

# Conclusion et perspectives

## Parallélisme et connexionnisme

Paralléliser une application est un travail délicat. Une bonne maîtrise des outils informatiques et de la programmation parallèle ne suffit pas. Pour atteindre une bonne accélération, il est nécessaire de connaître parfaitement l'algorithmique de l'application. Il en est de même pour la parallélisation d'un modèle connexionniste. Mais, alors que la tâche d'une application informatique classique est l'exécution d'une suite d'opérations précise, un réseau de neurones doit apprendre. Peu importent les valeurs exactes des poids associés aux connexions si elles donnent au réseau une bonne modélisation du problème à résoudre. Or, il existe de nombreuses manières d'apprendre, et une stratégie parallèle a toutes les raisons d'être au moins aussi bonne qu'une approche purement séquentielle.

## Parallélisme et classification incrémentale

Le classifieur incrémental a été un modèle intéressant à paralléliser, sa topologie dynamique devant être prise en compte pour ne pas rencontrer des problèmes d'équilibrage de la charge de travail des processeurs.

Nous avons étudié deux approches pour la parallélisation du classifieur incrémental, la première respecte parfaitement la dynamique d'apprentissage du modèle séquentiel, les poids évoluent exactement de la même manière. Le réseau parallélisé hérite de toutes les capacités du réseau séquentiel. Cependant, cette parallélisation implique de nombreuses communications de petite taille. Cette stratégie suppose donc une machine parallèle apte à communiquer rapidement (relativement à sa puissance de calcul). Par ailleurs, dans l'application d'OCR que nous avons étudiée, l'accélération maximale est limitée par le prétraitement. En effet, le réseau partageant l'espace d'entrée entre les processeurs, la dimension de cet espace limite le nombre de processeurs et donc l'accélération. Cependant, de nombreuses applications connexionnistes,

travaillant en haute dimension, peuvent faire appel au parallélisme avec succès (notamment en imagerie).

La deuxième parallélisation étudiée est modulaire, un classifieur complet est placé sur chaque processeur. Le travail est partagé entre les modules, chacun apprenant une partie des exemples. Régulièrement, les modules communiquent pour profiter de leurs progrès respectifs. Bien que ce partage du travail ne respecte pas la stratégie d'apprentissage séquentielle, les taux d'erreur en généralisation sont presque aussi bons qu'en séquentiel. Cependant, ces bons résultats sont conditionnés par des échanges soutenus entre les modules ; même sur une machine parallèle, tant de communications rendent l'apprentissage modulaire inefficace. En revanche, cette stratégie est très performante en généralisation : l'accélération est quasi-linéaire. De plus, le prétraitement est également parallélisé, sans devoir être modifié.

Afin d'améliorer le comportement de la parallélisation modulaire, nous avons fait le choix de nous éloigner davantage du fonctionnement séquentiel en spécialisant chaque module sur une des classes à apprendre. Les modules apprennent toujours sur un sous-ensemble des exemples, mais tous de la même classe. Bien sûr, le classifieur modulaire spécialisé est limité à un apprentissage supervisé. Loin d'être pénalisant, un espacement des communications conduit à un taux d'erreur en généralisation au moins aussi bon qu'en mode séquentiel. Les modules travaillent sur un sous-problème cohérent et s'échangent un travail synthétique. Bien évidemment, la diminution des communications conduit également à une accélération quasi-linéaire de l'apprentissage. Là encore, le prétraitement est également parallélisé sans aucun effort.

Cette efficacité nous a motivé pour porter le classifieur modulaire spécialisé sur une architecture distribuée (ordinateurs en réseau). Une fois encore, moyennant une prise en compte de la nature asynchrone du milieu, nous avons atteint d'excellentes performances, tout en préservant la qualité de l'apprentissage. De plus, outre le prétraitement, l'acquisition des données est parallélisée de fait. Cette version du classifieur parallèle, est potentiellement tolérante aux pannes. Enfin, elle peut également fonctionner directement sur de nombreuses machines parallèles (de type MIMD). La portabilité de cette version asynchrone vient notamment de l'implémentation d'un paramètre permettant de définir le degré de synchronisme adapté à l'architecture cible. En perspective, dans le but de faciliter la programmation de futurs développements, une version du classifieur parallèle modulaire spécialisé sur mémoire distribuée virtuellement partagée est en cours de réalisation (environnement DOSMOS [Lef97]).

## **Apports mutuels des sciences cognitives et du connexionnisme**

Au-delà de la parallélisation du modèle connexionniste, et dans le cadre d'un projet pluridisciplinaire, nous avons utilisé le classifieur comme outil dans le domaine des sciences cognitives : tout d'abord comme outil d'analyse en étudiant des données issues d'expérimentations sur la mémorisation d'odeurs, puis comme outil de modélisation en simulant un amorçage de répétition. Une de nos approches a été de demander au classifieur de discriminer les odeurs en fonction des descripteurs évoqués par les sujets. L'étude a mis en valeur l'inadéquation d'un simple jeu d'adjectifs pour caractériser une odeur, autant pour le classifieur que pour l'humain.

En dehors de toutes considérations cognitives, notre analyse des données olfactives a mis en évidence des incohérences qui rendent la base d'exemples difficile à apprendre. À la lumière de notre travail sur les odeurs, mais aussi sur la parallélisation du classifieur modulaire spécialisé, nous avons montré que trop d'incohérences, ou un chevauchement des classes, conduisent à une surabondance de prototypes qui nuit gravement aux performances en classification du modèle. Le classifieur a donc été adapté afin de pouvoir réaliser un apprentissage supervisé « cloisonné » (autant pour les versions séquentielle que parallèle). Le classifieur concentre son apprentissage sur une classe, partiellement ou totalement, afin de créer peu de prototypes.

Dans le cas des données olfactives (base incohérente), moins de prototypes perturbateurs sont créés, mais les prototypes créés sont de mauvaise qualité, les taux d'erreur restent donc bas. Pour le problème de classification des ondes de Breiman (base avec chevauchement), l'apprentissage cloisonné conduit à des prototypes en nombre suffisant. Pour ces deux applications un cloisonnement important a diminué les taux d'erreur. Dans le cadre d'une application réelle, comme la reconnaissance de formes manuscrites, il est intéressant de noter qu'un léger cloisonnement peut également améliorer la qualité de l'apprentissage.

## **Mémoire associative hétérogène**

Cette thèse s'achève sur la présentation d'un modèle connexionniste que nous proposons pour modéliser les processus de mémorisation humains. Ce modèle est une source de perspectives importantes, tant au niveau des applications potentielles que de la recherche nécessaire pour arriver à l'implémentation. Dans un premier temps, nous travaillons à la réalisation d'un classifieur parallèle partageant l'espace d'entrée en sous-espaces cohérents (modalité perceptive). Un module supplémentaire, à terme une mémoire associative, pourra intégrer la réponse de chaque classifieur pour donner une réponse globale.

Par ailleurs, cette recherche sur la spécialisation des tâches et l'intégration des réponses de modules, nous a conduit à nous intéresser à un autre modèle connexionniste, les mixtures d'experts. Une parallélisation modulaire de ces réseaux a été initiée lors d'une visite au DIM (*Departamento de Ingenieria Matematica*, Santiago du Chili) avec Estevez et a conduit à une publication.