UTILISATION DE LA "STRATÉGIE D'APPRENTISSAGE" DANS LES RÉSEAUX CONNEXIONNISTES*

Nicolas SZILAS

ACROE-LIFIA
INPG - 46, av. Félix Viallet
38031 GRENOBLE Cedex
FRANCE

Tél. : (33) 76 57 48 12
Email : Nicolas.Szilas@imag.fr

Résumé : D'un côté, nous constatons que de nombreux apprentissages connexionnistes se réduisent à l'acquisition et la représentation de données issues de l'environnement, selon un schéma "stimulus-réponse" totalement passif.

De l'autre, il apparaît deux grandes difficultés dans les réseaux actuels : la difficulté à gérer un grand nombre de poids (on cherche d'ailleurs toujours le nombre minimal de poids permettant de résoudre le problème) et la difficulté à apprendre des données complexes et structurées.

Nous sommes alors amenés à considérer que les deux difficultés ci-dessus mentionnées relèvent du même problème, à savoir la difficulté des réseaux à se *structurer*, et que l'approche "passive" des réseaux est à remettre en cause. Nous proposons d'élaborer des mécanismes de *stratégie* contrôlant le déroulemeint de l'apprentissage, qui interagissent avec le réseau *et* avec l'environnement afin de faciliter cet apprentissage.

Pour résoudre le problème de structuration ci-dessus, nous postulons qu'il faut limiter "l'effort" du réseau à chaque étape, ce qui nous conduit à la notion d'apprentissage progressif.

Ensuite, nous proposons quatre algorithmes d'apprentissage "stratégique" basés sur l'idée de progressivité, dont le dernier, le plus général (rien n'est donné a priori), agit sur l'environnement pour sélectionner des données à apprendre de plus en plus complexes, et par la même est amené à construire une hiérarchie des actions sur l'environnement, selon leur complexité (difficulté à apprendre les données qui en résultent). Enfin, nous exprimons notre souhait de rapprocher cette étude, encore embryonnaire, de la psychologie cognitive et développementale, de la didactique, et de toutes les autres recherches sur l'apprentissage interactif.

Mots clés : réseaux neuronaux, stratégie d'apprentissage, apprentissage interactif, apprentissage par l'action.

---

* article accepté en poster au Colloque jeunes chercheurs en sciences cognitives, La Motte d'Aveillans (Isère), France, 23, 24 et 25 mars 1994

PREMIERE PARTIE

I Introduction

Cet article a pour objectif de montrer l'importance de la composante "stratégique", dans les réseaux connexionnistes. Plus loin, nous définirons précisément ce que nous appelons stratégie d'apprentissage ; en simplifiant, nous regroupons sous ce terme toutes les techniques qui visent à ajouter un niveau supplémentaire dans le mécanisme d'apprentissage, afin de modifier de manière appropriée son déroulement.

Dans cette étude, nous nous intéressons aux réseaux qui ont pour vocation de représenter l'environnement, et plus précisément d'en construire une "copie" interne, sur laquelle pourra s'appliquer une "action intériorisée", choisie parmi un ensemble d'actions prédéfini.

En parcourant les différents écrits en connexionnisme, il est surprenant de constater que la facette "stratégique" est quelque peu mise de côté. Par exemple, la technique classique d'un apprentissage d'une base d'exemples en classification consiste à présenter dans un ordre séquentiel, éventuellement aléatoire, les exemples au réseau, qui doit modifier ses poids pour intégrer ces exemples. Le schéma d'apprentissage est alors de type "stimulus-réponse". Certains auteurs, conscients de la limite d'une telle approche, proposent des techniques pour exploiter *l'ordre* de présentation des exemples, afin d'améliorer les performances [Le Cun, 87][Haffner *et al.*, 88]. Cependant, la stratégie d'apprentissage reste souvent reléguée au rang d'une "technique" pour améliorer un algorithme, ce à quoi elle parvient d'ailleurs, au même titre que l'optimisation du pas d'apprentissage. Or, selon nous, la stratégie d'apprentissage est aussi importante que l'algorithme lui-même. En effet, notre objectif à moyen terme est de montrer que certains apprentissages ne peuvent réussir **que** si ils ont recours à une stratégie d'apprentissage. Cette approche est contraire à la méthode habituellement utilisée : les chercheurs ont tendance, afin de présenter des résultats positifs, à choisir des situations adaptées au réseau, c'est-à-dire où le réseau parvient à peu près à apprendre, pour ensuite appliquer diverses améliorations, dans le but d'optimiser différents critères (vitesse, stockage, précision...). Or nous préférons envisager le cas où le réseau échoue totalement, et utiliser une stratégie d'apprentissage pour parvenir à un résultat positif.

Dans cette première partie, après avoir caractérisé et défini le concept de stratégie d'apprentissage (section II), nous allons proposer des principes pour construire un "système stratégique" (section III), et enfin proposer quelques algorithmes permettant d'envisager une implémentation de ces principes (section IV).

II Caractérisation de la stratégie d'apprentissage

Nous allons partir des réseaux basés sur le principe de descente de gradient[1], et plus précisément de ses difficultés, pour saisir le rôle que peut avoir la stratégie d'apprentissage.

On recense en effet deux grandes difficultés, encore non résolues, quand on observe l'apprentissage dans ces réseaux supervisés.

- Difficulté à gérer un grand nombre de poids : bien qu'un point fort du connexionnisme soit sa capacité à traiter les dynamiques d'un grand nombre d'unités simples, il faut constater que les implémentations actuelles se limitent souvent, en ordre de grandeur, à 10, voire 100 neurones. Ceci n'est pas dû à des contraintes d'implémentations : les ordinateurs séquentiels actuels sont bien assez puissants pour simuler 100000 neurones. En fait, dans la pratique, il s'avère que si l'on double la taille d'un réseau, l'apprentissage peut devenir très lent et a beaucoup plus de chances d'être de moins bonne qualité en terme de généralisation [Kämmerer & Küpper, 89]. Ceci amène les concepteurs de programmes "neuronaux" à optimiser le nombre d'automates, c'est-à-dire trouver le nombre minimal d'automates permettant d'apprendre les données. Cette tendance peut être interprétée comme une sérieuse atteinte aux notions de redondance et de robustesse, chères au connexionnisme, voire même au principe de distribution de l'information : dans un tel réseau, "optimisé", chaque automate joue le rôle d'un détecteur de traits spécifique, rôle que nous ne connaissons pas, mais qui est localisé dans cet automate et dans celui-là seul. Les réseaux supervisés, tels qu'ils sont souvent implémentés, évitent donc

[1] dans des structures unidirectionnelles [Le Cun, 85] ou récurrentes [William & Zipser, 89]

le problème du grand nombre de poids. Plus précisément, on peut interpréter la difficulté à gérer un grand nombre de poids comme un phénomène de "cible mobile" [Fahlman & Lebiere, 91] : chaque unité du réseau tente de se constituer en un certain détecteur de trait, alors que le signal d'erreur qui lui est rétropropagé est sans cesse variable (apprentissage dit stochastique, ou on modifie le réseau après chaque exemple) ; ainsi, l'unité oscille entre différents états avant de trouver son "rôle". Ce même comportement est décrit dans [Jacobs *et al.*, 91] en tant que "temporal crosstalk". Ce phénomène, considéré par Fahlman et Lebiere [91] comme un facteur de ralentissement de l'apprentissage, justifie certainement les difficultés rencontrées par les "grands" réseaux. Les solutions à ce problème que l'on trouve dans la littérature sont de trois types :

• concevoir une architecture modulaire [Jacobs *et al.*, 91] : cela amène évidemment des difficultés pour choisir la taille des modules, leur articulation, ainsi que le moyen de faire apprendre ces modules.

• utiliser des réseaux incrémentaux qui "gèlent" les poids [Fahlman & Lebiere, 91], [Ayesteran & Prager, 93] : le réseau s'agrandit au cours de l'apprentissage, en fixant définitivement les poids appris à chaque étape : ainsi, le réseau adapte peu de poids à la fois. Ce type de réseaux, en gelant ses poids, présente néanmoins le défaut de principe de ne pas autoriser de restructurations profondes au cours du temps.

• utiliser des mécanismes de sélection de l'attention, qui consistent à focaliser l'apprentissage sur une zone limitée du réseau, à chaque instant, afin de limiter la complexité de l'apprentissage à chaque étape [Mjolsness, 87].

- Difficulté à apprendre des données complexes et structurées : Fodor avait affirmé que les réseaux connexionnistes étaient incapables de saisir les structures d'un environnement. Depuis, on a prouvé que ces critiques étaient injustifiées [Memmi, 93], grâce à l'étude des capacités de certains réseaux récurrents en inférence grammaticale [Elman, 90]. Cependant, il faut remarquer que ces performances restent limitées, et qu'il demeure difficile de faire converger un réseau confronté à des données trop complexes [Cleermans *et al.,* 89] [Smith & Zipser, 89] [Giles *et al.*, 92]. Les solutions apportées sont de plusieurs types :

• apprentissage progressif des données, qui permet de ne pas livrer au réseau, d'un coup, toute la complexité des données, mais lui fournir, progressivement, des données simples au départ, puis de plus en plus complexes [Elman, 91][Cloete & Ludik, 93]. Ce genre de technique nécessite de connaître *a priori* quelles données sont faciles, lesquelles sont difficiles. L'apprentissage progressif fait référence à tout ce que l'on pourrait regrouper sous le terme "d'apprentissage par l'action", en ce sens que le réseau n'est plus un simple récepteur d'informations, mais qu'il émet des informations vers l'environnement (ici en sélectionnant des données de plus en plus complexes).

• apprentissage à précision croissante, qui consiste à commencer à apprendre grossièrement les données, puis de plus en plus précisément, jusqu'à la précision requise [Smith & Zipser, 89][Allred & Kelly, 89].

A la lecture de ces difficultés et de leurs remèdes, il apparaît clairement, et cela constitue le point central du présent article, que les deux problèmes sont étroitement liés l'un à l'autre. D'un côté, on constate en effet, les difficultés d'un réseau à se structurer de manière complexe, et de l'autre, ses difficultés à appréhender un environnement complexe. Nous supposons ici que traiter l'un des problèmes sans l'autre risque de conduire à des solutions limitées. Ce n'est qu'en considérant les deux difficultés ci-dessus décrites comme relevant d'un même problème, celui finalement de la *structure*, que nous espérons trouver des algorithmes à la fois simples et puissants. Il est à remarquer que très peu d'études s'intéressent à la fois aux aspects "internes" (problème des grands réseaux) et "externes" (données complexes) de l'apprentissage connexionniste [Fahlman & Lebiere, 91] [Elman, 91].

Nous sommes maintenant en mesure de définir avec précision ce que nous appellerons *stratégie* dans le reste de l'article : "étant donné un algorithme d'adaptation, capable de constituer une représentation à partir des données reçues, la *stratégie* consiste à contrôler le déroulement de l'apprentissage pour augmenter ses performances, en agissant *sur le réseau* (sur le déroulement de l'algorithme) et/ou *sur l'environnement* (sur les données qui en sont issues)". A partir de cette définition assez vaste, la stratégie peut prendre des formes multiples, que nous tentons de résumer par un schéma (fig. 1), qui loin de constituer *le* schéma de l'apprentissage stratégique, peut servir de point de repère.

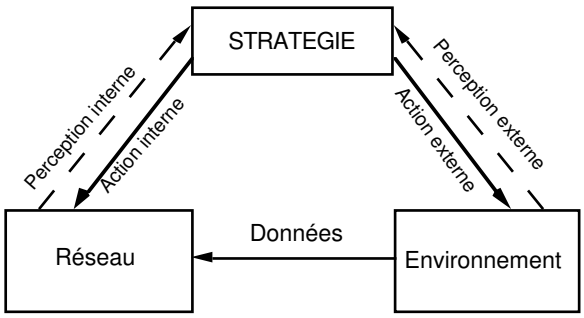

Figure 1 : apprentissage stratégique

Il y donc deux facettes dans la stratégie : la stratégie *interne*, et la stratégie *externe*.

Dans ce qui suit, nous exposons quelques principes qui pourraient gouverner un tel apprentissage stratégique.

III Principes pour construire un système "stratégique"

III.1 Aspect dynamique de la stratégie

Telle que nous l'avons définie ci-dessus, la stratégie consiste à contrôler le déroulement d'un apprentissage. *A chaque instant*, la "composante stratégique" prend en compte l'état actuel du réseau ainsi que l'environnement pour agir, soit vers l'intérieur, soit vers l'extérieur. Par conséquent, *la dimension temporelle est inhérente à un apprentissage stratégique.* L'apprentissage ne se limite pas ici à "réaliser un mapping", "minimiser un critère d'erreur", mais s'inscrit dans une *dynamique temporelle*, dont l'objectif est de réaliser un mapping, minimiser une erreur...

En plus de la *dynamique d'adaptation* (dynamique des poids du réseau), la stratégie gère en effet deux dynamiques, que nous appelons stratégiques, concernant :

- le réseau : il ne s'agit pas de la dynamique des activités ou des poids, mais celle de paramètres influant sur ces deux dynamiques : plasticité des "neurones" (mécanismes de focalisation), connectivité (mécanismes de suppression de connections), nombre de neurones (mécanismes de croissance, d'élagage)... Nous l'appelons la *dynamique de la structure*.

- les données issues de l'environnement : les informations en provenance de l'environnement et qui parviennent au réseau ne témoignent que d'une partie de cet environnement ; les apprendre revient à apprendre une *sous-partie de l'environnement*. On peut sélectionner une sous-partie de l'environnement en agissant directement sur l'environnement (apprentissage par l'action) ou en filtrant les données qui en sont issues (mécanismes "top-down").

Si ces trois dynamiques évoluent avec la même "rapidité", on peut penser, de manière intuitive, que l'apprentissage aura des difficultés à se stabiliser. En effet, dans ce cas, alors même que la dynamique d'adaptation tentera d'identifier une sous-partie de l'environnement, cette sous-partie évoluera, ainsi que la structure chargée de représenter cette sous-partie : la dynamique d'adaptation ne pourra se stabiliser. Pour qu'a chaque instant, cette dynamique tende vers un objectif stable, il est nécessaire que les deux autres dynamiques (stratégiques) soient dans une certaine mesure gelées, en attendant cette stabilisation. Ainsi, la dynamique stratégique n'évoluerait que lorsque la dynamique d'adaptation est stabilisée. Cela revient à dire que l'apprentissage se décompose en *phases*, pendant chacune desquelles la sous-partie de l'environnement et la structure du réseau sont fixes. Une phase est définie comme "le triplet composé d'une *configuration* (ce qui n'est ni poids ni activité) du réseau, d'une sous-partie de l'environnement et d'une durée d'apprentissage". Plusieurs phases se succèdent pour constituer un apprentissage entier, et le rôle de la stratégie d'apprentissage est de déterminer ces phases, ainsi que leur enchaînement.

III.2) Un apprentissage progressif

Face aux difficultés des réseaux qui ont été décrites dans la section II, nous répondons par un postulat, c'est-à-dire par une affirmation non démontrée qui nous permettra de déduire les grandes lignes d'un "système stratégique".

En effet, étant données les difficultés à gérer aussi bien un grand nombre de paramètres que des données complexes, il nous semble intéressant de poser que : "la durée d'une phase doit être limitée". Pour une plasticité donnée du système (sa "vitesse d'adaptation"), ceci revient à *limiter la longueur de la trajectoire dans l'espace des poids, pendant une phase d'apprentissage*. De manière imagée, ce postulat impose au réseau de "limiter son effort" au cours de chaque phase d'apprentissage. Pour limiter la trajectoire dans l'espace des poids, on pourra par exemple limiter le *nombre* de paramètres adaptables au cours d'une phase donnée ou bien limiter les *amplitudes maximales* de chaque paramètre. Ainsi dispose-t-on d'un premier critère pour définir la stratégie "interne".

Quelles conséquences en tire-t-on pour la stratégie "externe", celle qui agit sur l'environnement ? Il est clair que si le réseau ne peut faire qu'un "faible effort" à chaque phase (rigoureusement un faible déplacement dans l'espace des poids), alors, d'une phase à l'autre, le changement de sous-partie de l'environnement doit être tel que les nouvelles données à apprendre soient assimilables par le réseau "bridé". En effet, dans le cas contraire, le réseau ne réussira pas à apprendre les données avant la fin de chaque phase, puisque la trajectoire dans l'espace des poids est limitée. La stratégie externe consistera donc, dans la mesure ou cela est possible, à découper l'ensemble des données à apprendre en sous-parties "aisément assimilables" successivement présentées au réseau, à chaque phase ; c'est ce que l'on peut définir comme l'apprentissage progressif.[2] Dans la mesure où, à la fin, le réseau doit avoir appris l'ensemble des données, une solution consiste à partir d'une première sous-partie de l'environnement simple, de la compléter en la complexifiant progressivement, jusqu'à atteindre l'environnement en entier. Ainsi, la sous-partie de l'environnement correspondant à une phase p est un sur-ensemble de celle correspondant à la phase p-1. De ce fait, on appelle cette stratégie l'apprentissage progressif additif[3]. Nous disposons maintenant d'un critère pour la stratégie "externe".

Pour résumer, nous dirons que le principe d'apprentissage "stratégique" que nous proposons consiste à limiter l'adaptation du réseau, et en face la "difficulté" des données à apprendre à chaque instant, dans le but de garantir le succès de l'apprentissage à chaque étape, et donc de l'apprentissage global. C'est donc paradoxalement en "bridant" le réseau que nous pensons parvenir à des performances plus élevées.

IV Quelques algorithmes

Pour élaborer un algorithme stratégique, il nous reste à savoir limiter la trajectoire des poids à chaque phase et à découper l'ensemble d'apprentissage (l'environnement). Sur ces deux aspects, l'un interne, l'autre externe, il faut gérer le *"comment"*, c'est-à-dire la manière de limiter la trajectoire des poids et découper l'environnement, et le *"combien"*, c'est à dire les valeurs des limitations et découpages en question. Selon les situations, on peut connaître ou non **a priori** le *"comment"* et/ou le *"combien"* des composantes internes et/ou externes de l'apprentissage stratégique. Cela nous conduit à de nombreux cas à étudier ; dans ce qui suit, nous exposons les quatre cas les plus caractéristiques, qui s'enchaînent (progressivement...) du plus simple au plus évolué.

IV.1 Limitations et découpages sont entièrement connus a priori

Nous nous plaçons dans le cas où l'on connaît a priori les différentes phases de l'apprentissage stratégique. L'algorithme n'a plus qu'à gérer le passage d'une phase à l'autre :

```
Début
   Pour i allant de 1 à n faire
   début
      sélectionner la ième configuration du réseau
      sélectionner la ième sous-partie de
         l'environnement
      répéter
         effectuer l'apprentissage
      jusqu'à stabilisation
   fin
Fin
```

On suppose donc les phases prédéfinies telles que la stabilisation s'effectue rapidement (postulat de la partie III.1).

Pour établir a priori les sous-parties successives de l'environnement, on peut, s'il s'agit de l'apprentissage d'une base d'exemples, diviser cette base en *n* sous-ensembles et constituer ainsi des sous-parties de l'environnement comprenant un, puis deux, trois... et enfin *n* de ces sous-ensembles. (Certains découpages sont plus élaborés...)

Pour établir a priori les configurations successives du réseau, on peut par exemple considérer que l'apprentissage final nécessite $N=n.a$ automates et utiliser une architecture incrémentale démarrant avec *a* automates, puis passant à $2.a$ automates... pour arriver à $n.a$ automates. Notons que de nombreuses variantes sont encore à explorer pour limiter la trajectoire dans l'espace des poids entre deux stabilisations.

---

[2] Une variante consiste à jouer sur la précision requise : au fur et à mesure de l'apprentissage, on exige une précision de plus en plus élevée [Allred & Kelly, 89].

[3] Il s'agit ici d'un choix : d'autres possibilités existent, et sont même peut-être nécessaires pour complexifier progressivement les données.

IV.2 Sélection active dans l'environnement

L'algorithme précédent ne s'applique que dans des situations très circonscrites, où l'on en connaît déjà beaucoup sur l'environnement et la manière de le représenter. L'algorithme que nous allons présenter maintenant suppose toujours que l'on fixe a priori comment et de combien limiter "l'effort" du réseau à chaque phase, et que l'on connaisse a priori *comment* découper l'environnement en sous-parties successivement assimilables par le réseau. Mais on suppose que l'on ignore *à quel rythme* présenter au réseau ces sous-parties.

Le fait que l'on connaisse comment découper l'environnement suppose que l'on dispose d'un critère de *complexité* des sous-parties de l'environnement. Cette complexité augmente au fur et à mesure que le réseau assimile l'environnement. Si la complexité de l'environnement en entier est C, la particularité de l'algorithme qui suit est de gérer la complexité c_i de la sous-partie de l'environnement associée à la phase i, c_i variant de 0 à C. Par exemple, si il s'agit d'apprendre à compter, la complexité pourra être le nombre jusqu'auquel il faut savoir compter, à une étape donnée de l'apprentissage ; dans le cas où il faut apprendre à reconnaitre les lettres de l'alphabet, la complexité pourra être le nombre de lettres à apprendre : 1, 2, puis 3, ... jusqu'à 26... Partant d'un c_1 arbitraire, le système doit choisir entre augmenter c_1 (car il lui faut aboutir à la complexité $C > c_1$ à la fin de l'apprentissage), et diminuer c_1, pour faciliter l'apprentissage.

Nous proposons donc l'algorithme ci-contre. Comme on peut le remarquer, le coefficient Δc représente l'augmentation de la complexité de l'environnement de l'apprentissage à chaque phase. A chaque configuration du réseau, l'algorithme trouve l'environnement le plus difficile qu'il peut assimiler (soit en partant d'un environnement trop difficile, soit le contraire), avant de passer à la configuration suivante (phase suivante).

Ce type de sélection active pourrait s'appliquer au problème de l'inférence grammaticale. Elman [91] montre l'importance de commencer par un environnement simple, puis de le complexifier progressivement. Pour cela, il joue sur la longueur des phrases présentées au réseau. Son algorithme s'apparente plutôt au cas de la partie IV.1, puisqu'il fixe a priori les longueurs de phrases pour chaque phase (mais le réseau, lui, n'évolue pas en même temps). Cependant, on peut aisément adapter l'algorithme de sélection active ci-dessus à ce problème de l'inférence grammaticale, tout simplement en considérant c_i comme la longueur des phrases présentées au réseau. Ainsi, l'algorithme gère automatiquement son rythme d'apprentissage, ralentit quand il a des difficultés, accélère dans le cas contraire.

```
Début
  c0 <- 0
  i <- 1
  initialiser Δc à une faible valeur
  répéter
  début
      ci = ci-1 + Δc
      sélectionner la sous-partie de
          l'environnement selon ci
      sélectionner la ième configuration du réseau
      répéter
          effectuer l'apprentissage
      jusqu'à fin de la phase
      si (apprentissage réussi  et Δc diminue)
       ou (apprentissage échoué et Δc augmente)
        ou (c >= C)
          i <- i+1
      sinon
      début
          si apprentissage réussi
              augmenter Δc
          si apprentissage échoué
              diminuer Δc
      fin
      sinon diminuer Δc
  fin
  jusqu'à i>n
Fin
```

IV.3 Sélection et focalisation actives

Comme pour la composante externe de l'action stratégique, nous nous proposons ici de gérer en plus l'aspect quantitatif de la composante *interne*. L'algorithme devrait être assez similaire au cas de la sélection active seule, si ce n'est qu'il nous faut résoudre le dilemme suivant : dans le cas où l'apprentissage a échoué, faut-il attribuer cet échec à des données trop complexes, ou à un effort trop important demandé au réseau ?

Dans ce qui précède, d'un côté nous avons défini une phase comme l'intervalle entre deux états stabilisés, et de l'autre nous avons limité sa longueur. Cela nous amène, dans le cas où l'apprentissage a échoué, à distinguer deux cas :

- la dynamique des poids s'est stabilisée (dans une mauvaise solution)

- la dynamique des poids ne s'est pas stabilisée.

Nous considérons, dans le premier cas, que l'échec est dû à des données issues de l'environnement trop complexes, et dans le deuxième cas, qu'il est dû à une trajectoire dans l'espace des poids pas assez limitée. Nous en déduisons l'algorithme suivant :

```
Début
  a  <- 0 (a, amplitude de la trajectoire dans
          l'espace des poids)
  c  <- 0 (c, complexité de la sous-partie de
          l'environnement)
  initialiser Δa et Δc
  répéter
      a  <- a + Δa
      c  <- c + Δc
      sélectionner la configuration du réseau
         selon a
      sélectionner la sous-partie de
         l'environnement selon c
      répéter
         effectuer l'adaptation
            (modification des poids)
      jusqu'à fin de la phase
      si apprentissage réussi
         augmenter Δa et Δc
      sinon
      début
         a <- a - Δa
         c <- c - Δc
         si dynamique des poids stabilisée
            diminuer Δc
         sinon
            diminuer Δa
      fin
  fin
  jusqu'à c >= C (C, complexité de
      l'environnement  entier).
Fin
```

IV.4 Recherche de l'action "stratégique"

On suppose ici, comme dans ce qui précède, que la hiérarchie de l'action interne est connue a priori (par exemple, que plus il y de neurones, plus l'espace des poids est grand). On manipule donc, comme précédemment, l'amplitude a, ainsi que son incrément Δa.

Restant dans l'optique d'un apprentissage progressif additif (voir III.2), il nous faut d'un côté augmenter progressivement l'amplitude de la trajectoire dans l'espace des poids, ce qui se réduit à augmenter a, et de l'autre la complexité de la sous-partie de l'environnement sélectionnée. Cette complexité de la sous-partie de l'environnement est la conséquence directe de l'ensemble des actions appliquées à l'environnement lors d'une phase donnée ; nous parlerons donc aussi, par extension, de la complexité d'un ensemble d'actions. Dans la cas présent, on ignore a priori la complexité de tel ou tel ensemble d'actions.

Il faut donc réaliser une *association* (un "mapping") entre l'amplitude a et l'ensemble des actions appliquées à l'environnement lors d'une phase donnée, cette association vérifiant la condition "plus a augmente, plus la complexité de l'ensemble d'actions associé augmente". Ce mapping doit se réaliser par apprentissage, sous forme d'essais successifs d'actions sur l'environnement (tâtonnement).

Nous proposons donc la structure et l'algorithme qui suivent, sachant qu'au stade actuel de l'étude, il s'agit plus d'une ébauche que d'un résultat définitif.

Considérons que l'amplitude a est codée sur une première couche de neurones d'un réseau connexionniste, et que l'ensemble des actions qui peuvent être émises vers l'environnement à un certain instant (correspondant à une sous-partie de l'environnement) est codé sur une deuxième couche. L'action effectivement émise vers l'environnement est codé sur une troisième couche (fig. 2).

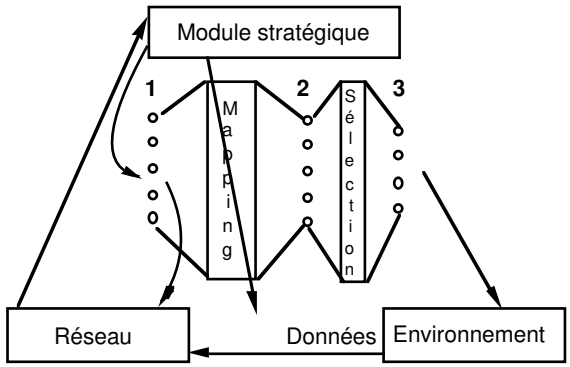

Figure 2 : l'action pour apprendre

Entre la première et la deuxième couche , on réalise le mapping évoqué ci-dessus, à l'aide, par exemple, d'un réseau à couche à rétropropagation de gradient[4].

Entre la deuxième et la troisième couche, il y a simplement sélection d'un action au hasard parmi l'ensemble d'actions codé dans la deuxième couche.

[4] Ici, ce n'est pas l'algorithme (de rétropropagation) qui compte, mais la structure dans laquelle il s'insère...

La première couche (codage de *a*) s'applique également sur le réseau initial, pour réaliser le mécanisme de limitation de la trajectoire dans l'espace des poids.

Un "module stratégique" gère *a* et l'apprentissage du mapping entre la première et la deuxième couche, en fonction de l'état du réseau et de l'ensemble prédéfini de toutes les actions à apprendre. C'est dans ce module qu'est implémenté l'algorithme ci-contre.

Si l'apprentissage a échoué à cause d'un ensemble d'actions trop complexe, le mapping est modifié de manière à associer à cet ensemble une amplitude *a* plus élevée, correpondant à une phase ultérieure.

Notons qu'une fois le mapping réalisé, la deuxième couche du réseau code l'équivalent du coefficient *c* dans les algorithmes précédents.

La particularité de cet algorithme est donc de réaliser deux apprentissages simultanément, le deuxième servant le premier :

- l'apprentissage de la réaction de l'environnement à un ensemble d'actions (comme tout algorithme d'apprentissage supervisé par l'environnement),

- l'apprentissage d'une hiérarchie de complexités des actions à apprendre.

```
Début
  a <- 0
  initialiser Δa
  répéter
  début
      a <- a + Δa
      répéter
          effectuer l'apprentissage
      jusqu'à fin de la phase
      si apprentissage réussi
          augmenter Δa
      sinon
      début
          a <- a - Δa
          si dynamique des poids stabilisée
              | modifier le mapping en faisant
              | apprendre l'association entre une
              | valeur de l'amplitude plus élevée que
              | a (valeur maximale par exemple) et
              | l'action considérée.
          sinon
              diminuer Δa
      fin
  fin
  jusqu'à toutes les actions apprises
Fin
```

DEUXIEME PARTIE

I Statut méthodologique de l'étude

Bien souvent, les travaux relatifs aux sciences de la cognition présentent le défaut de ne pas *se situer* par rapport aux autres disciplines et approches, conduisant à une science cognitive encore bien trop plurielle... Nous allons tenter ici au contraire de nous positionner.

Cette étude se situe dans l'une des disciplines de la cognition artificielle : les Réseaux de Neurones Formels. Néanmoins, alors qu'une caractéristique importante des "sciences de la cognition artificielle" est d'utiliser des *formalismes* de description, c'est par une démarche plutôt intuitive que nous avons pu proposer un modèle, sans l'appuyer sur une analyse mathématique. D'un autre côté, ce modèle n'est pas non plus directement issu de données humaines, quoiqu'il s'en soit inspiré. Quelle est donc le statut d'une telle étude ? Il est très simple : les Sciences Cognitives ont actuellement besoin de modèles généraux, et nous proposons justement une modélisation d'un système cognitif capable de représenter son environnement par interaction avec celui-ci. Au stade actuel de l'étude, le modèle est hypothétique, et il convient de le confronter à des données expérimentales.

D'un côté, il faut par conséquent implémenter les algorithmes et tester leur efficacité. Mais de l'autre, il convient de valider le modèle sur d'autres formes de la cognition, et en particulier la cognition humaine. Si le fait d'appliquer un modèle "artificiel" sur un système naturel choque, il faut réaliser que tout modèle est en fin de compte artificiel... De plus, c'est le propre des Sciences Cognitives de croiser les modèles, les idées, les thématiques...

La question qui se pose est donc : "le modèle de stratégie d'apprentissage de la première partie est-il un bon modèle (partiel) pour la cognition en général, et en particulier la cognition humaine ?". Nous ne pouvons répondre directement à cette question, mais pouvons chercher quelques indices dans certains travaux antérieurs.

II Psychologie de l'apprentissage...

Depuis une vingtaine d'années, la psychologie cognitive délaisse quelque peu l'étude de l'apprentissage, pour s'intéresser à des sujets, non moins intéressants, tels que la perception, la mémoire, le raisonnement... Elle a suivi en cela le mouvement de l'I.A. classique (purement déductive), ainsi qu'une conception nativiste des capacité humaines [Bates & Elman, 92]. Pour cette raison, il faut revenir longtemps en arrière pour trouver des études théoriques sur le sujet. En particulier, les travaux de Piaget restent la référence dans le domaine, en ce qu'ils présentent une théorie globale de l'apprentissage et du développement de l'intelligence [Piaget 47]. Malgré ses défauts, à savoir sa difficulté d'accès, son manque de formalisation, son manque d'expérimentations rigoureuses et ses légères erreurs, défauts qui ont parfois conduit à son rejet, la théorie piagétienne, et les courants qui l'ont suivie, restent une référence incontournable dès que l'on s'intéresse à l'apprentissage et au développement.

Une première caractéristique de cette théorie est de considérer que le développement de l'intelligence s'effectue selon une succession d'équilibres et de déséquilibres : à un stade de développement donné s'établit un certain équilibre ; le système cognitif est ensuite perturbé par l'acquisition de nouvelles aptitudes (schèmes), et s'achemine vers un nouvel équilibre incluant ces schèmes.

Ceci s'avère analogue au découpage en phases décrit dans la première partie : au passage d'une phase à l'autre, le système se déséquilibre, puis rétablit son équilibre pendant le déroulement de la phase (accommodation).

En postulant dans la section II.2 de la première partie que la durée d'une phase doit être limitée, nous rejoignons le fait que, chez l'homme, les périodes de déséquilibre (accommodation) sont relativement brèves, condition essentielle pour que l'enfant puisse utiliser ses connaissances (ses schèmes), tout en s'intégrant dans un processus d'évolution.

Ainsi, les algorithmes de la première partie constituent peut-être les prémisses d'une modélisation efficace de l'apprentissage humain, insistant sur l'importance de l'interaction entre le sujet apprenant et son environnement et en accord avec les idées piagétiennes.

Mais les liens entre notre modèle et la théorie piagétienne sont encore ténus, et nous souhaiterions les approfondir...

Actuellement, sur un plan plus appliqué, ce sont surtout les chercheurs en didactique et en E.I.A.O.[5] qui d'une part utilisent les bases piagétiennes et d'autre part mettent en valeur deux caractéristiques fondamentales de l'apprentissage sur lesquelles nous avons insisté dans la première partie : l'interaction et la progressivité. Bien que ces disciplines concernent l'acquisition de connaissances conceptuelles, donc symboliques, le modèle de stratégie d'apprentissage présenté ici nous semble pouvoir y trouver des applications.

D'une part, les algorithmes présentés sont comparables à des sessions d'apprentissage interactif, si l'on remplace la stratégie interne par l'élève et la stratégie externe par le tuteur informatique "intelligent" : l'algorithme de la section IV.1 constitue une session entièrement fixée a priori et celui de la section IV.2 constitue un apprentissage à vitesse variable selon les performances de l'élève.

D'autre part, l'idée de "brider" le réseau, c'est-à-dire de limiter son amplitude d'adaptation, rejoint le concept de maturation chez l'homme : les structures cognitives de l'apprenant ne sont pas les mêmes chez l'enfant que chez l'adulte, mais elles évoluent [Elman 91][Marchman 92].

De manière plus générale, nous souhaiterions approfondir les liens entre notre modèle et les formes plus symboliques d'apprentissage, qu'il s'agisse de disciplines appliquées (E.I.A.O., didactique) ou théoriques (théories de l'apprentissage en informatique théorique [Boucheron, 92]), et distinguer par là même ce qui est général à toute forme d'apprentissage, de ce qui est spécifique à l'apprentissage de concepts.

Enfin, le dernier algorithme présenté, basé sur l'apprentissage d'une hiérarchie de difficulté des données à apprendre, pourrait être la base d'un *modèle de l'apprenant*. Celui-ci prendrait en compte le fait que la représentation de l'environnement s'accompagne d'une représentation de l'action vers cet environnement, comme le montrent certains travaux en psychologie cognitive [Viviani & Stucchi, 91].

---

[5] Enseignement Intelligent (ou Interactif) Assisté par Ordinateur

Conclusion

Nous avons présenté la nécessité puis les principes de l'utilisation de la stratégie d'apprentissage dans les réseaux connexionnistes, pour finir par quelques propositions d'algorithmes.

Cependant, le modèle présenté n'est pas spécifique aux réseaux de neurones. Nous estimons qu'il s'applique à d'autres formes d'apprentissage en cognition artificielle : apprentissage symbolique, robotique "sensori-motrice", algorithmes génétiques... Il conviendrait maintenant de rapprocher la présente étude des travaux portant sur l'apprentissage actif dans ces différentes disciplines.

De plus, nous constatons de fortes parentés entre notre modèle et certains aspects du développement de l'intelligence humaine, et nous souhaiterions approfondir ce lien.

Comme nous l'avons déjà mentionné, cette étude en est encore au stade d'ébauche, et bien des étapes sont encore nécessaires avant son aboutissement : épuration du cadre théorique, formalisation du problème, mise en pratique des algorithmes (expérimentations) et étude de la validité du modèle pour d'autres formes (naturelles et artificielles) de cognition que les réseaux de neurones artificiels. Ceci nous amènera à revoir tout ou partie de ces premiers travaux.

Bibliographie

[Allred & Kelly, 89] Lloyd Allred, Gary E. Kelly. Supervised Learning Techniques for Backpropagation Networks. IJCNN, I, pp 721-728, 1989.

[Ayesteran & Prager, 93] The logical gates growing network. Tech. report CUED/ F_INFENG/TR 137, July 1993.

[Bates & Elman, 92] Elisabeth A. Bates, Jeffrey L. Elman. Connectionism and the study of change. CRL Tech. Report 9202, Feb. 92.

[Boucheron, 92] Stéphane Boucheron. Théorie de l'apprentissage - de l'approche formelle aux enjeux cognitifs. Hermès, Paris, 1992.

[Cleeremans *et al.*, 89] Axel Cleeremans, David Servan Schreiber, James L. McClelland. Finite State Automata and Simple Recurrent Networks. *Neural Computation*, **1**, pp 372-381, 1989.

[Cloete & Ludik, 93] Ian Cloete, Jacques Ludik. Increased Complexity Training. IWANN'93, Sitges, Spain, June 1993.

[Elman, 90] Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, **14**, pp 179-211, 1990.

[Elman, 91] Jeffrey L. Elman. Incremental learning, or the importance of starting small. CRL Tech. report 9101, Mars 1991.

[Fahlman & Lebiere, 91] Scott E. Fahlman, Christian Lebiere. The Cascade-Correlation Learning Architecture. Tech. Report, CMU-CS-90-100, August 1991.

[Giles *et al.*, 92] C. L. Giles, C. B. Miller, D. Chen, H. H. Chen, G. Z. Sun, Y. C. Lee. Learning and Extracting Finite State Automata with Second Order Reccurrent Neural Networks. *Neural Computation*, **4**, pp 393-405, 1992.

[Haffner *et al.*, 88] P. Haffner, A. Waibel, H. sawai, K. shikano. Fast Back-Propagation Learning Methods for Neural Networks in Speech. ATR Tech. Report TR-I-0058, 1988.

[Jacobs *et al.*, 91] Robert A. Jacobs, Michael I. Jordan, Andrew G. Barto. Task Decomposition Competition in a Modular Connectionist Architecture: The What and Where Vision Tasks. *Cognitive Science*, **15**, pp 219-250, April 1991.

[Kämmerer & Küpper, 89] Bernhard, R. Kämmerer, Wolfang A. Küpper. Design of hierarchical structures and their application to the task of isolated-word recognition. IJCNN, 1989.

[Le cun, 87] Yann Le Cun. Modèles connexionistes de l'apprentissage. thèse de doctorat de l'université Paris VI, 1987.

[Marchman, 92] Virginia A. Marchman. Language learning in children and neural networks: plasticity, capacity and the critical period. CRL Tech. report 9201.

[Mjolsness, 87] Eric Mjolsness. Control of Attention in Neural Networks. IJCNN, 1987.

[Memmi, 93] Daniel Memmi. Representations in connectionist networks. conf. Neuronîmes, October 1993.

[Piaget, 47] Jean Piaget. Psychologie de l'intelligence. Armand Colin Editeur, 1947.

[Smith & Zipser, 89] Anthony W. Smith & David Zipser. Encoding sequential structure: experience with the real-time recurrent learning algorithm. IJCNN, I, pp 645-648, 1989.

[Viviani & Stucchi, 91] Paolo Viviani, Natale Stucchi. Biological Movement Look Uniform: Evidence of Motor-perceptual Interactions. *Journal of experimental Psychology: Human Perception and Performance,* 1991.

[William & Zipser, 89] Ronald D. Williams & David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, **1**, pp 270-280, 1989.