurs applications) voit le jour en 1988. Cette manifestation sera d'une aide considérable pour promouvoir l'équipe RNA naissante du LERI. Un autre aspect de notre travail aura été d'obtenir dès 1988 l'aide de l'ANVAR pour notre projet de simulateur de RNA. Cette aide permettra entre autres choses d'assurer ma participation au titre d'ingénieur de recherche payé "sur contrat" durant ma thèse. Lorsque je soutiens celle-ci en décembre 1990, une autre thèse démarre en RNA, et il y a quantité de stages de DEA, pratiquement tous dans un cadre contractuel avec une entreprise ou un organisme de recherche. En particulier, l'équipe RNA du LERI participe au développement du système de reconnaissance de caractères manuscrits réalisé par ITECA (1990), système qui deviendra au fil des ans, l'un des piliers de son développement (- de 10 personnes en 1990, + de 50 en 1997).

¹ Découverte fin 1985 de l'algorithme de la rétro-propagation du gradient d'erreur qui permet l'apprentissage sur des réseaux multicouche de perceptrons. Les premières applications, comme par exemple Nettetalk en 1986 aux USA, sont résolument médiatiques. 1987 est l'année premier congrès international (IEEE) sur les réseaux de neurones artificiels.

Entre 1991 et 1994, je suis chercheur-enseignant à l'EERIE (École d'Ingénieur créée en 1988, dont fait partie le LERI). En particulier, j'assure un enseignement en RNA d'environ 60 heures. Le support de cours, TD et TP sera édité sous la forme d'un livre par EC2² en novembre 1992. Grâce à François Blayo venu renforcer l'équipe RNA du LERI en 1992, j'ai la possibilité de participer aux développements du robot Khepera³ lors d'un séjour (Professeur Invité) à l'École Polytechnique Fédérale de Lausanne (Suisse), dans le laboratoire de Micro-Informatique du Professeur Jean-Daniel Nicoud. Ce robot miniature, d'une fiabilité totale, sera d'une grande aide pour tester nos RNA face au monde réel. Le développement d'un support de type carte auto-organisatrice pour l'apprentissage par renforcement, qui constitue le sujet d'une troisième thèse faite au LERI, utilisera Khepera de manière intensive. L'idée d'associer une carte auto-organisatrice à un robot nous avait été fourni quelques années plus tôt (1989-1990) par Yves Coiton, à l'époque doctorant au Laboratoire de Neurobiologie Humaine sous la direction de Jean-Claude Gilhodes. A cette même époque (1989), nous avons fondé avec Yves Coiton une association qui a pour but de rassembler et faire se rencontrer les étudiants en thèse dans le connexionnisme : l'ACTH⁴.

Lors du rattachement de l'EERIE à l'École des Mines d'Ales (EMA), le LGI2P⁵ devient le nouveau laboratoire de recherche en informatique de l'EMA, avec une vocation affirmée vers le transfert de technologie. Je rejoins donc à la même époque (fin 1994), le Département de Recherche en Informatique, Automatique et Mécatronique (DIAM) de l'IUSPIM (Univ. d'Aix-Marseille III) où je suis, toujours⁶, dans l'équipe de Norbert Giambiasi. Notre installation à Marseille reçoit l'appui de la Région, à la fois sous la forme de subventions d'équipements qui nous permettent de disposer de plusieurs robots Khepera et de stations de travail, et aussi sous la forme de participation dans un projet de recherche mené avec la SOLLAC (Fos-sur-Mer). Durant mon séjour au DIAM, mon statut sera celui d'ATER et d'ingénieur de recherche payé sur contrat. La recherche contractuelle en RNA se fera principalement avec le projet Sachem-SOLLAC (avec qui nous collaborons depuis 1993). Un brevet est d'ailleurs en cours de dépôt sur le thème des RNA pour la perception dans Sachem (conjointement avec Claude-Charles Thirion, directeur du projet Sachem, et Marc Le Goc⁷). Notre second important contractant sera la DCA-DCAN (Le Brusc, Var) : une quatrième thèse ayant pour thème l'application des RNA aux antennes

² Suite à la faillite de l'éditeur, et l'impossibilité de faire réimprimer l'ouvrage, celui-ci est maintenant accessible sur Internet (<http://avalon.epm.ornl.gov/~touzetc/Book/Bouquin.htm>).

³ Depuis, l'équipe qui a développé Khepera au sein du LAMI-EPFL a créé la société K-Team S.A. et emploie environ 4 personnes à temps plein (<http://www.k-team.com/>).

⁴ L'ACTH (Association des Connexionnistes en Thèse) existe toujours (<http://www.supelec-rennes.fr/acth/>). Elle est forte de plus de 80 membres et, comme à ses débuts, se réunit chaque année. Nous avons organisé les journées Valgo 90 à Port-Camargue (Gard).

⁵ Le LGI2P rassemble les effectifs du LERI et ceux du précédent laboratoire de l'EMA.

⁶ Le Professeur Norbert Giambiasi m'ayant proposé de le suivre dans cette nouvelle affectation.

⁷ Ingénieur SOLLAC et, depuis, aussi professeur associé à l'IUSPIM.

d'hydrophones a été soutenue en 1997. Le doctorant, embauché par la société Vibria, doit sous peu participer à l'industrialisation des travaux RNA menés pour Sachem.

En août 1997, je suis invité par le Dr. Jacob Bahren, directeur du Center for Engineering Science Advanced Research (CESAR) à Oak Ridge National Laboratory (ORNL) dans le Tennessee (USA) à venir effectuer un séjour de recherche post-doctorale. Je travaille dans l'équipe Robotique Collective dirigée par Lynne Parker, au sein d'un environnement matériel exceptionnel⁸. Je n'en continue pas moins à assurer un certain nombre de missions dans le cadre du DIAM, comme de participer à l'organisation du congrès NEURAP (le successeur de NEURO-NIMES, qui se tient à Marseille depuis 1994), et assurer la co-direction de la thèse de Juan-Miguel Santos, inscrit en co-tutelle avec l'Université de Buenos Aires (Argentine).

⁸ Par exemple, nous disposons de 4 robots Nomad 200 pour nos expérimentations.

GRADES ET TITRES UNIVERSITAIRES

- **Baccalauréat** section C (1981).
- **DEUG A** (Physique-Chimie), Univ. d'Aix-Marseille I (1983).
- **Licence** Neurosciences du Comportement, Univ. d'Aix-Marseille III (1984).
- **Maîtrise** Neurosciences du Comportement, Univ. d'Aix-Marseille III, *mention AB* (1985).
- **DESS** Informatique double compétence, Université d'Aix-Marseille II, *mention B* (1986).
- **DEA XIAO** (Systèmes Intelligemment Assistés par Ordinateur), Univ. d'Aix-Marseille III, *mention TB* (1987).
- **Thèse de doctorat**
Spécialité : Composants, Signaux et Systèmes.
Formation doctorale : Conception assistée des systèmes informatiques, automatiques et micro-électronique.
Université des Sciences et Techniques du Languedoc, Montpellier II, *mention TH*,
soutenue le 14 décembre 1990, directeur de thèse : Prof. Norbert Giambiasi.
- **Qualification** à postuler sur les postes de Maître de Conférences (1993).

SITUATIONS SUCCESSIVES

- 1987-1990 : Ingénieur de recherche au LERI (laboratoire d'Étude et Recherche en Informatique) de l'EERIE (École pour les Études et la Recherche en Informatique et Électronique, Nîmes).
- 1991-1994 : Chercheur-enseignant au LERI-EERIE (Nîmes).
- 1994-1996 : ATER à l'IUSPIM (Institut Universitaire des Sciences Pour l'Ingénieur de Marseille).
- 1996-1997 : Ingénieur de recherche au DIAM (payé sur contrat), chargé de cours à l'IUSPIM.
- 1997-1998 : Chercheur invité dans le cadre des séjours de recherche post-doctorale au CESAR (Center for Engineering Science Advanced Research) à ORNL (Oak Ridge National Laboratory, Tennessee, USA).

1. ACTIVITÉS SCIENTIFIQUES

La **Vie** n'est pas un accident, elle est la conséquence inévitable de propriétés de la matière et des interactions entre molécules aidée par une absence complète de délais à respecter. Le développement de la **Conscience** semble à tout le moins plus hasardeux ; celui de l'**Intelligence** (humaine) est, dans l'opinion actuelle, un effet de bord dans lequel, bien sur, la nature humaine se plaît.

L'homme a entamé très tôt la conquête de son environnement matériel, mais ce serait une erreur de croire qu'il n'a accordé que très récemment une quelconque attention à son ego cognitif. Ainsi, il est édifiant de constater la qualité des travaux philosophiques de l'Antiquité (grecque) et, conséquemment, le peu de progrès réalisés depuis. En fer de lance de cette volonté de se comprendre, nous trouvons aujourd'hui les **Sciences Cognitives** (en direct continuation des Neurosciences, ou Sciences du Cerveau). L'**Intelligence Artificielle** (IA) occupe une place importante au sein des Sciences Cognitives pour au moins deux raisons :

- Notre actuelle "domination" sur notre environnement matériel découle bien certainement de notre intelligence. Des recherches visant à accroître celle-ci d'un artefact "matériel" (ou "artificiel") semblent donc désirables.
- La démarche scientifique, héritée des siècles passés, impose d'expérimenter pour valider. La validation indiscutable d'une notable compréhension de notre intellect se doit d'être de recréer l'Intelligence.

L'IA est née il y a déjà quelques décennies, cependant un regard quelque peu objectif se doit de constater que les résultats ou retombées, mêmes si elles sont loin d'être négligeables en termes économiques, ne sont pas remarquables. Un tel constat ne doit cependant pas nous alarmer. Dans l'intervalle de temps (100 ans au maximum), un certain nombre d'idées toutes faites⁹, de fausses analogies, d'a priori non explicites, etc., ont été

⁹ Ainsi, la solution aux problèmes de l'IA n'est pas uniquement liée à la puissance de calculs des ordinateurs.

identifiés. Des concepts nouveaux ont vu le jour et ont été explicités. Ils font sans aucun doute partie des pré-requis indispensables à une véritable compréhension de l'Intelligence¹⁰. Parallélisme massif, calcul sub-symbolique, auto-organisation sont des notions qui nous apparaissent "non-naturelles" (sic), qui vont à l'encontre du sens commun et des résultats de notre introspection¹¹. Plus grave, 4000 ans de tradition d'enseignement (une vingtaine d'années à l'échelle individuelle) s'ingénient à développer les¹² capacités cognitives de l'individu en le dotant de Logique. Hors s'il est une chose dont nous sommes sûrs, c'est que l'Intelligence n'est pas logique¹³. Le chercheur en IA se doit donc de réussir le difficile exercice de remettre en question ses mécanismes intellectuels, pour les adapter aux nouveaux paradigmes, tout en continuant à respecter l'orthodoxie logique dans la communication de ses travaux. Les Mathématiques, confrontées à ce même type de problème il y a déjà longtemps, ont développé un langage adapté dans lequel la communauté mathématique communique "naturellement"¹⁴. Le progrès en IA est peut-être à ce prix : se doter d'esprits neufs et entraînés à penser avec ces concepts. Dans l'attente, nous pouvons certainement préparer le terrain.

Les efforts menés avec les **systèmes à base de connaissance** ont permis de mettre en évidence leurs inadéquations avec le monde réel, du fait de la question toujours ouverte relative au passage du numérique (capteur) au symbolique (symbols grounding problem). L'absence de réponse adéquate a permis la rapide diffusion d'une approche sub-symbolique (numérique) connue sous le nom de **réseaux de neurones artificiels** (ou connexionnisme).

Définition : Les réseaux de neurones artificiels (RNA) sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau.

Bien entendu, les premiers travaux autour des RNA se sont cantonnés à quelques régions de l'espace des possibles. En particulier, les aspects temporels ont été ignorés, le nombre même d'éléments (neurones et synapses) en jeu reste faible, la dynamique de fonctionnement du processeur élémentaire est éventuellement non-linéaire¹⁵ mais certainement pas chaotique, etc. Pourtant, même simples, les travaux en RNA ont permis d'expérimenter et de comprendre les notions importantes de **traitements parallèles massifs, d'apprentissage sub-symbolique, d'auto-organisation**. Ils ont permis de plus, pour la première fois,

¹⁰ Tout le monde s'accorde à reconnaître sa multiplicité : il serait plus exact de parler des Intelligences.

¹¹ Il n'y a, bien sûr, pour personne meilleure Intelligence que la sienne.

¹² Un nombre limité d'entre elles.

¹³ A l'exception de l'Intelligence logique.

¹⁴ La place accordée aux Mathématiques par l'Enseignement en fait la priorité. Je n'en veux pour preuve que le système de notation du bac américain (1600 points maxi : 800 en Maths, 800 en Anglais).

¹⁵ Sigmoïde, gaussienne, etc.

de nous confronter avec des outils aux performances intéressantes, mais non aisément démontrables. Si l'on s'accorde à penser que quelque chose que l'on comprend ne peut pas être intelligent, il est instructif de remarquer les efforts qui sont réalisés : par les chercheurs pour expliciter autant que possible le comportement des RNA et par l'industrie pour valider aussi extensivement que possible leurs applications connexionnistes avant emploi.

Bien sur, quelques pionniers ont résolument effectué leurs travaux de recherches en mettant directement en contact RNA et monde réel. Le choix d'un système sensori-moteur dans lequel embarquer le RNA avait déjà un nom : robot. La **Robotique** n'a pas attendu les RNA, ni même l'IA pour faire ses débuts. Cependant, elle aussi est bien loin des ses objectifs initiaux. Le robot industriel d'aujourd'hui tient plus de l'automate que du Cyborg hollywoodien. Il pêche principalement par son (manque) d'Intelligence¹⁶. Ce manque de performance est dû à la volonté de programmer le comportement du robot ; d'où l'emploi de la programmation algorithmique en robotique, directement issue de la Logique. Dès le début des années 1980, un certain nombre de chercheurs ont plaidé pour une robotique différente, sans représentation de haut-niveau (i.e., symbolique). Le paradigme proposé est de considérer un comportement comme l'activation séquentielle de comportements élémentaires (simples). L'Intelligence s'exprime alors au niveau des mécanismes de sélection et d'activation des comportements élémentaires. Là encore, des mécanismes figés ou rigides ne permettent pas de prendre en compte la diversité et la nouveauté des situations rencontrées. Si la Robotique tient à réaliser certains de ses objectifs, alors les mécanismes de sélection, de même que l'acquisition des comportements élémentaires, doivent être acquis automatiquement, en situation, selon les besoins ; cad par **apprentissage**.

Définition : L'apprentissage en robotique a pour objectif la modification automatique du comportement du robot pour améliorer son comportement dans son environnement.

Malheureusement, il n'est pas facile de directement utiliser l'apprentissage tel qu'il est mis en oeuvre dans les RNA (supervisé ou auto-organisation) pour l'apprentissage en robotique. En effet, l'apprentissage supervisé nécessite la préalable définition d'un ensemble d'exemples représentatifs du comportement recherché, tâche difficilement conciliable avec les caractéristiques de l'apprentissage. Comment savoir à l'avance le comportement désiré (pour la génération de la base d'exemples) et dans le même temps assurer l'amélioration automatique des performances comportementales ? Un compromis a été proposé dès les années 1990, qui prône l'utilisation d'une fonction d'évaluation "générique"¹⁷ des performances, appelée fonction de renforcement. Cette fonction de renforcement permet une description qualitative¹⁸ (par opposition à quantitative) du comportement recherché. La correcte description de cette fonction de renforcement est au coeur des questions posées par l'**apprentissage par**

¹⁶ et aussi de capteurs et d'effecteurs.

¹⁷ Relativement indépendante de l'architecture du robot et de son environnement.

¹⁸ Par exemple, ternaire : bon (+1), mauvais (-1), sans opinion (0).

renforcement, avec celles relatives au traitement de nouvelles situations par rapport à celles déjà connues (**généralisation**) et au transfert d'information dans le temps (**programmation dynamique**). Aujourd'hui, l'implantation de l'apprentissage par renforcement sur carte auto-organisatrices est une solution élégante et efficace au problème de la généralisation, tandis que des algorithmes, tel que celui de la Bucket Brigade, le TD(1) ou le Q-learning, permettent de tenir compte des renforcements retardés dans le temps. Les algorithmes "génétiques" sont aussi des algorithmes d'apprentissage par renforcement, bien que ses tenants préfèrent appeler fonction d'évaluation¹⁹ la fonction de renforcement.

Les exemples d'apprentissage par renforcement en robotique restent pour l'instant modestes : évitement d'obstacles, suivi de murs, retour au nid, exploration, ramassage, balayage, etc. Ils n'en démontrent pas moins la possibilité de synthétiser des comportements à partir de la simple définition d'une mesure de performance, dans un certain nombre de cas, plus simples à écrire que l'algorithme solution et d'une plus grande généralité. En particulier lorsque le monde réel est difficile à modéliser préalablement à la mission, comme pour l'exploration des autres planètes du système solaire, le fond des océans, ou, plus simplement, hors de l'atmosphère contrôlée de l'atelier de production.

L'acquisition et la démonstration de comportements (intelligents) par un **robot "autonome"** est certainement un noble but. Pourtant, au même titre que l'on loue la notion du travail d'équipe face à celle du travail individuel, il semble qu'un groupe de robots est doué de plus de possibilités qu'un seul individu. Là encore, notre nature (humaine) et notre formation (scolaire) ne nous préparent pas au challenge de construire des systèmes multi-robots. Nous avons bon espoir d'obtenir, au fil des expérimentations en **robotique coopérative**, des données importantes sur les notions de coopératif/collectif, apprentissage du groupe, répartition/distribution automatique des tâches et génération automatique de plans d'action.

La robotique coopérative est un sujet ardu, mais les gains sont à la hauteur de la tâche. Une meilleure compréhension des mécanismes qui permettent l'émergence d'un comportement coopératif au sein d'un groupe de robots doit nécessairement fournir des informations importantes, susceptibles d'éclairer d'un jour nouveau le comportement collectif d'hommes et femmes travaillant en équipes. A court terme, il est aisé de prévoir des gains de productivité pour l'entreprise, éventuellement une diminution du stress individuel lié à un meilleur positionnement de chacun dans le groupe. A plus long terme, l'on peut imaginer progresser ainsi sur le chemin qui nous mène à l'étape suivante de notre évolution²⁰ telle que celle revendiquée par certains auteurs de science-fiction²¹: le groupe comme individualité.

¹⁹ Fitness rule (en anglais).

²⁰ Si tout va bien.

²¹ T. Sturgeon en particulier.

1.1 Réseaux de neurones séquentiels (Thèse 1987-1990)

L'histoire du domaine des réseaux de neurones artificiels appliqués aux Sciences Pour l'Ingénieur (SPI), aussi appelé connexionnisme, est récente. L'intérêt général pour cette thématique de recherche a été relancé en 1986, suite à la découverte de l'algorithme d'apprentissage par rétropropagation du gradient d'erreur pour les réseaux multicouches. Cette avancée a permis d'envisager l'application des techniques connexionnistes à de réels problèmes (par exemple industriels). Aujourd'hui, il est beaucoup plus difficile de recenser, in extenso, les domaines où le connexionnisme s'applique, que l'inverse. Toute application qui doit montrer des capacités d'apprentissage, de généralisation et où l'on dispose d'exemples du comportement souhaité est éligible. Les trois premiers grands domaines d'application sont : la vision (identification, classification, contrôle qualité, etc...), le traitement du signal (reconnaissance de la parole, séparation de sources, contrôle, etc...) et l'aide à la décision (prédiction, diagnostic, estimation, etc...).

La réalisation d'une application selon une approche connexionniste implique la synthèse d'une fonction de mise en correspondance des exemples d'entrée-sortie au cours de la phase d'apprentissage. L'apprentissage permet d'éviter la génération, coûteuse, par un opérateur humain d'un algorithme solution, dont la validité est limitée par des conditions d'utilisation précises (contraintes environnementales en particulier).

Les travaux menés autour du développement d'applications utilisant une approche IA ont permis de dégager trois problèmes fondamentaux :

- Le passage de valeurs numériques (données capteurs) à des valeurs symboliques reste le point d'étranglement que n'arrivent pas à franchir les systèmes à base de connaissances. Une approche qui évite le problème par un traitement numérique des informations est souhaitable.*
- La tendance au niveau applicatif est à la prise en compte, dans le cahier des charges lui même, de l'hétérogénéité des capteurs, du vieillissement et d'une nécessaire adaptabilité de l'application aux conditions locales d'utilisation.*
- Toutes les connaissances expertes ne sont pas formalisables par une approche à base de règles. Il existe des connaissances expertes, qualifiées d'intuitives, qui font partie du savoir-faire des entreprises, et qu'il faut capitaliser (stockage et utilisation).*

L'approche connexionniste répond, au moins en partie, à ces conditions. Aussi, une communauté de chercheurs connexionnistes s'est rapidement constituée. Les modèles neuronaux proposés aujourd'hui sont particulièrement représentatifs de la grande variété des domaines d'origine des chercheurs impliqués. Il manque une unité dans la démarche scientifique qui a pour objet la génération, la compréhension et l'utilisation de ces modèles.

Définition du contexte

La très grande majorité des modèles de réseaux de neurones artificiels sont dotés d'un fonctionnement combinatoire : la sortie du réseau est uniquement dépendante de l'entrée actuelle. Il existe cependant un grand nombre d'applications où la réponse doit être fonction des entrées actuelle et passées. Ainsi, les modèles connexionnistes montrent de grande aptitude à la reconnaissance de formes, mais la reconnaissance de mouvements vus comme une succession de formes est un domaine d'applications peu exploré, bien que très demandeur. D'autres implications importantes sont attendues dans les domaines de la prédiction d'éléments de séquences (météorologiques par exemple) et la génération de séquences (musicales par exemple). Quelques modèles connexionnistes ont été proposés pour traiter des problèmes de nature séquentielle. Cependant, aucun n'a été présenté dans un cadre formel permettant de connaître a priori ses propriétés et donc ses performances. De fait, une démarche guidée par le cahier des charges de l'application, définissant les propriétés et performances du modèles de réseaux de neurones artificiels à utiliser, est impossible.

Les approches proposées pour étendre le domaine d'application des réseaux neuronaux aux problèmes séquentiels se regroupent en trois classes. La première utilise un recodage spatial des entrées temporelles. Les limitations découlent évidemment du nombre restreint d'entrées pouvant être prise en compte. Le fonctionnement du réseau est combinatoire. La seconde approche fait appel à un codage de l'information temporelle localement sur le neurone. En fait, il y a ajout d'une boucle de rétroaction sur le neurone. Le comportement du réseau reste combinatoire, moyennant quelques aménagements des algorithmes d'apprentissage. La dernière classe regroupe les réseaux bouclés. Ces modèles sont les plus efficaces. Nous distinguons au sein de ceux-ci les réseaux séquentiels, qui utilisent le concept d'état interne. Cette approche au vu des résultats déjà obtenus et des concepts manipulés, apparaît la plus prometteuse pour résoudre des applications séquentielles. Les modèles les plus performants et les plus connus sont, dans un ordre chronologique, celui de Jordan et celui d'Elman.

L'architecture du modèle de Jordan est multicouche. La dernière couche est rebouclée sur la première (fig. 1). Les cellules d'entrée se répartissent en deux groupes : les cellules de plan et les cellules d'état. Les cellules de sortie sont rebouclées sur les cellules d'état par des connexions de poids fixes, de même pour les cellules d'état qui rebouclent sur elle-même par des connexions de poids fixes. L'état interne est défini par l'ensemble des cellules de sortie et des cellules de plan.

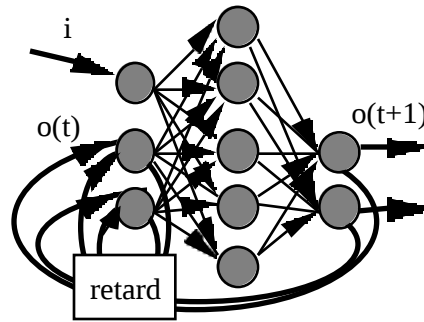


Figure 1. Réseau proposé par Jordan

L'architecture du modèle d'Elman est légèrement différente de la précédente. Il s'agit d'une structure multicouche (fig. 2.) où les boucles de rétroaction relient la couche cachée avec les cellules d'entrée. La couche d'entrée consiste en deux ensembles de cellules, les cellules de contexte et les cellules d'entrée. Il y a autant de cellules cachées que de cellules de contexte. Chaque cellule cachée est reliée à une seule cellule de contexte par une connexion fixe, de poids égal à 1. L'état interne est défini par l'ensemble des cellules de la couche cachée.

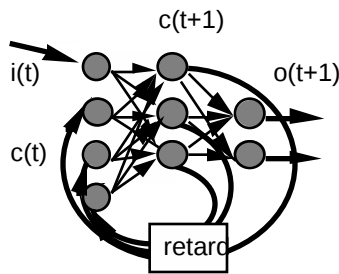


Figure 2. Réseau proposé par Elman

Pour ces deux modèles, l'algorithme d'apprentissage est la rétropropagation de gradient. Les performances de chacun de ces deux modèles sont déterminées par l'expérimentation. Leurs spécificités applicatives ne peuvent qu'être déduites d'expérimentations que l'on espère représentatives et exhaustives.

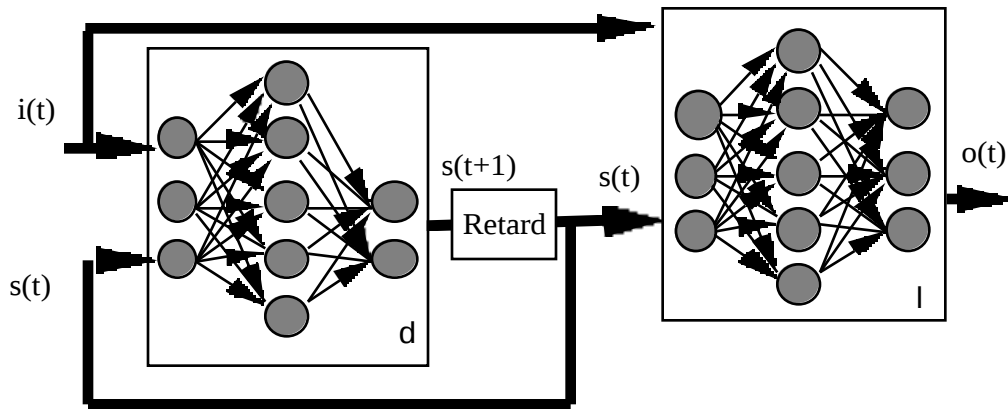
Définition de l'étude - Résultats

Figure 3. Machine séquentielle connexionniste

Le modèle de la machine séquentielle connexionniste que nous avons développé (fig. 3) constitue une évolution logique des approches connexionnistes du séquentiel. Inspiré par le modèle de machine séquentielle tel que le définit la théorie des Automates, il représente une généralisation d'une classe de modèles : les réseaux séquentiels. Il y a découpage en deux blocs fonctionnels, la fonction de transition et la fonction de sortie. Chaque fonction est réalisée par un réseau multicouche, permettant ainsi d'apprendre des fonctions de mise en correspondance non-linéaires. L'apprentissage est basé sur l'algorithme de la rétropropagation de gradient modifié. Trois algorithmes d'apprentissage différents ont été développés selon les informations disponibles au niveau du cahier des charges de l'application à résoudre.

La machine séquentielle connexionniste permet non seulement de décrire les différentes approches dans un cadre plus général, mais aussi de prédire et d'expliquer les applications possibles de chacune : les fonctions de transition et de sortie y sont plus explicites. Ainsi, le modèle de Jordan est une machine d'états connexionniste, il n'y a pas de fonction de sortie. La fonction de transition est réalisée par un réseau multicouche (3 couches). Il y a une connexion récurrente de chacun des neurones d'états internes sur lui-même. Ce modèle se comporte comme une machine d'état asynchrone. Pour un vecteur d'entrée donné, on observe l'évolution de la machine, qui passe par plusieurs états transitoires pour éventuellement aboutir dans un état stable.

Pour sa part, le modèle d'Elman correspond à une machine séquentielle connexionniste où les réseaux réalisant les fonctions de transition et de sortie sont des réseaux de neurones à deux couches. Il n'y a pas d'entrée primaire sur la fonction de sortie, c'est donc un modèle de type machine de Moore. La fonction de transition et la

fonction de sortie sont apprises. Cependant, comme il n'y a qu'une seule couche de poids modifiables pour chaque fonction, seules peuvent être réalisées des fonctions linéairement séparables. Par rapport à la machine séquentielle connexionniste, ce modèle ne peut pas réaliser d'associations non-linéaires au niveau de la fonction de transition.

Le modèle de Jordan et celui d'Elman présentent certaines limitations que nous avons explicitées. L'approche suivie par leurs auteurs est empirique : ayant imaginé une structure de réseau, ils vérifient qu'elle répond, ou non, au problème. La démarche qui a procédé au développement de la machine séquentielle connexionniste est inverse. Une structure générale pour la résolution de problèmes de nature séquentielle est proposée, structure que l'on adapte pour des applications particulières. De fait, la machine séquentielle connexionniste est une généralisation des réseaux de neurones séquentiels.

Les travaux expérimentaux que nous avons réalisés à l'aide du logiciel SACREN²² montre des propriétés nouvelles par rapport à la machine séquentielle (logique) :

- Généralisation de la reconnaissance à des séquences de valeurs discrètes alors que l'apprentissage a été réalisé sur des valeurs binaires.
- Réduction automatique de la table des états.
- Possibilité de retracer l'histoire des séquences soumises.
- Synthèse de l'automate sans connaissance à priori du graphe des états.
- Détermination automatique de la nature séquentielle ou combinatoire d'un problème.

²² SACREN : Système d'Aide au Choix d'un REseau de Neurones a été développé en Pascal sur station de travail Unix. Il permet de pallier au manque d'informations théoriques dans le domaine des réseaux connexionnistes en faisant appel de façon exhaustive à l'expérimentation. SACREN est basé sur la simulation événementielle, qui permet un gain de place mémoire important dans le stockage des architectures de réseaux à connexions non complètes (multicouches par exemple), une vitesse d'exécution élevée car seuls les éléments qui ont changé d'état sont remis à jour (nombre minimal de calculs), possibilités de gérer des délais au niveau des connexions (et donc tous types d'architectures est possible, en particulier bouclées). Le développement de SACREN a été l'objet d'un contrat avec l'ANVAR. Un certain nombre de modèles sont disponibles en bibliothèques comme les mémoires associatives, les cartes auto-organisatrices, les réseaux multicouches à rétropropagation de gradient et les modèles de réseaux séquentiels (dont la Machine Séquentielle Connexionniste).

1.2 Apprentissage par renforcement sur structure neuronale (1991-1993)

Dans le cadre de mes activités post-doctorales, je me suis intéressé à la coopération de plusieurs réseaux de neurones artificiels pour deux raisons. D'une part, dans le cadre des travaux prévus sur la compréhension et les possibilités d'utilisation du connexionnisme, il était nécessaire d'envisager l'implication au sein d'une application de plusieurs réseaux et non plus d'un seul, autant pour des raisons de modularité (développement, réutilisabilité) que pour des raisons de réduction de la complexité par une approche en décomposition de sous-problèmes. D'autre part, l'utilisation de la Machine Séquentielle Connexionniste mise au point durant mes travaux antérieurs m'avait déjà permis de mettre en oeuvre deux réseaux de neurones coopératifs, l'un pour la fonction de d'état, l'autre pour la fonction de sortie, au sein d'une application unique de reconnaissance de séquences. L'expérience acquise au cours du développement d'algorithmes d'apprentissage pour la Machine Séquentielle Connexionniste était un atout important pour la réussite de ce projet. Il était donc tout naturel que je poursuive mon activité scientifique dans ce sens.

S'il est vrai que de nombreux modèles de réseaux de neurones artificiels et de nombreux algorithmes d'apprentissage existent dans la littérature, peu de travaux ont été menés autour d'une coopération de réseaux pour réaliser une application. Ceci découle de l'impossibilité de réaliser sur les différents réseaux impliqués une distribution des informations quantitatives (gradient d'erreur par exemple) qui permette l'apprentissage supervisé. L'apprentissage par pénalité/récompense, aussi appelé apprentissage par essais et erreurs, manipule, non pas des grandeurs quantitatives pour réaliser l'apprentissage, mais des grandeurs qualitatives. Il s'agit, le plus souvent, d'un critère binaire qualifiant le bon ou mauvais comportement du système. Il nous est apparu que l'apprentissage par essais et erreurs permettait de résoudre le problème de la répartition de l'information critique entre les différents réseaux impliqués. Je me suis donc attaché à démontrer l'intérêt de cet apprentissage qualitatif pour une structure multi-réseaux. C'est le développement et la mise au point d'une implantation neuronale de l'apprentissage par essais et erreurs pour structures multi-réseaux qui ont constitué l'essentiel de mon activité scientifique post-doctorale.

Définition de l'étude - Résultats

La résolution d'un problème par un réseau de neurones peut se concevoir comme la découverte par apprentissage de la "bonne" relation d'entrée/sortie. Lorsque cette relation est complexe, le problème doit être décomposé en sous-problèmes terminaux. Chacun des comportements des sous-problèmes terminaux est alors réalisé par un réseau de neurones, que nous appelons réseau de base. Il faut donc déterminer, pour un problème complexe donné, d'une part la structure d'interconnexions des réseaux de base et d'autre part, les comportements que doivent réaliser chacun de ces réseaux. L'apprentissage peut être réalisé hors contexte si l'on connaît pour chacun des réseaux le comportement à réaliser, ou en contexte, à partir de la seule relation globale d'entrée/sortie²³. Nos travaux relatifs au développement de structures multi-réseaux se sont déroulés en quatre étapes :

- 1- étude des propriétés applicatives des différents modèles de réseaux aujourd'hui disponibles,
- 2- définition de critères permettant d'associer ces différents modèles les uns aux autres,
- 3- recensement des types de structures permettant les associations,
- 4- développement d'une technique d'apprentissage sur ces nouvelles structures.

1 - Cette étape est relative aux spécificités applicatives des divers modèles de réseaux de neurones disponibles. Nous avons étudié (1987-1993) les modèles suivants : les réseaux multicouches, les cartes auto-organisatrices, les mémoires associatives, les réseaux de type ART. La bonne compréhension, et donc la bonne utilisation des modèles connexionnistes, passe par une expérimentation exhaustive, telle que celle requise par des développements applicatifs. Dans tous les cas cités, l'étude de ces modèles a été réalisée au cours de recherches contractuelles (cf. contrats).

2 - Dans cette étape, nous avons cherché à caractériser le fonctionnement des modèles connexionnistes. L'objectif est de pouvoir spécifier les caractéristiques associatives des modèles neuronaux. Nous avons retenu huit critères de la théorie de la modélisation/simulation :

temps discret / continu	variables discrètes / continues
déterministe / stochastique	modèle autonome / non autonome
invariance / non invariance par rapport au temps	combinatoire / séquentiel
modèle instantané / non instantané	synchrone / asynchrone

²³ Dans l'approche liée au développement de la Machine Séquentielle Connexionniste que, la structure est fixée volontairement comme celle d'une machine séquentielle et nous avons étudié les possibilités d'apprentissage hors contexte lorsque l'on connaît pour le réseau de transition et le réseau de sortie les comportements à réaliser, et les possibilités d'apprentissage en contexte, lorsque le graphe d'états de l'automate à synthétiser est inconnu.

Ces critères nous ont permis de faire les constatations suivantes :

- La plupart des réseaux de neurones sont, aujourd'hui, des modèles à temps discret. L'unique raison semble être la facilité de programmation et de simulation.
- Les variables descriptives du modèle peuvent être soit discrètes, soit continues. Il ne s'agit pas là d'un critère très utile. Les réseaux neuronaux contiennent souvent des variables continues : les poids des connexions. Notons cependant qu'un réseau dont les variables de sortie sont continues peut fournir une infinité de réponses. Inversement un réseau dont les sorties sont discrètes ne peut fournir qu'un nombre fini de réponses.
- Certains modèles, tel que les réseaux à apprentissage par pénalités/récompenses, sont stochastiques. La phase d'apprentissage rend le fonctionnement de ces réseaux déterministe.
- Un modèle autonome, dans le cas des réseaux de neurones, est un réseau sans entrée. Les variations de la sortie résultent de phénomènes internes générés par des neurones oscillatoires par exemple.
- La plupart des réseaux utilisés par les SPI sont des modèles invariants par rapport au temps, sauf dans le domaine de la modélisation de réseaux de neurones biologiques.
- Les réseaux combinatoires avec retards sur les connexions, tels ceux nécessités par les boucles, sont non instantanés.
- La plupart des réseaux sont considérés, implicitement, comme synchrones : toutes les variables d'état évoluent à la même date.

Remarquons que l'on peut utiliser un neuvième critère selon la nature implicite ou explicite du temps dans les modèles. Souvent le temps est implicitement confondu avec la notion d'itération. Cependant, cette classe de modèles à temps implicite regroupe à la fois les modèles combinatoires instantanés et les modèles séquentiels synchrones à temps discret, ce qui la rend peu précise.

Ces huit critères nous ont permis de définir plus précisément chacun des modèles de réseaux de neurones artificiels. Leur utilité est évidente lorsqu'il s'agit d'envisager les possibilités de liaisons série ou parallèle entre réseaux. Ils ont de plus permis de recenser un certain nombre de fonctionnement "manquants", dont l'absence

semble uniquement liée au fait que la proposition des modèles et de leur fonctionnement est le reflet d'une méthode de conception au coup par coup. Ce constat confirme le besoin d'une démarche méthodologique scientifique pour le domaine du connexionnisme.

Le travail réalisé dans cette étape est partie intégrante d'une thèse co-encadrée (1990-1993, O. Sarzeaud) sur l'étude et le développement de réseaux de neurones artificiels modulaires.

3 - Notre étude des différentes structures d'interconnexions de réseaux a débutée par une analyse bibliographique visant à regrouper et répertorier toutes les possibilités d'interconnexions, depuis les plus classiques (série-parallèle) jusqu'aux structures intégrant des mécanismes de contrôle (champs de potentiels, addition vectorielle, hiérarchique, etc. ...). Nous avons porté un intérêt tout particulier à ces dernières car nous pensons qu'il est possible de remplacer la fonctionnalité de la structure de contrôle centrale (superviseur) par un algorithme d'apprentissage adéquat : l'apprentissage par pénalité/récompense, objet de l'étape suivante. Un exemple décrit plus loin illustre la possibilité d'éviter la présence d'un générateur central de formes locomotrices pour l'apprentissage de la marche hexapode (cf. 1.3).

4 - L'apprentissage par pénalité/récompense nous est apparu comme une solution technique permettant de répartir l'information critique entre les différents réseaux de la structure d'interconnexion. Originellement proposé en 1985, il s'applique à toute architecture de réseau, la seule contrainte est de disposer de neurones de sortie stochastiques binaires. Une fonction de renforcement analyse le comportement du système et fournit à chaque itération un signal de renforcement binaire qualifiant le comportement. Il ne s'agit plus, comme dans le cas de l'apprentissage supervisé, de diriger l'apprentissage par des grandeurs quantitatives significatives de l'erreur, mais plutôt de piloter l'apprentissage au moyen d'une information qualitative. Cette information qualitative est à la disposition du système dans son ensemble. Dans notre cas, ce système est une interconnexion de réseaux. Chaque réseau reçoit la même information qualitative dans le cas d'une unique fonction de renforcement commune. Il peut aussi exister une fonction de renforcement spécifique par réseau, nous retrouvons ici une déclinaison de l'apprentissage hors-contexte. L'avantage d'une répartition d'information qualitative réside dans l'absence de dilution d'information, à l'inverse de celle observée avec l'apprentissage supervisé. La modification des poids est réalisée par la classique méthode du gradient. La figure 4 résume le fonctionnement de l'algorithme.

L'étape 4 de ce travail est à l'origine d'une thèse co-encadrée (1992-1996, S. Sehad) sur l'étude et le développement de modèles connexionnistes à apprentissage par renforcement.

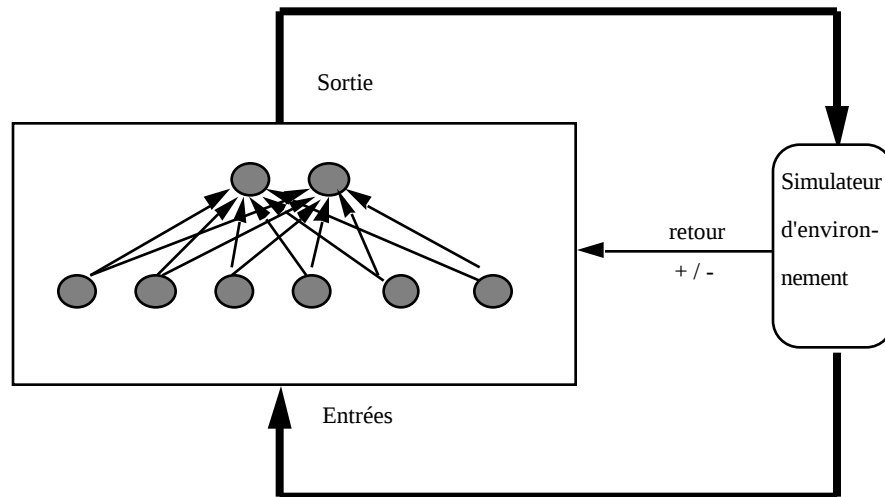


Figure 4. Algorithme d'apprentissage par pénalité/récompense

Le problème considéré pour illustrer nos travaux sur la coopération de réseaux de neurones est l'acquisition de la marche pour un robot hexapode. La marche étant un processus séquentiel, chacune des pattes du robot est pilotée par une machine séquentielle connexionniste. La structure comporte donc six machines séquentielles en parallèle, soit douze réseaux de neurones. L'entrée de chacune des machines est un ordre du type "marche" ou "stoppe". La fonction de renforcement analyse le comportement du robot et détermine s'il y a chute ou avancée. Dans le premier cas, un retour négatif est renvoyé vers tous les réseaux de neurones. Dans le second cas, un retour positif est envoyé. En quelques centaines d'itérations, les six machines séquentielles se coordonnent de façon à produire une marche tripode classique (une itération correspond à une configuration des pattes). Il y a dix marches possibles, qui sont toutes découvertes. L'espace de recherche contient 46656 successions de trois mouvements.

Une démonstration logicielle de cette application en C, sur station de travail Silicon Graphics, permet une visualisation 3D de la découverte de la marche. Ce logiciel d'animation a été retenu par le Ministère de la Recherche et de l'Espace pour être présenté dans ses jardins parisiens lors des journées de la Science 1993. Une collaboration avec l'École des Mines d'Ales a permis de réaliser un robot mobile doté de six pattes, des capteurs et de l'algorithme précédemment cité, qui valide nos résultats face au monde réel. Une seconde version pour Macintosh, réalisé avec un groupe d'élèves de l'école, a remporté le concours organisé par Apple France : Programmer un comportement intelligent (1993).

1.3 Synthèse neuronale de comportement adaptatifs (1994-1997)

La poursuite de mes activités scientifiques sur le thème de la synthèse neuronale de comportement adaptatifs est justifiée par les éléments suivants. Tout d'abord, la communauté scientifique a entériné l'intérêt porté aux systèmes multi-réseaux et aux systèmes hybrides face aux problèmes soulevés dans l'introduction générale, c'est à dire passage du numérique au symbolique, hétérogénéité du monde réel, adaptabilité, capitalisation de savoir-faire, etc... D'autre part, pour concluant que soient nos résultats, ceux-ci n'ont été vérifiés qu'en simulation. La simulation est une étape normale du développement d'une application industrielle, cependant elle ne possède pas le même degré de validation qu'une expérience menée dans le monde réel. Une étape importante de notre travail implique donc le développement et la mise au point de systèmes multi-réseaux en prise directe avec le monde réel.

Durant un séjour de deux mois (1993) comme professeur invité à l'EPFL (Lausanne, Suisse), j'ai découvert avec le robot Khepera une plate-forme d'expérimentation idéale. J'ai participé aux développements logiciels qui accompagnent Khepera et j'ai pu vérifier que l'implantation de nos algorithmes d'apprentissage par renforcement sur structures neuronales est aisée. L'utilisation de ce robot permet de vérifier l'intégration par les réseaux de neurones de l'hétérogénéité des capteurs, des imperfections de fonctionnement des effecteurs et de l'extrême diversité (situations, bruit) rencontrées dans le monde réel.

Un second intérêt de l'utilisation d'un robot est lié à son caractère pédagogique. Le transfert de nos résultats de recherche vers la communauté scientifique, le monde industriel et le public²⁴, est facilité par ce média.

Il est important de noter que notre domaine de recherche n'est pas la Robotique, et que nos résultats ne se comparent pas. Notre objectif initial est de développer et de tester nos résultats de recherche au sein d'un environnement hétérogène, imprévisible et changeant comme, par exemple, le monde réel. Par définition, un robot est aujourd'hui l'outil idéal pour intégrer et tester nos développements. Le nombre d'expérimentations aujourd'hui possibles avec un robot est réduit, ce qui nous conduit souvent à effectuer des expérimentations similaires à celles des roboticiens. Cependant, nos motivations et l'interprétation de nos résultats sont différents. Pour notre part, il s'agit de valider des capacités d'apprentissage et de généralisation démontrés par les réseaux de neurones artificiels impliqués et non de trouver des solutions à des problèmes de robotique.

²⁴ Ainsi que le démontre notre victoire au concours national Apple (Nunc Est Programmandum, 1993) sur le thème de la programmation d'un comportement intelligent. Ce logiciel, baptisé ANTROID, montre un robot hexapode muni d'un réseau de neurones artificiels apprenant à marcher par renforcement.

Définition de l'étude - Résultats

En continuité de nos travaux sur l'apprentissage par pénalité/récompense, nous nous sommes intéressé à l'**apprentissage par renforcement**. L'apprentissage par renforcement réalise la synthèse d'une fonction de mise en correspondance entre les situations et les actions qui maximise un signal de renforcement (positif). Au même titre que l'apprentissage par pénalité/récompense, l'apprentissage par renforcement fait appel à des critères qualitatifs pour réaliser l'apprentissage.

Historique

L'apprentissage par renforcement est tout à la fois un sujet nouveau et ancien pour l'Intelligence Artificielle. Le renouveau date des années 1980, avec dans la foulée la proposition d'une version neuronale par Barto et al. en 1985. En 1989, Watkins propose le Q-learning qui est vraisemblablement l'algorithme le plus étudié. Il existe plusieurs autres méthodes d'apprentissage par renforcement telles que le TD(0) et le AHC.

Objectif

L'apprentissage par renforcement est l'apprentissage d'une fonction de mise en correspondance entre des situations et des actions maximisant un signal de renforcement. C'est un apprentissage par essai et erreur. En effet, une fonction habituellement aléatoire permet de parcourir les différentes situations (essais). Des heuristiques, créées par l'opérateur humain, permettent de qualifier chaque action entreprise par rapport à chaque situation (erreur). L'objectif est d'obtenir un comportement acquis par l'agent qui soit le plus récompensant possible. Le signal de renforcement est un critère qualitatif (par exemple binaire par opposition à un signal quantitatif). La qualification des combinaisons n'est pas toujours immédiate. Elle peut être retardée dans le temps car dépendante des états futurs. Lorsque le signal de renforcement n'est disponible qu'après une longue séquence d'actions, comment noter individuellement chacune des actions ? L'un des apports de l'apprentissage par renforcement réside dans la possibilité de mémoriser une information séquentielle au sein d'une implantation combinatoire.

Algorithme : le Q-learning

Le Q-learning associe à chaque paire de situation-action une valeur d'utilité. Trois fonctions différentes sont impliquées : mémorisation, exploration et mise à jour (Fig. 5). En réponse à la situation actuelle, une action est proposée par la mémoire de l'agent. Cette action est celle qui présente la meilleure probabilité de renforcement positif (récompense). Cette proposition est cependant modifiée pour permettre une exploration extensive de l'espace des paires de situation-action. Après l'exécution de l'action par le robot dans le monde réel, une fonction de renforcement, imaginée par un opérateur humain, fournit une valeur de renforcement. Cette valeur, un simple critère qualitatif (+1, -1 ou 0), est utilisée par l'algorithme d'apprentissage pour ajuster la valeur (Q) d'utilité associée à chaque paire de situation-action. L'apprentissage est incrémental car l'acquisition des exemples est

réalisée séquentiellement dans le monde réel. La qualification des paires de situation-action n'est pas toujours immédiate: seul un petit nombre de cas sont suffisamment signifiants pour la fonction de renforcement (+1 ou -1).

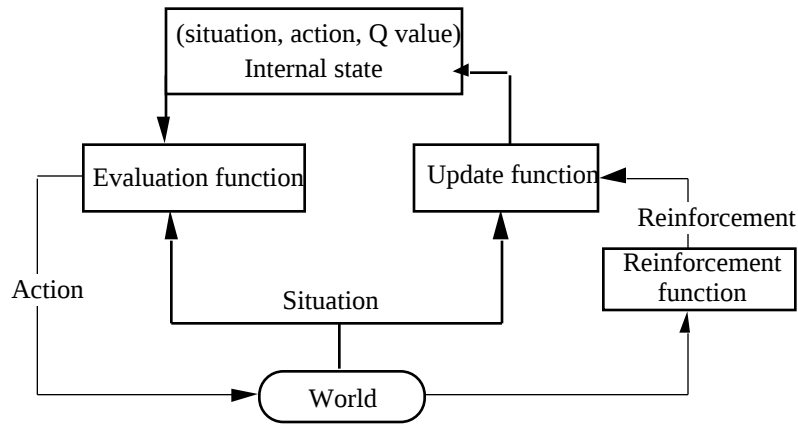


Figure 5. Modèle général pour les algorithmes de renforcement (Q-learning)

Résultats théoriques

Il a été démontré que l'application de l'équation suivante, de mise à jour des valeurs d'utilité (Q), permet la synthèse d'un comportement optimal :

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha (r + \partial \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

$Q(s_t, a_t)$ est l'utilité d'une action particulière a_t dans une situation donnée s_t , r le signal de renforcement envoyé par la fonction de renforcement, α et ∂ des constantes positives ($\in [0, 1]$)

Le dernier terme de cette expression est une estimation de l'erreur réalisée entre la prédiction de la valeur d'utilité attendue et la valeur courante $Q(s_t, a_t)$. La prédiction comprend le retour instantané r plus la valeur maximale d'utilité que l'on peut espérer du futur, représentée par la valeur d'utilité maximale pour la situation suivante (quelquesoit l'action a) $\max_a Q(s_{t+1}, a)$. La valeur Q maximale suivante est donc considérée comme la valeur d'utilité maximale possible pour le futur. Cette règle de mise à jour a pour effet de propager le renforcement reçu pour une paire de situation-action particulière $(s_{t+1}, \arg\max_a Q(s_{t+1}, a))$ à d'autres paires (ici, $Q(s_t, a_t)$). Il s'agit donc d'une rétropropagation des renforcements reçus dans le temps (les valeurs à $t+1$ influent sur le calcul des valeurs à la date t).

A l'avenir, si l'agent se retrouve dans la même situation s_t , la remise à jour de la valeur d'utilité qui vient d'avoir lieu pour l'accomplissement de l'action a_t l'aidera certainement dans la sélection de l'action la plus récompensante.

Convergence

L'aptitude de l'apprentissage à construire le comportement optimal a été démontré dans le cas d'une implantation sous forme de tableau (autant de lignes que de situations, autant de colonnes que d'actions, les cases de la matrice sont les valeurs Q), pour peu que chaque paire de situation-action soit vu un nombre infini de fois.

Les difficultés rencontrées dans la mise en oeuvre de ce type d'apprentissage résultent du fait que l'espace des situations possibles combiné avec celui des actions potentielles est si grand, qu'une exploration exhaustive de toutes les paires de situation-action est impossible, de même qu'une mémorisation exhaustive. C'est sur ces points que nous avons porté nos efforts.

MISE EN OEUVRE EN ROBOTIQUE

Même un robot doté d'un petit nombre de capteurs (N_{capteurs}) de résolution (D) est capable de percevoir un grand nombre de situations différentes $N_{\text{situations}} = ((2)^D)^{N_{\text{capteurs}}}$. Celles-ci se combinent avec les actions potentielles (N_{actions}) pour définir le cardinal du extrêmement grand espace de recherche des comportements potentiels : $(N_{\text{actions}})^{N_{\text{situations}}}$. D'autre part, un robot est par définition un artifact matériel dont les contraintes mécaniques ne permettent pas plus de quelques mouvements par seconde (moins d'une dizaine habituellement). Cette contrainte ajoutée à la durée de vie des batteries (habituellement quelques heures) ne permet pas d'envisager plus de quelques milliers de mouvements par expérience. Il faut donc obligatoirement généraliser à partir des informations acquises, et essayer d'acquérir les informations les plus pertinentes. Nos travaux ont porté en particulier sur la développement d'une implantation neuronale du Q-learning qui autorise une "bonne" généralisation. Durant la même période, des algorithmes ont été proposé par ailleurs pour optimiser l'exploration (par exemple Queue-Dyna).

Mémorisation/Généralisation

Plusieurs améliorations des implantations du Q-learning ont été proposé pour répondre au problème posé par la généralisation. Mahadevan et al. utilisent une distance de Hamming pour généraliser entre des situations similaires. Les mêmes auteurs utilisent aussi des clusters pour généraliser entre paires de situation-action similaires. Sutton propose l'architecture Dyna qui améliore la remise à jour de la mémoire par la réutilisation des expériences déjà rencontrées. Pour toutes ces améliorations, la généralisation est limitée à des critères syntaxiques (entre situations) ou semi-sémantiques (clusters de situation-action).

Les implantations neuronales du Q-learning sont peu nombreuses et impliquent uniquement des réseaux multicouches (en 1994). Dans ce cas, l'algorithme de remise à jour est un algorithme de descente de gradient

d'erreur. Un signal d'erreur sur les neurones de sortie doit donc être défini. La définition de cette erreur est limitée aux cas simples où seules deux actions sont possibles. De fait, Lin propose le modèle QCON : une implantation sur perceptron multicouche de l'algorithme du Q-learning qui présente pour particularité de n'avoir qu'un unique neurone de sortie. Il y a donc autant de réseaux QCON qu'il y a d'actions potentielles. Ce faisant, la généralisation entre paires de situation-action est impossible. D'autres implantations sur réseaux multicouches ont été proposés, notamment par notre équipe, mais elles ne sont pas encore satisfaisantes. De plus, il est impossible d'interpréter explicitement les connaissances acquises par un réseau multicouche. Ce qui, lorsque l'apprentissage s'est correctement déroulé, est une limitation à l'emploi de la solution trouvée par le réseau.

Nous avons tenté d'éviter ce problème de généralisation et d'interprétation par l'utilisation de carte auto-organisatrice pour l'implantation du renforcement. Ce développement constitue le coeur du travail présenté dans la thèse de S. Sehad (soutenue en 1996).

Une carte auto-organisatrice est utilisée pour stocker les valeurs Q. La phase d'apprentissage associe à chaque neurone de la carte une paire de situation-action et sa valeur Q. C'est une méthode de regroupement des états qui implique des critères de similarité syntaxique et de localité. Le nombre de paires mémorisées est égal au nombre de neurones. La propriété de voisinage de la carte de Kohonen permet de généraliser entre paires de situation-action similaires. A la différence de l'implantation sur perceptron multicouche, l'interprétation des poids est ici possible. De plus, si un comportement correct est synthétisé (cad. seuls des renforcement positifs ou nuls sont reçus), alors tous les neurones codent des valeurs Q positives. Cette dernière propriété permet l'optimisation des connaissances mémorisées.

La carte auto-organisatrice est utilisée de la manière suivante : la meilleure action à entreprendre dans une situation donnée est fournie par le neurone qui montre la distance la plus courte avec la situation d'entrée et une valeur de renforcement Q égale à +1. Le neurone sélectionné correspond à un triplet (situation, action, valeur Q). C'est cette action particulière qui devrait offrir la meilleure récompense dans la situation actuelle.

L'algorithme d'apprentissage remet à jour le poids correspondant à la valeur Q et aussi les poids relatifs à la situation et à l'action. C'est le neurone représentant la situation et l'action effectivement réalisée qui est sélectionné. La mesure de distance utilisée est différente de celle utilisée par la fonction d'exploration. Elle inclut les vecteurs de situation et d'action, mais pas la valeur de renforcement Q. En même temps que le neurone sélectionné, ses quatre voisins sont aussi remis à jour. Durant l'apprentissage, l'influence sur les voisins décroît proportionnellement à l'inverse du nombre d'itérations.

La fonction de mémorisation agit sur les poids du réseau de neurones artificiels. La taille mémoire requise par le système pour stocker les connaissances est définie, *a priori*, par le nombre de connexions dans le réseau. Elle est indépendante du nombre de paires de situation-action explorées, à la différence d'une implantation matricielle.

Comparaison entre implantations non-neuronales et neuronales

Les expérimentations menées avec le robot Khepera se déplaçant dans un environnement réel montrent que l'implantation sur carte auto-organisatrice du Q-learning requiert moins de mémoire (nombre de réels) et apprend plus vite que toutes les autres²⁵. Le comportement à synthétiser dans ce cas précis est un comportement d'évitement d'obstacles. Les résultats de la table 1 démontrent que cette implantation du renforcement est simple, rapide et efficace.

	Q-learning	+ Hamming	+ clusterisation	Dyna-Q	Multicouche	carte
Mémoire	6400	6400	$1.6 \cdot 10^6$	6400	56	176
Nbre d'it.	7500	3500	4000	6000	2000	500
Durée	55 mn	25 mn	30 mn	45 mn	8 mn	2 mn

Table 1. Comparaison de plusieurs implantations du Q-learning lors de la synthèse d'un comportement d'évitement d'obstacle.

Devant le succès rencontré par l'implantation à l'aide de carte auto-organisatrice, nous avons démarré en 1995 une thèse co-encadrée en co-tutelle avec l'Université de Buenos-Aires (Argentine) sur l'implantation du renforcement sur des réseaux à fonctions radiales de base (RBF). Les réseaux RBF présentent des propriétés intéressantes de couverture de l'espace de sortie par des valeurs réelles (les fonctions de transition sont des gaussiennes) alors que les cartes auto-organisatrices utilisent des valeurs discrètes. La définition de la fonction de mise en correspondance entre situations et actions pourrait être plus fine, plus rapide et plus efficace. Dans le cadre de ce travail de thèse, une automatisation de la définition des valeurs de seuils de la fonction de renforcement est aussi recherchée (cf. p. 24).

²⁵ Il faut noter que pour cet exemple, les dimensions de l'espace des situations et des actions sont respectivement de 1023^8 situations (approximativement 10^{24}) et, chaque moteur pouvant prendre 20 valeurs, 400 actions possibles par situation. La même fonction de renforcement a été utilisée pour toutes les implantations (+1 s'il évite, ou -1 s'il y a collision, ou 0 sinon). Les capteurs infra-rouges ont une portée utile de 5 à 2 cm ([0 .. 1]). Nous considérons que le robot évite lorsque la somme actuelle des valeurs sur les capteurs est inférieure à la somme précédente, la différence étant supérieure à 0.06. Il y a collision lorsque la somme des six capteurs frontaux est supérieure à 2.90 ou la somme des deux capteurs arrière est supérieure à 1.95. Les valeurs de seuil (0.06, 2.90, 1.95) ont été déterminées après de nombreuses expérimentations.

En parallèle de ce travail sur l'apprentissage par renforcement mettant en oeuvre pour sa partie expérimentale un bras de robot, nous avons co-encadré une thèse soutenue en 1997 ayant pour objectif l'étude de la reconfiguration d'une antenne de capteurs hydrophoniques à l'aide d'une approche neuronale²⁶. Une partie de ce travail utilise une carte auto-organisatrice avec un apprentissage par renforcement : il s'agit d'améliorer par l'utilisation de critères de continuité locaux de la fonction de coût, la définition des pondérations nécessaires à la formation de voies. La fonction de renforcement est alors définie comme une fonction de coût impliquant la largeur du lobe principal et le niveau des lobes secondaires.

²⁶ en collaboration avec la DGA/DCE-DLSM (Le Brusc, 83).

2. PROGRAMME DE RECHERCHE 1998-

Aujourd'hui, le connexionnisme a démontré suffisamment de qualités pour pénétrer l'industrie. Cependant, la quasi totalité des applications font appel au modèle du perceptron multicouche et à l'apprentissage supervisé (rétropropagation du gradient d'erreur). L'usage des méthodes connexionnistes peut être qualifié de pragmatique : les réseaux de neurones sont utilisés conjointement à d'autres techniques, chacune dans sa spécialité.

Les réseaux de neurones supervisés ont ouverts de nouveaux domaines d'applications aux ingénieurs en leur fournissant un outil simple et efficace pour l'apprentissage par l'exemple et la généralisation. Il est de ce fait possible d'abandonner la définition d'un algorithme solution pour une programmation à l'aide de exemples représentatifs du comportement souhaité. De la même manière, l'apprentissage par renforcement sur support d'implantation neuronal (de type carte auto-organisatrice) nous semble être doté d'excellentes potentialités du fait de l'association de la généralisation avec la synthèse de la solution à partir de la simple définition d'une fonction de renforcement : une mesure qualitative de la performance de solution souhaitée. L'apprentissage par renforcement apparaît comme une autre alternative à la programmation algorithmique.

Cependant, bien que l'intérêt et les mécanismes fondamentaux soient aujourd'hui explicités dans le cas d'un seul réseau et d'une seule fonction de renforcement, un véritable développement applicatif au sein d'une réelle application nécessite certainement la mise en oeuvre conjointe de plusieurs comportements élémentaires au sein d'une approche par décomposition en sous-problèmes. C'est à ce niveau que, capitalisant sur nos résultats antérieurs, nous envisageons notre travail de recherche pour les années à venir.

Ce programme de recherche sera mené dans le cadre de l'équipe Modélisation et Simulation des Systèmes dirigée par le Professeur Norbert Giambiasi au DIAM-IUSPIM ainsi que durant mon séjour au CESAR-ORNL dirigé par le Dr. Jacob Barhen au sein du Computational Intelligence Group du Dr. L. E. Parker.

2.1 Sujet proposé

Il est considéré comme un fait établi (au sein de la communauté “robot learning”) que l’acquisition d’un comportement par apprentissage par renforcement nécessite l’introduction de biais, c’est à dire d’*a priori*. L’objectif est soit de réduire la taille de l’espace de recherche, soit de réduire la complexité du comportement solution. A notre avis, l’introduction de biais implique de fait un certain nombre de restrictions qui ne sont pas correctement évaluées :

1- Les *a priori* réduisent la taille de l’espace de recherche jusqu’à une taille pour laquelle le nombre d’exemples disponibles autorisent l’apprentissage. Cependant, cette réduction de la taille de l’espace de recherche est du même coup une réduction de la variété des comportements qui pourraient être solution. Même si nous sommes garantis de trouver le comportement optimal sur l’espace de recherche réduit, il n’y a aucune garanti que ce comportement soit optimal relativement à l’espace de recherche initial.

2- L’utilisation des techniques d’apprentissage est seulement considérée quand la définition de l’algorithme correspondant au comportement souhaité est trop difficile ou trop coûteuse. Ceci découle de notre incapacité à correctement modéliser l’environnement (y compris le robot). C’est pourquoi, la validité des *a priori* doit être questionné. Il n’y a pas de raison de penser que les *a priori* sont plus aisés à définir que l’algorithme solution.

A notre avis, il existe d’autres moyens d’introduire des connaissances qui sont moins sujettes à caution (et pourtant pas encore explorées), comme l’introduction de connaissance *a posteriori*, la mise au point automatique de la fonction de renforcement et l’augmentation du nombre d’exemples disponibles. C’est sur ces trois points que nous avons commencé à travailler

Ajout de biais *a posteriori*

L’ajout de connaissance *a posteriori* permet de profiter d’un espace de recherche complet pour l’apprentissage (et donc sans *a priori*), puis de contraindre le comportement appris pour obtenir une solution finale “optimale”. Ainsi, au lieu de réduire l’espace des solutions possibles puis de trouver la solution optimale restante, on recherche une solution dans l’espace des solutions complet que l’on va ensuite optimiser manuellement. Parmi les différentes contraintes *a posteriori* figure en particulier la possibilité d’exprimer des séquences comportementales (séquences d’actions), difficilement prises en compte par l’implantation (combinatoire le plus souvent). Ces contraintes constituent des modules externes à l’algorithme d’apprentissage qui permettent une définition plus

explicite du comportement séquentiel du système, aussi bien pour interdire que pour faciliter l'expression de certaines séquences comportementales.

Construction de la fonction de renforcement

Les fonctions de renforcement aujourd'hui utilisées sont toutes le résultat d'une grande part d'intuition complétée par une fastidieuse mise au point expérimentale sous forme d'essais et erreurs. La promotion vers des problèmes industriels des méthodes de renforcement ne peut se concevoir qu'avec la mise au point de concepts, outils et méthodes d'aide à la conception des fonctions de renforcement. Nous envisageons donc de définir des paramètres représentatifs de l'expression statique des fonctions de renforcement qui permettent d'inférer leurs comportements avant la phase d'apprentissage. Dans ses travaux de thèse que nous co-encadrons, J.M. Santos étudie le calcul automatique des valeurs de seuils pour l'expression de la fonction de renforcement suivante (un seuil par récompense positive ou négative) :

$$RF(s_1, \dots, s_u) = \begin{cases} \bar{b} + 1 & \text{if } g_1(s_1, \dots, s_u) > q_+ \\ \bar{b} - 1 & \text{if } g_2(s_1, \dots, s_u) < q_- \\ \bar{b} & \text{otherwise} \end{cases}$$

où $(s_1, \dots, s_u)$ est la situation telle que perçue par les capteurs, $g_1()$ et $g_2()$ sont des fonctions quelconques.

L'idée permettant le calcul des valeurs seuil est basée sur la nécessité de respecter un certain nombre de ratios si l'on veut obtenir convergence de l'apprentissage. Par exemple, du fait des caractéristiques de l'implantation par cartes auto-organisatrices, un ratio de 1 doit être respecté entre les renforcements positifs et les renforcements négatifs (au moins au début). De la même manière, un nombre minimal de renforcements non nuls est nécessaire pour permettre l'apprentissage.

Une version dynamique de cet algorithme devrait permettre une optimisation des valeurs de seuils de la fonction de renforcement tout au long de l'apprentissage. Aujourd'hui les seuils ne sont optimaux que par rapport aux conditions initiales de l'apprentissage.

Accroître le nombre d'exemples disponibles

Plutôt que de s'appliquer à réduire la taille de l'espace de recherche pour le rendre compatible avec le faible nombre d'exemples d'apprentissage disponibles, il nous semble intéressant d'envisager l'approche baptisée "lazy learning". Cette approche n'est à ce jour pratiquement pas utilisée dans le contexte de l'apprentissage par renforcement. Lors d'une phase exploratoire conduite aléatoirement, une mémoire séquentielle des couples de

situation-action est constituée indépendamment de toute utilisation. Dans un cadre applicatif précis, cette mémoire questionnée avec soin est alors utilisée pour générer le comportement désiré. La même mémoire peut être utilisée pour différentes applications.

L'approche "lazy learning" ne permet pas de résoudre le problème posé par les limites mécanique et énergétique du robot, mais elle permet de réduire la durée d'expérimentation pour toutes les expériences suivantes. Il ne s'agit pas non plus d'une incorporation de connaissance *a priori* car il n'y a pas d'*a priori* durant la phase exploratoire (aléatoire). Il ne s'agit pas d'un modèle de l'environnement car il n'y a pas de phase de modélisation. Cependant, la mémoire construite est utilisée comme un modèle (modèle non explicite) au sens où elle fournit le résultat de l'exécution d'une action sans requérir d'effectivement réaliser cette action. A l'extrême, l'expérience effective est uniquement une phase de test, validant (ou non) la fonction de renforcement utilisée ici pour interroger la mémoire.

Bien entendu, nous rencontrons là encore les mêmes problèmes de place mémoire et généralisation, c'est pourquoi nous envisageons une implantation sur cartes auto-organisatrices (utilisées ici comme mémoires associatives).

Robotique coopérative

L'un des intérêts majeurs de l'apprentissage par renforcement sa relative indépendance par rapport à la plateforme d'expérimentation. Nous avons pu vérifier cette assertion en portant aisément nos développements initialement obtenus avec Khepera (un robot de 80 gr.) sur Nomad 200 (100 kg.). Dans le même ordre d'idée, la robotique coopérative où plusieurs robots, dotés chacun d'un certain nombre de comportements élémentaires, agissent conjointement pour réaliser une tâche (qu'un seul ne peut pas accomplir) est une plateforme expérimentale qui semble bien se prêter à nos recherches.

L'apprentissage en robotique coopérative présente au minimum toutes les contraintes associées à l'apprentissage pour le robot individuel. Nous pouvons donc là nous appuyer sur nos travaux passés et présents: le paradigme d'apprentissage doit être sub-symbolique, l'apprentissage doit se faire par renforcement (il requiert moins de contraintes que l'apprentissage supervisé) et la généralisation doit être partie intégrante des caractéristiques. Mais, l'apprentissage en robotique coopérative implique en sus une communication explicite entre les robots dont le principal effet est d'augmenter la taille de l'espace de recherche. Il semble que, parmi les diverses possibilités, l'utilisation d'un ensemble fixe de dimensions pour représenter la connaissance relative aux autres membres du groupe offre l'avantage de ne pas lier la taille de l'espace de recherche au nombre d'individus.

Une cadre applicatif pour nos recherche en robotique coopérative est l'observation par un groupe de robots de plusieurs cibles en mouvement. Chaque robot ne dispose bien sur que d'un champ de vision limité et doit se coordonner avec d'autres pour optimiser l'observation. Une mesure de la performance est le nombre de cibles observées par le groupe.

A PLUS LONG TERME

Distribution des récompenses aux membres du groupe

La mesure de performance que nous venons de décrire s'applique au groupe de robots dans son ensemble. La question importante est " Comment distribuer individuellement les signaux de renforcement reçus par le groupe ? " La plupart des chercheurs s'attachent à une description manuelle de fonctions de renforcement locales à chaque robot. Pourtant, cette méthode ne peut pas toujours être employée. L'information locale à un robot peut ne pas suffire à produire des récompenses locales cohérentes avec la tâche assignée au groupe. Ce problème est souvent appelé le "perceptual aliasing problem". Pour ce type de problème, les approches basées sur les chaines de Markov cachées sont prometteuses lorsqu'elles sont appliquées à un unique robot. Cependant, les spécificités de la robotique coopérative imposent un certain nombre de modifications du fait qu'une meilleure connaissance de la situation observée est impossible.

Pourtant, il faut obligatoirement résoudre cette question pour autoriser l'inscription du renforcement au registre des apprentissages disponibles pour la robotique coopérative. Les mécanismes de distribution qui seront développés dans ce but dépassent le simple cadre de la robotique coopérative²⁷.

Réduction de la taille de l'espace de recherche

La nécessité de réduire la taille de l'espace de recherche fait suite au calcul de sa taille selon la règle suivante: $(N_{actions})^{N_{situations}}$. Ceci découle du fait que le comportement est vu comme une fonction mise en correspondance capteurs-moteurs. Le cardinal de l'ensemble capteurs est exponentiel relativement au nombre de capteurs. Pourtant, il est possible d'imaginer un comportement complexe comme une succession de comportements élémentaires, chacun n'utilisant qu'un nombre restreint de capteurs et moteurs. Dans ce cas, la taille de l'espace de recherche global (incluant tous les comportements élémentaires) n'est que polynomial relativement au nombre de capteurs.

La mise en pratique de cette idée nécessite cependant la proposition de mécanismes (critères d'enchaînement) permettant l'activation successive des différentes mémoires associatives implantant chacune un comportement

²⁷ Cette question est aussi au coeur des recherches sur les systèmes multi-agents.

élémentaire. L'utilisation de marqueurs temporel et spatial des événements mémorisés au sein des différentes mémoires associatives est envisagée. Chaque mémoire est définie "naturellement" par une modalité sensorielle. Les travaux de Gallistel en éthologie tendent à prouver l'existence de cartes de modalités sensorielles variées, mise en relation par ces deux aspects prédominants du réel que sont l'espace et le temps.

Il est important de noter qu'ici aussi la mise en séquence des comportements élémentaires est effectuée postérieurement à la construction des cartes sensori-motrices. Nous retrouvons donc là une composante forte de nos travaux, à savoir le "lazy learning".

Les coordonnées spatiale et temporelle semblent jouer un rôle pivot en permettant le passage d'un comportement élémentaire au suivant, par contre rien quel peut être le mécanisme qui permette la sélection du comportement élémentaire suivant ? Nous pensons qu'une activation automatique des sélections est possible à partir du moment où l'objectif final a été défini comme une position à atteindre sur une des cartes sensori-motrices. Si cela est possible, nous aurions alors à notre disposition un mécanisme d'apprentissage d'un niveau d'abstraction plus élevé à l'apprentissage par renforcement (bien que toujours sub-symbolique) : au lieu de nécessiter la définition d'une mesure de la performance du comportement recherché, il suffirait de spécifier le but à satisfaire.

3. DIFFUSION DES CONNAISSANCES

- 3.1 Thèse de doctorat**
- 3.2 Ouvrages scientifiques (2)**
- 3.3 Brevet (1 en cours de dépôt)**
- 3.4 Revues scientifiques avec comité de lecture (2) (+ 2 soumis)**
- 3.5 Conférences avec actes et comité de lecture (21) (+ 2 soumis)**
- 3.6 Colloques nationaux et internationaux avec actes à diffusion restreinte (9)**
- 3.7 Rapports de contrats (14)**
- 3.8 Tutoriaux (11)**

3.1 Thèse

[1] *Thèse de doctorat* *

"Contribution à l'étude et au développement de modèles connexionnistes séquentiels"

soutenue le 14 décembre 1990, à l'Université des Sciences et Techniques du Languedoc, Montpellier, devant le jury composé de :

M. C. Durante,	Professeur à l'Université de Montpellier II (Directeur du LAMM)
M. N. Giambiasi,	Professeur à l'Université d'Aix-Marseille III (Directeur du LERI)
M. J. Hérault,	Professeur à l'INP de Grenoble (Directeur du TIRF)
M. C. Jutten,	Professeur à l'Université J. Fourier (LTIRF, Grenoble)
M. F. Prunet,	Professeur à l'Université de Montpellier II (LAMM)
M. O. Sébilleau,	Ingénieur (Directeur Général de DIGILOG, Aix-les-Milles)

3.2 Ouvrages scientifiques

[2] C. TOUZET, Les réseaux de neurones artificiels, Introduction au connexionnisme, 150 pages, Préface de J. Hérault, EC2 éditeur, Paris, 1992 (cet ouvrage est aussi accessible sur Internet à l'adresse : <http://avalon.epm.ornl.gov/~touzetc/Book/Bouquin.htm>) *

[3] C. TOUZET & N. GIAMBIASI, "Application of Connectionist Models to Fuzzy Inference Systems", in Parallelization in Inference Systems, Lectures Notes in Artificial Intelligence 590, B. Fronhöfer & G. Wrightson Eds., Springer Verlag, April 1992. *

* Disponible sur Internet: <http://avalon.epm.ornl.gov/~touzetc/publications.html>

3.3 Brevets

Un brevet est en cours de dépôt par la SOLLAC relativement à l'utilisation des RNA pour la validation de données capteurs dans SACHEM (avec Marc Le Goc et Claude-Charles Thirion).

3.4 Revues scientifiques avec comité de lecture

C. TOUZET, "Communication in Cooperative Mobile Robot Learning," Soumis à la revue Autonomous Robots, 1998.

C. TOUZET, "Robot Autonomy: *Tabula Rasa* Learning vs. Bias Incorporation," Soumis à la revue IEEE Trans. on Systems, Man and Cybernetics, 1998.

[4] P. DROUOT, C. TOUZET & F. BRIOLLE, "Approche connexionniste pour le détection de pannes des antennes SONAR passif," Soumis à la revue Traitement du Signal, 1997.

[5] J.-M. SANTOS & C. TOUZET, "Automatic Tuning of the Reinforcement Function," to appear in Neurocomputing, 1998.

[6] C. TOUZET, "Neural Reinforcement Learning for Behaviour Synthesis, " Robotics and Autonomous Systems, Special issue on Learning Robot: the New Wave, N. Sharkey Guest Editor, vol. 22, Nb 3-4, December 1997, pp 251-281. *

[7] A. BRONS, G. RABATEL, F. ROS, F. SEVILA & C. TOUZET, "Plant grading by vision using neural networks and statistics", Computers and Electronics in Agriculture, N° 9, 1993.

3.5 Conférences avec actes et comité de lecture

* Disponible sur Internet: <http://avalon.epm.ornl.gov/~touzetc/publications.html>

- [C2] C. TOUZET, "Learning Approaches in Cooperative Robotics: A Few Facts," CONALD Workshop on Robot Exploration and Learning Pittsburgh, PA, June 11-13, 1998.
- [C3] C. TOUZET & J. M. SANTOS, "Reinforcement Function Design and Bias for Efficient Learning in Mobile Robots," WCCI-FUZZ-IEEE'98, Anchorage, Alaska, May 4-9, 1998. *
- [C4] J. M. SANTOS & C. TOUZET, "Automatic Tuning of the Reinforcement Function," NEURAP'98, Marseilles, France, March 11-13, 1998. *
- [C5] M. LE GOC, C. TOUZET & C.-C. THIRON, "The SACHEM Experience on Artificial Neural Networks Application," NEURAP'98, Marseilles, France, March 11-13, 1998.
- [C6] C. TOUZET, "Neural reinforcement learning for behavior synthesis," CESA'96 (Computational Engineering in Systems Applications), IMACS Multiconference, Lille, France, 9-12 July 1996, pp 734-739. *
- [C7] P. DROUOT, C. TOUZET & M. LE GOC, "Artificial Neural Networks for Data Validation in Blast Furnaces Monitoring," NEURAP'96, Marseille, France, March 1996.
- [C8] S. SEHAD & C. TOUZET, "Neural Reinforcement Path Planning for the Miniature Robot Khepera," WCNN'95, Washington D.C., USA, 17-21 July 1995. *
- [C9] C. TOUZET, S. SEHAD & N. GIAMBIASI, "Improving Reinforcement Learning of Obstacle Avoidance Behavior with Forbidden Sequences of Actions," International Conference on Robotics and Manufacturing," Cancun, Mexico, 14-16 June 1995. *
- [C10] C. TOUZET, N. KIEFFER & M. LE GOC, "Artificial Neural Networks Forecasting and Monitoring Scaffold and Scaffolding Phenomena in Blast Furnaces," IEEE International Conference on SMC, Vancouver, Canada, 22-25 October 1995.
- [C11] C. TOUZET, F. BLAYO, C. LEHMANN & N. GIAMBIASI, "Neural Networks in Simulation," European Simulation Symposium, Istanbul, Turkey, October 1994.

* Disponible sur Internet: <http://avalon.epm.ornl.gov/~touzetc/publications.html>

[C12] C. TOUZET, "Extending Immediate Reinforcement Learning on Neural Networks to Multiple Actions," ESANN 94, Bruxelles, April 1994. *

[C13] S. SEHAD & C. TOUZET, "Reinforcement Learning and Neural Reinforcement Learning," ESANN 94, Bruxelles, April 1994. *

[C14] C. TOUZET, "Apprentissage par renforcement neuronal : avantages et limitations", Second European Congress on Systems Sciences, Prague, October 1993.

[C15] A. BRONS, G. RABATEL, F. ROS, F. SEVILA & C. TOUZET, "Multi-layer neural networks and statistical methods for quality control", Neuro-Nîmes 93, Nîmes.

[C16] C. TOUZET & N. GIAMBIASI, "The Connectionist Sequential Machine: a General Model of Sequential Networks", Australian Conf. on Neural Networks, in Canberra, P. Leong & M. Jabri Eds. Sydney University Electrical Engineering, NSW 2006, Australia, February 1992.

[C17] A. BRONS, G. RABATEL, F. SEVILLA & C. TOUZET, "Evaluation de la qualité des plantes en pots par un réseau multicouche, assisté par des méthodes statistiques," Neuro-Nîmes 91, Nîmes, France, novembre 1991.

[C18] O. SARZEAUD, Y. STEPHAN & C. TOUZET, "Finite Element Meshing using Kohonen's Self-Organizing Maps", ICANN91, Helsinki, Finland, juin 1991.

[C19] O. SARZEAUD, Y. STEPHAN & C. TOUZET, "Application des cartes auto-organisatrices à la génération de maillage aux éléments finis", Neuro-Nîmes 90, Nîmes, France, novembre 1990.

[C20] C. TOUZET & N. GIAMBIASI, "Connectionist finite-state machines", IJCNN 1990, Washington DC, 15-19 janvier 1990.

[C21] N. GIAMBIASI, R. LBATH & C. TOUZET, "Une approche connexionniste pour calculer l'implication floue dans les systèmes à base de règles", Neuro-Nîmes 89, Nîmes, novembre 1989.

[C22] C. TOUZET & N. GIAMBIASI, "Neuromimetic Sequential Machines" IASTED expert systems, Zurich, Suisse, June 1989.

[C23] C. TOUZET & N. GIAMBIASI, "Reconnaissance de séquences par des réseaux de neurones", Neuro-Nîmes 88, Nîmes, France, novembre 1988.

3.6 Colloques nationaux et internationaux avec actes à diffusion restreinte

[30] C. TOUZET & S. SEHAD, "Apprentissage par renforcement neuronal pour l'acquisition d'un comportement d'évitement d'obstacle," NSI'96, Marly-le-Roi, mai 1996.

[31] C. TOUZET, "KHEPERA : un outil pour l'étude du hasard et des contraintes dans la genèse des formes collectives," Journées de Rochebrune, AFCET Cognition, Janvier 1995.

[32] S. SEHAD & C. TOUZET, "Apprentissage par renforcement pour l'acquisition de comportements en robotique," NSI'94, Chamonix, mai 1994.

[33] P. COUTURIER, A. JOHANNET, I. SARDA & C. TOUZET, "Robot hexapode : réalisation et apprentissage de la marche," NSI'94, Chamonix, mai 1994.

[34] C. TOUZET, "Apprentissage par renforcement neuronal d'un comportement d'évitement d'obstacles pour le mini-robot Khepera", Journées de Rochebrune, AFCET Cognition, Janvier 1994.

[35] C. TOUZET & F. MONDADA, "Quelques comportements adaptatifs pour le robot miniature Khepera", Annales du Groupe CARNAC, n° 6, EPFL (Suisse), Décembre 1993.

[36] C. TOUZET & N. GIAMBIASI, "L'émergence de comportements intelligents dans le domaine du connexionnisme : réflexion sur la taille", Émergence dans les modèles de la cognition, ENST, Paris, 15-16 avril 1992.

[37] C. TOUZET & O. SARZEAUD, "Application d'un algorithme d'apprentissage par pénalité/récompense à la génération de formes locomotrices hexapodes", Journées de Rochebrune, AFCET Intelligence Artificielle, 20-24 janvier 1992.

[38] C. TOUZET & N. GIAMBIASI, "Quelques approches neuromimétiques pour le traitement des séquences", NSI 90, Centre Paul Langevin, Aussois, 7-10 mai 1990.

3.7 Rapports de contrats

- [R1] Validation de la validation des données HF par réseaux de neurones artificiels, contrat avec la SOLLAC (Fos), 1996-1997 (en collaboration avec N. Giambiasi et M. Le Goc).
- [R2] Étude et développement d'une modalité de communication sonore pour le robot Khepera, contrat avec l'ANVAR, 1997 (en collaboration avec P. Drouot).
- [R3] Étude d'une approche par réseaux de neurones pour le développement d'antennes sous-marines intelligentes, contrat avec la DGA, 1996 (en collaboration avec P. Drouot et N. Giambiasi).
- [R4] Prédiction par réseaux de neurones du Marché Toutes Marques, contrat avec la régie RENAULT 1994-1995 (en collaboration avec F. Blayo, P. Drouot et B. Girard).
- [R5] Apport d'une approche connexionniste à la conduite de HF, contrat avec la Région PACA (Fos), 1994-1995 (en collaboration avec N. Kieffer).
- [R6] Approche connexionniste pour la conduite d'un four de cuisson de biscuits, contrat MRE avec le CEMAGREF, 1992-1995 (en collaboration avec E. Rannou).
- [R7] Application des réseaux de neurones artificiels à l'analyse de la cohérence des données au centre d'un HF, contrat avec la SOLLAC (Fos), 1994 (en collaboration avec P. Drouot).
- [R8] Aide à la conduite d'un haut-fourneau par réseaux de neurones artificiels, contrat avec la SOLLAC (Fos), 1993 (en collaboration avec N. Kieffer).
- [R9] Connexionnisme et mécanisme de l'inférence, contrat avec l'ANRT : projet PROCOPE avec le FKİ (Forschungsgruppe Kunstliche Intelligenz) de l'Institut für Informatik (Tech. Univ. Munich, Allemagne) en 1990-1992.
- [R10] Développement d'un simulateur de structures arborescentes de réseaux de neurones (sur Sun), contrat ANVAR-NEUROSYSTEMES , 1991-1992 (en collaboration avec N. Giambiasi et O. Sarzeaud).
- [R11] Mirage de l'albumine par réseaux de neurones,

contrat FEDER-CRTS, 1990-1991(en collaboration avec N. Giambiasi et P. Montesinos).

[R12] Étude de faisabilité de la compression d'images par cartes auto-organisatrices,
contrat FEDER-GENESYS, 1990-1991 (en collaboration avec N. Giambiasi et S. Dellaporta).

[R13] Un réseau multicouche pour la reconnaissance de caractères manuscrits,
contrat avec la société BEC Frères, 1990 (en collaboration avec N. Giambiasi).

[R14] SACREN : Système d'Aide au Choix d'un Réseaux de Neurones,
contrat ANVAR, 1988-1989 (en collaboration avec N. Giambiasi).

3.8 Tutoriaux

[T1] C. TOUZET, "Learning Robot," Dept. of Computer Science, University of North Dakota, Grand Forks, ND, USA, October 1997 (1h 30).

[T2] C. TOUZET, "Neural reinforcement learning for behaviour synthesis," Robotique 2000, CogniSud, Marseille, June 1997. (30')

[T3] C. TOUZET, "Application des réseaux de neurones artificiels à la conduite de HF", séminaire MacSim, Toulon, Mars 1997. (30')

[T4] C. TOUZET, "Carte auto-organisatrice et apprentissage par renforcement", Université des Antilles-Guyane, Pointe-à-Pitre, Guadeloupe, Février 1997. (4h)

[T5] C. TOUZET, "Learning agents," Autonomous Agents'97, Marina del Rey, CA, USA, in collaboration with S. Sen, February 1997 (1 full day).

[T6] C. TOUZET, "Neural reinforcement learning for obstacle avoidance," Neuroscience and Mobile Robotics, CNRS Workshop, Marseille, March 1996. (30')

[T7] C. TOUZET, "Neural reinforcement learning for behavior synthesis," IEE seminar on Self-Learning Robots, London, England, February 1996 (45'). *

[T8] C. TOUZET, "Neural reinforcement learning for robot control," CESAR, Oak Ridge National Laboratory, TN, USA, November 1995 (1h)

[T9] C. TOUZET, "Réseaux de neurones séquentiels et à renforcement," Neuro-Nîmes 93, Nîmes, Novembre 1993 (3h).

[T10]C. TOUZET, "Neuromimétique," Séminaire Intelligence Artificielle, Haut Commissariat à la Recherche, CDTA, Alger, Algérie, mai 1991, (1h30).

[T11]C. TOUZET, "Introduction au connexionnisme," Journées IA2 (Pôle Informatique avancée et Intelligence Artificielle du Languedoc-Roussillon), Novembre 1990 et Novembre 1991 (1h30).

* Disponible sur Internet: <http://avalon.epm.ornl.gov/~touzetc/publications.html>

4. ENCADREMENT DE TRAVAUX DE RECHERCHE

- 4.1 Co-Encadrement de thèses**
- 4.2 Encadrement de DEA**
- 4.3 Participation à des contrats d'études**
- 4.4 Participation à des comités de programme et d'organisation**

4.1 Co-Encadrement de thèses

Participation à l'encadrement de la thèse de M. O. Sarzeaud (50%),

Thèse de Doctorat : "Contribution à l'étude et au développement de systèmes connexionnistes modulaires"

Soutenue en décembre 1993 à l'USTL (Montpellier).

Participation à l'encadrement de la thèse de Mlle S. Sehad (85%),

Thèse de Doctorat : "Contribution à l'étude et au développement de modèles connexionnistes à apprentissage par renforcement : Application à l'acquisition de comportements adaptatifs"

Soutenue en décembre 1995 à l'USTL (Montpellier).

Participation à l'encadrement de la thèse de M. P. Drouot (75%),

Thèse de doctorat : "Reconfiguration de SONAR passif comportant des hydrophones en panne : approche connexionniste"

Soutenue en juillet 1997 à l'Université d'Aix-Marseille III.

Participation à l'encadrement de la thèse de M. J. Santos (co-tutorat avec l'Université de Buenos-Aires, Argentine) (90%)

Thèse de doctorat : "Contribution to the study and development of the design of reinforcement functions"

Soutenance prévue en décembre 1998 à l'Université d'Aix-Marseille III.

NB : Co-encadrement des travaux de thèse avec le Professeur Norbert Giambiasi

Membre de Jury de thèse de :

- M. A. Brons, Université de Montpellier II, décembre 1992.
- M. M. Crucianu, LIMSI, Orsay, juin 1994.
- Mme C. Esczut, I3S, Sophia-Antipolis, mai 1995.
- M. O. Michel, I3S, Sophia-Antipolis, novembre 1996.
- Mme C. Toffano-Nioche, LIMSI, Orsay, décembre 1996.
- M. P. Zhang, UTC, Compiègne, février 1997.

4.2 Encadrement de stages de DEA

Encadrement du stage de M. S. Sorrentino, 1989-1990

Modèles connexionnistes séquentiels basés sur l'algorithme de la rétropropagation de gradient,
DEA Conception Assistée des Systèmes Informatiques, Automatiques et Micro-électroniques, U.S.T.L.,
Montpellier.

Encadrement des stages de M. Y. Stéphan et O. Sarzeaud, 1989-1990

Application des cartes auto-organisatrices à la génération de maillage aux éléments finis,
DEA Génie Électrique, INPG, Grenoble.

Encadrement du stage de Mlle. S. Dellaporta, 1990-1991

Compression d'images par carte auto-organisatrice,
DEA Conception Assistée des Systèmes Informatiques, Automatiques et Micro-électroniques, U.S.T.L.,
Montpellier.

Encadrement du stage de M. S. Benyacoub, 1991-1992

Coopération de réseaux de neurones à travers une décomposition structurelle,
DEA Intelligence Artificielle et Algorithmique, Université de Caen.

Encadrement du stage de M. E. Kotter, 1991-1992

Étude de l'algorithme ARP et application à l'apprentissage de la marche hexapode,
DESS double compétence Informatique et Applications aux sciences de la vie, EHEI, Paris.

Encadrement du stage de M. M. Pollizzi, 1991-1992

Réseaux de neurones séquentiels : comparaison de l'apprentissage sur la Machine Séquentielle Connexionniste et
de l'algorithme Real-Time Recurrent-Learning,
DEA Conception Assistée des Systèmes Informatiques, Automatiques et Micro-électroniques, U.S.T.L.,
Montpellier.

Encadrement du stage de Mlle S. Sehad, 1991-1992

Réseaux neuronaux "ART", application à la coalescence de données génétiques,
DEA Conception Assistée des Systèmes Informatiques, Automatiques et Micro-électroniques, U.S.T.L.,
Montpellier.

Encadrement du stage de M. E. Rannou, 1993-1994

Application des méthodes de minimisation à l'apprentissage connexionniste pour la classification en agro-alimentaire,

DEA Conception Assistée des Systèmes Informatiques, Automatiques et Micro-électroniques, U.S.T.L., Montpellier.

Encadrement du stage de M. P. Drouot, 1993-1994

Application de l'apprentissage par renforcement sur réseaux de neurones artificiels à la validation de données,

DEA Conception Assistée des Systèmes Informatiques, Automatiques et Micro-électroniques, U.S.T.L., Montpellier.

Encadrement du stage de M. R. Bernhard, 1994-1995

Application de l'apprentissage de type Hopfield au test des circuits,

DEA Productique et Informatique, Aix-Marseille III.

Encadrement du stage de M. C. Rubini, 1995-1996

Les cartes auto-organisatrices de très grande dimension : application à la supervision de procédés,

DEA Productique et Informatique, Aix-Marseille III.

Encadrement du stage de Mlle S. Meyer, 1995-1996

Analyse de performances de réseaux multicouches sur des bases d'apprentissage de plusieurs milliers d'exemples,

DEA Productique et Informatique, Aix-Marseille III.

4.3 Participation à des contrats d'études

- Contrat avec la SOLLAC (Fos), 1996-1997 : validation de la validation des données capteurs d'un haut-fourneau par réseaux de neurones artificiels. (200 KF) (supervision 100% et développement 100%)
- Contrat avec l'ANVAR, 1997 : étude et développement d'une modalité de communication sonore pour le robot Khepera. (25 KF) (supervision 100%)
- Contrat avec la DGA (le Brusc, 83), 1995-1996 : étude d'une approche par réseaux de neurones pour le développement d'antennes sous-marines intelligentes. (400 KF) (supervision 75%)

- Contrat avec la Région PACA (et Sollac), 1994-1995 : apport des réseaux de neurones à la perception dans SACHEM. (200 KF) (supervision 90% et développement 25%)
- Contrat avec la régie RENAULT pour la prédiction par réseaux de neurones du Marché Toutes Marques, 1994-1995. (100 KF dont 66 KF pour notre labo) (supervision 30% et développement 30%)
- Contrat avec la SOLLAC (Fos), 1994-1995-1996 : validation des données capteurs d'un haut-fourneau par réseaux de neurones artificiels. (200 KF) (supervision 75% et développement 25%)
- Projet Capital Humain et Mobilité en Réseaux de neurones artificiels, 1994. Notre laboratoire a reçu la visite pendant 1 an d'un chercheur belge (A. Terao) en 1993 et durant 3 mois celle d'un chercheur suisse (C. Lehmann).
- Contrat avec l'ANRT 1990-1992 : projet PROCOPE avec le FKI (Forschungsgruppe Kunstliche Intelligenz) de l'Institut für Informatik (Tech. Univ. Munich, Allemagne) sur le sujet : Connexionnisme et mécanisme de l'inférence. (70 KF) (supervision 100% et développement 100%)
- Contrat avec la SOLLAC (Fos), 1993 : aide à la conduite d'un haut-fourneau par réseaux de neurones artificiels. (50 KF) (supervision 100%)
- Contrat ANVAR-NEUROSYSTEMES , 1991-1992 : développement d'un simulateur de structures arborescentes de réseaux de neurones (sur Sun). (200 KF) (supervision 10% et développement 25%)
- Contrat FEDER-CRTS, 1990-1991 : mirage de l'albumine par réseaux de neurones. (100 KF) (supervision 50%)
- Contrat FEDER-GENESYS, 1990-1991 : étude de faisabilité de la compression d'images par cartes auto-organisatrices. (100 KF) (supervision 75% et développement 25%)
- Contrat avec BEC Frères, 1990 : développement en collaboration avec la société ITECA d'un réseau multicouche pour la reconnaissance de caractères manuscrits. (30 KF) (supervision 100% et développement 25%)
- Contrat ANVAR, 1988-1989 : développement d'un simulateur de réseau de neurones : SACREN. (500 KF) (supervision 50% et développement 100%)

4.4 Participation à des comités programme et d'organisation

Éditeurs invités (avec I. Cloete) d'un numéro spécial de la revue *Neurocomputing* (Elsevier) à paraître 1998.

Éditeurs invités (avec N. Giambiasi) d'un numéro spécial de la revue *Journal of Intelligent and Robotic Systems* (Kluwer) sur le thème : “Applications of Artificial Neural Networks,” February 1998.

Participation au comité programme et au comité d'organisation du congrès international NEURAP (Marseille, 1994-1998).

Participation au comité programme du congrès international Neuro-Nîmes (les réseaux de neurones artificiels et leurs applications, Nîmes) 1990 - 1993.

Participation au comité programme du congrès ESANN (European Symposium on Artificial Neural Networks, Bruxelles, 1993-1998).

Participation au comité programme du congrès PERAC (From Perception to Action, Lausanne, septembre 1994).

Participation au comité programme du congrès CES (Systems Science European Congress, Prague, 1993 et Rome, 1996).

Participation au comité d'organisation de l'école de printemps NSI (Neurosciences et Sciences de l'Ingénieur, St Jean du Gard, mai 1993).

Participation à la création de l'Association des Connexionnistes en THèse, organisation de Valgo90 (ACTH) à Port-Camargue (avril 1990).

5. ACTIVITÉS D'ENSEIGNEMENTS

Dès ma première année de thèse, j'ai été chargé d'encadrer des TP (classe de Math Sup et 2ème année d'ingénieur). Puis j'ai été chargé d'enseignement dès la création de l'école EERIE en septembre 1988. En complément de mon activité de recherche, j'ai assuré un demi-service d'enseignement. A l'issue de ma thèse en 1990, j'ai obtenu un poste d'ingénieur chargé de cours à l'EERIE. J'ai occupé ce poste jusqu'en août 1994 (en 1993, j'ai été durant deux mois professeur invité du cours postgrade sur les réseaux de neurones artificiels à l'EPFL, Suisse). En septembre 1994, j'ai rejoint l'IUSPIM en tant qu'ATER. J'ai été ATER pendant deux années, puis ingénieur de recherche chargé de cours pendant l'année scolaire 1996-1997.

5.1 Tableau récapitulatif

Matière	Type	Public	Année	Total horaire (eq. TD)
Informatique	TP	Math Sup	87-88	20
Système Temps Réel	TP	IUSPIM, 2ème année	87-88	42
Logique	TD, TP	EERIE 1ère année	88-89-90	60
Réseaux de neurones artificiels	Cours, TD, TP	EMA (3ème année)	89-90	21
	Cours, TD, TP	EERIE 3ème année	90-91-92-93- 94-95	60
	Cours, TD, TP	IUSPIM 2ème année	94-95-96-97	36 (50 en 96-97)
	Cours	DEA CASIAM (USTL)	91-92-93-94- 95	24
	Cours	DEA Génie Productique	95-96-97	18
	Cours, TP	IMERIR 3ème année	92-93-94-95- 96-97	20
Base de données	Cours	DU Informatique	94-95	30
Simulation	Cours, TD, TP	IUSPIM 2ème année	95-96	36

5.2 Stages

Pour compléter mon demi-service d'enseignement, j'ai encadrer des projets de fin d'études en 1991, 1992, 1993 1994, 1995 (60 h. eq. TD/année) et 1996 (40 h. eq. TD). Il est à noter que le logiciel d'animation 3-D développé durant le stage 1993 a été retenu par le Ministère pour être présenté dans ses jardins lors des journées de la Science 1993. D'autre part, le logiciel réalisé durant le stage de 1994 a gagné le concours national de la société Apple "Programmer un comportement intelligent" doté de 200 000 F pour le vainqueur.

J'ai aussi assuré des actions de formation permanente à destination des industriels dans le domaine des réseaux de neurones artificiels (chaque année depuis 1989).

5.3 Matières

Système Temps Réel

En 1987, dans le cadre de la troisième année d'ingénieur de l'IUSPIM (Faculté Saint Jérôme, Marseille), j'ai assuré un encadrement de travaux pratiques en systèmes temps réel. Cet encadrement représentait un volume de 42 heures. Il s'agissait de guider les étudiants dans la manipulation des sémaphores, tâches, coordinations et interruptions au sein de l'environnement de développement temps réel iRMX 86 (INTEL).

Logique

En 1988 et 1989, dans le cadre de la première année d'ingénieur à l'EERIE, j'ai assuré la création et l'encadrement de travaux dirigés et travaux pratiques en logique. Cet encadrement représentait un volume horaire de 60 heures. Il s'agissait d'aborder les techniques de synthèse des circuits digitaux (combinatoires et séquentiels). Les travaux pratiques, avec l'aide des pupitres de simulation, permettent de réaliser des circuits logiques arithmétiques (additionneurs par exemple) et des circuits logiques séquentiels (bascules, registres, compteurs). Enfin, à l'aide de pupitres de simulation dédiés, l'architecture des machines était abordée avec notamment la réalisation incrémentale d'une unité centrale (unité arithmétique et logique, adressage, mémoire, programme).

Base de données

A destination d'étudiants en formation continue pour la plupart, j'ai particulièrement axé ce cours sur les SGBD relationnels afin d'en faire un outil utile dans leurs activités.

Simulation

Dans le cadre de l'aide à la décision, j'ai bâti cet ensemble cours, TD, TP autour du logiciel SIMULINK (Matlab). Le modèle du système à simuler est un couplage mécanique entre un moteur et une charge. Le problème que l'on cherche à résoudre est celui de la détermination des plages optimales d'utilisation, comme des plages à éviter (c.a.d., caractéristiques du moteur pour une charge donnée et inversement, pour un moteur donné, les charges utilisables et les contraintes).

Réseaux de neurones

En 1989, j'ai eu la tâche de concevoir un cours de 60 heures "réseaux de neurones artificiels : introduction et spécialisation", pour la troisième année d'ingénieur. Ce cours a été proposé aux élèves de l'option Intelligence Artificielle dès 1990. Il se découpe en 30 heures de cours théoriques et 30 heures de travaux pratiques.

Le support de cours de la partie théorique s'est affiné au cours des deux années d'enseignement qui ont suivi. Il a donné lieu à la publication d'un livre introductif au domaine du connexionnisme de 150 pages édité par EC2 (sortie le 2 novembre 1992). Il fournit à l'étudiant, ou à l'ingénieur, une connaissance théorique et pratique des modèles de réseaux de neurones artificiels. L'objectif de cet ouvrage n'est pas de décrire in extenso tous les modèles de réseaux de neurones artificiels existant, mais plutôt d'étudier les plus représentatifs, avec le souci pédagogique de conserver le même formalisme. Les modèles détaillés incluent : Perceptron, Mémoires associatives, Cartes auto-organisatrices, Adaptive Resonance Theory, Apprentissage par pénalité-récompense et Réseaux multicouches. Nous montrons, au travers d'exemples d'applications, l'intérêt de ces modèles aux composants fortement connectés pour l'ingénieur dans des domaines tels que : la vision, le traitement du signal ou l'aide à la décision. La capacité d'apprentissage par l'exemple des réseaux neuronaux autorise le développement d'applications sans nécessiter une connaissance très exhaustive du domaine par l'ingénieur. Par contre, une grande aisance dans la manipulation des modèles et de leurs paramètres, dans la constitution des bases d'apprentissage et d'autres facteurs moins identifiés, est indispensable. L'acquisition de cette connaissance nécessite actuellement une expérience personnelle de la mise en oeuvre des techniques neuronales. Travaux dirigés et pratiques émaillent le texte afin de répondre à cette attente. Afin de conserver un langage clair et concis, tout le formalisme mathématique a été reporté en annexe. D'autre part, questions récapitulatives, références, informations pratiques et glossaire permettent au lecteur de poursuivre son instruction.

Les travaux pratiques ont nécessité la mise au point de simulateurs de réseaux neuronaux. Certains d'entre eux sont dérivés du projet SACREN, que j'ai conduit entre 1987 et 1989 (cf. contrats). D'autres sont des simulateurs disponibles dans le commerce, mais qui ont nécessité un important investissement personnel : une semaine de stage de formation au produit ExploreNet chez Hecht-Nielsen Co. à San Diego (USA), une semaine de stage au produit Nestor development System (NDS 1000) à Montpellier chez Europixels. Enfin, les résultats de nos travaux contractuels (cf. contrats) sont souvent repris pour finaliser le thème de l'exercice (compression d'images et reconnaissance de caractères manuscrits, par exemple). Initialement écrit en langage Pascal sur station de travail Apollo, les simulateurs sont aujourd'hui disponibles en C sur station de travail Sun. L'un des principaux avantages liés à l'utilisation de simulateurs développés en interne réside dans la possibilité de fournir à l'étudiant le programme source. A partir de celui-ci, il lui est beaucoup plus facile de comprendre et d'expérimenter individuellement en temps réel.

6. ANNEXES

A - Exemples et résumés de quelques travaux contractuels

- Contrat ANVAR, 1988-1989 Développement d'un simulateur de réseau de neurones (SACREN).	45
- Contrat avec BEC Frères, 1990 (collaboration avec la société ITECA) Développement en d'un réseau multicouche pour la reconnaissance de caractères manuscrits.	52
- Contrat avec l'ANRT 1990-1992 (Projet PROCOPE avec Tech. Univ. Munich, Allemagne) Connexionnisme et mécanisme de l'inférence.	56
- Collaboration avec le LEG-ENSIEG, 1990 Application de cartes auto-organisatrices à la génération de maillages aux éléments finis.	57
- Contrat FEDER-GENESYS, 1990-1991 Compression d'images visiophoniques par RNA.	60
- Collaboration avec le CIRAD, 1991 Etude et développement d'une méthode d'analyse par coalescence pour la détermination des espèces génétiques basée sur les RNA.	67
- Contrat NEUROSYSTEMES, 1991-1992 Maquette de démonstration de l'apprentissage par RNA, application à la locomotion hexapode.	72
- Contrats avec la SOLLAC (Fos), 1993-1997 Apport des réseaux de neurones à la perception dans SACHEM.	75
- Contrat avec l'ANVAR, 1997 Etude et développement d'une modalité de communication sonore pour le robot Khepera.	76

SACREN (Système d'Aide au Choix d'un Réseau de Neurones)

(Contrat ANVAR n° A8801006JAL)

SACREN : Système d'Aide au Choix d'un Réseau de Neurones est né du recours obligatoire à l'expérimentation dans le domaine des réseaux de neurones artificiels pour le choix du modèle connexionniste le plus adapté à une application particulière. Sous la dénomination générale de système connexionniste, modèle neuromimétique ou plus communément réseau de neurones artificiels se regroupe un grand nombre de modèles de réseaux différents. Les spécifications applicatives de chaque architecture de réseau ne reposent pas sur des résultats théoriques (sauf pour les plus simples : Perceptron). Le recours à l'expérimentation s'avère être le moyen le plus approprié pour choisir le type de réseau à adopter pour une application déterminée. SACREN fournit un environnement permettant à l'utilisateur/développeur de tester facilement les diverses possibilités neuromimétiques pour une application donnée. SACREN offre ainsi la possibilité de sélectionner une structure de réseau de neurones plus adaptée au problème industriel. Dans ce but, SACREN permet de simuler tous les types de réseaux neuronaux, certains d'entre eux ont fait l'objet d'une implantation informatique : les réseaux multi-couches et la rétro-propagation de gradient, les mémoires associatives linéaires et les cartes auto-organisatrices de Kohonen.

Le principal critère guidant le développement de SACREN a été la flexibilité. SACREN offre les possibilités suivantes (que l'on ne retrouve pas totalement au niveau des produits commercialisés) :

Facilités pour la conceptualisation :

- Une description informatique pour chaque neurone et chaque synapse (structure de données).
- Simulation événementielle (comportement dynamique).

Facilités pour la modification :

- Toutes topologies de réseaux autorisées.
- Chacun des neurones et des synapses pouvant être doté d'un comportement particulier.

SACREN a été développé sur station de travail graphique APOLLO en Pascal sous UNIX. Il a aussi été transporté sur SUN. Depuis sa création en 1989, un certain nombre de modèles ont été transcrits en C : les réseaux multicouches et la rétropropagation de gradient, les mémoires associatives linéaires, les cartes auto-organisatrices et la machine séquentielle connexionniste.

L'utilisation de la simulation événementielle présente des avantages par rapport aux techniques classiquement utilisées (calcul matriciel). Les neurones et les synapses sont effectivement représentés comme des entités au sein du modèle simulé, ce qui facilite la conceptualisation par l'opérateur. Le nombre de calculs est réduit et

l'espace mémoire nécessaire est minimal. Enfin, il est possible de manipuler des retards au niveau des connexions, ce qui permet d'envisager la simulation de réseaux dotés d'un comportement asynchrone.

SACREN permet la construction et la simulation d'une grande variété de réseaux. Le principal critère guidant son développement a été la flexibilité (chaque cellule possède son comportement propre, tout type de schéma de connexions peut être spécifié), ainsi que le montre la figure 1.

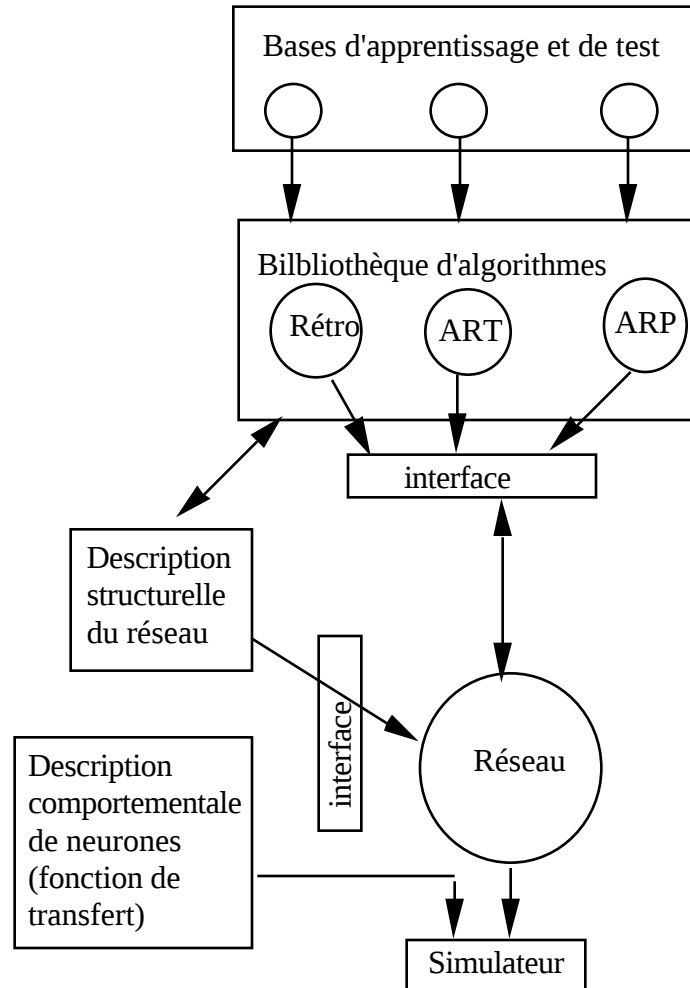


Figure 1. Structure générale du simulateur

Le module désigné sous le terme de description structurelle de neurones regroupe les utilitaires permettant de construire l'architecture du réseau de neurones : les données (schéma des connexions) utilisées pour la simulation. Le second module de description comportementale de neurones a pour objet la spécification du comportement de chacune des cellules. La bibliothèque des algorithmes regroupe différentes procédures

d'apprentissage pour le réseau de neurones : la rétropropagation de gradient, etc. La procédure d'apprentissage travaille à partir d'une base d'exemples d'apprentissage. Le module simulateur est composé d'un simulateur événementiel à événements discrets qui génère, classe et traite les événements consécutifs à l'application sur le réseau d'une forme en entrée. Les interfaces sont des procédures proposant à l'utilisateur, sous forme de menu, toutes les actions impliquées dans le déroulement d'une session. Chacun de ces modules est repris plus en détail dans les paragraphes suivants.

Description structurelle de neurones

Il s'agit d'un module évolutif destiné à décrire l'architecture de chacun des modèles implantés. On entend par modèle implanté, tout modèle dont l'algorithme d'apprentissage a été décrit et ajouté à la bibliothèque des algorithmes. Ce module permet à la date actuelle la construction de réseaux à une ou plusieurs couches, sans connexion intracouche et à connexion complète intercouches, avec des options pour décrire des réseaux bouclés (boucles depuis la couche de sortie ou la couche cachée), des cartes topologiques de Kohonen et des mémoires auto ou hétéro-associative. De plus, le stockage de l'architecture du réseau dans des fichiers aux normes ASCII permet à l'utilisateur une modification aisée des caractéristiques, l'ajout ou l'annulation de certaines synapses, la spécification de valeurs particulières à certains poids, ... En fait tous les composants relatifs à la description structurelle d'un réseau sont accessibles et modifiables par l'utilisateur.

Description comportementale de neurones

Il existe plusieurs fonctions différentes pour le calcul de la valeur de l'état du neurone:

- fonction signe,
- fonction sigmoïde avec prise en compte du passé : l'état précédent du neurone intervient dans le calcul,
- fonction sigmoïde classique.

Il est possible de rajouter de nouvelles fonctions. Chaque neurone peut avoir une fonction de transfert différente de celle des autres neurones.

Nous proposons aussi plusieurs variantes au niveau des informations transmises depuis le neurone vers les autres neurones. Ceci décrit le comportement du réseau. Toutes les cellules du réseau transmettent la même information :

- propagation de l'état du neurone,
- propagation de la variation d'état (état précédent - état présent).

Bibliothèque des algorithmes

Parmi les algorithmes d'apprentissage implanté, nous recensons les cartes auto-organisatrices, les mémoires associatives et plusieurs versions de la rétropropagation de gradient. Chacune relève de la même philosophie

générale, mais se distingue soit par le comportement des cellules (fonction de transfert), soit par l'architecture du réseau (réseaux bouclés).

Saisie des formes à apprendre

Le passage de la forme brute (pixels de l'image par exemple) à sa représentation sur la couche d'entrée du réseau dépend principalement des caractéristiques architecturales du réseau. L'apprentissage supervisé impose que l'on dispose, pour chaque forme d'entrée, de la forme de sortie désirée. Il est nécessaire d'avoir une procédure de codage qui permette le passage de la forme désirée au codage sur la couche de sortie du réseau. D'autre part, la base de données est fonction de l'application envisagée. Ainsi, une application de reconnaissance des formes ne requiert comme exemples d'apprentissage et de test que des images (ensemble de pixels), tandis qu'une application de diagnostic impose que cette même base de données soit composée d'un grand nombre de couples (symptômes, diagnostic associé). La structure de données informatique pour mémoriser les bases d'exemples de ces deux types d'application est incompatible, bien que le réseau de neurones reste identique à tous points de vue. Le type d'application envisagée engage une certaine spécificité des outils de saisie de la base d'exemples.

Interfaces

L'interface permet, en sélectionnant au sein d'un menu général d'une vingtaine d'options, de réaliser toutes les fonctions de base du système : apprentissage, simulation, affichage de la table des cellules, affichage de la table des synapses, quitter le système, sauvegarde de l'état du réseau, affichage des formes d'entrée/sortie, acquisition des formes d'entrée et de sortie, saisie interactive de forme d'entrée, construction d'une architecture de réseau, acquisition d'un réseau existant, changement de valeurs des paramètres de la rétropropagation.

Les fonctions liées à la vérification du bon fonctionnement du simulateur (debug) sont incluses dans chacune des procédures. Il existe aussi des procédures d'affichage des valeurs des tables des neurones et des synapses.

Les fonctions liées au test des performances des réseaux simulés (test) répondent à l'objectif de notre logiciel : "destiner aux laboratoires et à l'industrie, il doit principalement permettre de tester différents algorithmes d'apprentissage afin de déterminer expérimentalement celui qui est le mieux adapté à leur problème". Les outils de test sont regroupés dans une procédure qui calcule l'écart entre la réponse fournie par le réseau et celle désirée. Cet écart est en fait une mesure des performances du système. Il est possible de mesurer les performances du système à n'importe quel instant de la phase d'utilisation ou de la phase d'apprentissage.

Structure de données

La principale structure de données est un tableau des cellules (neurones) comprenant une liste des connexions (synapses) au départ de celle-ci. Un second tableau regroupe les synapses avec le délai qui leur est attaché ainsi que le numéro du neurone destination. Les neurones et les synapses contiennent de plus diverses

informations comme, pour les neurones, le potentiel, le type de la fonction de transfert, la somme pondérée des entrées et quelques valeurs relatives au passé de la cellule.

SACREN fait appel à la simulation dirigée par les événements. La simulation événementielle présente de multiples avantages par rapport au calcul matriciel (matrice des poids) classiquement utilisée. Rappelons brièvement ici, les trois principaux avantages liés à son emploi :

- Gain en facilité de conceptualisation : les neurones et synapses sont effectivement représentés comme des entités au sein du modèle simulé.

- Gain de temps de calcul : seules les cellules ayant changé d'états vont créer un événement, ainsi le minimum de calcul est effectué. Rappelons que le calcul matriciel oblige de recalculer à chaque interaction la valeur de tous les neurones du réseau. Ceci est particulièrement pénalisant dans le cadre des réseaux multicouches sans boucle où les couches sont activées les unes après les autres (et une seule à la fois).

- Gain de place mémoire : la structure de données décrivant le réseau est minimale, en effet seules les cellules et les connexions existantes sont décrites. La technique classique (matricielle) impose de manipuler des matrices de connexions très creuses. Ainsi dans le cas d'un réseau de neurones multicouches à connexions complète intercouches sans boucles comprenant n cellules, la taille de la matrice est n^2 alors qu'il existe environ $n^2/10$ connexions effectives. Pour des réseaux de grandes dimensions, le gain de taille mémoire est considérable.

Événements : la simulation du fonctionnement des réseaux de neurones nécessite trois types différents d'événements que nous avons baptisés : événement_synapse, événement_neurone, événement_zéro. Nous décrivons les actions réalisées par chacun de ces événements.

- L'événement_synapse est chargé de transmettre depuis un neurone source jusqu'à un neurone destination, la valeur de l'état du neurone source pondérée par le poids de la connexion. De plus, cet événement permet l'introduction explicite au niveau de la synapse du concept de temps. Il est possible d'attribuer un retard à la transmission entre deux neurones. Cet apport de la simulation événementielle n'est pas encore exploité au sein des modèles neuromimétiques développés actuellement. Chaque événement_synapse crée un événement_neurone.

- L'événement_neurone assure la mise à jour de la somme pondérée du neurone.

- L'événement_zéro assure le synchronisme de la mise à jour du traitement des neurones.

Échéancier : c'est la structure dans laquelle sont stockés les événements. Du fait de la manipulation explicite du temps dans notre modèle de réseau de neurones, à chaque événement est associé une date. L'échéancier doit assurer le classement des événements en fonction de leur date afin que soit traité en premier l'événement le plus proche dans le temps. La structure informatique de l'échéancier est présentée figure 2.

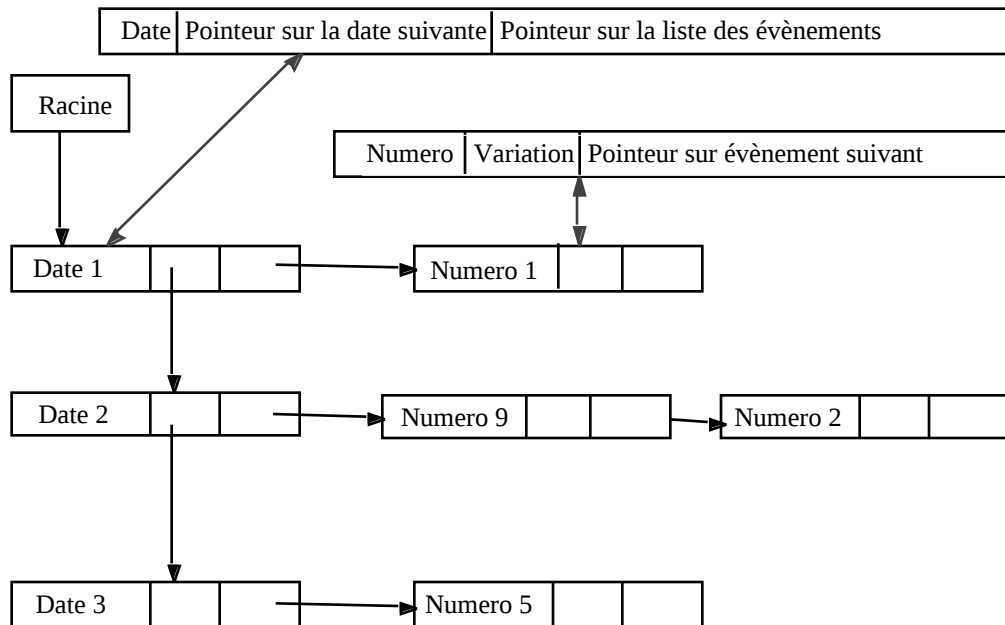


Figure 2. Échéancier utilisé dans SACREN : Structure informatique de l'échéancier. Le numéro de l'événement permet de connaître son type (événement_zéro, événement_neurone, événement_synapse).

L'utilisation du simulateur est schématisée figure 3. Le travail débute par le choix de la structure de réseau (modèle, nombre de neurones, comportement de la fonction de transfert) et l'algorithme d'apprentissage (nombre d'itération d'apprentissage, pas de modification des poids, constitution des bases d'exemples). Puis, la phase d'apprentissage sur la base d'exemples est exécutée. L'étape suivante est l'utilisation du réseau qui permet de valider ou d'invalidier les différents choix effectués dans les phases précédentes.

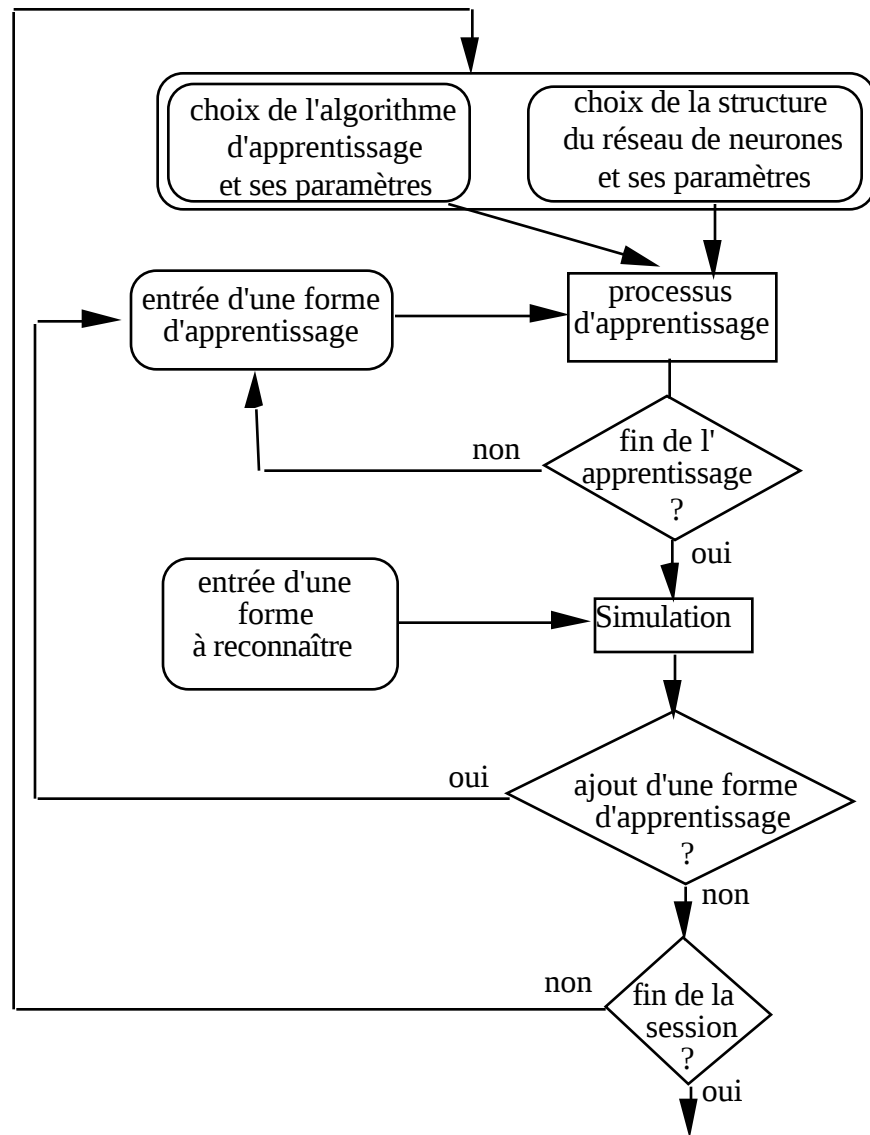


Figure 3. Organigramme d'une session

Bibliographie

C. Touzet, N. Giambiasi, "SACREN : Système d'Aide au Choix d'un Réseau de Neurones," Rapport de contrat ANVAR n° A8801006JAL, Nîmes, France, juillet 1989.

D'autre part, les travaux de recherche que nous avons menés ont utilisés ce logiciel et ont été décrits dans plusieurs publications notamment au congrès Neuro-Nîmes en 1988 et 1989.

Reconnaissance de caractères manuscrits par réseaux de neurones artificiels

(Contrat ITECA-BEC Frères; 1990-91)

Développement d'un module connexionniste pour la reconnaissance de caractères manuscrits. Le projet global est un système automatique de saisie de formulaires de chantier. L'apprentissage des écritures est réalisé sur site d'exploitation. La reconnaissance de forme est un domaine d'application privilégié pour les techniques connexionnistes du fait de leurs propriétés d'apprentissage à partir d'exemples, de généralisation, de résistance au bruit et de traitement en parallèle. Les réseaux multicouches avec la rétropropagation de gradient sont les plus utilisés aujourd'hui. De bonnes performances sont obtenues du fait notamment au nombre restreint (2 à 3) d'écrivains possibles. Un logiciel de saisie automatique de formulaire (FORMSCAN) a été réalisé par ITECA. Il est depuis 1991 installé sur site (société BEC Frères). Les performances mesurées sont excellentes (98 %).

Les différentes étapes de notre projet :

- 1/ Définition du cahier des charges : quelle doit être la vitesse de traitement, quel est le nombre d'écrivains, quelles sont les performances attendues (taux de rejet, taux de mal classés) ?
- 2/ Construction des bases d'apprentissage et de test. Comment répartir les exemples entre ces deux bases, quel nombre est suffisant, lesquels sont les plus représentatifs ?
- 3/ Sélection d'un modèle de réseau. Il s'agit de réaliser la mise en correspondance de l'image "pixel" avec le caractère.
- 4/ Définition des prétraitements : squelettisation, seuillage, détection de contours, normalisation, ...
Définition du traitement : nombre de réseaux de neurones, association avec traitement classique (vérification, contrôle de vraisemblance).
Codage des informations d'entrée et de sortie (type thermomètre, pixel, ASCII, ...).
- 5/ Définition : nombre de couches cachées, nombre de neurones par couche (entrée, cachée(s), sortie), constante d'apprentissage (μ), nombre d'itérations d'apprentissage.
Retour à l'étape 4 pour ajustement.
- 6/ Mise au point sur la base de test
Quelles sont les lettres qui posent problème, pourquoi ?
Sur quels critères peut-on autoriser l'apprentissage en cours d'utilisation, avec quels valeurs de paramètres ?

Vérification du respect des performances précisées dans le cahier des charges.

7/ Essais sur site.

8/ En parallèle, rédaction de la documentation, de la conclusion, etc.

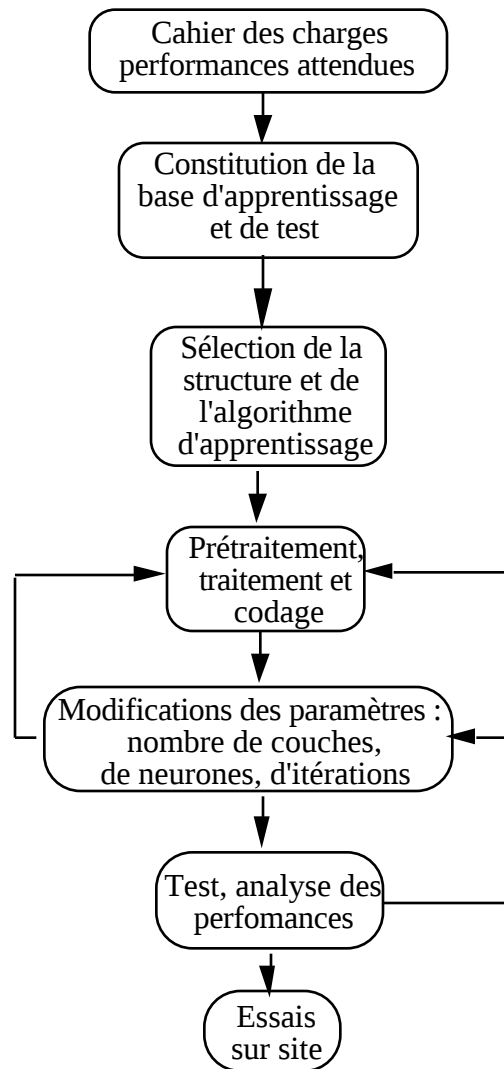


Figure 1. Les étapes du développement de l'application (connexionniste)

La figure suivante montre quelques exemples de chiffres manuscrits constituant la base d'apprentissage. Un codage de chaque image est réalisé en entrée du RNA. Il s'agit de découper l'image ".tif" en 15 cases (3 X 5), de compter le nombre de pixels noir dans chaque case. Chaque chiffre manuscrit est donc codé sous la forme d'un vecteur de dimension 15. On ajoute après chaque vecteur la classe de sortie (composante no. 16).

Étude des apports des réseaux de neurones séquentiels pour la réduction des arbres de recherche d'un démonstrateur de théorème

(Projet PROCOPE financé par l'ANRT)

Les recherches actuelles concernant l'utilisation de réseaux connexionnistes pour guider la recherche d'un démonstrateur de théorèmes montrent que l'emploi de ces techniques permet d'améliorer les performances sans accroître le temps de calcul. Cependant, la génération de ces heuristiques est réalisée hors-contexte : les informations qu'elles fournissent sont indépendantes de la partie de la démonstration déjà réalisée. De meilleures performances doivent être possibles par l'utilisation d'heuristiques dynamiques générées par des réseaux de neurones séquentiels. Ce sont des étapes préliminaires au développement de techniques connexionnistes de raisonnement. Les réseaux de neurones séquentiels participent à la réduction dynamique des arbres de recherches d'un démonstrateur de théorème. La machine séquentielle connexionniste apprend sur une base d'exemples constituée de séquences de démonstrations. En phase d'utilisation, le réseau de neurones, par références à l'expérience acquise et à l'arbre de recherche déjà parcouru, résout les embranchements non déterministes. Le modèle que nous utilisons est celui de la machine séquentielle connexionniste, issu du concept de machine séquentielle. C'est un modèle général de réseau séquentiel qui inclut tous les autres et est doté de possibilités supérieures.

Équipes impliquées

FKI (Forschungsgruppe Kunstliche Intelligenz) de l'Institut für Informatik de l'Université Technique de Munich (R.F.A.). 15 jours de mission au FKI (Munich, R.F.A.) en 1990 (C. Touzet), trois semaines de mission en 1991 (C. Touzet). Pour 1992, deux semaines de mission (C. Touzet, M. Polizzi). Visites réciproques au LERI pour une semaine de W. Ertel (FKI) en novembre 1990, Ilko Michler en novembre 1991 (2 semaines) et Bertram Fronhöfer en mai 1992.

Publications

C. TOUZET, N. GIAMBIASI, "Application of Connectionist Models to Fuzzy Inference Systems", in Parallelization in Inference Systems, Lectures Notes in Artificial Intelligence 590, B. Fronhöfer & G. Wrightson Eds., Springer Verlag, April 1992.

C. TOUZET, N. GIAMBIASI, "The Connectionist Sequential Machine: a General Model of Sequential Networks", ACNN'92, in Canberra, P. Leong & M. Jabri Eds. Sydney University Electrical Engineering, NSW 2006, Australia, February 1992.

Application des cartes auto-organisatrices à la génération de maillage aux éléments finis

(collaboration avec le LEG- ENSIEG)

L'idée essentielle est d'interpréter géométriquement la répartition des neurones des cartes auto-organisatrices dans l'espace synaptique. On utilise les cartes auto-organisatrices de Kohonen pour leur propriété d'arrangement spatial optimal. La répartition des neurones dans l'espace synaptique se superpose à la géométrie d'étude pour constituer le maillage. Il ne s'agit pas ici d'une application classique au sens mise en correspondance de deux espaces. Dans de nombreuses disciplines (électrotechnique, mécanique, thermique...), les techniques de résolution numérique font appel à un maillage de la géométrie d'étude. La précision des résultats et le temps de calcul dépendent fortement du maillage qui constitue une étape-clé de la résolution. Les cartes auto-organisatrices du fait de leurs propriétés d'auto-organisation et d'arrangement optimal satisfont aux critères usuels de maillage. Pour le maillage en électrotechnique, les critères géométriques sont les suivants : aucun " trou " ni recouvrement n'est toléré, les éléments doivent être le plus régulier possible (triangles équilatéraux, quadrilatères proches de rectangles pas trop plats, ...). Les critères physiques sont fonction du problème. En règle générale, le maillage doit être plus fin (éléments plus petits) dans les zones critiques où peuvent se produire des phénomènes tels que : effet de pointe, frontières entre régions, propriétés des matériaux (saturation), ...

Pour mailler une pièce avec une carte auto-organisatrice, la première approche qui vient à l'esprit montre que le réseau ne respecte pas la concavité. Certains neurones sont en dehors de la géométrie. Si on essaye alors de supprimer les neurones en dehors de la pièce et de reprendre l'apprentissage en ayant une forme de réseau plus adaptée à la pièce, les résultats ne sont pas satisfaisants. Il faut réaliser une découpe préalable de la géométrie à mailler en sous-parties convexes et fixer les neurones sur la périphéries (Figure 1).

Dans le but d'améliorer les performances des cartes auto-organisatrices, nous avons introduit deux nouveaux paramètres d'action comparables au phénomène biologique d'accoutumance. Certains neurones peuvent répondre de moins en moins activement à un stimulus lorsque celui-ci se présente fréquemment. Une fréquence de sélection est introduite, qui s'incrémente ou se décrémente selon que le neurone est le foyer d'activation (cluster) ou non. Ensuite, la corrélation des neurones aux entrées est pondérée par un facteur dépendant de la fréquence d'activation. Le choix du réseau est purement lié à des critères géométriques de la pièce pour la structure et à des critères physiques du problème pour le nombre de neurones (dont va dépendre le nombre de mailles). La forme

des mailles est donnée par les relations de voisinage entre neurones. Par exemple, un voisinage de 6 donne des mailles triangulaires.

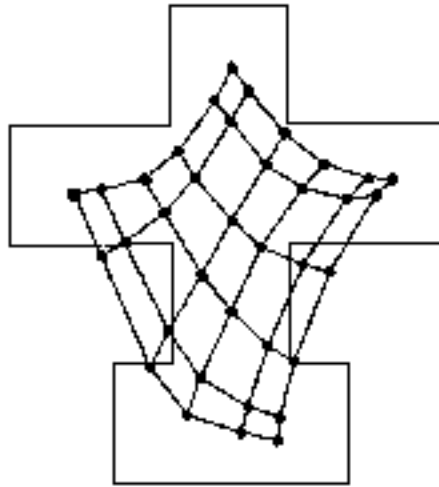


Figure 1. Maillage d'une géométrie concave par une carte auto-organisatrice

Algorithme

A Préparation au maillage réalisée par l'utilisateur :

A.1 Découpe de la géométrie en sous-domaines convexes.

A.2 Discrétisation des contours.

B Pour chaque sous-domaine :

B.1 choix du réseau (architecture, type de mailles).

B.2 détermination des poids des neurones périphériques.

B.3 choix du nombre d'itérations d'auto-organisation.

B.4 auto-organisation du réseau (maillage).

C Assemblage des sous-domaines.

Résultats expérimentaux

Les résultats obtenus (figure 2) montrent la validité d'une telle approche. La qualité géométrique du maillage est satisfaisante. La maîtrise du nombre de mailles est possible. La souplesse au niveau de la non-uniformité du maillage est accrue par rapport aux mailleurs automatiques classiques. La combinaison des éléments (triangles et quadrilatères) est possible. Enfin, cette approche permet une numérotation optimale des noeuds. Le nombre

d'éléments singuliers est réduit, le mixage des éléments triangulaires et carrés est possible et les propriétés d'arrangement spatial des cartes permettent de respecter simplement les critères physiques.

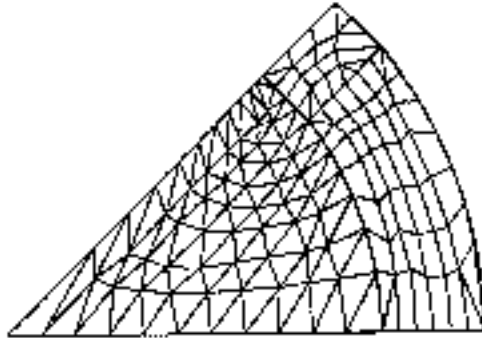


Figure 2. Maillage d'un huitième d'alternateur découpé en trois sous-domaines. L'encoche est maillée par un réseau carré 5 x 5, la partie extérieure du rotor par un réseau rectangulaire 7 x 8. Enfin, le triangle intérieur est maillé par un réseau triangulaire de 12 neurones de côté, les mailles étant triangulaires. Le maillage se compose de 187 éléments. La discrétisation du contour est géométrique aux alentours de l'encoche. On a de plus défini une zone critique maillée plus finement dans le coin supérieur droit du triangle. Le maillage nécessite 500 itérations pour l'encoche, 2000 pour la partie extérieure et 5000 pour le triangle intérieur.

Publications

O. SARZEAUD, Y. STEPHAN, C. TOUZET, "Finite Element Meshing using Kohonen's Self-Organizing Maps", ICANN91, Helsinki, Finland, juin 1991.

O. SARZEAUD, Y. STEPHAN, C. TOUZET, "Application des cartes auto-organisatrices à la génération de maillage aux éléments finis", Neuro-Nîmes 90, Nîmes, France, novembre 1990.

Compression d'images visiophoniques par RNA

(Contrat FEDER-GENESYS, Montpellier)

Les images qui proviennent de la télévision, de visioconférence, de satellite, ou d'applications médicales ... quelles qu'elles soient, représentent une quantité énorme de données après digitalisation. Diminuer les coûts de transmission ou de stockage est d'un intérêt très ancien (par exemple, le code Morse). En ce qui concerne les applications aux communications, le but recherché est alors de minimiser la durée de transmission d'un certain volume d'informations. Ceci apporte à la fois une économie sur le coût, une diminution des risques d'erreurs, une plus grande ergonomie et une plus grande performance, puisque les données sont acheminées en un temps plus court. Les diverses méthodes de compression sont basées sur les techniques de prédiction, de transformation, ou de quantification vectorielle, avec des possibilités de combinaisons entre elles. Elles sont de deux types. Si la transformation réalisée est réversible, la réduction utilise la redondance d'informations et aucune information n'est perdue. Dans l'autre cas, la transformation est irréversible. C'est une réduction d'entropie et il y a perte d'informations. Nous avons étudié l'utilisation des techniques connexionnistes pour la compression d'images visiophoniques : utilisation des cartes auto-organisatrices pour la quantification vectorielle d'images.

L'image à transmettre est découpée en blocs ou matrices de pixels. Chaque pixel peut être codé par les poids de composantes RVB ou par une composante de luminance et deux composantes différentielles de couleur (Y, Cb, Cr). La technique de codage fait appel à la capacité du réseau de neurones à différencier et classer les blocs en considérant :

- qu'au sein d'une même image, de nombreux blocs sont identiques,
- qu'une image est composée pour une part importante de blocs déjà présents dans l'image.

Les cartes auto-organisatrices permettent de projeter, en respectant la densité des exemples, les données d'entrées sur l'espace du réseau. Dans le cas de la quantification vectorielle, il y a autant de neurones que de mots dans le dictionnaire. Les exemples d'apprentissage sont constitués de tous les blocs de l'image. La quantification vectorielle réalise la mise en correspondance d'un ensemble de vecteurs d'entrée avec un ensemble de vecteurs de sortie, selon une mesure de distorsion. Tous les échantillons d'une région (de l'espace des blocs) sont représentés par un seul code. Ce code est un index du dictionnaire. Le dictionnaire est composé des vecteurs les plus fréquents de l'image (en terme de fréquence), ce sont les "mots" du dictionnaire. Sur la ligne de transmission, les codes correspondant à chaque bloc de l'image initiale sont envoyés séquentiellement. A la réception, on utilise le dictionnaire pour reconstruire l'image, qui est donc établie seulement à partir des mots du dictionnaire.

Les essais réalisés, à l'aide d'une image test de 172 x 128 (22 016 pixels), comparent les performances obtenues pour les différents paramètres suivants :

- dictionnaires de tailles différentes (liées à la taille de la carte, c'est-à-dire le nombre de neurones qui coderont les blocs),
- dictionnaires de dimensions différentes (l'espace sera projeté sur un réseau à 1 dimension, 2 dimensions et 3 dimensions),
- longueur de la séquence d'apprentissage (nombre d'itérations). On cherche à optimiser la durée d'apprentissage),
- taille des blocs , c'est-à-dire la dimension des vecteurs d'entrée,
- les variations du voisinage (par a et b qui sont les paramètres de changement de poids:

Les performances de la compression sont mesurées par :

MSE (Mean Square Error) qui représente la différence entre l'image initiale et l'image reconstituée :

$\hat{A} \text{ erreurs}^2$ ou bien, $\hat{A} \text{ erreurs}^2 / \text{pixel}$.

Le taux de transmission (Bit Rate) = Nombre de bits / pixel (bpp)
 $= (\log_2 \text{ nbre de vecteurs} / \text{nbre de pixels par bloc})$.

La quantification vectorielle

Une image est un objet analogique. Sa représentation spatiale est continue de même que la valeur de la couleur en chacun de ses points. Pour modéliser une image, il faut transformer l'image analogique par une fonction d'échantillonnage S, en une fonction discrète $f(x,y)$ (x et y étant les coordonnées des pixels) dont l'amplitude représente l'intensité lumineuse. Ce processus est décrit par la figure 1. Dans le cas particulier des images n'ayant qu'une seule couleur, l'intensité en chaque point est appelée niveau de gris. Le nombre plus ou moins élevé de niveaux de gris spécifie la qualité de l'image (par exemple : 256 niveaux de gris sont un critère de qualité).

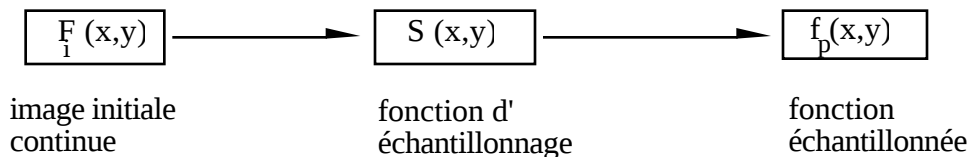


Figure 1. Échantillonnage

Pour résumer, l'échantillonnage est une numérisation :

- en espace : discrétiser les coordonnées (x,y), c'est échantillonner l'image.
- en fréquence : discrétiser la couleur d'un point, c'est quantifier les niveaux de gris.

Lors de la quantification vectorielle, l'image à transmettre est découpée en blocs de pixels. Chaque pixel peut être codé par les poids des composantes RVB (Rouge, Vert, Bleu) ou par une composante de luminance et deux composantes différentielles de couleur (Y, Cb, Cr). La technique de codage est basée sur le fait, qu'au sein d'une même image de nombreux blocs sont identiques ou peu différents.

La quantification vectorielle réalise une mise en correspondance d'un ensemble de vecteurs d'entrée avec un ensemble de vecteurs de sortie, selon une mesure de distorsion. Tous les échantillons d'une région (en fait les blocs semblables) sont représentés par un seul code. Ce code est un index du dictionnaire. Le dictionnaire est composé des blocs les plus fréquents de l'image, ce sont les "mots" du dictionnaire. Sur la ligne de transmission, les codes correspondant à chaque bloc de l'image initiale sont envoyés séquentiellement. A la réception, on utilise le dictionnaire pour reconstruire l'image, qui est donc établie seulement à partir des mots du dictionnaire.

Le principal problème de la quantification vectorielle réalisée par l'algorithme classique (Linde-Buzo-Gray ou LBG) est la non-invariance du dictionnaire par rapport aux conditions initiales. En effet, on est obligé de fixer des valeurs initiales pour les mots du dictionnaire, valeurs qui influencent la solution finale. De fait, cette solution n'est pas toujours optimale. D'autre part, pour construire un dictionnaire optimal, on utilise une mesure de distorsion entre l'ensemble d'entraînement et le dictionnaire. L'algorithme est sous-optimal puisqu'il tend à sur-coder les régions denses de points alors que les régions clairsemées sont sous-codées. Les blocs peu représentés (et éventuellement très significatifs pour l'oeil) sont rejetés au profit de blocs plus courants (codant le fond par exemple).

3/ Principe de la quantification vectorielle par carte auto-organisatrice (fig. 2)

Le nombre de mots du dictionnaire est égal à la taille du réseau (nombre de neurones). Le dictionnaire est donc composé des blocs les plus représentatifs de l'image.

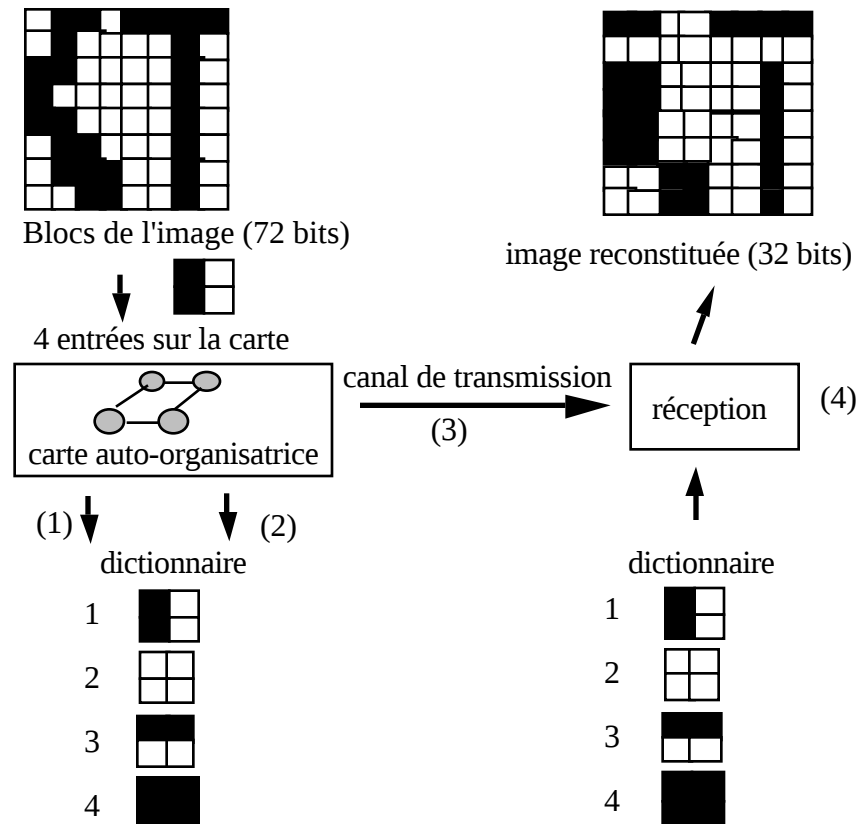


Figure 2. Principe de la quantification vectorielle par carte auto-organisatrice

- (1) Construction du dictionnaire : la carte sélectionne les mots du dictionnaire à partir des blocs les plus représentatifs de l'image.
- (2) Codage de l'image (par le dictionnaire) : la carte sélectionne pour le bloc de l'image qui lui est présenté le numéro du mot du dictionnaire le plus proche.
- (3) Transmission : le numéro des vecteurs est transmis par le canal.
- (4) Décodage (reconstitution).

Expérimentations²⁸

1/ Variation de la taille de la carte

La dimension des vecteurs est de $2 \times 2 = 4$ pixels. Il y a 20 itérations d'apprentissage ($\mu = 0.8$ et $\beta = 0.7$). Plus le nombre de neurones est important, meilleure est l'image restituée ; mais le taux de compression varie à l'inverse. Il faut trouver un compromis entre la qualité de l'image et le taux de compression (fig. 3). Nous avons choisi pour la suite de l'expérimentation 30 neurones (6×5) et des blocs de 2×2 .

²⁸ Nos expérimentations utilisent une image de 172×128 (22 016 pixels) sur 8 niveaux de gris.

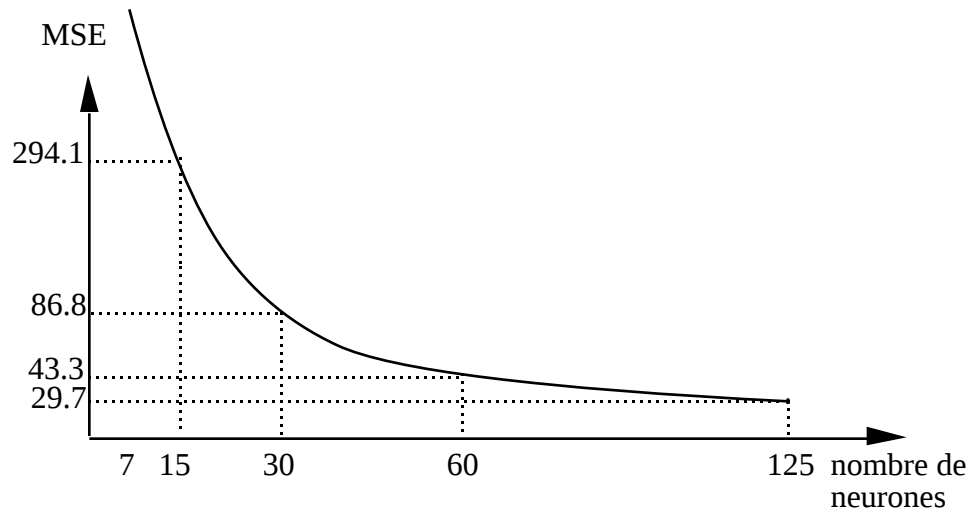
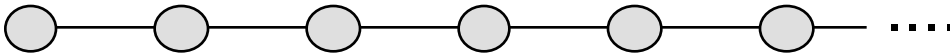


Fig. 3 Variation de la taille de la carte auto-organisatrice

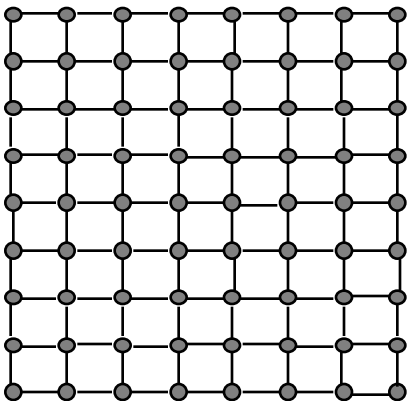
2/ Variation des dimensions du réseau : réseau 1D, 2D, 3D

Nous choisissons d'effectuer seulement 10 itérations et les valeurs suivantes : $\mu = 0.5$ et $\beta = 0.1$, les blocs sont de 2×2 . La dimension de l'espace d'entrée est égale à la taille des blocs. L'hypothèse que nous testons est : la projection faite sur l'espace de ce réseau est-elle d'autant meilleure que les dimensions du réseau et des entrées sont plus proches.

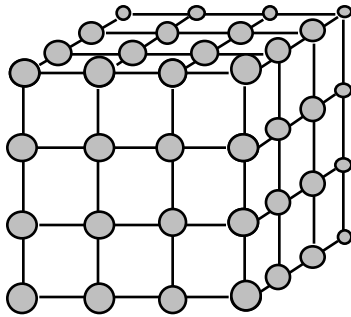
- Réseau à 1D : ligne de 64 neurones



- Réseau à 2D : carte de $8 \times 8 = 64$ neurones



- Réseau à 3D : cube de $4 \times 4 \times 4 = 64$ neurones



Le meilleur résultat est obtenu pour le réseau ligne (1D). Le voisinage est réduit et tend vers 0 (cf. variation du voisinage). Notre objectif est de tester la carte auto-organisatrice pour la compression d'image, nous conservons donc le réseau à 2D pour la suite des tests.

3/ Variation de la taille des blocs (dimension des vecteurs)

Carte de 30 neurones (6 x 5), 20 itérations, $\mu = 0.8$ et $\beta = 0.7$. Plus les blocs sont gros, plus le taux de compression est important. Le problème est qu'il existe une taille limite pour ces blocs à déterminer. Les blocs 4 x 4 sont très grands, mais ils offrent un bon taux de compression (12.8). Les blocs 2 x 2 donnent un meilleur résultat visuel que les blocs 4 x 1 car dans ce cas, les lignes horizontales sont trop visibles bien qu'ayant le même taux de compression et une erreur sensiblement identique.

4/ Variation du nombre d'itérations

Carte de 30 neurones (6 x 5), blocs de 2 x 2 pixels, $\mu = 0.8$ et $\beta = 0.7$. Plus la durée d'apprentissage est réduite, plus une intégration dans un système réel devient envisageable. Nous choisissons de nous limiter à 15 itérations (fig. 4) pour la suite de nos expérimentations, c'est le nombre pour lequel la distorsion est minimale.

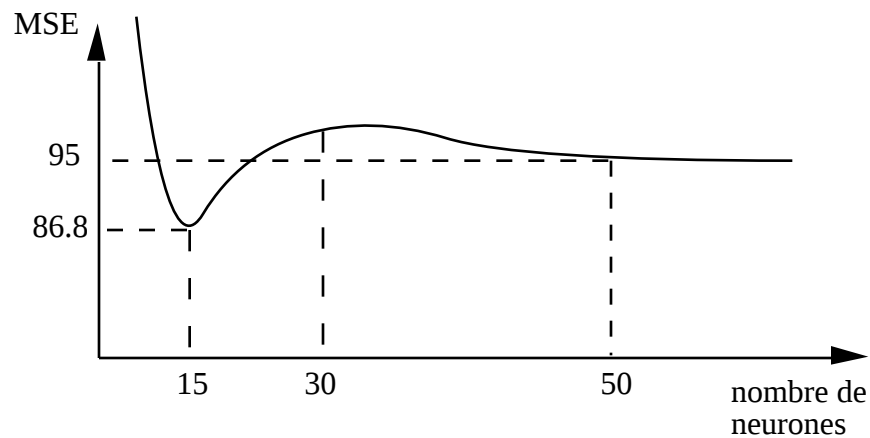


Figure 4. Variation de la séquence d'entraînement

5/ Variation du voisinage

Le meilleur résultat (en terme de minimisation de la distorsion) est atteint pour les valeurs ($\mu = 1$; $\beta = 0$), c'est-à-dire sans voisinage. Cependant, bien que l'erreur soit supérieure en présence d'un voisinage, à l'oeil l'image semble meilleure. Ce phénomène se reproduit sur plusieurs autres images. Nous constatons ici le fait bien connu que la mesure de la distorsion n'est pas un critère de mesure de la qualité visuelle d'une image compressée.

Conclusion

La variation de la taille de la carte montre que plus celle-ci est de taille importante, plus la qualité de l'image restituée est bonne. Toutefois, plus il y a de neurones, moins le taux de compression est important. Il faut donc rechercher un compromis entre un taux de compression satisfaisant et une bonne qualité d'image.

Concernant la dimension du réseau, plus la dimension est faible, meilleure est la MSE. Cependant, le réseau à 2D est celui que nous avons choisi pour toute la suite de nos travaux, car bien que de qualité absolue inférieure (MSE), il est d'une *qualité visuelle meilleure*. C'est un constat de même type que pour la variation du voisinage.

La variation de la taille des blocs donne un bon résultat pour 2 x 1 pixels (mais le taux de compression serait alors seulement de 1,6), alors qu'à l'opposé les blocs 4 x 4 pixels donnent un mauvais résultat visuel mais la compression est de 12,8.

Le nombre d'itérations de l'apprentissage fait varier l'erreur totale. Pour 15 itérations, on obtient une distorsion minimale. Au delà de 15 itérations, c'est le phénomène du "par coeur" qui prend la place de la généralisation.

Enfin, les paramètres μ et β font eux aussi varier la distorsion : les meilleurs résultats en valeur de distorsion, sont obtenus pour $\mu = 1$ et $\beta = 0$, c'est-à-dire un réseau sans voisinage. Dans ce cas, on ne peut plus parler de réseau (il n'y a plus de voisinage), le comportement observé est identique à celui de l'algorithme de Linde-Buzo-Gray. Notons cependant que dans ce cas aussi, l'appréciation subjective fournie par la vision humaine ne suit pas toujours les critères quantitatifs.

Enfin, l'utilisation de la carte de Kohonen permet de palier à l'un des problèmes majeurs rencontrés par l'algorithme LBG : la non-invariance par rapport aux conditions initiales. Dans notre cas, le dictionnaire initial est choisi aléatoirement et n'influence pas le résultat final. Rappelons que dans les autres méthodes de quantification vectorielle, ce choix met en oeuvre de nombreux calculs et constitue un facteur déterminant pour la qualité du quantificateur final.

Publications

S. Dellaporta et C. Touzet, "Étude de faisabilité de la compression d'images visiophoniques par réseaux neuronaux," rapport de contrat FEDER - GENESYS, juillet 1991.

S. Dellaporta, "Compression d'images par cartes auto-organisatrices," rapport de DEA, Université des Sciences et Techniques du Languedoc, juillet 1991.

Étude et développement d'une méthode d'analyse par coalescence pour la détermination des espèces génétiques basée sur les RNA

(Collaboration avec le Centre de Coopération Internationale en Recherche Agronomique pour le Développement à Montpellier, CIRAD)

Nous étudions le problème du regroupement (coalescence) des distances de migrations des fragments d'ADN (données RFLP) à l'aide d'une approche basée sur les réseaux de neurones. Cette approche nous permet de stabiliser le comportement de l'expert dans sa tâche d'analyse. Il s'agit de trouver un moyen de classification des bandes révélées sur un autoradiogramme, qui soit basée sur des critères connus, objectifs mais modifiables, et qui permette à différents experts humains d'arriver à une même classification. Une première approche a été proposée qui utilise des algorithmes de classification hiérarchique basée sur un seuil déterminé expérimentalement, pour obtenir des classifications plus fiables que par simple comparaison de poids moléculaires calculés par interpolation. Cette méthode perd rapidement sa robustesse dès qu'il s'agit de classifier des séries de trois bandes ou plus dont la taille augmente successivement d'une valeur proche du seuil d'agrégation. Nous proposons de constituer une base d'exemples de coalescences réalisés par un expert humain afin de réaliser l'apprentissage d'un réseau neuronal ART. Suite à l'apprentissage, le réseau réalise en fonction d'une valeur de seuil l'association des mesures. Le comportement du réseau et ses performances sont influencés par de nombreux paramètres tels que le codage des informations d'entrées sur la couche d'entrée, l'interprétation des informations de la couche de sortie, les valeurs de gains, etc. Ces paramètres ne peuvent être déterminés que par une expérimentation extensive.

La RFLP (Restriction Fragment Length Polymorphism) est une des méthodes les plus puissantes de comparaison entre différents individus. Elle consiste en la mesure des variations de poids moléculaire de fragments de leur matériel génétique, l'ADN, après digestion de celui-ci par des enzymes dits de "restriction". Pour cela, on s'attache à déterminer les distances de migration de ces fragments d'ADN après séparation par électrophorèse. Les variations de taille et donc de migration des fragments entre deux individus sont appelées "polymorphisme de la taille de fragments de restriction" ou données "RFLP", et sont lisibles sur un autoradiogramme.

ART (Adaptive Resonance Theory) est un modèle de réseau à apprentissage compétitif, particulièrement bien adapté car il permet de déduire des valeurs de seuil pour l'agrégation des données à partir d'un algorithme d'apprentissage par l'exemple et il n'est pas restreint à un nombre de classes prédéfinies. Dans la plupart des réseaux de neurones, deux étapes sont considérées. La première est la phase d'apprentissage : les poids des

connexions sont modifiés selon une règle d'apprentissage. La deuxième est la phase d'exécution où les poids ne sont plus modifiés. Avec le réseau ART, ces deux étapes sont réalisées simultanément. Le réseau en phase de test, s'adapte à des entrées inconnues en construisant de nouvelles classes (ajout de neurones) tout en dégradant au minimum les informations déjà mémorisées. Il existe plusieurs versions de réseaux (ART1, ART2, ART3). Le réseau ART1 est un réseau à entrées binaires.

Le réseau ART1 est formé d'une couche d'entrée qui est aussi la couche de sortie et d'une couche cachée (fig. 1). Le terme de couche cachée est emprunté au réseau multicouche, il souligne le fait que cette couche n'est pas directement observable par l'utilisateur à la différence de l'entrée ou de la sortie. Il n'y a pas de connexion entre les neurones d'entrées. Par contre, la couche cachée est une couche d'activation compétitive, tous les neurones sont reliés les uns aux autres par des connexions inhibitrices de poids fixes. Chaque neurone de la couche d'entrée est relié à tous les neurones de la couche cachée et, réciproquement, chaque neurone de la couche cachée est relié à tous les neurones de la couche de sortie. A chaque connexion est associé un poids.

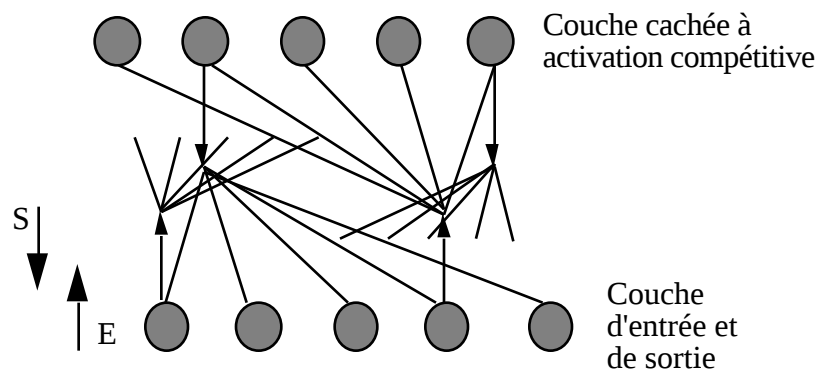


Figure 1. Architecture du réseau ART 1. La couche d'entrée est aussi celle de sortie. Tous les neurones de la couche d'entrée sont reliés à tous les neurones de la couche cachée et tous les neurones de la couche cachée à chacun de ceux de la couche de sortie. Il n'y a pas de relation entre les neurones d'entrée alors que la couche cachée est à activation compétitive.

La figure 2 montre un vecteur d'entrée E soumis au réseau. A cette entrée correspond, après compétition entre les neurones de la couche cachée, un unique neurone j gagnant. Ce gagnant est considéré par le réseau comme le plus représentatif du vecteur d'entrée E. Le neurone j génère en retour sur la couche de sortie un vecteur S binaire (seuillage). S est ensuite comparé au vecteur d'entrée E. Si la différence est inférieure à un seuil fixé pour le réseau, le neurone gagnant est considéré comme représentant de la classe du vecteur d'entrée. Dans ce cas, la modification des poids des connexions du neurone gagnant a pour effet de consolider ses liens d'activation avec l'entrée E ; en fait l'adéquation entre ce vecteur d'entrée et cette classe est améliorée. Dans le cas contraire, le

processus reprend avec les neurones de la couche cachée moins le neurone gagnant de l'étape précédente. Si tous les neurones cachés sont passés en revue sans qu'aucun ne corresponde à E, un nouveau neurone caché est ajouté, qui est initialisé comme représentant de la classe du vecteur d'entrée E.

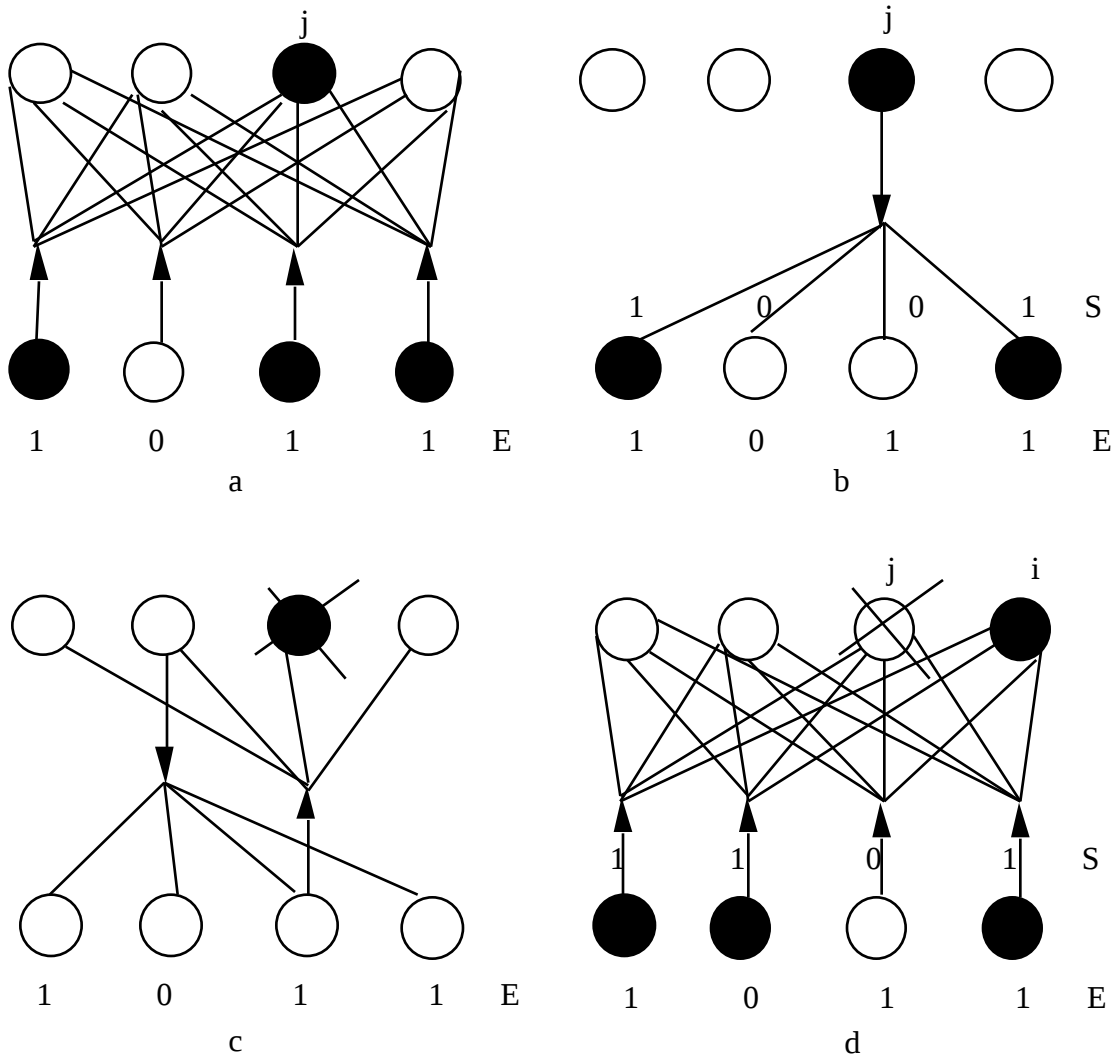


Figure 2. Fonctionnement du réseau ART1.

- Présentation du vecteur d'entrée E , un neurone gagnant j est sélectionné.
- Tentative d'unification entre S (retour du neurone j) et E .
- Échec : suppression du neurone gagnant, présentation de E .
- Unification : le neurone i est un représentant de la classe du vecteur d'entrée E .

Algorithme

L'apprentissage consiste tout autant dans la détermination des poids que de la valeur du seuil d'unification β .

- 1/ Initialisation des poids aléatoirement entre 0 et 1 et choix d'un seuil d'unification β .
- 2/ Présentation d'un vecteur d'entrée E_1 appartenant à la base d'apprentissage
- 3/ Calcul du neurone gagnant sur la couche cachée N_j .
- 4/ Génération en retour d'un vecteur de sortie S_j issu de ce seul neurone N_j . S_j a été seuillé afin de le rendre binaire.
- 5/ Tentative d'unification entre S_j et E_1 . Soit $|S|$ est la norme de S_j égale au nombre de composantes à 1, par exemple $|(1, 0, 1, 1)| = 3$.

Si $|S_j| / |E_1| \geq \beta$, l'unification est réalisée. Il faut modifier les poids : étape 7.

- 6/ Sinon $|S_j| / |E_1| < \beta$, le neurone gagnant N_j est inhibé.

S'il y a encore des neurones non inhibés sur la couche cachée alors retour à l'étape 3.

Sinon un nouveau neurone cachée est créé, initialisé comme représentant de la classe correspondant à la forme d'entrée E_1 en utilisant la loi de modification des poids de l'étape 7.

- 7/ Modification des poids

Couche des poids montants : (h neurone de la couche d'entrée, j neurone gagnant de la couche cachée)

$w_{jh} = 1 / |S_j|$ si le neurone h est actif (valeur 1),

$w_{jh} = 0$ sinon (valeur 0).

Couche des poids descendants: (j neurone gagnant de la couche cachée, k neurone de la couche de sortie)

$w_{kj} = 1$ si le neurone k est actif,

$w_{kj} = 0$ sinon.

Retour à l'étape 2.

- 8/ Quand le passage de tous les exemples de la base d'apprentissage n'occasionne plus aucun ajout de neurone, il faut mesurer les performances : contrôler le nombre et la qualité des classes construites. Si le nombre est trop faible, retour à l'étape 1 avec une augmentation de la valeur de β . Si ce nombre est trop élevé, retour à l'étape 1 en diminuant la valeur de β .

La valeur du seuil contrôle le degré d'unification recherché entre les formes à classer et les prototypes des classes. Plus la valeur du seuil est grande, meilleure est l'adéquation recherchée. La valeur du seuil doit être choisie entre 0 et 1. Le neurone i est rattaché à une classe dont le prototype générique à priori ne correspond précisément à aucune des formes de la base d'apprentissage. L'unification est réalisée lorsque le nombre d'entrées à 1 est comparable avec le nombre de retours à 1 (coactivation statistique).

Résultats

Un exemple de coalescence de données issues d'une distribution parabolique est réalisé (fig. 3). Les coordonnées d'un ensemble de point pris sur la parabole sont soumis en données d'entrée au réseau ART1 (fig. 2a). Après quelques itérations de l'ensemble de la base d'exemple, les classes construites par le réseau sont présentées sur la figure 2b. Ici quatre classes correspondants aux lettres a, b, c et d sont représentées, la valeur du seuil de vigilance est de 0.7. Plus la valeur de seuil est proche de 1, plus le nombre de classes créées est grand et réciproquement.

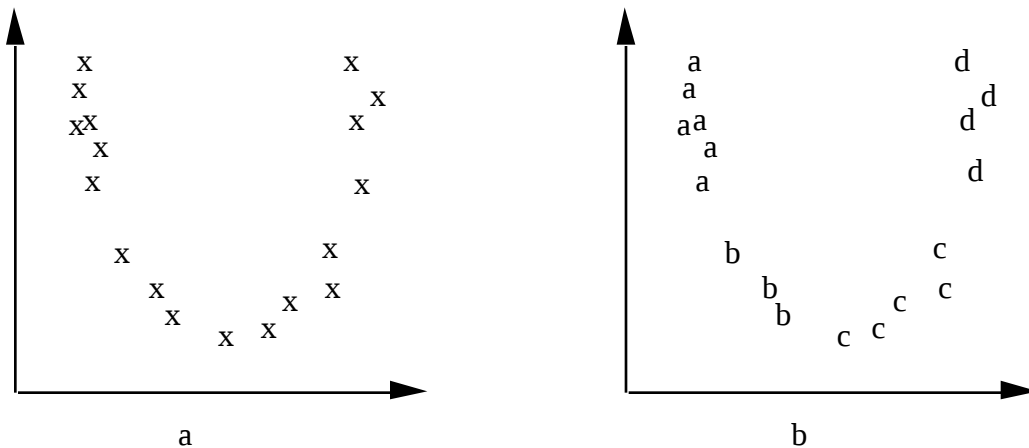


Figure 3. Exemple de traitement réalisé par le réseau ART1

- a) Base d'apprentissage (points extraits sur une parabole).
- b) Coalescence réalisée avec un seuil $\beta = 0.7$ (4 classes).

Les applications majeures du réseau ART ont été réalisées en reconnaissance de la parole, reconnaissance des formes visuelles, en détection d'image radar ainsi qu'en classification et en coalescence.

Conclusion

Le réseau ART1 a une architecture en deux couches qui interagissent entre elles. Le réseau se distingue aussi par deux caractéristiques: sa flexibilité et sa stabilité pour des entrées arbitraires. Il est capable de s'adapter à des entrées non familières en construisant de nouvelles catégories ou classes (flexibilité, plasticité) mais aussi d'adapter les classes déjà apprises tout en dégradant peu les informations déjà mémorisées (stabilité). Le problème posé par ces deux notions antagonistes (flexibilité-stabilité) est résolu par le principe de l'architecture évolutive.

Références

S. SEHAD, Réseaux neuronaux "ART" : application à la coalescence de données génétiques, DEA Conception Assistée des Systèmes Informatiques, Automatiques et Micro-électroniques, U.S.T.L., Montpellier, 1992.

Maquette de démonstration de l'apprentissage par RNA, application à la locomotion hexapode

(Contrat avec Neurosystèmes; 1991-92)

Avec l'objectif de réaliser une démonstration des capacités d'apprentissage des RNA et en particulier de l'apprentissage par renforcement, nous avons réalisé une simulation de l'apprentissage de la marche chez les insectes (figure 1). Chez la plupart des insectes, six pattes permettent la locomotion, chacune est dotée de son propre générateur de mouvement, relié par des connexions intra et inter-segments. On postule souvent l'existence d'un générateur central de formes locomotrices. Nous montrons qu'un générateur central de formes locomotrices n'est pas nécessaire, son rôle peut être tenu par l'algorithme d'apprentissage : le système de locomotion est composé de six réseaux neuronaux complètement interconnectés, sans générateur central d'activité. Les entrées du système sont les informations sensorielles relatives à la position des pattes. Sur la base de ces entrées, le système génère une nouvelle position des pattes. Ainsi, chaque patte a la possibilité de rester inactive, d'avancer ou de reculer. Seule l'action de recul est motrice. L'effet sur l'environnement peut être de trois types : aucun déplacement, avancée ou chute. A partir de cette information, l'apprentissage permet d'éduquer chacun des réseaux pour découvrir une marche. En fait, l'algorithme d'apprentissage explore de manière aléatoire l'espace des mouvements des six pattes. Chaque avancée dans la "bonne" direction est encouragée et on s'éloigne des états conduisant à la chute.

L'algorithme d'apprentissage par pénalité/récompense (**ARP**) s'applique à toute architecture de réseau, la seule contrainte étant d'avoir des neurones de sortie stochastiques. On introduit donc à ce niveau une part d'aléatoire dans le comportement du système. Si la réponse fournie par le système est considérée comme bonne, l'algorithme tend à favoriser l'apparition de ce comportement en réduisant l'aléatoire. Dans le cas où la réponse du système globale est considérée comme mauvaise, on cherche à éviter l'apparition ultérieure de ce comportement. Ce processus est itéré jusqu'à l'obtention du comportement désiré pour l'ensemble du système.

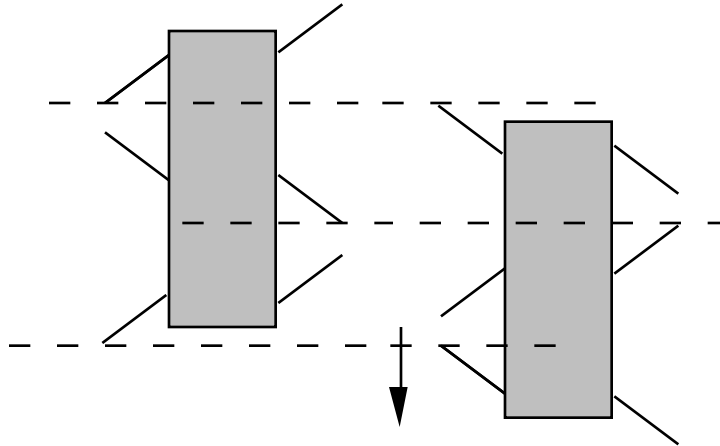


Figure 1. Une marche en deux temps.

Prolongements

Nous avons réalisé une démonstration logiciel de ce système, sur une Silicon Graphics, qui permet une visualisation 3D de la découverte de la marche qui a été exposé au congrès Informatique92, Interfaces des mondes réel et virtuel, Montpellier, mars 1992.

L'École **des Mines d'Ales** a réalisé un robot mobile doté de six pattes, de capteurs, d'effecteurs et de l'algorithme précédemment cité, qui valide nos résultats face au monde réel.

Publications

Claude TOUZET et Olivier SARZEAUD, "Application d'un algorithme d'apprentissage par pénalité/récompense à la génération de formes locomotrices hexapodes," Journées de Rochebrune, AFCET IA. et Cognition, 20-24 janvier 1992.

Récompenses

Il est à noter que le logiciel d'animation 3-D développé sur Silicon Graphics en 1993 a été retenu par le Ministère pour être présenté dans ses jardins parisiens lors des **Journées de la Science 1993**.

D'autre part, la version logicielle réalisée en 1994 a gagné le **concours national Apple** "Programmer un comportement intelligent" doté de 200 000 F pour le vainqueur (fig. 2).

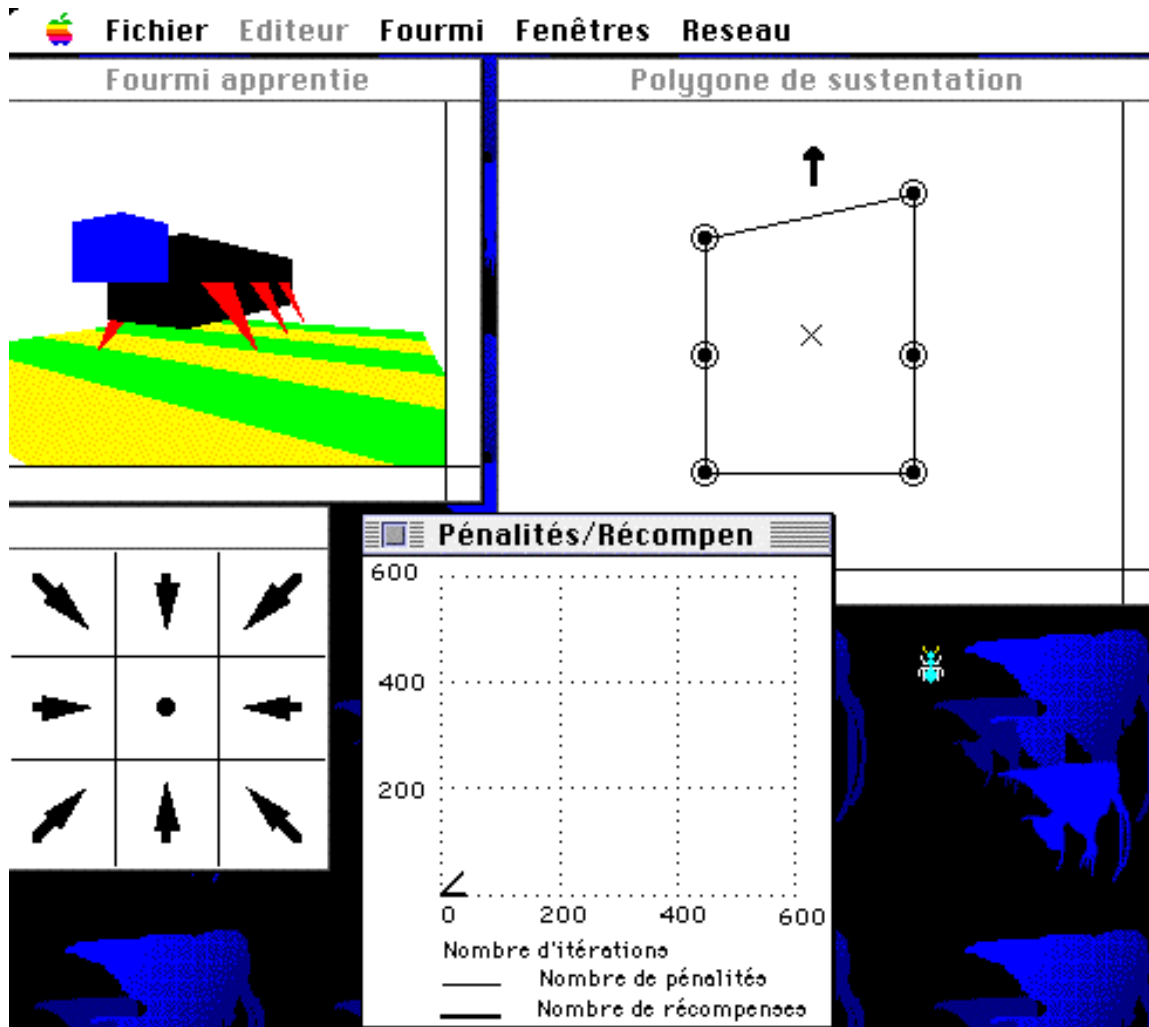


Figure 2. ANTROID, version Mac réalisée en 1994; qui a gagné le concours national Apple.

Aide à la conduite d'un haut-fourneau par réseaux de neurones artificiels

(Contrat avec SOLLAC-Sachem 1993-1997)

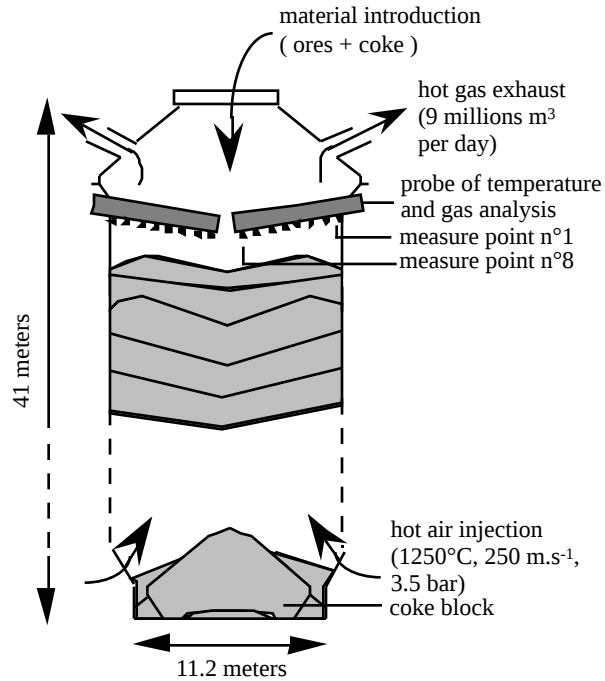


Fig. 1 Coupe schématique d'un haut fourneau. Du fait de la taille et de la complexité du procédé, seules des information capteurs sont utilisées pour le pilotage.

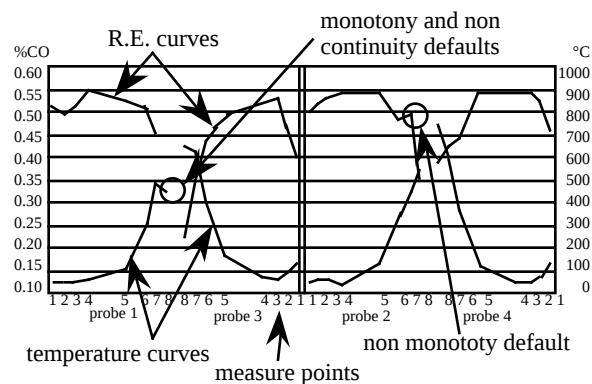


Fig. 2 Un exemple (2 “poutres”) des mesures réalisées sur le haut fourneau. Il y a 4 mesures “poutre” toutes les 16 minutes. Le problème est de déterminer les points de mesures invalides avant la mise en oeuvre des procédures à base de connaissance.

La description de la mise en oeuvre des RNAs est confidentielle. Elle fait l'objet d'un dépôt de brevet (en cours).

Réalisation d'une modalité de communication sonore pour le robot Khepera

(Contrat ANVAR)

L'objectif est de réaliser une maquette permettant la communication sonore entre plusieurs robots mobiles (en l'occurrence des Khepera). La communication sonore est tout à fait adaptés pour transmettre des messages d'ordres généraux sans présupposer de l'orientation ou de la position des agents, ce qui n'est pas le cas avec une communication électromagnétique (infrarouge ou lumière visible), et sans poser les problèmes de réglages délicats entre deux sources radio. De plus avec un tel système, l'opérateur sait tout de suite si les robots sont en communication ou non. Par contre, il sera difficile de localiser la source de l'appel, et impossible de laisser un message durable dans le temps. La communication sonore convient donc pour les messages d'alarme et de regroupement, mais ne pourra être efficacement utilisée comme signal de marquage.

Du plus en plus de chercheurs veulent utiliser des robots mobiles, non seulement en robotique, mais aussi en intelligence artificielle, biologie, sciences cognitives et autres domaines. De fait de nouveaux robots doivent être développés pour satisfaire les caractéristiques suivantes :

- * Facile à installer (plug and play)
- * Bonne puissance de calcul embarquée
- * Modulable(configuration flexible)
- * Pratique à utiliser et sans danger
- * Faible prix

La famille des K-robots (Khepera (Fig. 1) et Koala) est le résultat de cette motivation et satisfait les contraintes qui viennent d'être énumérées. Le grand nombre d'utilisateurs (plus de 400 vendus à travers le monde) et la diversité de leurs activités (des biologistes jusqu'aux mathématiciens) tendent à prouver que l'effort qui a été fait allait dans le bon sens, donnant la possibilité à toutes ces personnes de travailler avec un robot réel sans pour autant être roboticien. Les K-robots constituent une transition entre la simulation et les applications industrielles.



Fig. 1 Le robot miniature Khepera, avec module de vision et pinces

Avec un diamètre d'environ 55 mm., Khepera est un robot mobile très intéressant pour l'expérimentation. Il peut être utilisé sur une table, connecté à une station de travail par le biais d'une liaison série. Cette configuration est optimale pour l'expérimentation, avec tout à portée de main: le robot, l'environnement, et l'ordinateur de contrôle (fig. 2). Pour suivre le robot et afficher sa trajectoire, un scanner laser rend possible l'affichage des coordonnées x,y et l'orientation de chaque robot dans l'espace de travail.

Le module processeur est un microcontrôleur Motorola 68331 (il possède des performances similaires au processeur Motorola 68020 cadence à 16 Mhz) avec 256 kilo-octet de RAM, 256 Kilo-octet de ROM, six canaux

analogiques/numériques avec une résolution de 10 bits, et une liaison série. Le CPU présent sur le robot offre de multiples avantages : facilité de programmation (on peut utiliser tous les langages compatibles avec les processeurs de la famille 68000) et une bonne puissance de calcul. La base motrice est constituée de deux roues, quatre accumulateurs autorisant une autonomie de 30 à 40 minutes, et 8 capteurs de proximité par détection de lumière infrarouge. Ces capteurs sont suffisants pour la détection de lumière et d'obstacles simples.

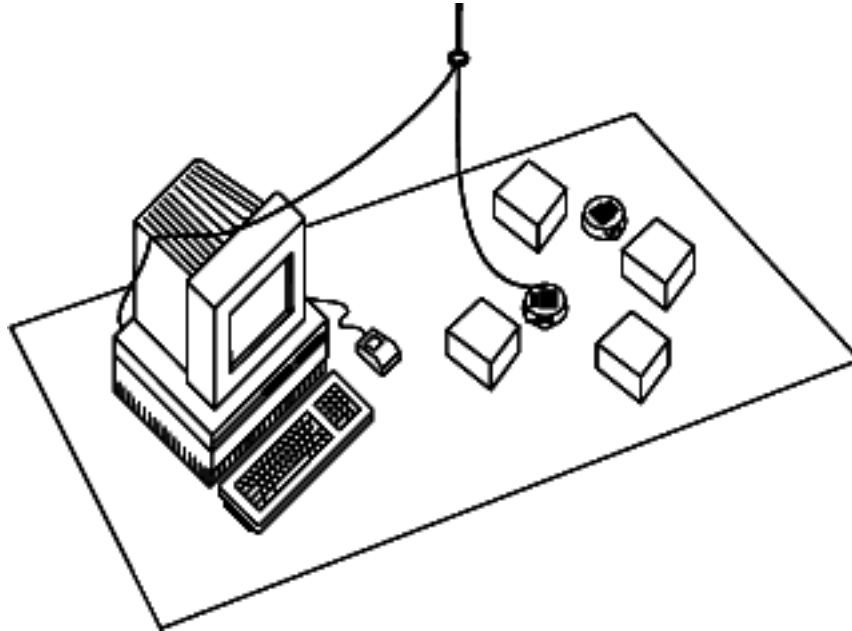


Fig. 2 Exemple de poste de travail avec Khepera

Le robot mobile Khepera a été développé pour tester les algorithmes de contrôle dans des environnements réels. Dans cette optique un soin particulier a été porté sur les aspects suivants :

- * Facilité d'utilisation: expérimentation et visualisation.
- * Flexibilité dans le choix des algorithmes grâce à la puissance de calcul embarquée
- * Facilité pour des modifications matérielles
- * Expérimentation avec plusieurs robots.

La taille du Khepera permet de résoudre les premier et dernier points. La structure du matériel résout en partie les second et troisième points mentionnés ci-dessus. Un bus d'extension donne la possibilité d'ajouter des modules additionnels tels que des pinces ou des caméras.

La solution préconisée pour réaliser une modalité de communication sonore repose sur l'emploi de codeurs et décodeurs DTMF (Dual Tone Mixed Frequency), utilisés couramment pour la numérotation téléphonique ainsi que sur l'utilisation de la tourelle d'extension du Khepera. Cet élément (fig. 3) est indispensable pour adapter un module additionnel sur le Khepera sans avoir à passer par une électronique complexe. De plus toutes les entrée/sortie présentes sur cette tourelle d'extension sont directement contrôlable par le programme de supervision fourni avec Khepera.

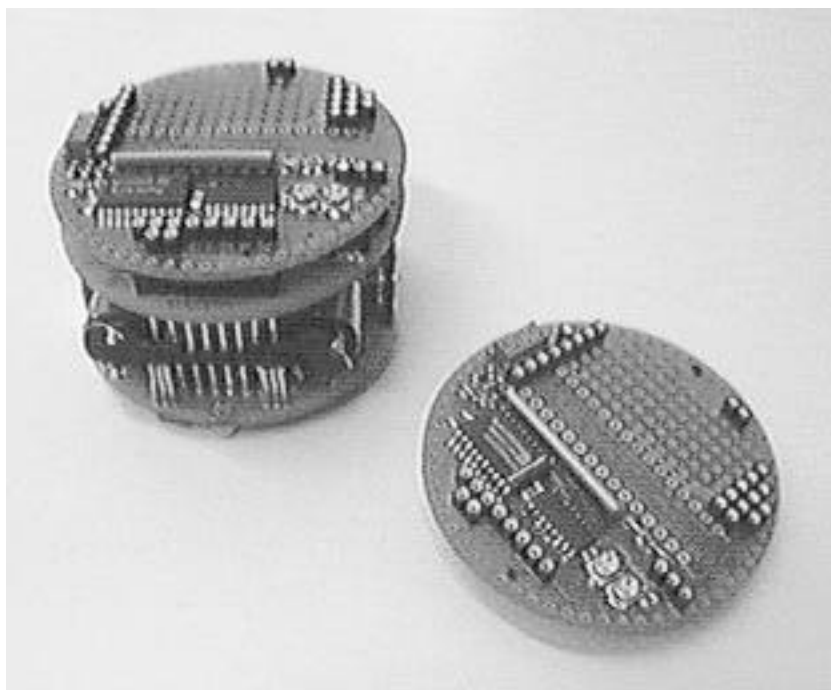


Fig. 3 Tourelle d'extension I/O

Le codage DTMF repose sur le mixage de deux fréquences parmi 8, Pour former 16 combinaisons comme cela est visible dans la table 1 ci-dessous :

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	0	1	2	3
770 Hz	4	5	6	7
852 Hz	8	9	10	11
941 Hz	12	13	14	15

Table 1.
Fréquences
utilisables

Conclusion

A l'issue de notre projet, nous disposons d'une maquette réalisant la communication sonore entre deux robots KHEPERA. La communication sonore est parfaitement adaptée à la transmission de messages généraux entre agents sans qu'il soit nécessaire de connaître leur orientation ou leur position. Nous nous affranchissons ainsi des deux conditions qui font obstacle à la transmission électromagnétique (infrarouge ou lumière visible). Par rapport à la communication radio, qui est elle aussi indépendante de la position et de l'orientation, la communication sonore ne pose sans doute pas autant de problèmes délicats de réglage entre les sources. De plus, il est tout à fait possible d'imaginer des lieux et des situations où les ondes radio sont perturbées, absorbées alors que les ondes sonores ne sont pas affectées (et réciproquement). Enfin, la communication radio nécessite un traitement du signal important et complexe, alors que la communication sonore est beaucoup moins gourmande en puissance de calcul.

La principale raison que nous pouvons envisager pour expliquer l'absence à l'heure actuelle de ce type de communication pour la robotique est due à la difficulté d'envisager pour l'ingénieur roboticien la possibilité de communiquer avec un répertoire limité (dans notre maquette, seulement 16 signaux possibles). Pourtant, les exemples pris dans la nature ne peuvent que confirmer le fait que disposer d'un vocabulaire réduit ne limite pas pour autant la communication et l'expression des comportements collectifs (alarme, recherche de nourriture, chasse collective, retour au nid, etc...). En dotant Khepera, qui est un outil de recherche et développement à la disposition de la nouvelle génération de roboticiens, d'une modalité de communication sonore, nous espérons générer de nouvelles habitudes de programmation robotique, moins gourmande en puissance de calculs. Offrir une palette d'outils et fonctionnalités toujours plus vaste autour de sa gamme de robots est une caractéristique essentielle de **K-TEAM S.A.** (le fabricant du KHEPERA) et **PREFIGURE** (son distributeur en France). C'est ce qui motive leurs intérêts pour notre projet.

Il est bien évident que les travaux menés pour doter Khepera d'une modalité de communication sonore peuvent facilement être étendus à d'autres robots, plus orientés vers l'industrie ou des applications très précises. C'est ce qui motive l'intérêt de la **Sécurité Civile** pour notre projet. Aujourd'hui, les applications faisant intervenir de multiples robots au sein la même et unique application (surveillance d'entrepôts, ramassage divers, guidage, etc.) sont en cours de développement. La communication sonore trouve là un très vaste champ d'applications.

B - Publications jointes

1 - C. TOUZET, N. GIAMBIASI, "The Connectionist Sequential Machine: a General Model of Sequential Networks", *Australian Conf. on Neural Networks*, in Canberra, P. Leong & M. Jabri Eds. Sydney University Electrical Engineering, NSW 2006, Australia, February 1992.

2 - C. TOUZET, N. GIAMBIASI, "Application of Connectionist Models to Fuzzy Inference Systems", in *Parallelization in Inference Systems*, Lectures Notes in Artificial Intelligence 590, B. Fronhöfer & G. Wrightson Eds., Springer Verlag, April 1992.

3 - C. TOUZET, N. KIEFFER and M. LE GOC, "Artificial Neural Networks Forecasting and Monitoring Scaffold and Scaffolding Phenomena in Blast Furnaces," *IEEE International Conference on SMC*, Vancouver, Canada, October 1995.

4 - C. TOUZET, "Neural Reinforcement Learning for Behaviour Synthesis," *Robotics and Autonomous Systems*, Special issue on Learning Robot: the new wave, N. Sharkey Guest Editor, Vol. 22, Nb. 3-4, pp. 251-281, December 1997.