Republique Algérienne Démocratique Et Populaire
Ministère De L'enseignement Supérieur Et De La Recherche Scientifique

Université Frères Mentouri Constantine
Faculté Des Sciences Exactes
Département De Mathématiques

Numéro d'ordre :

Numéro de série :

THÈSE

En vue de l'obtention de Grade de docteur en Mathématiques
Option : Statistique Appliquée

L'APPROCHE NEURONALE DE L'INFÉRENCE STATISTIQUE

Présentée par :
Dalel ZERDAZI

Dirigée par :
Ahmed Chibat

• Devant le jury :

Nahima Nemouchi	Prof	U. Frères Mentouri Cne	Présidente
Ahmed Chibat	MCA	U. Frères Mentouri Cne	Rapporteur
Abdelhamid Ayadi	Prof	U. Larbi Ben M'Hidi OEB	Examinateur
Ahmed Nouar	MCA	U. 20 Août 1955 Skikda	Examinateur
Fouad Lazhar Rahmani	Prof	U. Frères Mentouri Cne	Examinateur

Soutenue le : 24 / 04 / 2017

Intitulé de la thèse :

L'APPROCHE NEURONALE DE L'INFÉRENCE STATISTIQUE

Réalisée par : Dalel Zerdazi
Année : 2017

Remerciements

En tout premier lieu, je remercie Allah, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.
C'est avec une certaine émotion et beaucoup de sincérité que je voudrais remercier toutes les personnes ayant soutenu et apprécié mon travail.
À l'issue de la rédaction de cette recherche, je suis convaincue que la thèse est loin d'être un travail solitaire. En effet, je souhaite remercier mon directeur de thèse, Monsieur Chibat Ahmed, pour son soutien tout au long de ce doctorat. Je lui suis également reconnaissante pour le temps conséquent qu'il m'a accordé, ses qualités pédagogiques et scientifiques, sa franchise et sa sympathie. J'ai beaucoup appris à ses côtés.
J'exprime tous mes remerciements à l'ensemble des membres de mon jury : Madame Nemouchi Nahima et Messieurs : Rahmani Fouad Lazhar, Nouar Ahmed et Ayadi Abdelhamid.
Le seul moyen de se délivrer d'une tentation, c'est d'y céder paraît-il ! Alors j'y cède en disant un grand merci aux personnes qui ont cru en moi et qui m'ont permis d'arriver au bout de cette thèse. Je remercie mes parents ainsi que mon frère et sœurs pour leur soutien au cours de ce doctorat et sans lesquels je n'en serais pas là aujourd'hui.

À mes Parents, Frère et Sœurs,
et à tous ceux que je ne nomme pas, mais qui se reconnaîtront.

Mise en situation

L'enjeu central de la statistique est l'inférence.
L'abord répétitif de cette question se fait à travers les théories, les méthodes et les outils qui se développent dans le temps ; qui émergent et s'améliorent au fur et à mesure. Ce qui pousse toujours à revenir aux questions centrales mais sous de nouveaux éclairages apportés par les innovations successives. L'idée cruciale de l'inférence revient toujours en question.
Les réseaux de neurones artificiels, aujourd'hui admis comme extensions des outils statistiques, soulèvent à leur tour ces questions récurrentes de l'inférence. L'enjeu est de construire une base théorique solide pour ce nouvel outil, mais cela impose de revenir aux questions communes et fondamentales à toutes les techniques et méthodes qui naissent de la pensée statistique depuis ses origines.

Notre travail est une tentative pour contribuer en certains points à ce courant.

TABLE DES MATIÈRES

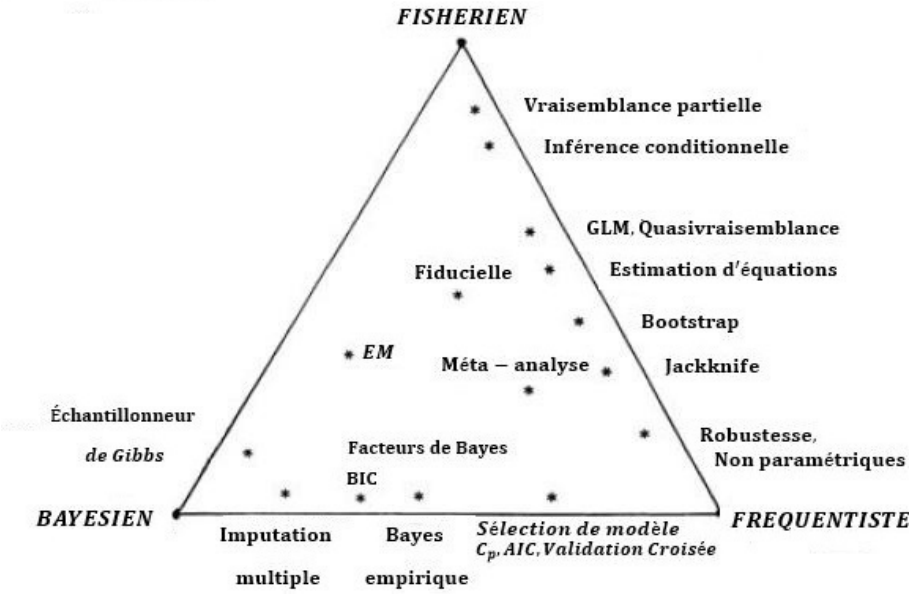
CHAPITRE 1

INFÉRENCE STATISTIQUE

1.1 Introduction

Les concepts fondamentaux de la théorie de l'inférence statistique ont commencé à voir le jour depuis plus de 200 ans. Différentes approches et différentes écoles se sont affrontées. Enormément de débats passionnés et de controverses ont émaillé cette longue époque. Par le fait que la mésentente prend source à la racine, c'est-à-dire dans le sens même de la notion de probabilité, les querelles ne sont pas prêtes à s'apaiser ni le consensus à s'établir. Débordant le niveau philosophique, les grandes questions restent sujettes à controverse jusque dans la mise en œuvre et l'application des outils et des méthodes de la statistique dans différents contextes.
Les différences fondamentales concernent l'interprétation de la notion de probabilité et les objectifs de l'inférence statistique.
De nos jours, la théorie statistique moderne se développe sous l'influence de trois grands courants de pensée : l'école bayésienne, l'école fréquentiste et l'école fisherienne. Si, en beaucoup de points, les deux premières écoles connaissent de fortes oppositions, l'école fisherienne s'instaure bien souvent en position de compromis. Bien que, certaines démarches lui restent spécifiques et se distinguent nettement des deux autres approches.
Les techniques et méthodes statistiques, qu'elles soient anciennes ou récentes, ne sont pas chacune le fruit exclusif d'une école par rapport à une autre. Elles sont plutôt le résultat d'une conjugaison des influences. Chaque technique et

chaque méthode se caractérise par sa relative proximité à un pôle par rapport à un autre.



Triangle de l'inférence : Efron, B. (1998). RA Fisher in the 21st century. Statistical Science, Vol. 13, No. 2, 95-122.

Les réseaux de neurones artificiels, reconnues comme extensions des outils statistiques à l'instar d'autres techniques et méthodes émergentes, se sont révélés d'une grande efficacité dans le traitement de problèmes complexes qui, souvent, sont réfractaires aux méthodes traditionnelles. Cependant, les soubassements théoriques, qui sont actuellement objets d'intenses recherches, ne sont pas encore arrivés au niveau de maturité dont jouissent les méthodes statistiques traditionnelles.
Si les avancées pratiques sont fulgurantes et marquent une incontestable puissance, la difficile manipulation de cette nouvelle créature mathématique fait que beaucoup de questions théoriques restent en suspens, soit partiellement, soit totalement. Parmi ces questions figure, en position centrale, celle de l'inférence :

- Quels sont les aspects de l'inférence statistique qui doivent être mis en lumière dans le cadre de cette technique neuronale ?
- Quels sont les entités à considérer (paramètres, modèles, cadre prédictif, ...) ?

- Quelles sont les spécificités caractérisant cette nouvelle technique, et relevant de l'inférence statistique ?
- Quels sont les outils, hérités de la théorie statistique générale, qui seraient mis en œuvre pour permettre la manipulation de la notion de l'inférence dans la théorie des réseaux neuronaux ?

1.2 Inférence

En toute généralité, le rôle de l'inférence statistique est de produire un cadre dans lequel des conclusions inductives peuvent être formulées au sujet du mécanisme ayant généré les données. Ces données étant assimilées à des réalisations de variables aléatoires (Young et Smith). Nous avons en notre possession un ensemble de données, $x = (x_1, x_2, ..., x_n)$. Leur analyse s'articule sur deux considérations :

- Le vecteur x est assimilé à la réalisation d'une variable aléatoire $X = (X_1, X_2, ..., X_n)$ ayant une densité de probabilité inconnue $f(x)$.
- Cette densité de probabilité devrait appartenir à une certaine famille appropriée $\mathfrak{E}$.

En inférence statistique paramétrique, $f(x)$ possède une forme analytique connue, mais implique des paramètres inconnus en nombre fini $\theta = (\theta_1, ..., \theta_d)$. La région de $\mathbb{R}^d$ de toutes les valeurs possibles de θ sera notée Θ et sera appelée l'espace des paramètres. Dans ce cadre, la représentation paramétrique du modèle est $f(x; \theta)$.

L'inférence statistique est dite non paramétrique, lorsque tout simplement le modèle n'admet pas une représentation paramétrique.

En inférence paramétrique, dans la mesure où le modèle est fixé, toutes les questions tournent autour des paramètres θ. L'objectif est de trouver le moyen d'utiliser l'ensemble des données x pour évaluer certains aspects de θ. On distingue cinq principaux types d'inférences :

- Estimation ponctuelle
- Estimation par intervalle de confiance
- Tests d'hypothèses
- Prédiction de la valeur d'une variable aléatoire dont la distribution dépend de θ.
- Vérification de l'adéquacité d'un modèle spécifié par $\mathfrak{E}$ et Θ.

Dans l'estimation ponctuelle, une seule valeur est calculée à partir des données x. Cette valeur sera utilisée pour estimer θ. Dans l'estimation par in-

tervalle de confiance, les données sont utilisées pour déterminer un intervalle qui possède une grande probabilité prédéterminée d'inclure la vraie valeur de θ. Les tests d'hypothèses posent des hypothèses a priori concernant le paramètre θ, et évaluent leur plausibilité en vérifiant si elles sont soutenues ou non par les données en notre possession. Le quatrième et le cinquième type peuvent être considérés dans beaucoup de situations comme l'enjeu central de l'inférence, Ils peuvent s'appuyer sur les trois premiers types, tout comme ils peuvent être développés d'une manière autonome, Leur considération a abouti à la construction d'un axe spécifique qui est celui de l'inférence prédictive, on distingue au moins six dérivations de cet axe :

- Approche bayésienne totale
- Méthodes exactes
- Approche par la théorie de la décision
- Méthodes basées sur la vraisemblance prédictive
- Approches asymptotiques
- Méthodes du bootstrap.

En outre, l'inférence prédictive ne se restreint pas uniquement aux modèles paramétriques mais peut être également conçue pour les modèles non paramétriques.

1.3 La sélection de modèle

Un autre domaine, qui revêt une grande importance dans le contexte de cette thèse est celui de la sélection de modèle. C'est un domaine qui connaît des développements importants et qui est l'objet d'une recherche très active, mais qui néanmoins n'enregistre pas une percée définitive. La question qui s'y pose est de savoir comment sélectionner le modèle lui-même à partir des observations, et non uniquement les paramètres continus d'un modèle donné. Nous allons voir, par la suite, que les caractéristiques et les particularités des réseaux de neurones artificiels font que la question de l'inférence se trouve naturellement confinée dans cette voie.

Il est bien connu déjà que des problèmes relativement simples peuvent se compliquer rapidement avec la question du choix des variables explicatives lorsque leur nombre initial est conséquent. Si, en outre, s'ajoute des considérations sur la structure formelle du modèle, qui n'est plus préétablie mais arbitraire, la complexité atteint de hauts niveaux.

1.4 Les axes actuels de la recherche

Les théories classiques sur l'estimation et les tests ne sont opérantes que lors de l'entame d'un problème de sélection de modèle mais pas plus loin. Les statisticiens commencent à rencontrer aujourd'hui des problèmes réellement compliqués, avec un nombre considérable d'observations et des centaines de candidats pour le modèle. Un domaine en plein essor, appelée apprentissage machine, est en train de se développer pour traiter ce genre de problème. Cependant, il n'est pas encore très bien lié à la théorie statistique.

1.5 Spécificité des RN

Le traitement de la question de l'inférence dans le cadre des réseaux de neurones impose de répondre au préalable à trois questions essentielles :
1. Où situer les RN entre l'inférence paramétrique et l'inférence non paramétrique ?
2. Comment formuler la problématique de l'inférence dans le cadre des RN ?
3. Quel critère serait le plus adéquat pour juger de la qualité de l'inférence ?

1.5.1 Première question : Où situer les RN entre l'inférence paramétrique et l'inférence non paramétrique

La grande particularité des RN réside dans le fait que la forme analytique du modèle qui en résulte n'est pas stable et définitivement fixé à l'avance. Bien que le modèle ne dépend effectivement que de l'ajustement de ses paramètres, le nombre de ces paramètres n'est pas fixe a priori comme dans les méthodes statistiques standards.
Si en statistique paramétrique nous pouvons dire que l'espace des paramètres Θ est une partie de $\mathbb{R}^d$, ($\Theta \in \mathbb{R}^d$) avec d connu, nous pouvons encore le dire pour les RN, sauf que d est désormais inconnu jusqu'à la fin du processus d'identification du meilleur modèle. Ce qui ne correspond pas exactement à la définition de l'inférence statistique paramétrique dans la mesure où la représentation paramétrique du modèle $f(x;\theta)$ n'est pas établie à l'avance, et la forme analytique n'est pas connue avec précision. De l'autre côté, nous ne pouvons pas pour autant situer les RN dans le cadre de l'inférence statistique non paramétrique dans la mesure où, à l'achèvement du processus

d'identification, le modèle possède une représentation ne dépendant que de ses paramètres Θ. Ainsi, pour les RN, l'extraction du modèle ne se fait pas uniquement parmi ceux d'une seule famille (par l'ajustement des paramètres), mais il se fait parmi ceux d'une suite de familles emboîtées (voir figure ci-après). Les deux opérations, du choix de la famille et de l'ajustement des paramètres, se font de manière simultanée.
À noter, comme nous allons le voir dans le développement de la thèse, qu'un nombre réduit de paramètres conduit à une mauvaise qualité de l'inférence, tout autant qu'un nombre excessif. La question du nombre optimal de paramètres va se révéler centrale pour notre problématique.

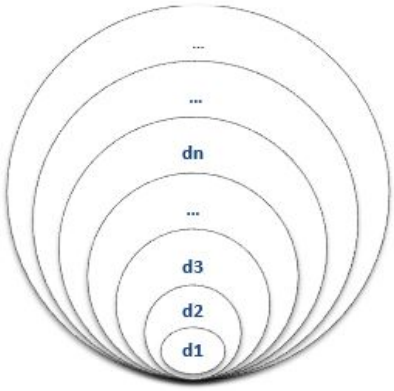


FIGURE 1.1 – Emboîtement des différents modèles neuronaux

1.5.2 Deuxième question : Comment formuler la problématique de l'inférence dans le cadre des RN ?

Les RN se sont développés de sorte à apporter des outils comparables à ceux élaborés dans le cadre standard pour répondre à toute sorte de problématique formulée dans la science statistique. S'il y a diversité d'objectifs et de questions posés dans le cadre de la statistique traditionnelle, cette même diversité se retrouve et se prolonge dans le cadre des RN.
Nous ne pouvons pas parler des RN en les réduisant à une entité unique, ce qui aura le même sens que de confiner la statistique à une seule technique. Les réseaux de neurones se diversifient en fonction des problématiques à traiter. Dans chaque cas de figure, les caractéristiques et les propriétés sont spécifiées pour produire l'entité répondant au besoin. La question de l'inférence

se retrouve ainsi sujette à plusieurs angles d'attaque. Chacun approprié au contexte requis. Néanmoins, une approche globale peut être envisagée sous réserve d'une formulation générale du problème de l'inférence.
En effet, la statistique, à la lumière de la définition générale de l'inférence, construit sa logique, ses outils et ses techniques pour répondre à la considération suivante : "Comment, à partir du **fragment**, extraire une information valable et sûre touchant le **tout** ?". C'est cette question qui est au cœur de la pensée statistique. À partir de quelques données récoltées sur un échantillon, extraire une information rigoureuse, utile et sûre qui porte sur toute la population. Or, qu'est-ce qui peut nous autoriser en cela ?
Les outils qui, à partir de "l'observable", nécessairement limité, permettent de se renseigner sur "l'inaccessible", puisent leur puissance dans ces subtilités, qui sont autant de principes.
Estimer, tester, modéliser, prévoir ..., sont des actes qui découlent d'une même philosophie unificatrice. Le fragment recèle une information qui lui est spécifique et une information qu'il partage avec le tout. Tout l'art consiste à isoler et extraire cette information commune, extrapolable et généralisable.
Ainsi, la crédibilité que l'on accorde aux résultats touchant la population générale tient de la conviction que le canal qui permet l'extrapolation du **fragment** vers le **tout** est d'une fiabilité affirmée. C'est ce concept que nous allons identifier avec le terme **"généralisation"**, et autour du quel va s'articuler tout notre travail.

Unicité de l'inférence dans la régression, la classification et le clustering

Si, souvent le couronnement d'un travail statistique est l'aboutissement à un modèle (explicatif ou prédictif), les techniques centrales sont celles de la régression, la classification et le clustering. Pour cette raison, nous allons ci-après clarifier l'idée de l'unicité de la question de l'inférence dans ces divers contextes. Bien entendu, quand nous abordons des problèmes relevant des trois catégories citées plus haut, nous sommes toujours mis dans la même situation : une situation d'inférence !
Dans le cas de la régression, les individus sont décrits au moyen de variables. Certaines de ces variables sont sensées expliquer d'autres. Elles sont liées. Il s'agit de déterminer cette liaison. Cette liaison que nous découvrons doit rester vraie sur toute la population dont est issu l'échantillon.
Dans le cas de la classification, les individus sont aussi décrits au moyen

de variables. Mais l'une d'entre elles est catégorielle. Elle représente les classes auxquelles appartiennent les individus. L'identification de la classification revient à trouver dans les variables explicatives le moyen de prédire les classes auxquelles appartiennent les éléments de toute la population dont est issu l'échantillon.

Dans le cas du clustering, les individus sont aussi décrits au moyen de variables. Mais ici, il s'agit d'identifier et de discerner les populations dont le mélange a donné l'échantillon. Ce qui conduit naturellement à la question suivante : Quelle relation existe-t-il entre la découverte des structures dans les données et l'estimation des densités de probabilité ?

C'est toujours le principe fondamental de l'utilisation de l'échantillon pour extraire de l'information concernant la population mère. La découverte de la structure dans l'ensemble des données signifie la "subdivision" de cet ensemble en sous-ensembles ayant une certaine différence "naturelle" entre eux. Autrement dit, la répartition de l'ensemble en parties ou chacune d'elles contient des éléments ayant une certaine ressemblance (ou proximité) entre eux et une dissemblance avec les éléments des autres parties. Dire cela c'est assumer que les éléments de chacune des parties proviennent d'une population spécifique, différente des populations qui ont fournit les éléments des autres parties.

Si maintenant on admet que cette population est caractérisée par sa distribution de probabilité, alors identifier cette population revient à identifier sa distribution de probabilité. Vu sous cet angle, nous admettons que l'ensemble soumis à l'étude résulterait d'un mélange de populations. Il s'agit alors d'identifier le comment de ce mélange. Le cerner signifierait identifier la structure dans les données. Une manière de le cerner c'est de procéder à l'estimation des densités de probabilité. Il faut insister sur le fait que l'échantillon n'est que l'instrument et non le but. La recherche de la structure dans l'ensemble en notre possession n'est pas une finalité en soi. La finalité est la structure dans la population qui a engendré notre échantillon.

Notre travail est pour faire une inférence sur ce qui n'est pas observable. C'est dans la même manière et avec le même esprit que pour la classification ou pour la régression. Ce que nous trouvons sur l'échantillon est une indication sur ce que nous nous attendons à trouver dans la population. En filigrane, c'est toujours la question de la généralisation qui est posée. Ainsi, dans les cas de la régression et de la classification, l'inférence se fait en direction d'une seule population. Mais dans le troisième cas elle s'est faite en direction de plusieurs populations.

Dans les deux premiers cas, il s'agit de découvrir une "vérité" dans l'échantillon, mais qui reste valable dans toute la population. Dans le troisième cas, il s'agit d'identifier quelles sont les populations qui sont entrées en jeu. Dans l'ensemble des trois cas, nos outils sont les mesures faites sur un certain nombre de variables.

1.5.3 Troisième question : Quel critère serait le plus adéquat pour juger la qualité de l'inférence ?

À la lumière de la discussion précédente, le critère avec lequel nous pouvons juger la qualité de l'inférence peut se présenter sous des formes diverses, mais néanmoins il doit répondre à un principe unique de base qui est celui de la mesure de la qualité de généralisation.
Le critère de la qualité de généralisation sera donc celui sur lequel nous allons focaliser notre travail relatif aux RN. Mais avant cela, nous allons démontrer un résultat unificateur et universel. Ce résultat est valable pour toute méthode de modélisation, car il est démontré dans le cas le plus général. Malgré la similitude éventuelle, dans la forme, avec des résultats connus liés aux estimateurs, ce résultat s'établit pour les méthodes mêmes génératrices des estimateurs. Il établit que quelle que soit la méthode d'estimation, minimiser l'erreur de généralisation revient à résoudre une "forme particulière" du dilemme Biais/Variance. Ce résultat vient donc généraliser des résultats connus qui ont été établis pour des estimateurs particuliers (voir par exemple (Hagiwara) et la question de la régularisation). Ainsi, quelle que soit la méthode d'estimation, l'équivalence est démontrée entre la minimisation de l'erreur de généralisation et l'équilibre Biais/Variance.

1.6 Le coût de la généralisation

Proposition 1.1 *Le coût de la généralisation s'écrit en termes de biais et variance.*

Remarque 1.1 *Cette démonstration ne tient pas seulement pour le modèle linéaire ni pour les réseaux de neurones. Elle est valable dans le cas le plus général de la régression linéaire ou non linéaire.*

Le modèle régressif est le suivant :

$$y = f(x) + \varepsilon$$

Où $f(x)$ est une fonction déterministe du régresseur x et ε est une erreur aléatoire. x est un vecteur à p composantes (qui doivent correspondre à p variables). Nous avons également les hypothèses suivantes :

$$\mathbb{E}(\varepsilon) = 0$$

$$\mathbb{E}(\varepsilon^2) = \sigma^2$$

Le modèle estimé sera noté par f_R.

1.6.1 Problématique :

L'objectif ici est d'évaluer la qualité de la généralisation de la méthode qui génère les estimateurs. On ne mesure pas la qualité de généralisation d'un modèle particulier f_R, produit par une méthode, mais plutôt la qualité de généralisation de la méthode même qui produit ce modèle. Chaque méthode d'estimation produit une "procédure" qui permet de construire un estimateur f_R étant donné un échantillon $\mathfrak{E}$ (composé de n observations sur p variables). La méthode peut être : La méthode des moindres carrés, la méthode ridge, les réseaux de neurones, ...etc.

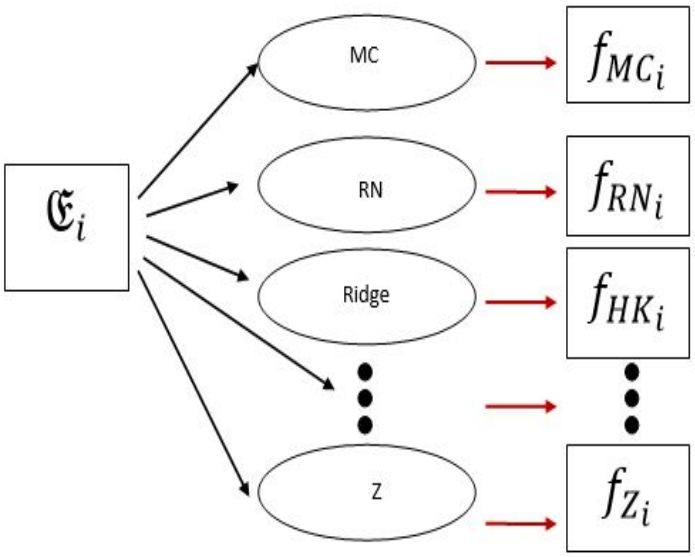


FIGURE 1.2 – Schéma explicatif

Disposant d'un échantillon $\mathfrak{E}$, chacune des méthodes fournit un estimateur qui lui est propre. Pour une même méthode M, à chaque échantillon $\mathfrak{E}_i$ correspond un estimateur f_{M_i}. Des échantillons différents engendrent des estimateurs différents. Ainsi, la qualité de généralisation d'une méthode est la qualité de généralisation au vu de tous les échantillons possibles.
Une méthode est meilleure qu'une autre si sa qualité de généralisation est meilleure que celle de l'autre sur l'ensemble des échantillons possibles.

1.6.2 Ecriture formelle du coût de la généralisation

Une méthode généralise bien lorsque les estimateurs qu'elle génère généralisent bien, dans le sens d'une espérance sur tous les estimateurs. En outre, ceci doit l'être quel que soit le point supplémentaire (x, s) sur lequel sera mesurée cette capacité de généralisation. De cette façon, une autre espérance entre en jeu : c'est celle relative au point supplémentaire (x, s). La fonction de coût se conçoit donc avec une double espérance :

- Par rapport à l'estimateur (et donc par rapport à l'échantillon ayant conduit à l'élaborer).
- Par rapport au point supplémentaire.

Au sujet du point supplémentaire (x, s), si x peut être fixé, s ne peut pas l'être. Il est la vraie réponse du modèle régressif $s = f(x) + \varepsilon$. Si le même x est repris, s sera différent à cause de l'existence de ε.
Si maintenant, nous définissons la fonction de coût, pour une méthode donnée M, comme étant l'erreur quadratique moyenne entre la vraie réponse s et l'estimation $f_M(x)$, nous devons l'écrire comme suit :

$$C(M) = \mathbb{E}\big([s - f_M(x)]^2\big) \tag{1.1}$$

Cependant, il faut correctement préciser le sens de l'opérateur espérance, $\mathbb{E}$.
s est une variable aléatoire (dont la loi de probabilité est induite par celle de ε).
$f_M(x)$ est une autre variable aléatoire (dont la loi de probabilité est induite par celle de l'échantillon $\mathfrak{E}$).
$\mathbb{E}$ est une espérance à deux niveaux :

- D'abord pour chaque valeur particulière de s, l'espérance est alors prise par rapport aux ensembles $\mathfrak{E}$ (qui génèrent les estimateurs f_M qui de leur côté génèrent les estimations $f_M(x)$).

$$C_s(M) = \mathbb{E}_{\mathfrak{E}}\big([s - f_M(x)]^2\big) \tag{1.2}$$

Remarque 1.2 *C_s est le coût pour un point particulier s. C'est l'espérance par rapport à la méthode, puisqu'elle est prise sur tous les estimateurs susceptibles d'être générés par la méthode.*

– Ensuite sur toutes les valeurs possibles de s.

$$C(M) = \mathbb{E}_{\varepsilon}\big(C_s(M)\big) \tag{1.3}$$

Ainsi, quand on écrit $\mathbb{E}$, c'est pour écrire :

$$\mathbb{E}\big([s - f_M(x)]^2\big) = \mathbb{E}_{\varepsilon}\big(\mathbb{E}_{\mathfrak{E}}([s - f_M(x)]^2)\big) \tag{1.4}$$

C'est le coût pour la méthode M, pour tous les estimateurs possibles qu'elle peut générer (en fonction des échantillons $\mathfrak{E}$), et pour toutes les valeurs possibles s obtenus quand on injecte une valeur particulière (et quelconque) x.

Remarque 1.3 *La même formule serait obtenue si nous aurions commencé d'abord par calculer l'espérance par rapport à tous les réponses possibles s, mais pour le même estimateur f_M. Nous obtiendrons le coût pour un estimateur particulier (en fonction de toutes les réponses possibles s).*

$$C_{fM}(s) = \mathbb{E}_{\varepsilon}\big([s - f_M(x)]^2\big)$$

Dans un deuxième temps, nous calculons l'espérance de ce coût par rapport à tous les estimateurs susceptibles d'être généré par la méthode :

$$C(M) = \mathbb{E}_{\mathfrak{E}}\big(C_{fM}(s)\big)$$

Passons maintenant à la décomposition de la fonction de coût $C(M)$:

– Pour le premier niveau de l'espérance, c'est-à-dire $C_s(M)$, nous avons :

$$s - f_M(x) = [s - f(x)] + [f(x) - f_M(x)] = \varepsilon + [f(x) - f_M(x)]$$

Et donc :

$$\mathbb{E}_{\mathfrak{E}}\big([s - f_M(x)]^2\big) = \mathbb{E}_{\mathfrak{E}}\big(\big[\varepsilon + [f(x) - f_M(x)]\big]^2\big)$$

$$\mathbb{E}_{\mathfrak{E}}\big([s-f_M(x)]^2\big) = \mathbb{E}_{\mathfrak{E}}(\varepsilon^2)+\mathbb{E}_{\mathfrak{E}}\big([f(x)-f_M(x)]^2\big)+2\times\mathbb{E}_{\mathfrak{E}}\big(\varepsilon\times[f(x)-f_M(x)]\big)$$

Mais comme ε est une valeur particulière et fixe (puisque correspondant à une valeur fixe de s), nous avons :

$$\mathbb{E}_{\mathfrak{E}}\big([s - f_M(x)]^2\big) = \varepsilon^2 + \mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]^2\big) + 2\varepsilon \times \mathbb{E}_{\mathfrak{E}}\big[f(x) - f_M(x)]\big)$$

– Pour le deuxième niveau de l'espérance, c'est-à-dire $C(M)$, nous avons :

$$C(M) = \mathbb{E}_{\varepsilon}\big[\varepsilon^2 + \mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]^2\big) + 2\varepsilon \times \mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]\big)\big]$$

$$C(M) = \mathbb{E}_{\varepsilon}[\varepsilon^2] + \mathbb{E}_{\varepsilon}\big[\mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]^2\big)\big] + \mathbb{E}_{\varepsilon}[2\varepsilon \times \mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]\big)\big]$$

Le premier terme est σ^2, le deuxième terme est égal à $\mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]^2\big)$, car ne dépendant pas de ε. Le troisième terme es : nul car

$$\mathbb{E}_{\varepsilon}[2\varepsilon \times \mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]\big)\big] = 2 \times \mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]\big)\mathbb{E}_{\varepsilon}[\varepsilon] = 0$$

Donc, en conclusion :

$$C(M) = \sigma^2 + \mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]^2\big) \tag{1.5}$$

Notons que cette fonction de coût se décompose en deux termes :

– La variance de l'erreur ε.
– Une erreur quadratique moyenne au sujet de laquelle il y a plusieurs remarques :
  – Ce n'est pas l'erreur quadratique moyenne usuellement prise entre les estimations et les vraies valeurs, mais plutôt l'espérance du carré des écarts des estimations par rapport à l'unique valeur vraie de la fonction de régression (sans partie aléatoire).
  – L'espérance est prise par rapport aux échantillons, chaque échantillon $\mathfrak{E}$ fournit un estimateur f_M et chaque estimateur f_M fournit une estimation $f_M(x)$.

Reste maintenant à décomposer l'erreur quadratique moyenne en variance et carré du biais. Nous avons :

$$EQM = \mathbb{E}_{\mathfrak{E}}\big([f(x) - f_M(x)]^2\big)$$

Que nous pouvons écrire :

$$EQM = \mathbb{E}_{\mathfrak{E}}\big([f_M(x) - f(x)]^2\big)$$

Nous avons :

$$f_M(x) - f(x) = f_M(x) - \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) + \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - f(x)$$

D'où

$$EQM = \mathbb{E}_{\mathfrak{E}}\Big(\big[f_M(x) - \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big)\big]^2\Big) + \mathbb{E}_{\mathfrak{E}}\Big(\big[\mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - f(x)\big]^2\Big)$$

$$+2 \times \mathbb{E}_{\mathfrak{E}}\Big(\big(f_M(x) - \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big)\big) \times \Big(\mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - f(x)\big)\Big)$$

Le premier terme est la variance $\big($de la variable aléatoire $f_M(x)\big)$. Le deuxième terme est :

$$\mathbb{E}_{\mathfrak{E}}\Big(\big[\mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - f(x)\big]^2\Big) = \big[\mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - f(x)\big]^2$$

Et c'est le carré du biais. Nous allons montrer que le troisième terme est nul, en effet,

$$\mathbb{E}_{\mathfrak{E}}\Big(\big[f_M(x) - \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big)\big]\Big) \times \Big(\mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - f(x)\Big)$$

$$= \mathbb{E}_{\mathfrak{E}}\Big[f_M(x) \times \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - f_M(x) \times f(x) - \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big)^2 + \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) \times f(x)\Big]$$

$$= \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big)^2 - f(x) \times \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big)^2 + \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) \times f(x) = 0$$

À la lumière de cela, nous pouvons dire que minimiser le coût :

$$C(M) = \sigma^2 + \mathbb{E}_{\mathfrak{E}}\big([f_M(x) - f(x)]^2\big)$$

Revient à minimiser la quantité :

$$EQM = \mathbb{E}_{\mathfrak{E}}\big([f_M(x) - f(x)]^2\big)$$

Qui n'est autre que la somme de la variance et du carré du biais :

$$EQM = \mathbb{E}_{\mathfrak{E}}\Big(\big[f_M(x) - \mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big)\big]^2\Big) + \big[\mathbb{E}_{\mathfrak{E}}\big(f_M(x)\big) - f(x)\big]^2$$

$$EQM = V + B^2 \tag{1.6}$$

Ce qui revient à la résolution du dilemme biais/variance.

Commentaire :

La quantité EQM (erreur quadratique moyenne) est pour juger de la qualité de généralisation des méthodes, les unes par rapport aux autres, et non simplement des estimateurs. Une méthode qui fournit un EQM moindre est préférable aux autres, dans le sens de la qualité de généralisation. Mais si deux méthodes ont **la même valeur de l'EQM** alors cela appelle la remarque suivante :

Remarque 1.4 *La méthode qui a le plus grand biais aurait la plus petite variance.*

Une petite variance signifie **une plus grande stabilité**. Des échantillons différents produisent (à travers les estimateurs construits) des estimations $f_M(x)$ proches les unes des autres (car proches de leur espérance).

1.7 La question de l'Amélioration de la qualité de généralisation : Méthodes traditionnelles et méthodes neuronales : Unicité des objectifs et similitude des solutions.

Nous allons nous concentrer essentiellement sur les questions de régression et de modélisation. C'est un sujet très riche errecelant beaucoup d'enjeux tant théoriques que pratiques.
Le point de départ est le modèle linéaire. C'est un choix naturel car le cheminement historique des idées et des concepts s'est d'abord tracé dans son cadre. Les méthodes pour résoudre les problèmes généraux sont d'abord apparues dans le contexte du modèle linéaire.
Bien que l'objet de notre analyse soit la méthode neuronale, spécifiquement conçue pour traiter de problèmes difficiles restés largement réfractaires aux méthodes traditionnelles, nous devons accorder une grande attention au modèle linéaire. Le parallèle entre les deux approches traditionnelle et neuronale serait plus visible en raison de la maturité des travaux élaborés dans son cadre.
Les outils développés pour traiter des problématiques plus complexes que celles formulées dans le cadre de la régression linéaire ne sont souvent qu'une sophistication de ceux déjà existants. Si les problèmes hautement complexes

(tels que la forte non linéarité et la multitude de variables prises en considération) apporte un nouveau lot de difficultés qui leur sont propres, ils héritent également de quelques questions qui sont restées coriaces même dans le cadre linéaire (tels que la multicollinéarité et la sélection de variables pertinentes). Des méthodes palliatives ont été élaborées pour corriger la défaillance dans certaines circonstances de la méthode des moindres carrés. De la même manière, des méthodes palliatives ont été élaborées pour les mêmes raisons pour les méthodes neuronales. Même si les sources des solutions apparues dans les deux cadres sont différentes et indépendantes, les techniques résultantes présentent des analogies remarquables.
Notre travail s'est fixé comme objectif de mettre en lumière ces similitudes, d'apporter une explication à certaines d'entre elles et de formuler en problèmes ouverts celles qui sont restées en suspens. Nous allons construire notre travail comme suit :
Nous allons d'abord introduire la régression linéaire multiple et la méthode phare des moindres carrés avec ses propriétés.
Le chapitre qui suivra sera consacré à une introduction brève des réseaux de neurones, de ses principales propriétés. Il sera fait cas des problèmes majeurs rencontrés et des solutions envisagées, en présentant des algorithmes d'apprentissages en mettre l'accent sur une méthode neuronale toute récente, l'extreme learning machine, et des problèmes théoriques ouverts qu'elle a généré.
Ensuite, on a présenté les méthodes concurrentes de celle des moindres carrés et les méthodes neuronales palliatives avec les situations dans lesquelles elles sont les plus performantes.
Le chapitre suivant situera le cadre de comparaison entre les solutions apportées suivant les deux approches.
En final, nous présenterons une application des méthodes neuronales dans un cas réel qui illustre la grande efficacité de ces techniques pour des problèmes de prédiction.

CHAPITRE 2

LA RÉGRESSION LINÉAIRE MULTIPLE

2.1 Le modèle linéaire

Nous avons choisi de présenter le modèle linéaire dans sa forme la plus générale pour de multiples raisons :
Notre travail est une recherche de similitudes et d'analogies entre les méthodes traditionnelles de la statistique et les méthodes neuronales. Les méthodes classiques proposent des solutions analytiques aux problèmes de la régression, tandis que les méthodes neuronales proposent des solutions itératives et viennent se surclasser l'une et l'autre par de meilleures propriétés et de plus grandes performances. Néanmoins, la toute dernière théorie, celle de "L'extreme learning machine", bien qu'adoptant la même architecture de réseaux, revient aux solutions analytiques mais avec de bien meilleures propriétés et de bien plus grandes performances.
Nous allons montrer dans le chapitre 3 que cette méthode, au moins dans sa forme basique, peut s'exprimer au moyen d'un modèle linéaire généralisé.
Notre souci de recourir au modèle le plus général n'est donc pas seulement pour souligner l'idée que la linéarité n'est entendue que par rapport aux coefficients (ou paramètres) et non par rapport aux variables explicatives, mais il vise surtout à fixer un modèle dans lequel s'insèrent toutes les méthodes que nous nous proposons d'étudier. Ainsi, le modèle linéaire est :

$$Y = X\beta + \varepsilon$$

Où X est une matrice $(n \times p)$ de rang p, β est un vecteur inconnu de dimensions $(p \times 1)$, ε est le vecteur de dimensions $(n \times 1)$ qui représente les erreurs aléatoires et qui possède les propriétés :

- $\mathbb{E}[\varepsilon] = 0$
- $\mathbb{E}[\varepsilon\varepsilon'] = \sigma^2 I_n$

À souligner cependant les précisions suivantes :

- X ne représente pas les variables indépendantes originelles mais elle est compris dans le sens où la forme générale de $X\beta$ est

$$\sum_{i=1}^{p} \beta_i \theta_i(x_\nu) \tag{2.1}$$

  Où x_ν est la $\nu^{ème}$ observation ; $x_\nu = \{x_{1\nu}, x_{2\nu}, ..., x_{p\nu}\}$.
  Les θ_i sont des fonctions arbitraires des variables explicatives originelles.
- Le fait de supposer que l'erreur ε est sans biais ne diminue en rien la généralité dans la mesure où le modèle peut comporter une constante (et que donc le biais de l'erreur peut être englobé dans cette constante). Dans le cas des méthodes neuronales, ce biais peut être perçue comme le poids associée à une entrée constante qui vaut 1 pour toutes les observations.
- Au sujet de l'hypothèse $\mathbb{E}[\varepsilon\varepsilon'] = \sigma^2 I_n$, l'indépendance des erreurs n'est pas restrictive dans la mesure où ne devons considérer que la méthode des moindres carrés ordinaires pour ne traiter que ses concepts de base (et non ses variantes techniques). De même, et pour la même raison, nous supposons également que σ^2 est constante et que nous n'avons pas de problèmes d'hétéroscédacité.
- Nous pouvons au besoin revenir au modèle linéaire simple, lorsqu'il n'y a pas nécessité de généralisation, en posant :
  Pour tout $i : \theta_i(x_\nu) = x_{i\nu}$.
  Nous retrouvons de cette manière le modèle le plus usuel, où la variable expliquée est une fonction directe des variables explicatives.

De cette manière nous pouvons dire que nous sommes en possession d'un échantillon $[(X_{1,i}, ..., X_{p,i}), Y_i]_{i=\{1,..,n\}}$ d'observations indépendantes et identiquement distribuées. Le modèle de la régression linéaire s'écrit :

$$Y_i = \beta_0 + X_{1i}\beta_1 + X_{2i}\beta_2 + ... + X_{pi}\beta_p + \varepsilon_i \tag{2.2}$$

Où $\beta = (\beta_1, \beta_2, ..., \beta_p)'$ sont les paramètres réels du modèle à estimer, et les ε_i sont les erreurs aléatoires. D'un point de vue matriciel, le modèle de

régression linéaire multiple s'écrit :

$$\begin{pmatrix} Y_1 \\ . \\ . \\ . \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & X_{11} & ... & X_{p1} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ 1 & X_{1n} & ... & X_{pn} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ . \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ . \\ . \\ \varepsilon_n \end{pmatrix}$$

$$Y = X\beta + \varepsilon$$

Y étant un vecteur colonne de dimensions $(n \times 1)$, la matrice X est souvent appelée matrice de design de dimensions $(n \times p)$, β est un vecteur colonne de dimensions $(p \times 1)$ et ε est un vecteur colonne de dimensions $(n \times 1)$.
Certaines hypothèses supplémentaires sont souvent mises en évidence dans le cadre du modèle linéaire, à savoir :

- Une hypothèse principale : la non collinéarité des variables indépendantes afin de garantir la régularité de la matrice $X'X$ et dont son inversibilité, propriété essentielle pour les méthodes de résolution.
- Une hypothèse parfois supplémentaire : $\varepsilon_i \sim N(0, \sigma^2)$ (Les erreurs suivent une loi normale). C'est sous cette hypothèse qu'il y a une identification des deux solutions : celle des moindres carrés ordinaires et celle du maximum de vraisemblance.

2.2 Estimateurs des moindres carrés ordinaires MCO

La procédure usuelle pour l'estimation des paramètres inconnus β est la méthode des moindres carrés ordinaires qui a produit l'estimateur du même nom.
Cet estimateur jouit de bonnes propriétés lorsqu'un certain nombre de conditions favorables sont réunies, principalement lorsque la matrice $X'X$ dans la forme d'une matrice de corrélation et proche de la matrice unité. Il possède l'avantage d'être le meilleur estimateur linéaire sans biais dans un sens qui sera précisé par le théorème de Gauss-Markov et qui sera énoncé plus bas. Cet estimateur étant obtenu par la procédure suivante : Il s'agit de trouver les

valeurs des éléments de β qui minimisent la somme des résidus. Le problème peut s'écrire comme suit, déterminer $\hat{\beta}$ tel que :

$$(Y - X\beta)'(Y - X\beta) = min(Y - X\beta)'(Y - X\beta) \tag{2.3}$$

Ce qui est équivalent à :

$$min(Y'Y - \beta'X'Y - Y'X\beta + \beta'X'X\beta) \tag{2.4}$$

Dérivant par rapport à β, nous obtenons :

$$-X'Y - X'Y + X'X\beta + (X'X)'\beta = 0 \tag{2.5}$$

$$-2X'Y + 2X'X\beta = 0 \tag{2.6}$$

Pour trouver $\hat{\beta}$ nous devons résoudre le système d'équations suivantes :

$$X'X\hat{\beta} = X'Y \tag{2.7}$$

Cet ensemble d'équations s'appelle communément les "équations normales" de l'estimation MCO. Et qui donne

$$\hat{\beta} = (X'X)^{-1}X'Y \tag{2.8}$$

C'est l'estimateur des moindres carrés ordinaires.

2.2.1 Interprétation géométrique

Orthogonalité et somme des résidus

Nous avons :

$$X'X\hat{\beta} = X'Y \tag{2.9}$$

$$\Rightarrow X'(X\hat{\beta} - Y) = 0 \tag{2.10}$$

$$\Rightarrow X'(Y - X\hat{\beta}) \tag{2.11}$$

$$\Rightarrow X'\hat{r} = 0 \tag{2.12}$$

Où $\hat{r}$ est le vecteur de résidus de la régression, les résidus sont orthogonaux aux variables explicatives. Cela veut dire que chaque variable explicative (chaque colonne de la matrice X) est orthogonale aux résidus de la régression. Ce résultat est une généralisation du résultat de régression simple de l'orthogonalité entre la seule variable explicative (à part la constante) et les

résidus. C'est une généralisation, mais la preuve est beaucoup plus succincte que celle que nous connaissons dans la régression simple. La Figure suivante représente le cas où il y a deux variables explicatives :

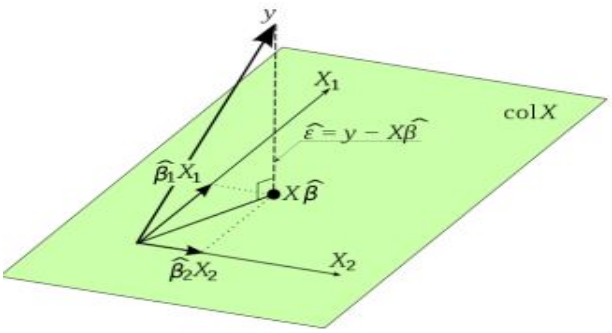


FIGURE 2.1 – La représentation géométrique des estimateurs MCO

Proposition 2.1 *(Biais d'estimateur)*
L'estimateur des moindres carrés ordinaires du vecteur de paramétres β est sans biais c'est à dire que $\mathbb{E}(\hat{\beta}) = \beta$.

Preuve

$$\hat{\beta} = (X'X)^{-1}X'Y \tag{2.13}$$

$$\hat{\beta} = (X'X)^{-1}X'(X\beta + \varepsilon) \tag{2.14}$$

$$\hat{\beta} = \beta + (X'X)^{-1}X'\varepsilon \tag{2.15}$$

$$\mathbb{E}(\hat{\beta}) = \beta + \mathbb{E}\Big((X'X)^{-1}X'\varepsilon\Big) \tag{2.16}$$

$$\mathbb{E}(\hat{\beta}) = \beta + \mathbb{E}\Big((X'X)^{-1}X'\mathbb{E}(\varepsilon)\Big) \tag{2.17}$$

$$\mathbb{E}(\hat{\beta}) = \beta \tag{2.18}$$

Proposition 2.2 *(Variance de $\hat{\beta}$)*
La variance de l'estimateur des moindres carrés ordinaires $\hat{\beta}$ du vecteur de paramètre β est égale à :

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$$

Preuve

$$Var(\hat{\beta}) = \mathbb{E}([\hat{\beta} - \mathbb{E}(\hat{\beta})]\ [\hat{\beta} - \mathbb{E}(\hat{\beta})]') \tag{2.19}$$

$$Var(\hat{\beta}) = \mathbb{E}([\hat{\beta} - \beta]\ [\hat{\beta} - \beta]') \tag{2.20}$$

Or on a :

$$\hat{\beta} = \beta + (X'X)^{-1}X'\varepsilon \tag{2.21}$$

$$\hat{\beta} - \beta = (X'X)^{-1}X'\varepsilon \tag{2.22}$$

Ce qui permet d'écrire :

$$Var(\hat{\beta}) = \mathbb{E}[(X'X)^{-1}X'\varepsilon\varepsilon'X(X'X)^{-1}] \tag{2.23}$$

$$Var(\hat{\beta}) = (X'X)^{-1}X'\mathbb{E}(\varepsilon\varepsilon')X(X'X)^{-1} \tag{2.24}$$

Sous l'hypothèse $\mathbb{E}(\varepsilon\varepsilon') = \sigma^2 I_n$, nous obtenons finalement :

$$Var(\hat{\beta}) = (X'X)^{-1}X'\sigma^2 I_n X(X'X)^{-1} \tag{2.25}$$

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1}X'X(X'X)^{-1} \tag{2.26}$$

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1} \tag{2.27}$$

2.2.2 Théorème de Gauss Markov

Ce théorème postule que $\hat{\beta}$ est un estimateur efficace, c'est-à-dire l'estimateur de variance minimale parmi la classe des estimateurs linéaires en Y et sans biais. On parle d'estimateur $BLUE$ ($Best\ Linear\ Unbiased\ Estimator$). Pour démontrer ce théorème, il suffit de prouver que chaque composantes $\hat{\beta}_j$ de $\hat{\beta}$ avec $j = \{1, ..., k\}$ est de variance minimale, c'est-à-dire :

$$Var(\tilde{\beta}_j) \geq Var(\hat{\beta}_j)$$

Avec $\tilde{\beta}_j$ un estimateur sans biais.

Preuve

$$\hat{\beta} = Ay, o\grave{u} \quad A = (X'X)^{-1}X' \tag{2.28}$$

Considérons maintenant un autre estimateur $\tilde{\beta}$ linéaire en y avec :

$$\tilde{\beta} = (A + C)y \tag{2.29}$$

$$\tilde{\beta} = [\ (X'X)^{-1}X' + C\ ]y \tag{2.30}$$

$$\tilde{\beta} = [\ (X'X)^{-1}X' + C\ ]\ (X\beta + \varepsilon) \tag{2.31}$$

$$\tilde{\beta} = \beta + (X'X)^{-1}X'\varepsilon + CX\beta + C\varepsilon \tag{2.32}$$

$$\tilde{\beta} = \beta(I_k + CX) + [\ (X'X)^{-1}X' + C\ ]\varepsilon \tag{2.33}$$

L'estimateur $\tilde{\beta}$ est sans biais si $CX = 0$. Nous supposons que cette condition est vérifiée et donc :

$$\tilde{\beta} = \beta + [\ (X'X)^{-1}X' + C\ ]\varepsilon \tag{2.34}$$

$$\tilde{\beta} - \beta = [\ (X'X)^{-1}X'\varepsilon + C\varepsilon\ ] \tag{2.35}$$

Calculons $Var(\tilde{\beta})$:

$$Var(\tilde{\beta}) = \mathbb{E}[\ (\tilde{\beta} - \beta)\ (\tilde{\beta} - \beta)'\ ] \tag{2.36}$$

$$Var(\tilde{\beta}) = \mathbb{E}\left[\left((X'X)^{-1}X'\varepsilon + C\varepsilon\right)\left(\ (X'X)^{-1}X'\varepsilon + C\varepsilon\right)'\right] \tag{2.37}$$

$$Var(\tilde{\beta}) = \mathbb{E}\left[\left((X'X)^{-1}X'\varepsilon + C\varepsilon\right)\left(\varepsilon'(X'X)^{-1}X' + C'\varepsilon'\right)\right] \tag{2.38}$$

$$Var(\tilde{\beta}) = \sigma^2\left[(X'X)^{-1} + (X'X)^{-1}X'C' + CX\ (X'X)^{-1} + CC'\right] \tag{2.39}$$

$$Var(\tilde{\beta}) = \sigma^2\left[(X'X)^{-1} + \sigma^2 CC'\right] \tag{2.40}$$

$$Var(\tilde{\beta}) = Var(\hat{\beta}) + \sigma^2 CC' \tag{2.41}$$

Remarquons que les éléments diagonaux de CC' sont des sommes de carrés, donc positifs. On en déduit que les composantes diagonales de $Var(\tilde{\beta})$ sont donc supérieures ou égales aux composantes diagonales de $Var(\hat{\beta})$, c'est-à-dire :

$$Var(\tilde{\beta}_j) \geq Var(\hat{\beta}_j)_{j=\{1,\ldots,k\}}$$

D'où la preuve du théorème.

Proposition 2.3 *(Résidus et variance résiduelle)*
Le vecteur des valeurs ajustées est le vecteur des prédictions de Y au moyen de X et de β, c'est-à-dire

$$\hat{Y} = X\hat{\beta} = X(X'X)^{-1}X'Y$$

Le vecteur des valeurs ajustées peut être interprété comme la projection de Y sur le sous-espace engendré par les colonnes de la matrice X.

$$\hat{Y} = P_X Y \tag{2.42}$$

Où P_X est un projecteur (c'est-à-dire une matrice idempotente) sur le sous-espace engendré par les colonnes de X.

$$P_X = X(X'X)^{-1}X'. \tag{2.43}$$

Les résidus sont définis par la relation suivante : $\hat{\varepsilon} = Y - \hat{Y}$. Le vecteur des résidus est la différence :

$$\hat{\varepsilon} = Y - X\hat{\beta} = Y - X(X'X)^{-1}X'Y = (I - P_X)Y \tag{2.44}$$

Le vecteur des valeurs ajustées peut également être interprété comme la projection de Y dans le noyau de X' (ou l'orthogonal du sous-espace engendré par les colonnes de X).

$$\hat{\varepsilon} = P_X^{\perp} Y \tag{2.45}$$

Où $P_X^{\perp}$ un projecteur (c'est-à-dire une matrice idempotente) sur le noyau de X'

$$P_X^{\perp} = I - X(X'X)^{-1}X' \tag{2.46}$$

Proposition 2.4 *(Estimateur de variance de bruit)*
Nous permet de construire un estimateur sans biais pour σ_ε^2 qui est :

$$\hat{\sigma}_\varepsilon^2 = \frac{\hat{\varepsilon}'\hat{\varepsilon}}{n-p}$$

La quantité $(n-p)$ est le nombre de degré de liberté avec un rang de $(I-P_X)$ Notre objectif est de calculer $\mathbb{E}(\hat{\varepsilon}'\hat{\varepsilon})$ Pour obtenir le résultat, on utilisera le thérorème général suivant

Théorème 2.2.1 *Soit un vecteur u composé de n variables aléatoires d'espérances nulles, et tel que $var(u) = \sigma_u^2 I$ et A une matrice symétrique non-aléatoire,alors*

$$\mathbb{E}(u'Au) = \sigma_u^2 Tr(A) \tag{2.47}$$

Preuve

Nous avons vu que $\hat{\varepsilon}$ peut également s'écrire :

$$\hat{\varepsilon} = (I - P_X)Y \tag{2.48}$$

Où P_X est un projecteur sur le sous-espace engendré de X :

$$P_X = X(X'X)^{-1}X' \tag{2.49}$$

Donc :

$$\hat{\varepsilon} = (I - P_X)Y = (I - P_X)(X\beta + \varepsilon) = X\beta - P_X X\beta + \varepsilon - P_X\varepsilon \tag{2.50}$$

Or $P_X X = X$,ce qui donne

$$\hat{\varepsilon} = \varepsilon - P_X\varepsilon = (I - P_X)\varepsilon \tag{2.51}$$

On obtient

$$\hat{\varepsilon}'\hat{\varepsilon} = \varepsilon'(I - P_X)'(I - P_X)\varepsilon \tag{2.52}$$

Et comme $(I - P_X)$ est symétrique et idempotente, on a :

$$\hat{\varepsilon}'\hat{\varepsilon} = \varepsilon'(I - P_X)\varepsilon = \varepsilon' I\varepsilon - \varepsilon' P_X\varepsilon \tag{2.53}$$

Par (2.47), on obtient

$$\mathbb{E}(\hat{\varepsilon}'\hat{\varepsilon}) = \sigma_\varepsilon^2 Tr(I) - \sigma_\varepsilon^2 tr(P_X) \tag{2.54}$$

$\mathrm{Tr}(I) = n$ et $\mathrm{Tr}(P_X) = p$, car la trace d'une matrice idempotente est égale à son rang. Donc :

$$\mathbb{E}(\hat{\varepsilon}'\hat{\varepsilon}) = n\sigma_\varepsilon^2 - p\sigma_\varepsilon^2 = (n - p)\sigma_\varepsilon^2 \tag{2.55}$$

2.3 Estimateurs du maximum de vraisemblance

On se place sous les hypothèses fortes, c'est-à-dire que les erreurs ε_i sont supposées gaussiennes. Nous avons donc :

$$\varepsilon_i \sim N(0, \sigma^2) \Rightarrow y_i = x_i'\beta + \varepsilon_i \sim N(x_i'\beta, \sigma^2)$$

Et mutuellement indépendants puisque les erreurs ε_i le sont. La vraisemblance s'en déduit :

$$l(Y,\beta,\sigma^2)=\sum_{i=1}^{n}\Pi f(y_i)=\big(\frac{1}{\sqrt{2\pi\sigma^2}}\big)^n exp\Big[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i-x_i'\beta)^2\Big] \tag{2.56}$$

$$l(Y,\beta,\sigma^2)=\big(\frac{1}{\sqrt{2\pi\sigma^2}}\big)^n exp\Big[-\frac{1}{2\sigma^2}\|Y-X\beta\|^2\Big] \tag{2.57}$$

D'où l'on déduit la log-vraisemblance :

$$logl(Y,\beta,\sigma^2)=-\frac{n}{2}log2\pi-\frac{n}{2}log\sigma^2-\frac{1}{2\sigma^2}\|Y-X\beta\|^2 \tag{2.58}$$

On cherche les estimateurs $\hat{\beta}_{mv}$ et $\hat{\sigma}^2_{mv}$ qui maximisent cette log-vraisemblance. Il est clair qu'il faut minimiser la quantité $\|Y-X\beta\|^2$, ce qui est justement le principe des moindres carrés ordinaires, donc :

$$\hat{\beta}_{mv}=\hat{\beta}=(X'X)^{-1}X'Y.$$

Dérivant (2.58) par rapport à σ^2 on trouve :

$$\frac{\partial logl(Y,\beta,\sigma^2)}{\sigma^2}=-\frac{n}{2\sigma^2}+\frac{1}{2\sigma^4}\|Y-X\beta\|^2 \tag{2.59}$$

À partir de cette équation on trouve :

$$\hat{\sigma}^2_{MV}=\frac{\|Y-X\hat{\beta}_{MV}\|^2}{n} \tag{2.60}$$

Donc l'estimateur $\hat{\sigma}^2_{MV}$ est biaisé.

2.4 Compléments sur la régression linéaire multiple : Anova et inférence sur les paramètres

2.4.1 Test de Fisher et analyse de variance de la régression

L'analyse de la variance permet de décomposer la variance de la variable dépendante en deux composantes : la variance expliquée par le modèle et

la variance résiduelle. Comme on le verra plus loin, cette analyse permet de défnir le coeffcient de détermination multiple qui est un indicateur de la qualité d'ajustement du modèle de régression linéaire multiple. La relation fondamentale de l'analyse de la variance s'établit en remarquant que :

$$\| \underbrace{Y - \bar{Y}}_{SCT} \|^2 = \| \underbrace{\hat{Y} - \bar{Y}}_{SCE} \|^2 + \| \underbrace{Y - \hat{Y}}_{SCR} \|^2 \tag{2.61}$$

On a donc :

$$SCT = SCE + SCR$$

Avec :
$SCT = (Y - \bar{Y})^2$ qui sera d'autant plus grande que les valeurs des Y_i seront éloignées de la moyenne $\bar{Y}$.
$SCE = (\hat{Y} - \bar{Y})^2$ Somme des carrées due au modèle.
$SCR = (Y - \hat{Y})^2$ Somme des carrées résiduelle qui sera proche de 0 si le modèle de régression explique quasi-complètement la variable Y et de ce fait il y aura très peu de fluctuations aléatoires.
On teste $H_0 : \beta_1 = \beta_2 = ... = \beta_{p-1} = 0$ Contre ,
H_1 pour lequel $\beta_j \neq 0, j = \{1, ..., p-1\}$.

Source de variation	SC	ddl	CM
régression SCE	$\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$	$p-1$	$\frac{CME}{CMR}$
résiduelle SCR	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	$n-p$	$SCR/(n-p)$
totale SCT	$\sum_{i=1}^{n}(y_i - \bar{y})^2$	$n-1$	

Nous décidons de rejeter l'hypothèse nulle H_0 et d'accepter l'hypothèse alternative H_1 au seuil $\alpha = 5\%$, si :

$$\mid F_{obs} \mid \geq F_{1-\alpha}(p-1, n-p)$$

Nous décidons de ne pas rejeter H_0 et donc de l'accepter si

$$\mid F_{obs} \mid \leq F_{1-\alpha}(p-1, n-p)$$

Remarque 2.1 ***Le coefficient de détermination***
Le coefficient de détermination généralement noté R^2 est défini par :

$$R^2 = \frac{SCE}{SCT} \tag{2.62}$$

Intuitivement ce coefficient de détermination quantifie la capacité du modèle à expliquer les variations de Y.
Si R^2 est proche de 1 alors le modèle est proche de la réalité.
Si R^2 est proche de 0 alors le modèle explique très mal la réalité. Il faut alors trouver un meilleur modèle.
Le problème en régression multiple avec l'interprétation du R^2 est qu'il varie de façon monotone avec le nombre de variables explicatives. On lui préfèrera donc pour l'interprétation le R^2 ajusté.

Remarque 2.2 ***Le coefficient de détermination ajusté***
Le coefficient de détermination ajusté généralement noté R^2_{adj} est défini par :

$$R^2_{adj} = \frac{SCE/n-p-1}{SCT/n-1} \tag{2.63}$$

Si les variables ne contribuent pas toutes de manière significative il faudra sélectionner un sous-modèle. On cherchera alors à sélectionner le modèle dont le R^2_{adj} est le plus élevé.

2.4.2 Distribution des estimteurs et tests statistiques

Sous l'hypothèse de Normalité de la distribution des Y , pour $i = \{0, ..., n\}$

$$\hat{\beta}_i \sim N(\beta_i, \sigma^2(\hat{\beta}_i))$$

On estime alors $\sigma^2(\hat{\beta}_i)$ par $\hat{\sigma}^2(\hat{\beta}_i)$ et on obtient :

$$\frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}(\hat{\beta}_i)} \sim t(n-p-1) \tag{2.64}$$

Pour $i = \{0, ..., n\}$

$$\begin{cases} H_0 : \beta_i = 0 \\ H_1 : \beta_i \neq 0 \end{cases}$$

Statistique de test et sa loi sous H_0 :

$$t = \frac{\hat{\beta}_i}{\hat{\sigma}(\hat{\beta}_i)} \sim t(n-p-1) \tag{2.65}$$

Donc pour un paramètre $\beta_i, i = \{1, ..., k\}$

$$IC = [\hat{\beta}_i - t_{1-\alpha/2}\hat{\sigma}(\hat{\beta}_i), \hat{\beta}_i + t_{1-\alpha/2}\hat{\sigma}(\hat{\beta}_i)] \tag{2.66}$$

Où $t_{1-\alpha/2}$ est le quantile d'ordre $1-\alpha/2$ d'une student à $(n-p-1)$ *ddl*.

Remarque 2.3 ***Intervalle de confiance d'un point de la droite***
Soit $x_t = (x_{t,1}, x_{t,2}, ..., x_{t,p})$ un vecteur donné.
On cherche un IC pour $E(y_t/x_t) = \beta_1 + \beta_2 x_{t,2} + ... + \beta_p x_{t,p}$ que l'on estime par $\hat{y}_t = \hat{\beta}_1 + \hat{\beta}_2 x_{t,2} + ... + \hat{\beta}_p x_{t,p}$, on a :

$$\hat{y}_t \sim N\big(x_t'\beta, \sigma^2(\hat{y}_t)\big)$$

Avec

$$\sigma^2\big(\hat{y}_t\big) = \sigma^2(x_t'(X'X)^{-1}x_t\big)$$

Comme σ^2 est inconnu, on l'estime avec $\hat{\sigma}^2$:

$$\hat{\sigma}^2\big(\hat{y}_t\big) = \hat{\sigma}^2(x_t'(X'X)^{-1}x_t\big) \tag{2.67}$$

On déduit que :

$$\frac{\hat{y}_t - x'x_t\beta}{\hat{\sigma}(\hat{y}_t)} \sim t(n-p-1) \tag{2.68}$$

D'où

$$IC = [\hat{y}_t - t_{1-\alpha/2}\hat{\sigma}(\hat{y}_t), \hat{y}_t + t_{1-\alpha/2}\hat{\sigma}(\hat{y}_t)] \tag{2.69}$$

Où $t_{1-\alpha/2}$ est le quantile d'ordre $1-\alpha/2$ d'une student à $(n-p-1)$ ddl.

2.5 Prévisions

Reprenons le modèle de régression linéaire multiple :

$$y_i = x_i\beta + \varepsilon_i$$

Pour lequel l'estimation des paramètres est conduite en utilisant n obervations, $i = \{1, ..., n\}$. Considérons que l'on dispose d'une observation supplémentaire pour la variable indépendante à savoir x_{n+1} et nous intéressons à la prévision de la variable dépendante en fonction de x_{n+1}. Cette prévision est égale à :

$$\hat{y}_{n+1} = x_{n+1}\hat{\beta} \tag{2.70}$$

La vraie valeur inconnue de y_{n+1} satisfait au modèle de régression linéaire simple, soit :

$$y_{n+1} = x_{n+1}\beta + \varepsilon_{n+1}, \tag{2.71}$$

Elle permet de défnir l'erreur de prévision :

$$\hat{\varepsilon}_{n+1} = y_{n+1} - \hat{y}_{n+1} = x_{n+1}\beta + \hat{\varepsilon}_{n+1} - x_{n+1}\hat{\beta}, \tag{2.72}$$

$$\varepsilon_{n+1} = y_{n+1} - \hat{y}_{n+1} = x_{n+1}(\beta - \hat{\beta}) + \varepsilon_{n+1}. \tag{2.73}$$

On obtient alors :

$$\mathbb{E}(\hat{\varepsilon}_{n+1}) = x_{n+1}\beta x_{n+1}\mathbb{E}(\hat{\beta}) + \mathbb{E}(\varepsilon_{n+1}) \tag{2.74}$$

$$\mathbb{E}(\hat{\varepsilon}_{n+1}) = x_{n+1}\beta - x_{n+1}\beta = 0, \tag{2.75}$$

$$Var(\hat{\varepsilon}_{n+1}) = \mathbb{E}\big[\big(\hat{\varepsilon}_{n+1} - \mathbb{E}(\hat{\varepsilon}_{n+1})\big)^2\big], \tag{2.76}$$

$$Var(\hat{\varepsilon}_{n+1}) = \mathbb{E}(\hat{\varepsilon}^2_{n+1}) \tag{2.77}$$

$$Var(\hat{\varepsilon}_{n+1}) = \mathbb{E}\big[(x_{n+1}(\beta - \hat{\beta}) + \varepsilon_{n+1})^2\big], \tag{2.78}$$

$$Var(\hat{\varepsilon}_{n+1}) = \mathbb{E}\big[(x_{n+1}(\beta - \hat{\beta})^2 x'_{n+1}\big] + \mathbb{E}(\varepsilon^2_{n+1}) \tag{2.79}$$

$$Var(\hat{\varepsilon}_{n+1}) = x_{n+1}\mathbb{E}\big[(\beta - \hat{\beta})^2\big]x'_{n+1} + \mathbb{E}(\varepsilon^2_{n+1}), \tag{2.80}$$

$$Var(\hat{\varepsilon}_{n+1}) = x_{n+1}\mathbb{E}\big[(\hat{\beta} - \mathbb{E}(\hat{\beta}))^2\big]x'_{n+1} + \mathbb{E}(\varepsilon^2_{n+1}), \tag{2.81}$$

$$Var(\hat{\varepsilon}_{n+1}) = x_{n+1}Var(\hat{\beta})x'_{n+1} + \mathbb{E}(\varepsilon^2_{n+1}), \tag{2.82}$$

$$Var(\hat{\varepsilon}_{n+1}) = \sigma^2 x_{n+1}(X'X)^{-1}x'_{n+1} + \sigma^2, \tag{2.83}$$

$$Var(\hat{\varepsilon}_{n+1}) = \big[1 + x_{n+1}(X'X)^{-1}x'_{n+1}\big]\sigma^2. \tag{2.84}$$

Sous l'hypothèse de normalité du terme d'erreur, on a alors :

$$\hat{\varepsilon}_{n+1} \sim N\big(0, Var(\hat{\varepsilon}_{n+1})\big),$$

$$\frac{\hat{\varepsilon}_{n+1}}{\sqrt{Var(\hat{\varepsilon}_{n+1})}} \sim N(0,1),$$

$$\frac{\hat{\varepsilon}_{n+1}}{\sigma\sqrt{1 + x_{n+1}(X'X)^{-1}x'_{n+1}}} \sim N(0,1),$$

$$\frac{\hat{\varepsilon}_{n+1}}{S\sqrt{1 + x_{n+1}(X'X)^{-1}x'_{n+1}}} \sim t(n-k),$$

$$Pr\big[-t_{1-\alpha/2} \leq \frac{\hat{\varepsilon}_{n+1}}{S\sqrt{1 + x_{n+1}(X'X)^{-1}x'_{n+1}}} \leq t_{1-\alpha/2}\big] = 1 - \alpha,$$

$$Pr\big[-t_{1-\alpha/2}S\sqrt{1 + x_{n+1}(X'X)^{-1}x'_{n+1}} \leq y_{n+1} - \hat{y}_{n+1} \leq t_{1-\alpha/2}S\sqrt{1 + x_{n+1}(X'X)^{-1}x'_{n+1}} = 1 - \alpha$$

Donc :

$$IC(1-\alpha) = \hat{y}_{n+1} \pm t_{1-\alpha/2}S\sqrt{1 + x_{n+1}(X'X)^{-1}x'_{n+1}} \tag{2.85}$$

Pour un niveau de confiance de $1 - \alpha = 95\%$, il y a donc 95% de chances que la valeur inconnue y_{n+1} soit comprise dans l'intervalle. Cet intervalle est un indicateur de la qualité de la prévision $\hat{y}_{n+1}$: plus (resp. moins) il est large, plus faible (resp. élevée) est la précision.

2.6 Multicollinéarité

Le terme "colinéarité" provient de l'algèbre linéaire et s'utilise pour dire qu'un vecteur est proportionnel à un autre. Nous ne traiterons pas ici d'algèbre linéaire, une interprétation plus pratique de ce que représente la colinéarité en statistique pourrait se réduire à un mot de **corrélation**.
Il faut distinguer entre ce qu'on appelle la multicollinéarité **parfaite** et la multicollinéarité **imparfaite**.

2.6.1 Multicollinéarité parfaite

Dans ce cas, il existe une relation linéaire exacte qui relie un sous-ensemble des variables explicatives. Dans la majorité des cas, il résulte d'un problème logique dans le choix des régresseurs. Il y a plusieurs types de situations où cela peut arriver.
L'exemple le plus connu de ce problème est la soi-disant "trappe des variables dichotomiques", que nous pouvons illustrer avec un exemple simple. Supposons que nous avons un échantillon avec des individus, et une des caractéristiques est le sexe de l'individu. Nous pourrions construire deux variables dichotomiques, dont la première prend la valeur de 1 lorsque l'individu est une femme et 0 autrement, et la deuxième prend la valeur de 1 lorsque l'individu est un homme et 0 autrement. Appelons ces deux variables X_1 et X_2, nous pourrions avoir, par exemple :

$$X_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ . \\ . \\ . \\ 0 \end{pmatrix} \quad , \quad X_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ . \\ . \\ . \\ 1 \end{pmatrix}$$

Il est évident que

$$X_1 + X_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ . \\ . \\ . \\ 1 \end{pmatrix}$$

Maintenant, si nous essayons d'estimer une régression et d'inclure une constante, X_1 et X_2 comme variables explicatives, la constante sera tout simplement la somme de X_1 et X_2. Donc, c'est le cas que nous pouvons exprimer une des variables explicatives comme une combinaison linéaire des autres variables explicatives du modèle de régression.
La multicollinéarité parfaite nous empêchera même d'estimer notre régression. Il est facile de montrer que, en présence d'un problème de multicollinéarité parfaite, la matrice $X'X$ n'est pas de rang plein. Il est impossible de calculer $(X'X)^{-1}$, et l'estimateur $\hat{\beta} = (X'X)^{-1}X'Y$ n'existe même pas.

2.6.2 Multicollinéarité imparfaite

Il s'agit maintenant d'une situation où ce n'est pas le cas qu'une variable explicative est une combinaison linéaire exacte des autres variables explicatives du modèle, mais plutôt une situation où une variable explicative est très fortement corrélée avec une autre variable explicative ou avec une combinaison linéaire de ces variables. Dans ce cas, la matrice $X'X$ n'est pas singulière, mais elle peut souvent être presque singulière. Elle aura une valeur caractéristique presque de 0, et beaucoup plus faible que les autres valeurs caractéristiques de la matrice $X'X$.
La multicollinéarité imparfaite n'est typiquement pas un signe d'une erreur logique dans le choix des variables explicatives du modèle, mais est due aux données utilisées et à la question à laquelle on essaie de répondre en spécifiant le modèle de régression multiple. Il y a une conséquence de cette situation qui est strictement dans le domaine de l'analyse numérique. Avec une matrice $X'X$ qui est presque singulière, même si l'ordinateur est capable de calculer son inverse, le résultat du calcul sera en général sujet à des erreurs

numériques importantes. Les coefficients estimés seront imprécis non au sens statistique mais plutôt au sens numérique.
L'autre conséquence de la multicollinéarité imparfaite est que les écarts types des coefficients estimés risquent d'être plutôt élevés. Par conséquent, les intervalles de confiance pour les coefficients individuels seront très larges et les tests d'hypothèse n'auront pas beaucoup de puissance.
Dans le cas d'un modèle de régression multiple avec deux variables explicatives et erreurs homoscédastiques ($Var(\varepsilon_i/X_{1,i}, X_{2,i}) = \sigma_\varepsilon^2$), nous avons :

$$\hat{\beta}_1 \sim N(\beta_1, \sigma^2_{\hat{\beta}_1})$$

Où

$$\sigma^2_{\hat{\beta}_1} = \frac{1}{n}\Big[\frac{1}{1-\rho^2_{X_1,X_2}}\Big]\frac{\sigma^2_\varepsilon}{\sigma^2_{X_1}}$$

Où ρ_{X_1,X_2} est la corrélation (dans la population) entre les deux variables explicatives de la régression. On voit à partir de cette équation que, toutes choses étant égales par ailleurs, plus élevée est la corrélation entre les deux variables explicatives, plus élevée est la variance de $\hat{\beta}_1$. Dans ce cas, le modèle de régression n'est pas forcément mal spécifié. Par contre, il peut être très difficile sinon impossible d'estimer avec précision et d'établir la significativité d'un coefficient d'une variable dont la corrélation avec au moins une autre variable explicative est très forte.

Preuve

La preuve de cette formule dans le cas où $k = 2$ est relativement facile. Le modèle au départ est donné par

$$Y_i = \beta_1 X_{1i} + \beta_2 X_{2i} + \varepsilon_i$$

Nous avons

$$\bar{Y} = \beta_1 \bar{X}_1 + \beta_2 \bar{X}_2 + \bar{\varepsilon}$$

Où, comme d'habitude, une barre indique la moyenne échantillonnale d'une variable. Ceci nous donne

$$Y_i - \bar{Y} = \beta_1(X_{1i} - \bar{X}_1) + \beta_2(X_{2i} - \bar{X}_2) + (\varepsilon_i - \varepsilon) \tag{2.86}$$

Où

$$Y = X\begin{pmatrix}\beta_1\\ \beta_2\end{pmatrix} + \varepsilon$$

Où par construction la variable dépendante et les variables explicatives ont des moyennes échantillonnales de zéro et où on soustrait la moyenne échantillonnale des erreurs de chaque ε_i (bien sûr, puisque nous n'observons pas les $\bar{\varepsilon}_i$. Nous n'observons pas non plus $\bar{\varepsilon}$). L'estimateur MCO est donné par la formule habituelle :

$$\begin{pmatrix}\hat{\beta}_1\\ \hat{\beta}_2\end{pmatrix} = (X'X)^{-1}X'Y$$

Dans ce cas, la matrice variance-covariance du vecteur de paramètres estimés est donnée par :

$$Q = \frac{\sigma_\varepsilon^2}{n}\begin{pmatrix}\sigma_{X_1}^2 & \sigma_{X_1,X_2}\\ \sigma_{X_1,X_2} & \sigma_{X_2}^2\end{pmatrix}^{-1} \tag{2.87}$$

$$Q = \frac{1}{\sigma_{X_1}^2\sigma_{X_2}^2 - (\sigma_{X_1,X_2})^2}\begin{pmatrix}\sigma_{X_1}^2 & \sigma_{X_1,X_2}\\ \sigma_{X_1,X_2} & \sigma_{X_2}^2\end{pmatrix} \tag{2.88}$$

Ce qui donne :

$$\sigma_{\hat{\beta}_1}^2 = \frac{\sigma_\varepsilon^2}{n}\left(\frac{\sigma_{X_2}^2}{\sigma_{X_1}^2\sigma_{X_2}^2 - (\sigma_{X_1,X_2})^2}\right) \tag{2.89}$$

$$\sigma_{\hat{\beta}_1}^2 = \frac{1}{n}\left(\frac{1}{\sigma_{X_1}^2 - \frac{(\sigma_{X_1,X_2})^2}{\sigma_{X_2}^2}}\right)\sigma_\varepsilon^2 \tag{2.90}$$

$$\sigma_{\hat{\beta}_1}^2 = \frac{1}{n}\left(\frac{1}{1 - \frac{(\sigma_{X_1,X_2})^2}{\sigma_{X_1}^2\sigma_{X_2}^2}}\right)\frac{\sigma_\varepsilon^2}{\sigma_{X_1}^2} \tag{2.91}$$

$$\sigma_{\hat{\beta}_1}^2 = \frac{1}{n}\left(\frac{1}{1 - \rho_{X_1,X_2}^2}\right)\frac{\sigma_\varepsilon^2}{\sigma_{X_1}^2} \tag{2.92}$$

Où ρ_{X_1,X_2}^2 est le coefficient de corrélation entre X_1 et X_2 au carré. En regardant cette expression, il est clair que la variance $\sigma_{\hat{\beta}_1}^2$ du paramètre estimé $\hat{\beta}_1$ va croître avec la valeur absolue du coefficient de corrélation entre X_1 et X_2. On peut aussi montrer que la variance de $\hat{\beta}_2$ est donnée par :

$$\sigma_{\hat{\beta}_2}^2 = \frac{1}{n}\left(\frac{1}{1 - \rho_{X_1,X_2}^2}\right)\frac{\sigma_\varepsilon^2}{\sigma_{X_2}^2} \tag{2.93}$$

Encore une fois, sa variance augmente avec la valeur absolue du coefficient de corréltation entre X_1 et X_2.
La multicollinéarité imparfaite traduit le fait qu'il peut être très difficile

(sinon impossible), statistiquement parlant, d'isoler l'impact individuel de chacune de deux variables explicatives qui sont fortement corrélées. C'est possible que chacune des deux variables soit non significative sur la base d'un test d'hypothèse simple (basé sur une statistique t), tandis qu'un test de l'hypothèse nulle jointe que les deux variables sont non significatives rejette cette hypothèse nulle sur la base d'une statistique F. En interprétant les résultats d'une telle régression, il est important d'insister sur l'importance du bloc de deux variables pour expliquer la variable dépendante, tout en soulignant l'impossibilité d'attribuer l'importance à une variable particulière à cause du problème de multicollinéarité imparfaite.

2.6.3 Comment remédier à la multicollinéarité ?

Lorsque le modèle est utilisé uniquement à des fins de prédictions, les effets de la multicollinéarité sur l'estimation des paramètres peuvent être négligés (Neter et al., 1985, Chap. Il). Par contre, si l'étude de l'effet des paramètres sur la variable dépendante (analyse fonctionnelle) présente un intérêt pour l'utilisateur, des mesures correctives doivent être envisagées. Différentes approches correctives ont été proposées afin de tenir compte du problème de multicollinéarité. La première, l'approche heuristique, est basée sur l'expérience de l'utilisateur, sur sa connaissance des causes de la multicollinéarité et du problème à l'étude. Cette approche, essentiellement subjective, repose sur un choix judicieux des variables explicatives qui seront utilisées dans le modèle. On propose, entre autres, de limiter le plus possible le nombre de variables explicatives, d'éviter l'utilisation de variables explicatives fortement corrélées entre elles, d'utiliser les variables les plus pertinentes pour le problème à traiter. D'autres approches interviennent directement sur l'estimation des paramètres en modifiant la méthode des moindres carrés. C'est le cas, en particulier : de la régression Ridge, Lasso et d'autres.

CHAPITRE 3

LES RÉSEAUX DE NEURONES ET LES ALGORITHMES D'APPRENTISSAGES

3.1 Introduction

La modélisation à partir des données empiriques est une tâche difficile, les méthodes conventionnelle sont souvent insuffisantes face à ce type de données, c'est pourquoi les chercheurs s'orientent vers le développement de nouveaux outils et de nouvelles méthodologies capables de traiter des données de plus en plus complexes. Les méthodes de traitement de la régression linéaire et les outils inéherents sont bien maitrisés dans le cadre de la statistique traditionnelle. Les difficultés du traitement de la régression non linéaire persistent.
Les réseaux de neurones constituent maintenant une technique de traitement de données bien comprise et maitrisée, la mise en œuvre des réseaux de neurones est généralement simple, ils sont composés d'éléments simples (ou neurones) fonctionnant en parallèle, ces éléments ont été fortement inspirés par le systéme nerveux biologique. On peut entrainer un réseau de neurones pour une tâche spécifique (classiffication, régression non linéaire, reconnaissance de caractère, ...) en ajustant les valeurs des connections (ou poids) entre les éléments (neurones).
L'objectif de ce chapitre est double : il s'agit tout d'abord de rappeler les définitions de base relatives aux réseaux de neurones ainsi que les propriétés mathématiques de certains d'entre eux. Ensuite, nous nous attacherons à

détailler certains aspects de leur mise en œuvre, et plus particulièrement de leur apprentissage.

3.2 La Régression et les Réseaux de neurones

Tout comme la plupart des modèles statistiques, les RN sont en mesure d'effectuer différentes catégories de tâches, notamment de la régression et de la classification. Les tâches de régression visent à mettre en relation un certain nombre de variables d'entrée x avec un ensemble de résultats continus d (les variables cible). Par opposition, les tâches de classification visent à affecter des observations aux classes d'une variable cible catégorielle en fonction d'un ensemble de valeurs d'entrée.
L'approche la plus directe, et sans doute la plus simple de l'inférence statistique consiste à considérer que nous pouvons modéliser les données en utilisant une forme fermée de fonction pouvant contenir un certain nombre de paramètres (poids) ajustables que nous pouvons estimer, de sorte que le modèle peut nous donner la meilleure explication possible de nos données. Par exemple, considérons une problématique de régression dans laquelle nous cherchons à modéliser ou approcher une variable cible unique, d, à l'aide d'une fonction linéaire de la variable d'entrée, x. La fonction mathématique que nous utilisons pour modéliser ces relations est donnée simplement par une transformation linéaire f à deux paramètres, connus sous le nom d'ordonnées à l'origine, β_1 et de pente, β_2.

$$d = f(x) = \beta_1 + \beta_2 x$$

Notre tâche consiste à trouver des valeurs de β_1 et de β_2 qui vont permettre de faire le lien entre une entrée x et la variable d. Comme on a vu dans le deuxième chapitre, cette problématique est connue sous le nom de régression linéaire.

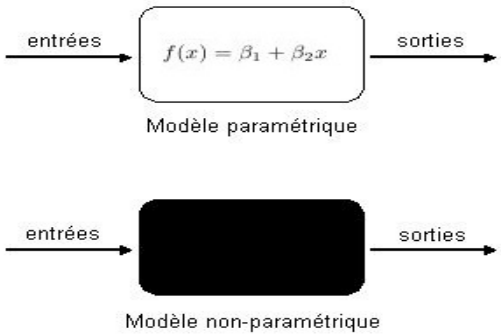


FIGURE 3.1 – La différence entre les modèles paramétriques et les modèles non-paramétriques

Dans les modèles paramétriques, la relation entre les entrées et les sorties s'exprime par une fonction mathématique de forme fermée. Par opposition, dans les modèles non-paramétriques, la relation entre les entrées et les sorties est pilotée par un approximateur (comme un réseau de neurones) que nous ne pouvons pas représenter par une fonction mathématique standard.

3.3 Les étapes de la conception d'un réseau

La première chose à faire n'est pas de choisir le type de réseau mais de bien choisir ses échantillons de données d'apprentissage, de tests et validation. Ce n'est qu'ensuite que le choix du type de réseau interviendra. Afin de clarifier un peu les idées, voici chronologiquement les quatres grandes étapes qui doivent guider la création d'un réseau de neurones.

Choix et préparation des échantillons

Le processus d'élaboration d'un réseau de neurones commence toujours par le choix et la préparation des échantillons de données. Comme dans les cas d'analyse de données, cette étape est cruciale et va aider le concepteur à déterminer le type de réseau le plus approprié pour résoudre son problème. La façon dont se présente l'échantillon conditionne : le type de réseau, le nombre de cellules d'entrée, le nombre de cellules de sortie et la façon dont il faudra mener l'apprentissage, les tests et la validation.

Elaboration de la structure du réseau

La structure du réseau dépend étroitement du type des échantillons. Il faut d'abord choisir le type de réseau : un perceptron standard, un réseau de Hopfield, un réseau à décalage temporel (TDNN), un réseau de Kohonen, un ARTMAP etc ..., dans le cas du perceptron par exemple, il faudra aussi choisir le nombre de neurones dans la couche cachée. Plusieurs méthodes existent et on peut par exemple prendre une moyenne du nombre de neurones d'entrée et de sortie, mais rien de vaut de tester toutes les possibilités et de choisir celle qui offre les meilleurs résultats.

Apprentissage

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré.
Une fois l'architecture d'un réseau de neurones choisie, il est nécessaire d'effectuer un apprentissage pour déterminer les valeurs des poids permettant à la sortie du réseau de neurones d'être aussi proche que possible de l'objectif fixé.

Validation et Tests

Alors que les tests concernent la vérification des performances d'un réseau de neurones hors échantillon et sa capacité de généralisation, la validation est parfois utilisée lors de l'apprentissage. Une fois le réseau calculé, il faut toujours procéder à des tests afin de vérifier que le réseau réagit correctement. Il y a plusieurs méthodes pour effectuer une validation : la validation croisée, la régularisation..., mais pour les tests, dans le cas général, une partie de l'échantillon est simplement écarté de l'échantillon d'apprentissage et conservé pour les tests hors échantillon. On peut par exemple utiliser 60% de l'échantillon pour l'apprentissage, 20% pour la validation et 20% pour les tests. Dans les cas de petits échantillons, on ne peut pas toujours utiliser une telle distinction, simplement parce qu'il n'est pas toujours possible d'avoir suffisamment de données dans chacun des groupes ainsi crées.

3.4 Architectures des réseaux de neurones

3.4.1 Les réseaux réccurents

Un réseau de neurones est dit récurrent s'il existe au moins un cycle dans la structure des liens entre neurones. Le fonctionnement d'un neurone en lui-même ne diffère pas nécessairement de celui des neurones des modèles antérieurs. L'existence de liens à l'envers permet de créer dans le réseau un comportement dynamique, mais lui rajoute aussi une fonction de mémorisation. Ainsi, si l'on voulait créer un réseau de neurones capable de réaliser une addition, seule une connexion vers l'arrière pourrait permettre de mémoriser la retenue et l'utiliser à l'étape suivante de l'addition. De tels réseaux sont donc plus complexes, mais également beaucoup plus puissants.
Le plus célèbre modèle de réseau de neurones récurrent est celui de "Hopfield", proposé en 1982. Dans ce modèle, le réseau est totalement connecté, hormis qu'il n'existe pas de connexion d'un neurone vers lui-même.

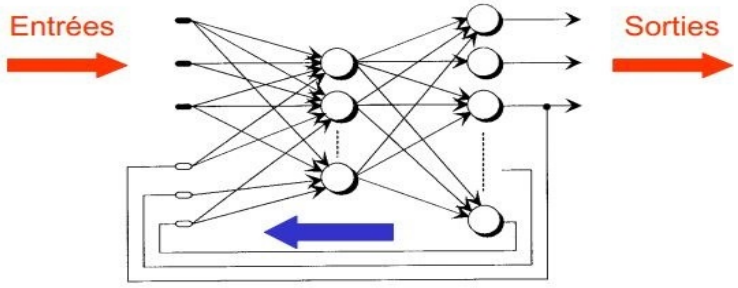


FIGURE 3.2 – Connexions réccurentes

3.4.2 Les réseaux à propagation avant

Appelés aussi "réseaux de type Perceptron", ce sont des réseaux dans lesquels l'information se propage de couche en couche sans retour en arrière possible. Dans un tel réseau, le flux de l'information circule des entrées vers les sorties sans "retour en arrière".

3.4.2.1 Types des réseaux à propagation avant

On peut distinguer dans cette catégorie deux structures différentes dépendant du nombre de couches dans le réseau :

Réseau monocouche

La structure d'un réseau monocouche est telle que des neurones organisés en entrée soient entièrement connectés à d'autres neurones organisés en sortie par une couche modifiable de poids.

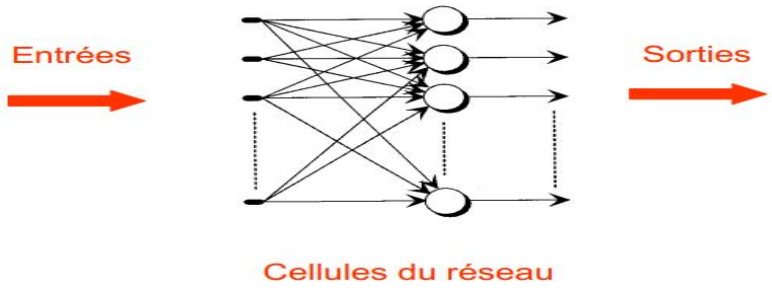


FIGURE 3.3 – Réseau monocouche

Réseau multicouche

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche, et les connexions ne se font qu'avec les neurones de couches avales. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées.

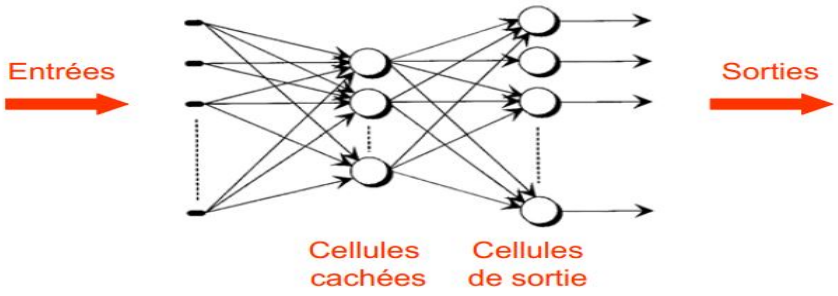


FIGURE 3.4 – Réseau multicouche

3.4.2.2 Propriété fondamentale

La propriété fondamentale des réseaux de neurones est l'approximation parcimonieuse. Cette expression traduit deux propriétés distinctes : d'une part, les réseaux de neurones sont des approximateurs universels et, d'autre part, une approximation à l'aide de réseau de neurones nécessite, en général, moins de paramètres ajustables que les approximateurs usuels.

L'approximation universelle

La propriété d'approximation universelle a été démontrée par (Cybenko 1989) et (Funahashi 89) et peut s'énoncer de la façon suivante :
"Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire".
Cette propriété est vraie pour les neurones présentés précédemment, c'est-à-dire à fonction d'activation sigmoïdale, mais aussi pour les fonctions radiales et les ondelettes.

La parcimonie

Lorsqu'on veut modéliser un processus à partir de données, on cherche toujours à obtenir les résultats les plus satisfaisants possibles avec un nombre minimal de paramètres ajustables. Dans cette optique, (Hornik 1994) a montré que :
Si le résultat de l'approximation (c'est-à-dire la sortie du réseau de neurones) est une fonction non linéaire des paramètres ajustables, elle est plus parcimonieuse que si elle est une fonction linéaire de ces paramètres. De plus, pour des réseaux de neurones à fonction d'activation sigmoïdale, l'erreur commise dans l'approximation varie comme l'inverse du nombre de neurones cachés, et elle est indépendante du nombre de variables de la fonction à approcher. Par conséquent, pour une précision donnée, donc pour un nombre de neurones cachés donné, le nombre de paramètres du réseau est proportionnel au nombre de variables de la fonction à approcher.
Ces résultats s'appliquent aux réseaux de neurones à fonction d'activation sigmoïdale, puisque la sortie de ces neurones n'est pas linéaire par rapport à leurs coefficients. Ainsi, l'avantage des réseaux de neurones par rapport

aux approximateurs universels (tels que les polynômes) est d'autant plus sensible que le nombre de variables de la fonction à approcher est grand : pour des problèmes faisant intervenir une ou deux variables, on peut généralement utiliser indifféremment des réseaux de neurones ou des polynômes. En revanche, pour des problèmes présentant trois variables ou plus, il est généralement avantageux d'utiliser les réseaux de neurones.

De l'approximation de fonction à la modélisation statistique

Les problèmes que l'on rencontre en pratique ne sont que très rarement des problèmes d'approximation de fonction connue. Dans la très grande majorité des cas, on cherche à établir un modèle à partir de mesures, ou, en d'autres termes, à trouver la fonction qui passe "au plus près" d'un nombre fini de points expérimentaux, généralement entachés de bruit. On cherche alors à approcher la fonction de régression du processus considéré, c'est-à-dire la fonction que l'on obtiendrait en calculant la moyenne d'une infinité de mesures effectuées en chaque point du domaine de validité du modèle. Le nombre de points de ce domaine étant lui-même infini, la connaissance de la fonction de régression nécessiterait donc une infinité de mesures en un nombre infini de points. Les réseaux de neurones, en raison de leur propriété fondamentale, sont de bons candidats pour réaliser une approximation de la fonction de régression. C'est ce qui justifie l'utilisation pratique des réseaux de neurones : la recherche d'une approximation de la fonction de régression à partir d'un nombre fini de points expérimentaux. L'utilisation des réseaux de neurones rentre donc complètement dans le cadre de méthodes statistiques d'approximation d'une fonction de régression. De telles méthodes ont été largement développées pour les fonctions de régression linéaires. L'apport des réseaux de neurones réside dans leur capacité à modéliser des processus non linéaires.

3.5 L'apprentissage des réseaux de neurones

L'apprentissage est un processus dynamique et itératif permettant de modifier les paramètres d'un réseau en réaction avec les stimuli qu'il reçoit de son environnement. Le type d'apprentissage est déterminé par la manière dont les changements de paramètre surviennent.

Cette définition implique qu'un réseau se doit d'être stimulé par un environnement, qu'il subisse des changements en réaction avec cette stimulation, et que ceux-ci provoquent dans le futur une réponse nouvelle vis-à-vis de l'environnement. Ainsi, le réseau peut s'améliorer avec le temps.
Dans la plupart des architectures, l'apprentissage se traduit par une modification de l'efficacité synaptique, c'est-à-dire par un changement dans la valeur des poids qui relient les neurones d'une couche à l'autre.
Soit le poids $w_{i,j}$ reliant le neurone i à son entrée j. Au temps t, un changement $\Delta w_{i,j}(t)$ de poids peut s'exprimer simplement de la façon suivante :

$$\Delta w_{i,j}(t) = w_{i,j}(t+1) - w_{i,j}(t) \tag{3.1}$$

et, par conséquent

$$w_{i,j}(t+1) = w_{i,j}(t) + \Delta w_{i,j}(t) \tag{3.2}$$

avec $w_{i,j}(t+1)$ et $w_{i,j}(t)$ représentant respectivement les nouvelle et ancienne valeurs du poids $w_{i,j}$.

Phases d'apprentissage et de test du réseau

Le calage des paramètres du modèle (essentiellement le poids des liaisons entre les différents neurones) est réalisé d'après un algorithme de calcul qui utilise la présentation répétée d'un ensemble de plusieurs couples entrée-sortie connus (exemples qui constituent l'ensemble d'apprentissage).
L'objectif de ce calcul est la minimisation d'une fonction d'erreur entre la réponse désirée et la réponse obtenue à la sortie du modèle.
La validation du modèle se réalisera ensuite sur des exemples (ensemble de test) non utilisés dans le calcul des poids. La performance du réseau est déterminée en fonction du nombre de succès et d'échecs dans la discrimination.
Les paramètres d'ajustement du réseau sont le nombre de neurones cachés et les fonctions d'activation.

3.6 Les types d'apprentissage

On distingue deux grandes classes d'algorithmes d'apprentissage :
L'apprentissage supervisé et l'apprentissage non supervisé.

3.6.1 L'apprentissage supervisé

Dans ce type d'apprentissage, le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie. Ainsi, le réseau va se modifier jusqu'a ce qu'il trouve la bonne sortie, c'est-à-dire celle attendue, correspondant à une entrée donnée.

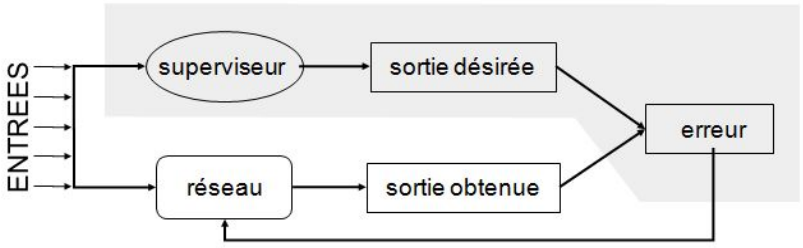


FIGURE 3.5 – L'apprentissage supervisé

But :

Faire la bonne action (le bon choix)

Retour :

La sortie qu'il aurait fallu avoir. On souhaite que le système **généralise**.

3.6.2 L'apprentissage non supervisé

L'apprentissage est qualifié de non supervisé lorsque seules les valeurs d'entrée sont disponibles. Dans ce cas, les exemples présentés à l'entrée provoquent une auto- adaptation du réseau afin de produire des valeurs de sortie qui soient proches en réponse à des valeurs d'entrée similaires (de même nature).

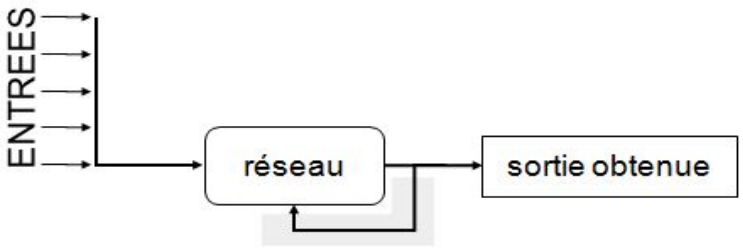


FIGURE 3.6 – L'apprentissage non supervisé

Le taux d'apprentissage

Cette valeur est modifiée -diminuée- manuellement au fur et à mesure que le réseau apprend. Il n'est pas simple de faire manuellement une modification optimale de cette valeur : une valeur trop importante va très vite faire diverger le réseau de la solution recherchée. Les poids tombent dans la zone de saturation des fonctions sigmoïdes utilisées comme fonctions de transition.
Une trop petite valeur ralentira énormément le processus d'apprentissage. Avec un peu d'habitude, on peut bien sûr devenir expert dans les modifications des valeurs de η pendant l'apprentissage et arriver ainsi à obtenir de très bonnes courbes de convergences. Toutefois, ce procédé n'est pas totalement satisfaisant et l'utilisateur novice aimerait bien disposer de procédures automatiques de variation de η.
La valeur optimale de E dépend à la fois du problème à résoudre et de la nature du réseau utilisé. Intuitivement, on peut très bien comprendre que dans une région relativement plate de la fonction de coût, on ait intérêt à aller beaucoup plus vite que dans le cas d'une région ayant des pentes raides. Une valeur importante de η accélère la convergence pour le premier cas tandis que pour le deuxième elle peut faire diverger la procédure.
La connaissance d'une information locale sur la forme de l'espace des erreurs pourrait nous permettre de choisir le bon η, itération après itération et cellule par cellule. Cette connaissance est possible en exploitant l'information des dérivées de la fonction de coût d'ordre supérieur à 1.

3.7 Les règles d'apprentissage

Les règles d'apprentissages diffèrent les uns des autres par la façon dans laquelle sont formulés les ajustements des poids synaptiques.

3.7.1 Apprentissage de Hebb :

L'idée d'un mécanisme "synaptique" de couplage avait été proposée par "Hebb" dès 1949. Cette règle, basée sur des données biologiques, modélise le fait que si des neurones, de part et d'autre d'une synapse, sont activés de façon synchrone et répétée, la force de la connexion synaptique va aller croissant. Il est à noté ici que l'apprentissage est localisé, c'est-à-dire que la modification d'un poids synaptique w_{ij} ne dépend que de l'activation d'un neurone i et d'un autre neurone j. Mathématiquement, cette règle peut s'exprimer de la façon suivante :

$$w_{ij}(t+1) = w_{ij}(t) + \eta y_j(t) x_i(t) \tag{3.3}$$

Où $x_i(t)$ et $y_j(n)$ sont les entrées et les sorties, au temps t des neurones i et j dont le poids de connexion (entre i et j) vaut $w_{ij}(t)$, η est le taux d'apprentissage.

3.7.2 Apprentissage par correction d'erreurs "Delta rule" :

Cette régle est aussi une version modifiée de la loi de Hebb. Les poids des liens entre les neurones sont continuellement modifiés de façon à réduire la différence (le delta) entre la sortie désirée et la valeur calculée de la sortie du neurone. Les poids sont modifiés de façon à minimiser l'erreur quadratique à la sortie du RN. L'erreur est alors propagée des neurones de sortie vers les neurones des couches inférieures, une couche à la fois.
La réponse désirée : $d_k(n)$
Un signal d'erreur est produit : $e_k(n) = d_k(n) - y_k(n)$
Le signal d'erreur $e_k(n)$ enclenche un mécanisme de commande dont le but est d'appliquer une séquence d'ajustement correctifs, pas à pas aux poids synaptiques du neurone k, donc le but est atteint en minimisant une fonction de coût.

$$\varepsilon(n) = \frac{1}{2} e_k^2(n) \tag{3.4}$$

Régle delta :

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \tag{3.5}$$

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) \tag{3.6}$$

η doit choisi avec préauction pour assure la stabilité de la convergence de processus iteratif d'apprentissage.

3.7.3 Règle de Hopfield :

Cette régle se base sur la même hypothèse que la loi de Hebb, mais ajoute une variable supplémentaire pour contrôler le taux de variation du poids entre les neurones avec une constante d'apprentissage qui assure à la fois la vitesse de convergence et la stabilité du RN.
La loi de Hebb et le modèle de Hopfield correspondent à un apprentissage non-supervisé, le réseau se corrigera lui-même jusqu'à atteindre un état stable.

3.7.4 Apprentissage basé sur la mémoire :

Dans l'aprentissage basé sur la mémoire tout ou la majorité d'éxperiences passées sont explicitemment stokées dans une grande mémoire d'éxemples d'entrées-sorties correctement classées :

$$\{(x_i, d_i)\}_i^N = 1 \tag{3.7}$$

Régle du plus proche voisin : le voisinnage locale est définis comme étant l'exemple d'apprentissage qui se situe dans le voisinnage immédiat ou valeur test X_{test}.
$X'_n \in \{X_1 ... X_n\}$ est dit plus proche voisin si :

$$min\ d(X_i, X_{test}) = d(X'_n, X_{test}) \tag{3.8}$$

Si on prend le problème de classification a deux classes C_1 et C_2 ,la réponse désirée d_i prend la valeur 0 pour C_1 et la valeur 1 pour C_2.
Pour la classification d'un vecteur test X_{test} , l'algorithme répond en cherchant les données d'entrainement dans un "voisinnage local" de X_{test}.
tous les algorithmes d'apprentissages basées sur la mémoire impliquent deux ingrédients essentiels :

- Un critère utilisé pour définir le voisinage local du vecteur test X_{test}.
- Une régle d'apprentissage appliquée aux exemples d'entrainement dans le voisinnage local de x_{test}.

3.7.5 Apprentissage compétitif :

Le principe de cet apprentissage est de regrouper les données en catégories. Les patrons similaires vont donc être rangés dans une même classe, en se basant sur les corrélations des données, et seront représentés par un seul neurone, on parle de « winner-take-all ».
Dans un réseau à compétition simple, chaque neurone de sortie est connecté aux neurones de la couche d'entrée, aux autres cellules de la couche de sortie (connexions inhibitrices) et à elle-même (connexion excitatrice). La sortie va donc dépendre de la compétition entre les connexions inhibitrices et excitatrices.
Donc, on peut écrire la règle d'apprentissage compétitif comme suit :

$$\Delta w = \begin{cases} \eta(x - w) \ si\ le\ neurone\ est\ vainqueur \\ 0\ autrement \end{cases} \tag{3.9}$$

Où $0 < \eta < 1$ correspond à un taux d'apprentissage, x est le vecteur d'entrée et w est le vecteur de poids.

3.8 La Fonction de coût

La définition d'une fonction de coût est primordiale, car celle-ci sert à mesurer l'écart entre la sortie du modèle et les mesures faites sur les exemples d'apprentissage. Il existe un grand nombre de fonctions possibles (Bishop 1995). La fonction la plus couramment utilisée, est la fonction dite des "moindres carrés". La fonction d'erreur permet d'évaluer la performance d'un réseau de neurones au cours de l'apprentissage. La fonction de coût indique dans quelle mesure les prévisions du réseau sont proches des valeurs cibles et donc, quel ajustement doit être apporté aux poids par l'algorithme d'apprentissage à chaque itération.

3.8.1 Fonction de coût des moindres carrés

Toutes les fonctions d'erreur utilisées pour l'apprentissage des réseaux de neurones doivent intégrer une certaine mesure des distances entre les valeurs cibles et les prévisions correspondant aux entrées. Une approche courante consiste à utiliser la fonction d'erreur dite de somme des carrés. Dans ce cas, le réseau va apprendre une fonction discriminante. L'erreur de la somme des

carrés est simplement donnée par la somme des différences entre les valeurs cibles et les sorties prévues définies pour l'ensemble d'apprentissage par :

$$E = \sum_{i=1}^{n} (d_i - y_i)^2 \ Pour\ un\ exemple \tag{3.10}$$

$$E = \sum_{i=1}^{n} \frac{1}{n} (d_i - y_i)^2 \ Sur\ un\ ensemble\ d'exemples \tag{3.11}$$

n représente le nombre d'observations d'apprentissage, y_i représente la prévision (sortie du réseau) et d_i représente la valeur cible pour la $i^{ème}$ observation. Plus la différence entre les prévisions du réseau et les valeurs cible sera importante, plus la valeur de l'erreur sera grande, ce qui nécessite alors un ajustement plus important des poids par l'algorithme d'apprentissage.
L'apprentissage consiste à minimiser une fonction de coût à l'aide des algorithmes d'optimisation qui vont être décrit par la suite. Si la sortie du modèle est linéaire par rapport aux paramètres, l'apprentissage s'effectue en une seule étape avec la méthode des moindres carrés ordinaire. Les méthodes itératives qui assurent la décroissance de la fonction de coût et convergent vers un minimum, sont utilisées dans d'autres cas.

3.8.2 Minimisation de la fonction de coût

Le modèle étant non linéaire en ses paramètres, la fonction de coût n'est pas quadratique. La méthode des moindres carrés n'est donc pas applicable. En conséquence, on a recours à des méthodes itératives d'optimisation de la fonction de coût.
Toutes les méthodes d'optimisation utilisent le gradient de la fonction de coût. La première étape de l'apprentissage d'un réseau de neurones consiste donc à calculer le gradient de la fonction de coût, à l'aide de l'algorithme de rétro-propagation.
Une fois le gradient calculé, un algorithme itératif de modification des paramètres est mis en œuvre. Parmi ces derniers, on distingue les méthodes itératives du premier ordre et les méthodes du second ordre.
Les méthodes du premier ordre modifient itérativement les paramètres de manière proportionnelle au gradient de la fonction de coût, avec un coefficient de proportionnalité fixe ou variable au cours du déroulement de l'optimisation. En d'autres termes, l'extrémité du vecteur des paramètres se déplace, à chaque itération, dans la direction du gradient de la fonction de coût.

Dans les méthodes du second ordre, la direction de déplacement du vecteur des paramètres est obtenue par une transformation linéaire du gradient de la fonction de coût, transformation qui fait intervenir la matrice des dérivées secondes de la fonction de coût par rapport aux paramètres (matrice Hessienne). Ces méthodes sont beaucoup plus efficaces que les méthodes du premier ordre. Le choix entre les diverses méthodes du second ordre dépend notamment du nombre de paramètres des modèles étudiés, elles sont reconnues comme étant les plus rapides à converger par rapport à celles du premier ordre.

3.9 Les algorithmes du premier ordre

3.9.1 Rétro-propagation de gradient

L'approche la plus utilisée pour la minimisation de la fonction de coût est basée sur la méthode du gradient. On commence l'entrainement par un choix aléatoire des vecteurs initiaux du poids.On présente le premier vecteur d'entrée, une fois on a la sortie du réseau, l'erreur correspondante et le gradient de l'erreur par rapport à tous les poids sont calculés. Les poids sont alors ajustés. On refait la même procédure pour tous les exemples d'apprentissage. Les poids du réseaux sont ajustés dans le sens du gradient négatif de la fonction coût. Ce processus est répété jusqu'à ce que les sorties du réseau soient suffisamment proches des sorties désirées.

L'entrainement de retro-propagation

On va présenter l'ordre du processus d'une manière plus détaillée dans cet exemple :

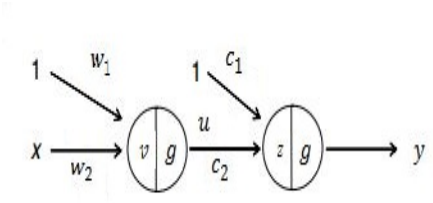


FIGURE 3.7 – Exemple formel

L'erreur de réseau vaut :

$$E = \frac{1}{2}(d-y)^2 \tag{3.12}$$

Le gradient d'erreur : Couches de sorties

On va utiliser le principe de la chaîne de dérivation de fonctions :

$$\frac{\partial E}{\partial c} = \frac{\partial E}{\partial y}.\frac{\partial y}{\partial z}.\frac{\partial z}{\partial c} \tag{3.13}$$

Sachant que :

$$\frac{\partial E}{\partial y} = y - d, y = \frac{1}{1+e^{-z}} \tag{3.14}$$

$$\frac{\partial y}{\partial z} = -\frac{e^{-z}(-1)}{(1+e^{-z})^{-2}} = \frac{e^{-z}}{(1+e^{-z})^{-2}} \tag{3.15}$$

$$1 + e^{-z} = 1/y, e^{-z} = 1 - y/y \tag{3.16}$$

Remplaçant ces derniers dans l'équation ci-dessus on trouve :

$$\frac{\partial y}{\partial z} = \frac{1-y/y}{1/y^2} = y(1-y). \tag{3.17}$$

On a,

$$z = c_1 + c_2 u \tag{3.18}$$

Et le troisième dérivé : $\frac{\partial z}{\partial c}$, vaut

$$\frac{\partial z}{\partial c_1} = u, \frac{\partial z}{\partial c_2} = 1 \tag{3.19}$$

Les trois dérivées ont été obtenus, la dérivée d'erreur pour les deux poids peut être présenté par :

$$\frac{\partial E}{\partial c_1}(y-d)y(1-y) = Q, \tag{3.20}$$

$$\frac{\partial E}{\partial c_2}(y-d)y(1-y)u = Qu. \tag{3.21}$$

Le gradient d'erreur : Couches cachées

La dérivée d'erreur peut être formulé comme suit :

$$\frac{\partial E}{\partial w} = \left(\frac{\partial E}{\partial y}.\frac{\partial y}{\partial z}.\frac{\partial z}{\partial u}\right).\frac{\partial u}{\partial v}.\frac{\partial v}{\partial w} \tag{3.22}$$

Le premier terme des trois dérivées entre parenthèses dans (3.22) présente éssentiellement $\frac{\partial E}{\partial u}$, les deux premières dérivées sont connues précédemment sous Q, il nous reste que le dernier terme $\frac{\partial z}{\partial u}$. On peut facilement calculer le, car $z = c_1 + c_2 u$, donc :

$$\frac{\partial z}{\partial u} = c_2 \tag{3.23}$$

Remplaçant les trois dérivées dans (3.22) on obtient :

$$\frac{\partial E}{\partial u} = \frac{\partial E}{\partial y}.\frac{\partial y}{\partial z}.\frac{\partial z}{\partial u} = Qc_2 \tag{3.24}$$

Le deuxième terme de $\frac{\partial E}{\partial w}$ et $\frac{\partial E}{\partial v}$ peut être écrit sous :

$$\frac{\partial u}{\partial v} = u(1-u) \tag{3.25}$$

Sachant que $v = w_1 + w_2 x$,le dernier composant $\frac{\partial v}{\partial w}$ est écrit sous la forme suivante :

$$\frac{\partial v}{\partial w_2} = x, \frac{\partial v}{\partial w_1} = 1 \tag{3.26}$$

On obtient finallement :

$$\frac{\partial E}{\partial w_2} = Qc_2 u(1-u)x = Kx \tag{3.27}$$

$$\frac{\partial E}{\partial w_2} = Qc_2 u(1-u) = K \tag{3.28}$$

La sortie de réseau :

$$y = \frac{\partial E}{\partial w_1} + \frac{\partial E}{\partial w_2} + \frac{\partial E}{\partial c_1} + \frac{\partial E}{\partial c_2} \tag{3.29}$$

Et à partir de (3.12) on trouve E.
Il est possible maintenant d'appliquer les régles d'apprentissage et de modifier les poids synaptiques.

L'apprentissage de retro-propagation

Il y a deux façons d'appliquer l'algorithme :

« Batch »

Mise à jour des poids après la présentation de tous les exemples, on additionne les erreurs/dérivées sur tous les exemples et on ajuste les poids à la fin du cycle (époque) comme suit :

$$d_m = \sum_{i=1}^{N} \left[\frac{\partial E}{\partial w_m} \right]_n \tag{3.30}$$

d_m représente le gradient total , où :

$$\Delta w_m = -\eta d_m, \tag{3.31}$$

η : le taux d'apprentissage et :

$$w_{m+1} = w_m + \Delta w_m \tag{3.32}$$

$(-)$ présente le signe de descent

« On-line »

Mise à jour des poids après chaque exemple, on calcule les dérivées et on ajuste les poids à chaque présentation d'un exemple, cet apprentissage est plus rapide au départ mais plus long en général.

Avantages et inconvénients de l'algorithme de rétropropagation :

Il y a deux **avantages** principaux de cet algorithme :
- Le premier c'est qu'il est classé parmi les méthodes les plus importantes pour l'approche séquentielle.
- Le deuxième avantage c'est que cet algorithme a employé l'information de gradient. Cet algorithme, en plus de permettre la propagation du signal provenant des cellules d'entrée vers la couche de sortie, permet, en suivant le chemin inverse, rétropropager l'erreur commise en sortie vers les couches internes.

Pour **les inconvénients** on peut citer les points suivants :

- Si η est trop grand, l'algorithme peut entrainer à une augmentation de E et probablement aux oscillations divergentes ayant pour résultat une panne complète dans l'algorithme.
- Si η est choisi très petit, la recherche peut etre extrémement lente (le temps de calcul est très long).
- Dans la pratique, une valeur utilisé de η constante mene, généralement, à améliorer les résultats quoique la garantie de la convergence est perdue.
  Cette situation est illustrée sur la figure (3.8), qui représente les lignes de niveau de la fonction de coût (fonction de deux variables w_1 et w_2) et l'évolution du point représentatif du vecteur w au cours du déroulement de l'algorithme.
- Au voisinage d'un minimum de la fonction de coût, le gradient de cette dernière tend vers 0 : l'évolution du vecteur des coefficients devient donc très lente. Il en va de même si la fonction de coût présente des "plateaux" où sa pente est très faible ; ces plateaux peuvent être très éloignés d'un minimum, et, dans la pratique, il est impossible de savoir si une évolution très lente du gradient est due au fait que l'on est au voisinage d'un minimum, ou que l'on se trouve sur un plateau de la fonction de coût.
- Si la courbure de la surface de coût varie beaucoup, la direction du gradient peut être très différente de la direction qui mènerait vers le minimum, c'est le cas si le minimum recherché se trouve dans une "vallée" longue et étroite (les courbes de niveau sont des ellipsoides allongés au voisinage du minimum), comme on le voit également sur la figure (3.8).

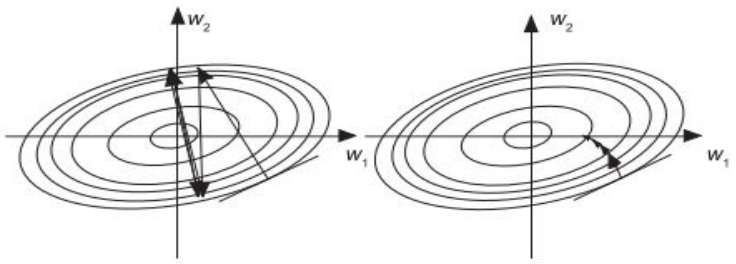


FIGURE 3.8 – Minimisation de la fonction de coût par la méthode du gradient simple

– Dans la plupart des points sur la surface d'erreur, le gradient local ne se dirige pas vers le minimum. Alors, la descente de gradient prend beaucoup de petites mesures pour atteindre le minimum, et c'est donc clairement un procédé totalement inefficace.

En conclusion

Bien que l'algorithme de rétropropagation soit le plus utilisé pour l'apprentissage supervisé, son implantation se heurte à plusieurs difficultés techniques. Il n'existe pas de méthodes permettant de :

– Trouver une architecture appropriée (nombres de couches, nombre de neurones).
– Choisir une taille et une qualité adéquate d'exemples d'entraînement.
– Choisir des valeurs initiales satisfaisantes pour les poids, et des valeurs convenables pour les paramètres d'apprentissage permettant d'accélérer la vitesse de convergence.
– Problème de la convergence vers un minimum local, qui empêche la convergence et cause l'oscillation de l'erreur.

Plusieurs approches ont été proposées pour remédier à ces problèmes. Une des techniques d'accélération est celle de la création dynamique des neurones, un neurone est ajouté chaque fois que l'erreur se stabilise à un niveau inacceptable. Les méthodes de recherche unidimensionnelle (notamment celle qui est présentée dans les compléments théoriques et algorithmiques dans ce qui suit), fondées sur des principes solides, sont recommandées. Pour faire face aux deux autres problèmes, on utilise des méthodes du second ordre qui, au lieu de modifier les coefficients uniquement en fonction du gradient de la fonction de coût, utilisent les dérivées secondes de cette dernière.
Les grandes lignes des méthodes du premier ordre les plus fréquemment utilisées, ainsi que les méthodes du seconde ordre, sont présentées dans les sections suivantes.
Dans les sections qui ce suit on va étudier des autres méthodes qui sont plus efficaces et qui deviennent actuellement les plus employées dans le domaine des réseaux de neurones.

Remarque 3.1 *Dans la rétropropagation, le même taux d'apprentissage s'applique à tous les poids. Par conséquent, tous les poids changent de la même manière. Cependant, en réalité, quelques poids peuvent être plus près de l'optimum ou ou avoir une influence plus forte sur l'erreur que les autres et, en*

conséquence, plus de flexibilité et une vitesse plus élevée de convergence pourraient être réalisées si chaque poids devaient être ajustés indépendamment d'une façon adaptative. Pour ajuster les poids, dans les deux algorithmes considérés au-dessous chaque poids w_m a un taux d'apprentissage associé de sorte que les mises à jour soient donné par

$$\Delta w_m = -\eta_i \frac{\partial E}{\partial w_m} \tag{3.33}$$

3.9.2 La méthode de Delta bar delta (taux d'apprentissage adaptatif)

Dans la rétropropagation discuté ci-dessus, le même taux d'apprentissage s'applique à tous les poids. Par conséquent, tous les poids changent de la même manière. Cependant, en réalité, quelques poids peuvent être plus près de l'optimum ou avoir une influence plus forte sur l'erreur que les autres et, en conséquence, plus de flexibilité et une vitesse de convergence plus élevée pourraient être réalisées si chaque poids devaient être ajustés indépendamment d'une façon adaptative.
La méthode connue sous le nom de "Delta bar delta" propose une telle taux d'apprentissage pour les différents poids. Dans cette méthode, chaque poids a son propre taux d'apprentissage et il est ajusté pendant chaque itération : Si la direction dans laquelle l'erreur diminue au point courant, comme indiqué par le gradient d'erreur, est identique que la direction dans laquelle l'erreur a été décroissante récemment, alors le taux d'apprentissage est augmenté. Cependant, si le gradient est dans la direction opposée, le taux d'apprentissage est diminué.
La direction récente dans laquelle l'erreur a été décroissante jusqu'à l'époque m est exprimée par f_m :

$$f_m = \mu f_{m-1} + (1-\mu) d_{m-1} \tag{3.34}$$

μ détermine les gradients les plus récents qui ont une influence plus forte sur le f_m, c.-à-d la direction dans laquelle l'erreur a été décroissante récemment. La durée de « récent » est déterminée par la valeur de μ, qui est une constante entre 0 et 1. L'algorithme est commencé par un taux d'apprentissage individuel, $\eta_1, ..., \eta_n$, tous les η_n sont associé à une petite valeur,

$$\eta_m = \begin{cases} \eta_{m-1} + u & si\ d_m.f_m > 0 \\ \eta_{m-1} \times d & si\ d_m.f_m < 0 \\ \eta_m\ autrement, \end{cases} \tag{3.35}$$

Où u (up) et d (down) sont des constants réglés à la main avec $u > 1$ et $d < 1$.

Avantages et inconvénients

- Dans cette méthode, on a besoin d'une seule session d'entrainement et l'apprentissage n'exige pas la recherche des paramètres optimaux à travers l'essai et l'erreur.
- Le problème avec cette approche est qu'une nouvelle constante doit être réglé à nouveau par l'utilisateur et sa valeur peut être également vu comme un problème fortement dépendant.
- Il est difficile de trouver un "u" approprié, les petites valeurs peut être vu comme une conséquence des adaptations lentes tandis que les grandes valeurs met le processus d'apprentissage en danger.

3.9.3 La méthode de la descente la plus raide (steepest descent)

Quand on connaît les premières dérivées, il est naturel de suivre la direction inverse du gradient pour chercher un minimum, puisque c'est dans cette direction que la fonction d'erreur décroit le plus (cette technique a été utilisée par Cauchy au $19^{ème}$ siècle). le même principe des deux premières méthodes est employés ici, l'erreur est toujours réduite si le gradient est négatif. Le taux d'apprentissage qui est le même pour tous les poids, est adapté intérieurement pendant l'entrainement.
La mise à jour des poids dans cette approche est différente de celui utilisé dans "Delta bar delta". Spécifiquement, commençant par une valeur initiale, η est doublé dans chaque étape (époque d'essai). Ceci rapporte une mise à jour préliminaire pour les poids.
Si l'EQM ne diminue pas avec ce taux d'apprentissage, les poids font un retour à leurs valeurs originales, le taux d'apprentissage est divisé en deux, et l'entrainement est continuée. Si l'EQM ne diminue toujours pas, η est divisé en deux jusqu'à ce qu'on atteigne un taux d'apprentissage auquel l'EQM est diminue.
l'ajustement final des poids est fait seulement après qu'on obtient un taux d'apprentissage approprié et qui réduit l'EQM. En ce moment, η est doublés encore et une nouvelle étape est commencée.

La recherche est continue jusqu'on obtient la relation suivante :

$$\frac{E(w_m) - E(w_{m+1})}{E(w_m)} \leq E_{min} \tag{3.36}$$

$$\eta \leq \eta_{min}$$

où $E(w_m)$, $E(w_{m+1})$ sont les erreurs pour les époques précédentes et les époques au point courant. E_{min} et η_{min} sont des constantes réglés par l'utilisateur.

Avantages et inconvénients

- Cette méthode est efficace plus loin du minimum.
- Elle est capable d'éviter les minimums locaux.
- Elle est plus lent près du minimum.
- La recherche linéaire peut poser des problèmes.
- Elle pourrait des vallées de « zigzag » vers le bas.
- Elle coûte cher en temps de calcul.

3.9.4 La méthode de QuickProp

La méthode suit cet argument : l'objectif de l'étude est de trouver rapidement les poids optimum aux quels le dérivé d'erreur est 0.
Supposons le dérivé après la dernière époque $m-1$ était d_{m-1} et cela il mené à une variation de poids de Δw_{m-1}, si le dérivé pour l'époque courante m est d_m, Δw_m est calculé par :

$$\Delta W_m = \frac{d_m}{d_{m-1} - d_m} \Delta w_{m-1} \tag{3.37}$$

Où (3.37) est une approximation de la courbure, qui est le dérivé du gradient de la surface d'erreur au poids w. Ainsi, plus la courbure est haute, plus la variation de poids est inférieure, et vice versa. Par conséquent, la méthode est répétée jusqu'à ce qu'on atteint les poids optimum.

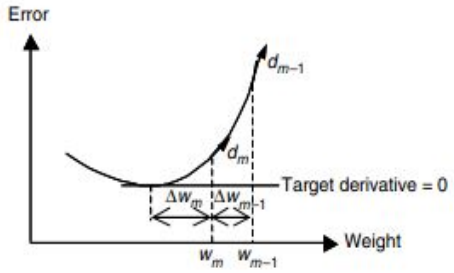


FIGURE 3.9 – La méthode de QuickProp et l'inclusion de la courbure de la surface d'erreur.

Avantages et inconvénients

- Cette méthode est très efficace parce qu'il n'y a aucun paramètre à ajuster et l'erreur descend beaucoup plus rapidement au début de l'entrainement.
- Dans QuickProp, la courbure de la surface d'erreur est implicitement impliquée, les méthodes plus avancées qui emploient explicitement l'information de courbure sont trés éfficaces pour produire une plus grandes accélération et exactitude.

3.10 Les méthodes de second ordre

Dans les méthodes de second ordre, la pente et la courbure au point courant dans l'espace de poids est déterminée. La pente, qui est la première dérivée de l'erreur, indique le taux de changement d'erreur. La courbure indique le taux auquel la pente elle-même change aux poids courants. La courbure de la surface d'erreur à un point, est exprimée par le deuxième dérivé de l'erreur par rapport aux poids $\frac{\partial^2 E}{\partial w^2}$, ainsi, la variation des poids peut être exprimé par :

$$\Delta w = -\frac{\partial E/\partial w}{\partial^2 E/\partial w^2} \tag{3.38}$$

Dans cette section, on va présenté EQM avec la fonction des poids par

$$E(w) = \frac{1}{N}\sum_{i=1}^{N}[d_i - y_i]^2 = \frac{1}{N}\sum_{i=1}^{N}[d_i - f(w, x_i]^2 \tag{3.39}$$

Toutes les méthodes de minimisation d'erreur sont semblables du fait où elles sont itératif. Commençant par les valeurs initiales pour les poids, elles mettent à jour incrémentalement les poids dans la direction négative du gradient.
Les algorithmes d'optimisation de seconde ordre se servant d'informations sur la courbure de la fonction d'erreur.

3.10.1 La méthode de Newton et de Gauss-Newton

Elle utilise la dérivée seconde (courbure) de la fonction de coût pour atteindre le minimum rapidement. La modification des paramètres se fait à travers la formule suivante :

$$\Delta w_m = -H^{-1} d_m \tag{3.40}$$

Dans ce cas, le taux d'apprentissage est constant et égal à 1. La direction de descente est une fonction du Hess et du Gradient. Dans GN, à chaque époque, le taux d'apprentissage η est 1, et il est seulement accepté si EQM diminue pour cette valeur. Autrement, il est divisé en deux maintes et maintes fois jusqu'à une valeur pour laquelle les diminutions de l'EQM est atteintes. Alors les poids sont ajustés et une nouvelle époque est commencé.

$$\Delta w_m = -\eta H^{-1} d_m \tag{3.41}$$

Les critères d'arrêt sont semblables à ceux dans la méthode de "steepest descent" et sont répétés ci-dessous :

$$\frac{E(w_m) - E(w_{m+1})}{E(w_m)} \leq E_{min} \tag{3.42}$$

$$\eta \leq \eta_{min}$$

Où E_{min} et η_{min} sont les niveaux minimum acceptables pour l'EQM et η, respectivement.

Avantages et inconvénients

– Si la fonction de coût est quadratique, l'algorithme atteint la solution en une seule itération.

- Si ce n'est pas le cas, cette méthode est efficace au voisinage d'un minimum. Néanmoins, la matrice Hessienne doit être définie positive pour que la méthode puisse converger vers le minimum. En pratique, cette condition n'est pas toujours vérifiée, et par la suite la méthode peut ne pas converger.
- La méthode de Newton n'est pas trop employée à cause des calculs compliqués du Hessien.
- Cet approche est bien adaptée surtout pour les problèmes de petites dimensions puisque le calcul de la matrice Hessienne est facile. Alors que si le problème présente un grand nombre de variables, il est généralement conseillé de coupler celle-ci avec la méthode du gradient conjugué ou une méthode de Quasi-Newton.

3.10.2 La méthode de Quasi-Newton

Dans de nombreux problèmes, le calcul et l'inversion de la matrice Hessienne du critère, à chaque étape de l'algorithme, risquent de rendre trop coûteuse, voire impossible, l'utilisation de la méthode de Newton, et la forme du critère ne suggère plus. Les méthodes de type Quasi-Newton pallient cette difficulté en remplaçant la matrice Hessienne du critère ou, plus directement encore, son inverse, par une approximation actualisée à chaque étape.
La modification des paramètres s'écrit :

$$\Delta w_m = -\eta_m M_{m-1} d_m \tag{3.43}$$

La suite M_{m-1} est construite de façon à converger vers l'inverse de la matrice Hessienne, l'approximation M_m de l'inverse de cette matrice au point x_m est calculée à partir de l'approximation précédente M_{m-1} et des accroissements $v_m = x_m - x_{m-1}$ et $a_m = d_m - d_{m-1}$ du point courant et du gradient de telle sorte que la matrice M_m reste symétrique, définie positive et vérifie la relation caractérisyique des méthodes de type Quasi-Newton.

$$M_m a_m = v_m \tag{3.44}$$

À la première itération, la matrice M_0 est prise égale à la matrice identité. Parmi toutes les méthodes de Quasi-Newton la méthode de BFGS permet de réduire très significativement le nombre d'itérations avant la convergence,cependant l'approximation de H^{-1} est correcte lorsque la méthode est

proche d'un minimum de la fonction d'erreur. Ainsi,comme le conseille (Dreyfus et al., 2002), ne disposant pas de règles théoriques pour connaître le moment du passage du gradient à BFGS, l'utilisateur doit par conséquent procéder partâtonnements. Cette méthode est équivalente au gradient conjugué selon Polak-Ribière.

Avantages et inconvénients

- L'utilisation de Quasi-Newton est restreint, dans la pratique, au réseau de neurone de petite taille.
- La vitesse de convergence est beaucoup plus grande que celle des méthodes précédentes, de plus elle est relativement insensible au choix du pas d'apprentissage.
- Il est préférable de commencer la minimisation par une méthode de gradient simple, puis d'utiliser la méthode BFGS lorsqu'on estime être proche d'un minimum.
- Cet algorithme converge en une seule itération pour une fonction quadratique. C'est donc un algorithme qui est inefficace loin du minimum de la fonction et très efficace près du minimum.
- Cette méthode est bien reconnue par son efficacité. Mais le principal inconvénient réside dans le calcul des dérivées secondes de f qui s'avère le plus souvent coûteux et très difficile à réaliser, Un certain nombre d'algorithmes se proposent ainsi de contourner cette difficulté en utilisant des approximations de la matrice Hessienne. On peut mentionner le cas particulier où f peut s'écrire sous forme de moindres carrés, on obtient alors une approximation de la matrice Hessienne en ne considérant que les produits des gradients.
- Une méthode Quasi-Newtonienne, n'est efficace que si elle est appliquée au voisinage d'un minimum.

3.10.3 La méthode de Gradient conjugué

Quand les premières dérivées sont calculées, une méthode plus élégante peut être utilisée, celle des gradients conjugués. Dans les méthodes de gradients conjugués (Hestenes et Stiefel, 1952) les directions de recherche successives sont mutuellement conjuguées par rapport à la matrice Hessienne. La méthode du gradient conjugué possède la plupart des avantages de la méthode de Newton et Quasi-Newton, mais sans l'inconvénient d'avoir à calculer

et à inverser la matrice Hessienne. Elle est basée sur le concept des vecteurs conjugués, d'une part, ainsi de calculer le minimum le long de cette direction de descente, d'autre part. Cette étape, appelée méthode de recherche d'une ligne, est une minimisation à une dimension de lafonction objectif. On obtient ainsi un nouveau point où l'on recalcule une nouvelle direction de descente et ainsi de suite jusqu'à la convergence.
Si on note p_{m-1} la direction conjuguée de recherche, on peut choisir la prochaine direction par

$$p_m = -d_m + \beta_m p_{m-1} \tag{3.45}$$

Dont le scalaire β_m peut calculé de manière suivante

$$\beta_m = \frac{d_m^t d_m}{d_{m-1}^t d_{m-1}} \tag{3.46}$$

La modification des poids vaut

$$\Delta w_m = \eta_m p_m \tag{3.47}$$

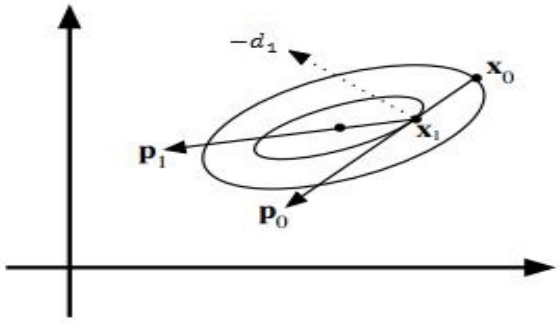


FIGURE 3.10 – Illustration de la méthode du gradient conjugué

Avantages et inconvénients

- L'algorithme de gradient conjugué fournit une technique de minimisation qui exige seulement l'évaluation de la fonction d'erreur et de son gradient.
- Dans l'algorithme de gradient conjugué et pour une fonction d'erreur non linéaire générale, La matrice Hessienne n'a pas besoin d'être définie positive.

- La technique est censée requérir un nombre d'étapes limité par le nombre de paramètres dans le cas d'un coût quadratique. Les méthodes vu précédemment sont appelés "méthodes de premier ordre". On va passer aux méthodes d'optimisation du second ordre qui sont des méthodes itératives de descente du gradient qui consistent à remplacer la fonction de coût par son approximation quadratique au voisinage de point courant.
- Dans les méthodes de second ordre, la courbure de la surface d'erreur, dénotée par le deuxième dérivé de la surface d'erreur est employée pour guider plus efficacement l'erreur au point minimal de la surface d'erreur.

3.10.4 Méthode de Powell-Beale (cgb)

Rétro-propagation du gradient conjugué avec redémarrage Powell-Beale. Pour tous les algorithmes du gradient conjugué, la direction de recherche est périodiquement remise à 0 à la borne négative du gradient. Le point de réinitialisation norme se produit lorsque le nombre d'itérations est égal au nombre de paramètres de réseau (poids et biais), mais il existe d'autres méthodes de réinitialisation qui peuvent améliorer l'efficacité de l'entraînement. Une telle méthode de réinitialisation a été proposée par Powell, sur la base d'une version antérieure posée par Beale. Cette technique redémarre s'il y a très peu orthogonalité laissé entre le gradient actuel et le gradient précédent. Cela a été testé par l'inégalité suivante :

$$\mid g_k^T - g_k \mid \geq 0.2 \parallel g_k \parallel^2 . \tag{3.48}$$

3.10.5 Méthode de Fletcher - Reeves(cgf)

Rétro-propagation du gradient conjugué avec les mises à jour Fletcher - Reeves.
Tous les algorithmes de gradient conjugué commencent par la recherche dans la direction de la plus grande pente (négative du gradient) de la première itération. $p_0 = -g_0$.
Une recherche en ligne est ensuite effectuée pour déterminer la distance optimale pour se déplacer le long de la direction de la recherche en cours :

$$x_{k+1} = x_k \alpha_k p_k. \tag{3.49}$$

Ensuite, la prochaine direction de recherche est déterminée de telle sorte qu'elle est conjuguée à des directions de la recherche précédente. La procédure

générale pour la détermination de la nouvelle direction de recherche est de combiner la nouvelle direction de descente plus raide avec la direction de la recherche précédente :

$$p_k = -g_k + \beta_k p_{k-1}. \tag{3.50}$$

Les différentes versions de l'algorithme du gradient conjugué se distinguent par la façon dont la constante β_k est calculée. Pour la Fletcher-Reeves mettre à jour la procédure est :

$$\beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{3.51}$$

Ceci est le rapport entre le carré de la norme du gradient de courant à la norme du gradient au carré du précédent.

3.10.6 Méthode de Polak-Ribière

Conjugué rétro-propagation du gradient avec les mises à jour Polak-Ribière
Une autre version de l'algorithme de gradient conjugué a été proposée par Polak et Ribière. Comme pour l'algorithme de Fletcher-Reeves, traincgf, la direction de recherche à chaque itération est déterminée par

$$p_k = -g_k + \beta_k p_{k-1}. \tag{3.52}$$

Pour la mise à jour Polak-Ribière, la β_k constante est calculée par

$$\beta_k = \frac{\Delta g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{3.53}$$

Ceci est le produit interne de la variation précédente du gradient avec le gradient actuel divisé par le carré de la norme du gradient précédent.

3.10.7 La méthode de Levenberg Marquardt

La méthode de Levenberg-Marquard est utilisée pour effectuer l'apprentissage du réseau de neurones par minimisation de la fonction de coût. Elle est basée sur la méthode de descente du gradient de second ordre.
L'algorithme de Levenberg-Marquardt (LM) est une amélioration de la méthode classique de Gauss-Newton dans la résolution des problèmes de régression non-linéaire des moindres carrés, Il s'agit de la méthode recommandée

pour les problèmes (de régression) non-linéaires des moindres carrés, car il est plus efficace par rapport aux autres algorithmes d'optimisation.
Cette méthode est très proche de la méthode de Gauss Newton décrite précédemment. La seule différence réside dans l'introduction d'un paramètre e^{λ}, appelé paramètre de Levenberg-Marquardt, permettant de stabiliser la méthode de Gauss-Newton. Ce paramètre est actualisé automatiquement en fonction de la convergence de chaque itération. il est ajouté à la dérivée seconde.
Une stabilisation est possible grâce à un procédé réitératif (si une itération diverge, on la recommence au départ en augmentant le paramètre e^{λ} jusqu'à obtenir une itération convergente).
La variation des poids pour tout les poids est exprimée sous la forme suivante

$$\Delta w_m = -\frac{d_m}{(H_m + e^{\lambda} I)} \tag{3.54}$$

Où I représente la matrice d'identité.
λ est choisit automatiquement, Commençant par une valeur initiale de λ,l'algorithme essaye à se diminuer sa valeur par des incréments de $\Delta\lambda$ dans chaque époque.
Si l'EQM n'est pas réduit, λ est augmenté d'une façon répetée jusqu'on atteint le minimum.
Lorsque λ est petit la méthode de LM est semblable au méthode de GN.
Lorsque λ est grand la méthode de LM est semblable au méthode de steepest descent.
Ainsi, la méthode de LM est un algorithme hybride qui combine les avantages de steepest descent et des méthodes de GN pour produire une méthode plus efficace que l'une ou l'autre de ces deux méthodes.
L'apprentissage se termine avant le nombre spécifique d'époques si les conditions suivantes sont remplies :

$$\lambda > 10\Delta\lambda + Max[H] \tag{3.55}$$

$$\frac{E(w_m) - E(w_{m+1})}{E(w_m)} \leq E_{min} \tag{3.56}$$

Où $Max[H]$ est le maximum des valeurs propres de la matrice Hessienne qui donne la garantit qu'une solution est atteinte d'une façon stable en bas de la courbe d'erreur.

Avantages et inconvénients

- La principale motivation du choix de l'algorithme de Lenvenberg-Marquardt repose sur la taille de la matrice du Hess en fonction de la quantité de données de la base d'apprentissage, du coût moindre des calculs et de la garantie rapide de la convergence vers un minimum.
- La matrice Hessiene est toujours définie positive ce qui assure la convergence vers un minimum de la solution .
- Il ne faut pas que le nombre de poids constituant le réseau soit plus grand que 200 car cette méthode devient inéfficace en terme de rapidité de calcul.

3.11 Extreme Learning Machine

Les réseaux de neurones, ont été largement utilisés dans de nombreux domaines en raison de leur capacité à approximer des fonctions non linéaires complexes directement à partir des données et pour fournir des modèles pour une grande classe de phénomènes naturels et artificiels qui sont difficiles à manipuler en utilisant des techniques paramétriques classiques. D'autre part, il y a un manque d'algorithmes d'apprentissage rapides pour les réseaux de neurones.
Les algorithmes d'apprentissage traditionnels sont généralement beaucoup trop lents que nécessaire. On peut voir que cela peut prendre plusieurs heures, plusieurs jours, et encore plus de temps pour entraîner les réseaux de neurones en utilisant des méthodes traditionnelles. D'un point de vue mathématique, la recherche sur les capacités d'approximation des réseaux de neurones à propagation avant a mis l'accent sur deux aspects :

- Approximation universelle sur ensembles d'entrée compacts.
- Approximation dans un ensemble fini des échantillons de formation.

De nombreux chercheurs ont exploré la capacité d'approximation universelle des RN à propagation avant.
Traditionnellement, tous les paramètres des réseaux doivent être ajustés et donc il existe la dépendance entre les différentes couches de paramètres (poids et biais).
Dans les dernières décennies différentes méthodes du gradient ont été utilisés dans divers algorithmes apprentissage des RN à propagation avant.
Cependant, il est clair que les méthodes d'apprentissage fondées sur la des-

cente du gradient sont généralement très lents, ils peuvent aussi facilement converger vers des minimaux locaux. Beaucoup d'étapes itératives peuvent être exigées par ces algorithmes d'apprentissage afin d'obtenir de meilleures performances d'apprentissage. Il a été démontré, que des réseaux à une seule couche cachée (SLFN) (avec Nœuds cachées) avec des poids d'entrée et des biais choisis aléatoirement peuvent apprendre exactement N observations distinctes. Contrairement à la croyance établie et aux mises en œuvre pratiques, que tous les paramètres des réseaux doivent être ajustés, il n'est pas nécessaire d'ajuster les poids d'entrée et les biais de la couche caché. En fait, certains résultats de simulation sur des données artificielles et réelles ont montré que cette méthode non seulement rend l'apprentissage extrêmement rapide mais produit également une bonne performance de généralisation.
Après que les poids et les biais de la couche cachée soient fixés aléatoirement, les réseaux peuvent être tout simplement considérés comme un système linéaire. Les poids de sortie (reliant la couche cachée à la couche de sortie) peuvent être déterminés analytiquement par l'opération de l'inverse généralisée de la matrice de sortie de la couche cachée.
Sur la base de ce concept, il est possible de construire un algorithme d'apprentissage pour les réseaux à une seule couche cachée appelé apprentissage Extreme Learning Machine (ELM) dont la vitesse d'apprentissage peut être des milliers de fois plus rapide que les algorithmes d'apprentissage traditionnels comme l'algorithme de rétro-propagation (BP), l'obtention d'une meilleure performance de **généralisation** est également garantie.
Cet algorithme (ELM) non seulement tend à atteindre la plus petite erreur mais il atteint aussi la plus petite norme de poids.
Ce nouvel algorithme d'apprentissage peut être :

- Facilement mis en œuvre.
- Atteindre l'erreur d'entraînement la plus petite.
- Obtenir la plus petite norme de poids.
- Avoir une bonne performance de généralisation.
- Fonctionner extrêmement rapidement.

3.11.1 Réseaux à une seule couche cachée(SLFN) avec neurones cachés aléatoires

Pour N échantillons arbitraires distincte (x_i, t_i) où $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \mathbb{R}^n$ et $t_i = [t_{i1}, x_{i2}, ..., t_{im}]^T \in \mathbb{R}^m$, un SLFN standard avec $(\tilde{N})$ neurones

cachés et une fonction d'activation $g(x)$ est mathématiquement modélisée comme :

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(w_i x_j + b_i) = O_j, j = 1...N \tag{3.57}$$

Où $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ est le vecteur des poids reliant le i^{me} neurone caché et les entrées, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ est le vecteur des poids reliant le i^{me} neurone caché et les neurones de sortie, et b_i est le biais du i^{me} neurone caché.
$w_i.x_j$ désigne le produit intérieur de w_i et x_j .
Le neurone de sortie est linéaire.
Un SLFN standard avec $\tilde{N}$ neurones cachés avec la fonction d'activation $g(x)$ peut approximer les N observations avec une erreur zéro. Ce qui signifie que $\sum_{j=1}^{\tilde{N}} \parallel O_j - t_j \parallel = 0$, i.e. il existe β_i , w_i et b_i telle que

$$\sum_{i=1}^{\tilde{N}} \beta_i g(w_i x_j + b_i) = t_j, j = 1...N \tag{3.58}$$

Ces équations ci-dessus peuvent être écrites d'une manière compacte $H\beta = T$.
Où

$$H(w_1, ..., w_{\tilde{N}}, b_1, ..., b_{\tilde{N}} x_1, ..., x_N) = \begin{bmatrix} g(w_1 x_1 + b_1) & ... & g(w_{\tilde{N}} x_1 + b_{\tilde{N}}) \\ . & . & . \\ . & . & . \\ . & . & . \\ g(w_1 x_N + b_1) & ... & g(w_1 x_N + b_1) \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ . \\ . \\ . \\ \beta_{\tilde{N}}^T \end{bmatrix} \; et \; T = \begin{bmatrix} t_1^T \\ . \\ . \\ . \\ t_N^T \end{bmatrix}$$

Théorème 3.11.1 *Soit un SLFN standard avec N neurones cachés et la fonction d'activation $g : \mathbb{R} \to \mathbb{R}$ qui est infiniment différentiable dans tout intervalle, pour N observations arbitraires distinctes (x_i, t_i) où $x_i \in \mathbb{R}^n$ et $t_i \in \mathbb{R}^m$, pour tout w_i et b_i choisis aléatoirement dans tout intervalle de $\mathbb{R}^n$ et $\mathbb{R}$ respectivement, selon une distribution de probabilité continue,*
Alors, avec probabilité 1, la matrice de sortie de la couche caché H du SLFN est inversible et $\parallel H\beta - T \parallel = 0$

Théorème 3.11.2 *Étant donné toute petite valeur positive $\varepsilon > 0$ et une fonction d'activation $g : \mathbb{R} \to \mathbb{R}$ qui est infiniment différentiable dans tout intervalle, il existe $\tilde{N} \leqslant N$ tel que pour N observations arbitraires distinctes (x_i, t_i) où $x_i \in \mathbb{R}^n$ et $t_i \in \mathbb{R}^m$, pour tout w_i et b_i choisis aléatoirement dans tout intervalle de $\mathbb{R}^n$ et $\mathbb{R}$ respectivement, selon une distribution de probabilité continue quelconque,*
Alors, avec probabilité 1 : $\| H_{N\times\tilde{N}}\beta_{\tilde{N}\times m} - T_{N\times M} \| < \varepsilon$

3.11.2 L'algorithme de l'Extreme Learning Machine

Étant donné un ensemble d'observations

$$\Im = \{(x_i, t_i)x_i \mid x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m, i = 1, ..., N \tag{3.59}$$

La fonction d'activation $g(x)$ et le nombre de neurones cachés est $\tilde{N}$
Étape 01 : Assigner aléatoirement des valeurs aux poids d'entrée w_i et aux bais b_i, $i = 1, ..., \tilde{N}$.
Étape 02 : Calculer la matrice H de la sortie de couche cachée.
Étape 03 : Calculer les poids de sortie $\beta = H^{\ddagger}T$, telle que $T = [t_1, ..., t_N]^T$.

Remarque 3.2 *Le théorème* 3.11.1*, prouve en théorie, que cet algorithme fonctionne pour toute fonction d'activation infiniment différentielle $g(x)$. Ces fonctions d'activation comprennent les fonctions sigmoïdes, ainsi que les fonctions à base radiale, sinus, cosinus, fonctions exponentielles, et beaucoup d'autres. D'après le théorème* 3.11.2*, la limite supérieure du nombre requis de neurones cachés est le nombre d'observations distinctes, qui est $\tilde{N} \leq N$.*

Remarque 3.3 *Plusieurs méthodes peuvent être utilisées pour calculer l'inverse généralisée de Moore-Penrose de H. ces méthodes peuvent inclure, mais ne sont pas limités, à :*
- *la projection orthogonale,*
- *méthode d'orthogonalisation,*
- *méthode itérative,*
- *décomposition en valeurs singulières.*

3.12 Conclusion générale

Dans cette section, plusieurs méthodes d'apprentissage ont été présentées. Ce sont les méthodes les plus couramment utilisées dans le domaine

des réseaux de neurones. Chacune de ces méthodes a ses avantages et ses inconvénients. Si la méthode du gradient est la plus simple à implémenter elle reste toutefois très lente. La méthode de Gauss-Newton est plus rapide mais nécessite le calcul du Hessien. La méthode de Lenvenberg-Marquardt reste la plus optimale mais nécessite une capacité d'espace mémoire importante , elle présente un intérêt pratique car elle peut être utilisée sans avoir à choisir le taux.

Globalement, la méthode de Levenberg-Marquardt ne présente pas cet inconvénient, mais elle devient lourde pour des "gros" réseaux (une centaine de paramètres), en raison de l'inversion de matrice nécessaire à chaque itération. On a donc intérêt à choisir la méthode de Levenberg-Marquardt si le réseau est "petit", et celle de BFGS dans le cas contraire. Si l'on dispose du temps nécessaire, il est recommandé d'essayer les deux.

La méthode du gradient conjugué peut également constituer une solution efficace au problème d'optimisation de la fonction de coût.

Bien entendu, aucune méthode ne conduit à coup sûr au minimum global. Il convient donc de se placer dans des conditions où les minima locaux sont aussi peu nombreux que possible. Pour pallier ce problème bien connu, diverses solutions ont été suggérées , en pratique , il suffit de réaliser plusieurs apprentissages en choisissant des paramètres initiaux différents. En procédant de la sorte, on possède une plus grande chance de trouver le minimum global.

Comme conclusion, nous voyons un progrès perpétuel dans les algorithmes successifs. Le progrès est compris dans plusieurs sens :

- Minimiser la fonction de coût
- Atteindre ce minimum le plus rapidement possible
- Assurer la meilleure qualité de généralisation

Les mécanismes instaurés pour atteindre ces objectifs ont été de plus en plus sophistiqués.

Parmi les algorithmes itératifs, c'est celui de Levenberg-Marquardt qui se distingue par sa rapidité et ses bonnes performances. L'extreme learning machine a atteint les meilleures performances mais avec encore plus de rapidité. Notre souci essentiel dans cette thèse focalise surtout sur la qualité de la généralisation. La question demeure commune pour tous les algorithmes cités. Le problème étant un problème de fond, dans le concept fondateur même de ces diverses techniques. Très analogique aux préoccupations qui se trouvent dans les méthodes statistiques traditionnelles, le problème ne se restreint pas à minimiser la fonction de coût, mais à voir si ce processus de minimisation

conduit également à une bonne qualité de généralisation. Ce qui n'est pas toujours le cas. Il est apparu alors la nécessité d'aborder spécifiquement cette question. Des palliatifs ont été proposés avec le même esprit qui à été à l'origine de la création des estimateurs concurrents des moindres carrés dans le cadre de la statistique traditionnelle, que nous allons aborder dans le chapitre suivant.
Parmi ces méthodes palliatives figurent les algorithmes avec régularisation ou avec arrêt prématuré qui constituent une forme de contraintes et qui transforment la question en un problème d'optimisation avec contraintes. Tout à fait de la même manière qui a présidé à l'élaboration des estimateurs concurrents des moindres carrés.
En outre, des manipulations d'un genre tout à fait nouveau et très spécifiques aux méthodes neuronales sont apparues. En effet, car le traitement des problèmes hautement complexes est devenu désormais possibles grâce aux réseaux de neurones qui, pour avoir une meilleure prise, doivent eux-mêmes avoir une grande complexité. Il s'agit de trouver celle qui est optimale. Ces méthodes sont celles de l'élagage.
Nous avons choisi de les exposer également, car même d'une manière très subtile, ils peuvent avoir des liens avec l'idée unificatrice que nous tentons de développer.

CHAPITRE 4

MÉTHODES CONCURRENTES DE L'ESTIMATEUR DES MOINDRES CARRÉS ET MÉTHODES NEURONALES PALLIATIVES

4.1 Introduction

Dans ce chapitre, nous allons présenter deux groupes de méthodes qui ont été créés pour répondre au même besoin : celui d'assurer une meilleure qualité de généralisation que celle produite par les méthodes usuelles et consacrées. Pour le premier groupe, il s'agit des estimateurs biaisés qui ont été construits pour suppléer à l'estimateur des moindres carrés lorsque les conditions de son application ne sont pas optimales, principalement lorsque les variables explicatives présentent de fortes multicollinéarités entre elles.
Pour le second groupe, ce sont les mêmes méthodes neuronales qui ont été améliorées pour forcer les algorithmes à ne pas aller trop dans le sens du surajustement, qui est un problème bien spécifique aux réseaux de neurones. En effet, ce qui en certaines circonstances est un point fort de ces techniques (la qualité d'approximateur universel) devient source de défaillance par le fait de permettre de trop bien ajuster, conduit à dégrader la qualité de généralisation.
Ce chapitre réuni les éléments qui vont servir à une étude comparative de toutes ces méthodes, avec le but ultime de les présenter comme découlant d'un même principe unificateur.

4.2 Les méthodes concurrentes aux estimateurs des MCO

4.2.1 La régression Ridge

La régression Ridge est une technique originaire des statistiques permettant de manipuler la colinéarité en régression. Elle est probablement la plus grande rivale de PLS (partial least square) en terme de fléxibilité et de robustesse du modèle prédictif. Mais cette méthode n'a pas de procédure de réduction de dimension. Dans cette situation, la matrice $(X'X)$ possède au moins une valeur propre proche de 0 (toutes les valeurs propres de cette matrice sont non-négatives) indiquant que son déterminant est proche de 0. L'un des inconvénients majeurs en cas de multicolinéarité est l'imprécision de l'estimation ℓ_2 des paramètres. En effet, la matrice de variance-covariance est donnée dans (5.2), Ainsi, la variance totale des paramètres obtenus par la méthode des moindres carrés est donné dans (5.7), d'après cette expression, on constate que la variance totale devient exagérément grande lorsqu'une ou plusieurs valeurs propres sont très petites. Pour traiter ce problème, (Hoerl., 1962) fut le premier à proposer comme alternative l'estimateur Ridge. Repris et développé par (Hoerl et Kennard 1970a,b) l'estimateur ridge a la forme suivante

$$\hat{\beta}^* = (X'X + kI)^{-1}X'y \tag{4.1}$$

Où $k > 0$. En d'autres termes, une constante a été ajoutée à chaque élément de la diagonale de $(X'X)$. Souvent, de façon à pouvoir comparer plus directement les paramètres à estimer, on standardise les variables en leur soustrayant leur moyenne et en les divisant par leur écart-type. Dans ce paragraphe, cette standardisation est implicite. La matrice $X'X + kI$ est donc inversible même si les vecteurs de $X'X$ sont non linéairement indépendants. Alors

$$\phi(\hat{\beta}) = (Y - X\hat{\beta})'(Y - X\hat{\beta})$$

est minimale. Sur, et dans, la sphère centrée à l'origine et de rayon $\| \hat{\beta} \|$. En outre, $\phi(\hat{\beta})$ est une fonction croissante en k.
Ainsi, la solution Ridge requiert une augmentation de la somme des carrés des résidus par rapport à celle des moindres carrés. Cette solution est adaptée au cas où des valeurs propres sont très petites, mais non nulles.

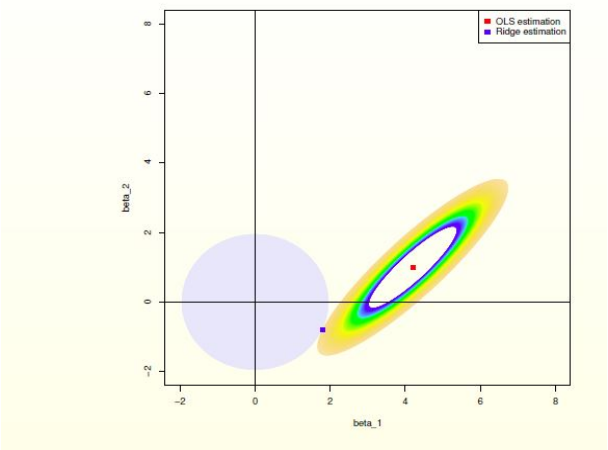


FIGURE 4.1 – Justification géométrique du Ridge

La variance totale de l'estimateur Ridge devient ainsi

$$Var(\beta^*) = \sigma^2 Tr(X'X + kI)^{-1} = \sigma^2 \sum_{i=1}^{p} \frac{1}{\lambda_i + k} \tag{4.2}$$

En comparant (5.7) et (4.2), on constate que la variance totale de l'estimateur Ridge est plus petite que celle de l'estimateur des moindres carrés. En fait ce gain de variabilité se fait au détriment du biais.
L'estimation des paramètres par la méthode des moindres carrés est connue pour produire des estimateurs non biaisés. Par contre, la procédure d'estimation ridge fournit des estimateurs avec biais, c'est-à-dire qu'en faisant la moyenne de ces estimateurs sur tous les échantillons possibles d'une population, cette moyenne ne sera pas égale à la valeur des paramètres de la population. Un résultat important est celui de (Hoerl et Kennard., 1970a), ils ont en effet montré qu'il existe toujours une constante $k > 0$ telle que l'erreur quadratique moyenne de l'estimateur Ridge est plus petite que celle de l'estimateur des moindres carrés. L'erreur quadratique moyenne (EQM) étant définie par :

$$EQM(\hat{\beta}) = Var(\hat{\beta}) + [Biais(\hat{\beta})]^2$$

Où

$$Biais(\hat{\beta}) = \mathbb{E}(\hat{\beta}) - \beta$$

Le principal problème de la procédure d'estimation Ridge est celui du choix de k. Une procédure graphique décrite dans (Hoerl et Kennard., 1970b) souvent proposée consiste à obtenir les estimateurs ridge pour différentes valeurs de k. Cette méthode est subjective et peut dépendre de l'échelle utilisée pour k. Cependant, depuis 1970, différentes méthodes pour déterminer k sont apparues dans la littérature. Dans ce travail, nous limiterons le choix de k à la proposition faite par Hoerl, En effet, nous utiliserons

$$k = \frac{p\hat{\sigma}^2}{\beta' \ell_2 \beta \ell_2}$$

Avec

$$\hat{\sigma}^2 = \frac{e' \ell_2 e \ell_2}{n - p}$$

Où p représente le nombre de variables explicatives. Ce choix de k, souvent évoqué dans la littérature, semble approprié dans le sens où l'estimateur Ridge correspondant a une erreur quadratique moyenne plus faible que l'estimateur ℓ_2 en présence du problème de la multicolinéarité.

4.2.2 L'estimateur de Marquardt

L'aspect du problème qui a motivé Hoerl et Kennard pour construire leur estimateur en ajoutant une petite quantité à la diagonale de $X'X$ (quand elle est présentée sous la forme d'une matrice de corrélation) est le même qui a motivé Marquardt dans la construction de son estimateur par inverse généralisé. En effet, on observe souvent que le spectre des valeurs propres de la matrice $X'X$ s'étant à partir de grandes valeurs positives jusqu'à des valeurs très proches de 0. Il est possible d'augmenter les valeurs de cette diagonale en lui ajoutant de petites quantités (comme pour le cas de l'estimateur Ridge). Mais l'idée de Marquardt est plutôt de supprimer les valeurs trop proches de zéro. La matrice $X'X$ ne devient plus carré mais rectangulaire, d'où la nécessité de recourir à l'inverse généralisée.
Nous savons que l'estimateur des moindres carrés s'écrit :

$$\hat{\beta} = (X'X)^{-1} X'Y$$

Et l'estimateur Ridge s'écrit :

$$\hat{\beta}^{\star} = (X'X + kI)^{-1} X'Y$$

L'estimateur par inverse généralisé s'écrit :

$$\hat{\beta}^{+} = A_r^{+} X'Y$$

Où A_r^{+} est obtenue par la manière suivante :
Soit D la matrice diagonale des valeurs propres, ordonnées de la manière comme suit :

$$\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p$$

Et soit la matrice des vecteurs propres S qui transforme la matrice $A = X'X$ en la matrice D. Alors :

$$S'AS = D$$

Où $S'S = I$, alors :

$$A^{-1} = SD^{-1}S'$$

En supposant que A est de rang r nous admettons que les derniers $(p-r)$ éléments de D sont nuls. Partitionnons S comme suit :

$$S = (S_r \vdots S_{p-r})$$

Où S_r est de dimensions $(p \times r)$ et S_{p-r} est de dimensions $(p \times (p-r))$. En partitionnant D d'une manière similaire :

$$\begin{bmatrix} D_r & \vdots & 0 \\ \cdots & \cdots & \cdots \\ 0 & \vdots & D_{p-r} \end{bmatrix}$$

Où D_r est de dimensions $(r \times r)$ et D_{p-r} est de dimensions $((p-r) \times (p-r))$ En supposons D_{p-r} une matrice nulle, nous avons également $D^{-1}_{(p-r)}$ comme matrice nulle. L'inverse A_r^{+} est donc

$$A_r^{+} = S_r D_r^{-1} S_r' \tag{4.3}$$

4.2.3 L'estimateur de James Stein

L'estimateur ridge et l'estimateur de Marquardt sont différents de celui de **James Stein**, $\beta_{JS} = c.\beta_{MCO}$ où $0 < c \leq 1$.

4.2.4 L'estimateur Lasso (Least Absolute Shrinkage and Selection Operator)

Cet opérateur a suscité une attention croissante depuis son introduction par (Robert Tibshirani., 1996), cet estimateur est défini comme l'estimateur des moindres carrés sous une contrainte de type ℓ_1

$$\hat{\beta}_{Lasso} = \begin{cases} min_{\beta \in R^p} \{ \| Y - X\beta \|_n^2 \\ s.c \| \beta \|_1 \leq t \end{cases} \tag{4.4}$$

Où t est un paramètre positif. Cet estimateur avait déjà été introduit en théorie du signal par (Chen et Donoho) sous le nom de Basis Pursuit De-Noising et défini sous sa forme pénalisée, il cherche β en résolvant le problème d'optimisation suivant :

$$\hat{\beta}_{Lasso} \in argmin_{\beta \in \Re^p} \{ \| Y - X\beta \|_n^2 + \lambda_1 \| \beta \|_1 \}$$

La fonction optimisée dans le Lasso est l'erreur empirique mesurée par $\| Y - X\beta \|^2$ auquel on ajoute un terme de régularisation $\lambda_1 \| \beta \|_1$ correspondant à la norme ℓ_1 de β pondérée par un paramètre λ_1. La norme ℓ_1 impose une certaine parcimonie à la solution du problème, autrement dit ce terme fait en sorte qu'on ait "beaucoup de zéros". La quantité de zéros dans la solution dépendra bien sûr de la valeur de λ_1.
Pour $\lambda_1 = 0$, on est dans le cas des moindres carrées, et il n'y a typiquement pas de zéro du tout (le vecteur β est plein). En faisant tendre λ_1 vers $+\infty$, le terme dominant devient la norme ℓ_1 de β, et la solution est le vecteur nul (on n'a que des zéros).
Une méthode pour sélectionner une "bonne" valeur λ_1 est d'effectuer une validation croisée pour chaque valeur de λ_1, et de prendre à la fin la valeur de λ_1 qui a une erreur de prédiction minimale.

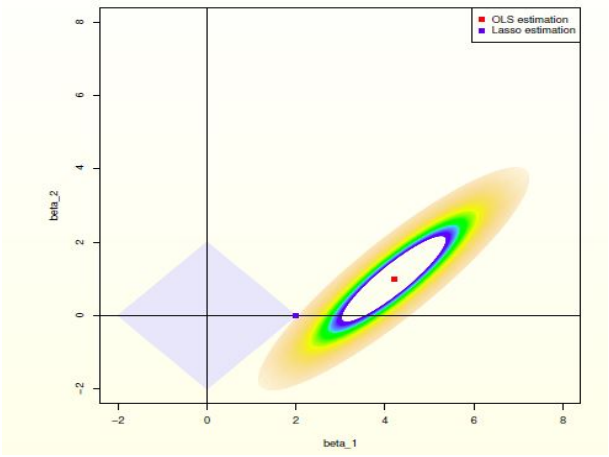


FIGURE 4.2 – Justification géométrique du Lasso

Limites de l'estimateur Lasso

- Les résultats théoriques relatifs à l'estimateur Lasso nécessitent une hypothèse sur la matrice de Gram. Cette hypothèse n'autorise que de faibles corrélations entre les variables. D'autre part, le Lasso n'intègre pas une connaissance a priori sur le modèle : il ne permet pas d'inclure la connaissance de structures particulières entre les variables, comme par exemple la prise en compte des corrélations connues entre certaines variables. Enfin, l'estimateur Lasso nécessite d'être adapté pour pouvoir prendre en compte des problèmes dans le cadre semi-supervisé.
- L'estimateur Lasso repose sur une hypothèse implicite sur la faible dépendance des variables explicatives. L'algorithme le plus populaire pour résoudre de critère de minimisation Lasso est le LARS, base également sur ces corrélations. Ainsi, dans des problèmes d'estimation avec de fortes corrélations entre les variables, l'algorithme LARS échoue à reconstituer le modèle.

4.2.5 Méthode Adaptative

Nous allons définir un nouvel estimateur $\hat{\beta}_N^{Lasso}$, Soit $\gamma > 0$ on définit le vecteur de poids $\hat{w} = \frac{1}{|\hat{\beta}^{ls}|^\gamma}$ On a alors :

$$\hat{\beta}_N^{Lasso} = argmin_\beta \sum_{i=1}^{n} \left(y_i - \sum_j x_{ij}\beta_j\right)^2 + k_N \sum_{j=1}^{p} \hat{w}_j \parallel \beta_j \parallel$$

Propriétés Oracle

La clé du theorème, et donc de la méthode adaptative, est la dépendance entre les poids et les données : le poids des coefficients estimés nuls est très grand ($\rightarrow \infty$) alors que le poids des coefficients estimés non nuls tend vers une constante. Les paramètres non significatifs sont donc ramenés à 0, la méthode est alors bien consistante.

4.2.6 Elastic Net

L'estimation elastic net est la solution au problème :

$$min_{\beta \in R^p}\{\parallel Y - X\beta \parallel^2 + \lambda_1 \parallel \beta \parallel_1 + \lambda_2 \parallel \beta \parallel_2^2\} \tag{4.5}$$

Ainsi l'elastic net pénalise aussi bien la norme ℓ_1 de β (comme dans Lasso), mais pénalise également la norme ℓ_2. L'elastic net combine en quelque sorte les vertus du Lasso et de la régression Ridge : D'une part il produit un vecteur β parcimonieux (et effectue donc naturellement une sélection de prédicteurs), et d'autre part, il tient compte des groupes éventuels de prédicteurs fortement corrélées entre eux.

4.3 Problèmes pour la généralisation

Les réseaux de neurones devant généraliser sur les exemples de l'ensemble d'apprentissage, il ne sont qu'une approximation des fonctions que l'on recherchait vraiment. Quelques problèmes de l'approximation de réseaux de neurones sont identifiés ici et des solutions sont proposées.

4.3.1 Choix de l'architecture

Le choix de l'architecture d'un réseau détermine la classe des fonctions calculables par celui-ci, ou encore sa complexité potentielle. C'est évidemment le premier paramètre sur lequel les utilisateurs de réseaux ont joué pour contrôler les performances d'un système. La démarche la plus évidente pour choisir la meilleure architecture est bien entendu de tester plusieurs modèles différents, changeants les types de neurones, le nombre de couches, le nombre de neurones cachés. Cependant, l'évaluation comparative des réseaux ainsi créé pose problème, de nombreuses méthodes existant mais étant beaucoup trop lourdes en calculs. Pour cette raison, la communauté de réseaux de neurones a adopté des procédures sous-optimales. La plus courante consiste en l'utilisation d'un ensemble de validation, le réseau offrant les erreurs les moindres sur cet ensemble étant considéré le meilleur. Cette méthode est également coûteuse en temps de calculs et soumise à de nombreux aléas.
Un problème qui apparaît lors d'un apprentissage est le problème du surapprentissage. Si le réseau de neurone apprend par coeur, il donnera de mauvais résultats quand on lui présentera des données un peu différentes.

4.3.2 Problème du surajustement

4.3.2.1 Définition du surajustement

Le surajustement se détecte sur la base d'une estimation des performances de généalisation du modèle. Le surajustement caractérise une fonction dont la complexité c'est-à-dire le nombre et la nature des degrés de libertés est telle qu'elle est capable de s'ajuster exactement aux exemples d'apprentissage, même si ceux-ci sont entachés de bruit. Ce phénomène est donc à l'origine un phénomène local : dans certains domaines des entrées, la fonction utilise localement certains de ses degrés de liberté de manière à passer précisément par certains exemples. Cette définition du surajustement suppose que tous les exemples ont la même importance et que l'on recherche effectivement une solution dont la réponse est en moyenne satisfaisante. Généralement, cette hypothèse est formalisée dans la fonction de coût choisie. Ainsi si l'on considère un ensemble d'apprentissage et une fonction de coût quadratique, en vertu de la propriété d'approximation universelle, il est toujours possible d'obtenir une fonction de coût aussi petite que l'on veut sur l'ensemble d'apprentissage, à condition de mettre suffisamment de neurones cachés. Cependant, le but de l'apprentissage n'est pas d'apprendre exactement la base d'apprentissage,

mais le modèle sous-jacent qui a servi à engendrer les données. Or, si la fonction apprise par le réseau de neurones est ajustée trop finement aux données, elle apprend les particularités de la base d'apprentissage au détriment du modèle sous-jacent : le réseau de neurones est surajusté .

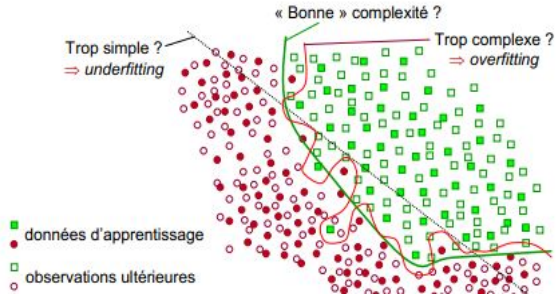


FIGURE 4.3 – Les bornes de la généralisation

4.3.2.2 Compromis Biais-Variance

Le paramètre à minimiser dans le cas de l'entraînement d'un RN est l'erreur sur les résultats donnés. Cependant, celle-ci ne tient compte que des valeurs de l'ensemble d'entraînement, alors qu'on devrait tenir compte de l'erreur sur toutes les données à traiter. Ceci étant impossible, on base plutôt l'optimisation sur l'erreur quadratique moyene. Comme il est mentioné dans (1.6), Cette erreur se décompose en deux termes : le biais et la variance. La difficulté de cette optimisation est de contrôler à la fois le biais et la variance. Alors que la variance est monotone croissante, a mis converge, le biais a un comportement non-monotone. Il commence par décroître, puis croît lentement.
Il faut donc trouver un compromis où la somme du biais et de la variance est minimale. Cela revient à accepter un certain biais pour maintenir la variance relativement faible.

4.4 Techniques pour améliorer la généralisation

4.4.1 L'arrêt prématuré

Il consiste à utiliser des données de test, Les données de test sont en fait un échantillon de réserve qui n'est jamais utilisé pour l'apprentissage. Il

est en revanche utilisé pour valider la manière dont un réseau se comporte pour modéliser la relation entre les entrées et les valeurs cible à mesure que l'apprentissage progresse. L'essentiel de la méthode permettant d'évaluer les performances en modélisation neuronale repose sur les différentes approches des données de test.
L'optimisation d'un RN s'effectue à l'aide d'un ensemble d'apprentissage. Un ensemble de test distinct est ensuite utilisé pour interrompre l'apprentissage afin de limiter le phénomène de surajustement.
L'erreur calculée sur l'ensemble d'apprentissage décroît d'une manière continue et se stabilise ensuite, mais l'erreur de test passe par un minimum avant de croître. Au début de l'apprentissage, cette erreur diminue au fur et à mesure que les paramètres sont modifiés pour s'approcher des sorties désirées données en apprentissage. Lorsqu'il y a surajustement, l'erreur de généralisation sur l'ensemble d'arrêt augmente. Il faut donc stopper l'apprentissage là où l'apprentissage produira les meilleures performances de généralisation. La sélection du meilleur moment pour stopper l'apprentissage pose quelques problèmes. L'utilisation d'un ensemble de validation reste une des méthodes les plus utilisées. Il suffit de stopper l'apprentissage avant que l'erreur sur l'ensemble de validation augmente.

Séparation des ensembles de test, d'apprentissage et de validation

La séparation de la base de données disponible en trois ensembles est une étape importante qui peut influer sur les performances du modèle choisi. Comme cela a été présenté, il est nécessaire que ces trois ensembles soient indépendants.
L'ensemble d'apprentissage doit être le plus complet possible afin que le modèle apprenne la diversité de tous les comportements du système.
L'ensemble de validation, étant donné le principe de l'arrêt prématuré, « spécialise » le réseau : le modèle est sélectionné pour donner les meilleures performances possibles sur cet ensemble. Il doit donc être choisi comme représentant de comportement le plus courant de la base de données dont on dispose (Toukourou et al., 2009). L'ensemble de test doit être choisi en fonction de l'objectif fixé.

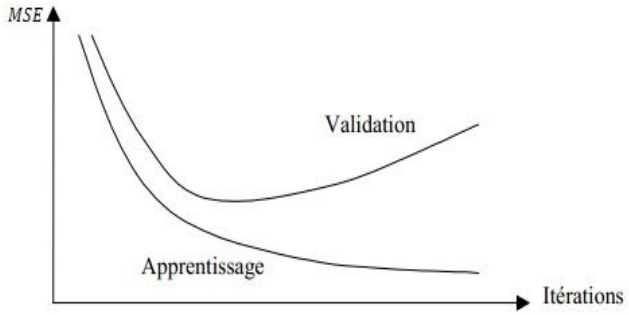


FIGURE 4.4 – Évolution typique des performances d'apprentissage et de validation

Cette technique modifie sensiblement l'algorithme d'apprentissage afin de :
1. Présenter au réseau un couple de valeurs entrées-cible issu de l'ensemble d'apprentissage.
2. Calculer les prévisions du réseau pour les valeurs cible.
3. Utiliser la fonction d'erreur pour calculer la différence entre les prévisions (sorties) du réseau et les valeurs cible.
4. Continuer avec les étapes 1 et 2 jusqu'à ce que tous les couples de valeurs entrées-cible de l'ensemble d'apprentissage aient été présentées au réseau.
5. Utiliser l'algorithme d'apprentissage pour ajuster les poids du réseau afin d'améliorer les prévisions pour toutes les valeurs entrée-cible.
6. Envoyer l'ensemble du jeu de test au réseau, effectuer les prévisions, et calculer la valeur de l'erreur de test du réseau.
7. Comparer l'erreur de test à celle de l'itération précédente. Si l'erreur continue de diminuer, l'apprentissage se poursuit, dans le cas contraire, l'apprentissage prend fin.

Remarque 4.1 *Le nombre de cycles nécessaires pour l'apprentissage d'un modèle de réseau de neurones avec des données de test et un arrêt prématuré peut varier. En théorie, nous devons poursuivre l'apprentissage de réseau pendant autant de cycles que nécessaire, tant que l'erreur de test diminue.*

L'arrêt prématuré s'appuie sur des méthodes statistiques connues qui sont basés sur la validation croisée.

La validation croisée

La validation croisée a été proposée par (Stone., 1974) afin d'estimer l'erreur de généralisation en utilisant l'ensemble de la base de données disponible. Cette méthode repose sur une estimation des performances à partir d'exemples n'ayant pas servi à la conception du modèle. Pour ce faire, on scinde la base d'apprentissage en D parties de taille (approximativement) égale. On effectue l'apprentissage du modèle sur $(D-1)$ sous-ensembles et on utilise le dernier sous-ensemble comme ensemble de validation, la performance du modèle s'obtient à partir des erreurs de validation constatées après les D apprentissages.

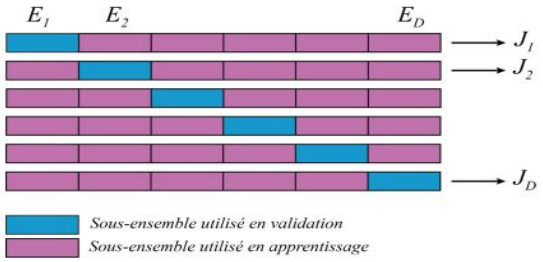


FIGURE 4.5 – Principe de la validation croisée,

En utilisant la fonction de coût des MCO, on procède comme suit :

– Pour chaque partie laissée de côté, on calcule l'erreur quadratique moyenne de validation (EQMV),

$$J_i = \frac{1}{N}\sum_{m=1}^{N}\left(d^m - y(x^m, w)\right)^2 \tag{4.6}$$

Pour le sous ensemble i qui comprend N exemples.

– À la fin, la performance de généralisation du modèle (appelée"score de validation croisée") est estimée en réalisant la moyenne quadratique des D erreurs (EQMV) précédentes.

$$S = \sqrt{\frac{1}{D}\sum_{i=1}^{D} J_i} \tag{4.7}$$

Donc à sélectionner, parmi des modèles candidats, le «meilleur» modèle compte tenu des données disponibles. On n'utilise donc pas l'erreur

d'apprentissage qui, n'est pas un bon estimateur de l'erreur de généralisation.

Dans le contexte de RN, la recherche de l'architecture optimale s'effectue souvent en partant d'un modèle linéaire et en augmentant progressivement le nombre de neurones cachés. Le modèle optimal est alors défini comme étant celui qui présente le meilleur score de validation croisée.

La limite naturelle de la validation croisée correspond au cas où D est égal au nombre d'exemples dans la base d'apprentissage. Cette méthode est connue sous le nom de "leave-one-out" voir (Plutowski., 1994) car chaque apprentissage n'est validé que sur un seul exemple.

Les difficultés de cette méthode sont de deux ordres :

- Le temps de calcul nécessaire, qui pour une même base d'apprentissage est d'autant plus grand que D est élevé (il est donc maximum dans le cas du leave-one-out)
- Des performances contrasté es en termes de taille de l'architecture sélectionnée et d'estimation des performances. À ce niveau, deux cas sont à distinguer :
  - Le nombre d'exemples est grand au regard de la complexité de la fonction à approcher (nombre d'entrées, non-linéarité) : dans ce cas, le phénomène de surajustement est difficile à mettre en évidence. La méthode donne certes de bons résultats (même avec un petit nombre de partitions) mais sans grand mérite car il y a peu de risque de surajustement.
  - Le nombre d'exemples est petit au regard de la complexité de la fonction à approcher : On est obligé d'augmenter le nombre de partitions de façon à garder un nombre suffisant d'exemples pour réaliser l'apprentissage des D modèles. Les résultats montrent alors une tendance à la surestimation de la taille des modèles nécessaires et à la sous-estimation des scores de validation croisée. Ceci traduit un phénomène mis en évidence par (Breiman., 1996) : une petite modification des données d'apprentissage peut entraîner de grandes différences dans les modèles sélectionnés. Autrement dit, si l'on raisonne en termes de fonction de coût, les exemples dont on se sert pour estimer les paramètres d'un modèle peuvent grandement influencer les minima vers lesquels convergent les différents apprentissages. On parle alors d'instabilité vis-à-vis des données d'apprentissage : les EQMV calculées à partir des différentes partitions ne peuvent donc

pas raisonnablement êtres moyennées pour estimer la performance de généralisation du modèle. La littérature conseille généralement d'utiliser $D = 10$. Cependant, ne sachant pas a priori s'il dispose de "peu" ou de "beaucoup" d'exemples, le concepteur essaiera souvent différentes valeurs de D. Si l'on se rappelle qu'à partir d'une base d'apprentissage, il est recommandé de procéder à plusieurs initialisations des poids de façon à diminuer le risque de minima locaux, on arrive très vite à un nombre d'apprentissages très élevé. En soi, ceci n'est pas grave si les résultats de ces différents essais sont cohérents. Dans le cas contraire, le découragement peut rapidement intervenir. Un cas particulier de la validation croisée est le leave-one-out où chaque sous-ensemble n'est composé que d'un seul exemple. Le score de leave-one-out est un estimateur non biaisé de l'erreur de généralisation. Il est coûteux en temps de calcul, si les données sont nombreuses, mais il est très utile lorsque les données sont peu nombreuses.

En conclusion

Nous pouvons dire que l'arrêt prématuré gére le nombre effectif de paramètres. Dans la pratique, l'arrêt prématuré est une méthode simple, mais efficace pour contrôler la flexibilité d'un réseau. d'autre part cette méthode peut être inapplicable, car il est difficile de déterminer avec précision le moment exact où il faut arrêter l'apprentissage puisque les performances sur la base de validation ne se dégradent pas nettement. On préfère donc utiliser les méthodes de régularisation, d'autant que (Sjoberg., 1995) a montré que l'arrêt prématuré était identique à un terme de pénalisation dans la fonction de coût.

4.4.2 La régularisation

Les méthodes de régularisation ne cherchent pas à limiter la complexité du réseau, mais elles contrôlent la valeur des poids pendant l'apprentissage. Il devient possible d'utiliser des modèles avec un nombre élevé de poids et donc un modèle complexe, même si le nombre d'exemples d'apprentissage est faible. (Bartlett., 1997) a montré que la valeur des poids était plus importante que leur nombre afin d'obtenir de modèles qui ne sont pas surajustés. Il montre, que si un grand réseau est utilisé et que l'algorithme d'apprentissage trouve une erreur quadratique moyenne faible avec des poids de valeurs

absolues faibles, alors les performances en généralisation dépendent de la taille des poids plutôt que de leur nombre. Cette technique de régularisation par termes de pénalité est probablement la manière la plus systématique de contrôler la flexibilité d'un réseau. Les méthodes de pénalisation ajoutent un terme supplémentaire à la fonction de coût usuelle afin de favoriser les fonctions régulières :

$$W = EQM + \lambda \sum_{i=1}^{m} w_i^2 \tag{4.8}$$

Où le deuxième terme de l'équation représente le terme de régularisation, w_j est un poids dans l'ensemble total des poids de m dans le réseau et λ est le paramètre de régularisation.
Cette méthode minimise la somme des carrés des poids avec la somme de l'erreur quadratique. Elle tire les poids qui n'ont pas une influence dans le réseau, tout en gardant les poids qui minimisent efficacement l'erreur. La quantité de la régularisation est contrôlée par le paramètre λ, et plus le λ est grand, la régularisation devient plus importante.

Remarque 4.2 *Lorsque les poids du réseau sont grands en valeur absolue, les sigmoïdes des neurones cachés sont saturées, si bien que les fonctions modélisées peuvent avoir des variations brusques. Pour obtenir des fonctions régulières, il faut travailler avec la partie linéaire des sigmoïdes, ce qui implique d'avoir des poids dont la valeur absolue est faible. La méthode de* ***régularisation du weight decay*** *limite la valeur absolue des poids.*

Régularisation par modération des poids (Weight Decay)

Cette méthode est appelée ridge regression dans le cas de modèles linéaires par rapport aux paramètres (Saporta., 1990). L'apprentissage s'effectue en minimisant

$$W = EQM + \frac{\lambda}{2} \sum_{i=1}^{m} w_i^2 \tag{4.9}$$

Si λ est trop grand, les poids tendent rapidement vers zéro, le modèle ne tient plus compte des données. Si λ est trop petit, le terme de régularisation perd son importance et le réseau de neurones peut donc être surajusté. Dans le cas intermédiaire, les poids après l'apprentissage ont des valeurs modérées.
Plusieurs variantes existent, notamment avec un λ différent selon la couche à laquelle appartiennent les paramètres. Le calcul de ces hyperparamètres peut être réalisé par des méthodes statistiques (McKay., 1992). McKay propose

une démarche fondée statiquement d'une manière solide, mais qui repose sur de nombreuses hypothèses et conduit à des calculs lourds. En pratique, il apparaît que les valeurs de ces hyperparamètres ne sont pas critiques : une démarche heuristique, qui consiste à effectuer plusieurs apprentissages avec des valeurs différentes des paramètres, à tester les modèles obtenus sur un ensemble des données de validation, et à choisir le meilleur, est généralement suffisante .

Conclusion : Étude comparative

L'arrêt prématuré est une méthode palliative, qui n'empêchent pas le surajustement, mais permettent de le détecter : on se sert d'une base de validation pour détecter les zones de forte courbure de la sortie du modèle, entraînées par l'ajustement trop précis de la sortie du modèle aux exemples d'apprentissage. Le weight decay est une heuristique : l'équivalence entre le fait que les poids du réseau soient "grands" et le surajustement n'a -à notre connaissance- jamais été démontrée ni dans un sens ni dans l'autre. Certes, elle se comprend intuitivement et (Bartlett., 1997) a montré que, pour avoir une bonne capacité de généralisation, la taille des poids est plus importante que la taille du réseau. Cependant, ceci pourrait très bien se révéler inexact dans tel ou tel cas particulier : cela dépend de la forme réelle de la fonction à approcher.

4.5 La complexité structurelle des réseaux de neurones

Les méthodes vu précédemment emploient tous les poids dans l'entrainement et ne réduisent pas la structure et la complexité du modèle. Les résultats obtenus en apprentissage peuvent, au vu de l'approximation universelle, être aussi proches des sorties désirées qu'on le souhaite, si la complexité du modèle est suffisante. Ainsi, la complexité du modèle doit être ajustée pour trouver un compromis entre le biais et la variance. Dans leur article (Geman et al., 1992) contrôlent la complexité du modèle et donc le surajustement en limitant le nombre de neurones cachés. Cependant (Gallinari et Cibas., 1999) ont montré que cette vision théorique avait des limites pour un réseau à couches dont l'apprentissage était effectué avec une base d'apprentissage comprenant peu d'exemples. En étudiant différentes architectures pour un

problème de régression, ils ont montré que le biais et la variance n'évoluent pas nécessairement en sens contraire lorsque le nombre de neurones cachés augmente. Dans leur cas, un modèle avec quinze neurones cachés à une variance plus élevée qu'un modèle avec soixante neurones cachés. En résumé, le surajustement ne s'explique pas seulement par le compromis biais-variance, notamment lorsque le nombre d'exemples est faible. De plus, l'interprétation du surajustement en ces termes a été développée pour les problèmes de régression et ne se transpose pas simplement aux problèmes de classification.

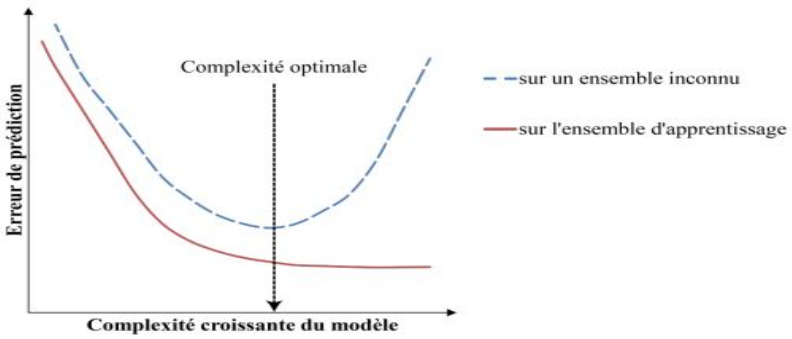


FIGURE 4.6 – Complexité du modèle et capacité de généralisation

Si un modèle est trop complexe, on observe le surajustement : le modèle reproduit très bien les exemples d'apprentissage mais il est incapable de généraliser. Or, la capacité de généralisation étant primordiale, il va falloir résoudre au mieux le dilemme biais-variance en trouvant une complexité conduisant à minimiser la somme du biais et de la variance.

4.5.1 Critère d'évaluation (mesure de pertinence)

Les mesures de pertinence associées aux méthodes de sélection de variables neuronales sont souvent basées sur des heuristiques calculant l'importance individuelle de chaque variable dans le modèle obtenu après apprentissage. Ces heuristiques sont nombreuses, mais peuvent être classées selon leurs similarités en quatre grandes familles :

- Les mesures d'ordre zéro (i.e. utilisant les valeurs des paramètres du réseau)
- Les mesures du premier ordre (i.e. utilisant les dérivées du premier ordre des paramètres du réseau)

- Les mesures du second ordre (i.e. utilisant les dérivées du second ordre des paramètres du réseau)
- Les termes de régularisation permettant de pénaliser les variables inutiles pendant l'apprentissage.

4.5.2 Critère d'arrêt

Une fois que la méthode d'évaluation et celle de recherche ont été fixées, certaines méthodes de sélection de variables examinent tous les sous-ensembles fournis par la méthode de recherche. Une bonne heuristique, dont la complexité est suffisamment raisonnable dans la plupart des applications, est d'estimer l'erreur de généralisation pour les différents sous-ensembles de variables sélectionnés. L'ensemble de variables idéal est celui qui donne les meilleures performances.
L'erreur de généralisation peut être estimée grâce à un ensemble de validation, par validation croisée ou par d'autres estimations algébriques. Plusieurs mesures ont été proposées en statistiques (Gustafson et Hajlmarsson., 1995) ou pour les réseaux de neurones (Moody.,1991).
La plupart des méthodes de sélection de variables utilisent des techniques assez rudimentaires pour arrêter la sélection : certaines méthodes fixent un seuil par rapport au critère de pertinence, d'autres classent juste les variables en fonction de l'estimation de l'erreur de généralisation.
D'un autre côté et lorsque les différents poids optimales sont dans les RN, le réseau produit différents ensembles de poids. Ceci doit être résolu pour faire un modèle idéal et pour faire des conclusions réalistes. Pour atteindre ce but on doit réduire la complexité structurelle des réseaux de sorte que seulement les poids et les neurones essentiels demeurent dans le modèle. Plusieurs méthodes de sélection de variables sont inspirées des techniques d'élagage des poids dans le réseau. La décision de supprimer un poids est faite selon un critère de pertinence. Une connexion est coupée si sa pertinence est faible. Après avoir présenté une technique d'élagage précise (Optimal Brain Damage), nous passerons en revue les différentes méthodes de sélection de variable qui en ont découlé. Nous proposerons aussi des variantes à ces méthodes à partir de considérations générales issues de (Leray, P. et Gallinari,P. 1997) et (Leray,P. et Gallinari,P. 1999).

4.6 Les méthodes d'élagages

4.6.1 L'élagage des poids par Optimal Brain Damage : OBD

(LeCun,Y et al., 1990) a proposé une technique d'élagage appelée OBD, Dans cet approche, les poids qui n'ont pas une importance pour la cartographie d'entrée-sortie d'un réseau sont sélectionné et enlevé. La pertinence d'un poids peut être calculé à partir de la matrice Hessienne introduite par les méthodes d'apprentissages de Newton et Levenberg-Marquardt.

$$H_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j} \tag{4.10}$$

Comme on a vu précédemment et pour les grands réseaux, le calcul peut devenir coûteux. Pour résoudre ce problème, une approximation locale qui utilise seulement les termes diagonaux est appliquée ici, où la pertinence d'une connexion est définit par :

$$Pertinence\ (w_j) = \frac{H_{jj}}{2} w_j^2 = \frac{\partial^2 EQM}{2\partial w_j^2} w_j^2 \tag{4.11}$$

Où le Hessien de la fonction de coût est utilisé pour calculer la dépendance du modèle par rapport aux poids. Il désigne les éléments diagonaux de la matrice Hessienne, qui contient le carré de la dérivée de l'erreur du réseau par rapport à chacun des poids individuels ,w_j.
Pour utiliser cette mesure de pertinence d'une connexion comme critère de sélection d'une variable, il faut calculer la pertinence d'un neurone de la couche d'entrée en utilisant l'approximation suivante :

$$Pertinence\ (x_i) = \sum_{j \in fan-out(i)} Pertinence\ (w_j) \tag{4.12}$$

Où $fan - out(i)$ représente l'ensemble des poids partant de la variable i .

4.6.2 La sélection de variables par Optimal Cell Damage : OCD

OCD a été proposé dans (Cibas, T et al., 1994,1996) une méthode équivalente étant proposée au même moment dans (Mao, J et al., 1994). Cette

méthode généralise la technique d'élagage OBD à la sélection de variables. En utilisant (4.11) et (4.12) nous obtenons :

$$S_i = Pertinence\ (x_i) = \sum_{j \in fan-out(i)} \frac{\partial^2 EQM}{2\partial w_j^2} w_j^2 \tag{4.13}$$

OBD et OCD considèrent que H, le Hessien de la fonction de coût (EQM) est une matrice diagonale (i.e. les termes croisés d'un développement de Taylor du second ordre sont négligés). Cette hypothèse revient à supposer que la fonction de coût est minimale et localement quadratique autour du minimum local.

Algorithme OCD
0.Atteindre un minimum local (fixé par un seuil θ)
1. Calculer la pertinence de chaque entrée grâce à (4.13)
2. Trier les entrées par ordre croissant de pertinence
Soit S_i la liste des pertinences classées par ordre croissant, on peut définir La pertinence cumulée par :

$$S'_i = \sum_{j=1}^{i} S_j$$

3.Supprimer les entrées dont la pertinence cumulée est inférieure à un seuil fixé q
4.Recommencer en 0 tant que les performances estimées sur une base de test ne chutent pas.

Remarque 4.3 *Les seuils θ et q sont déterminés par validation croisée.*

4.6.3 Une autre variante : N-OCD

Suite aux travaux de (Cibas, T et al., 1996), une variante d'OCD, N-OCD a été proposé, qui distingue bien les critères mis en œuvre dans la sélection de variable en essayant d'améliorer l'évaluation de la mesure de pertinence et celle du critère d'arrêt.

Évaluation de la mesure de pertinence

OCD donne de bons résultats mais possède aussi un inconvénient : si le seuil q est trop élevé, l'algorithme peut supprimer des variables significatives.

De même, si le nombre de variables de pertinence faible est élevé, cela ne signifie pas que toutes ces variables sont inutiles et à éliminer. Les variables ont été supprimés une par une en réapprenant le RN et en ré-estimant les mesures de pertinence à chaque fois.

Critère d'arrêt

Un autre problème se pose alors : quel critère d'arrêt utiliser ?. L'estimation de l'erreur de généralisation avec un ensemble de test fournit un critère d'arrêt non monotone, qui oscille au fur et à mesure de la sélection. Il est assez brutal de s'arrêter dès que les performances en test diminuent. N-OCD va donc supprimer toutes les variables jusqu'à la dernière et déterminer ensuite quel sous-ensemble de variables sélectionner grâce à un test statistique.

Algorithme N-OCD

0. Pour $p = k$ jusqu'à 1 (nombre de variables)
1. Atteindre un minimum local (déterminé grâce à un ensemble de validation)
2. Estimer l'erreur $EQM(p)$ (sur un ensemble de test)
3. Calculer la pertinence de chaque entrée grâce à (4.13)
4. Supprimer la variable la moins pertinente
5. Retourner en 0.
6. $M^* = min(EQM(p))$
7. $p_i = p/EQM(p) \approx M^*$ au sens de Fisher
8. $p_0^* = min(p_i)$

La solution classique est de sélectionner le réseau $F(p_0)$ qui obtient la plus petite erreur. Malheureusement, le nombre de données servant à estimer l'erreur est limité, il existe donc une incertitude sur l'estimation du critère de choix.
Pour tenir compte de ce problème, il faut chercher parmi tous les modèles $F(p)$ ceux qui sont statistiquement proches de $F(p_0)$ à l'aide d'un test de Fisher.
Nous obtenons ainsi un ensemble de modèles tels que $EQM(p_i) \approx EQM(p_0)$. En posant comme hypothèse que nous cherchons le plus petit ensemble de variables possible, il suffit de prendre $p_0^* = min(p_i)$, i.e. le plus petit modèle statistiquement proche du modèle obtenant une erreur en test minimale.

4.6.4 L'élagage des poids par Optimal Brain Surgeon : OBS

Cette méthode, qui n'est pas envisageable pour un grand nombre de poids, a cependant un avantage : elle permet de mettre à jour immédiatement les poids du réseau lorsqu'une connexion est supprimée. De plus, (Hassibi et al., 1994) insistent sur le fait qu'un apprentissage utilisant OBD peut conduire à une baisse de performances de généralisation, ce qui n'est pas le cas d'OBS. De la même manière qu'OCD utilisait le calcul de pertinence des poids donné par OBD, Unit-OBS proposé dans (Stahlberger, A et al., 1997) se sert du calcul de pertinence des poids donné par OBS pour supprimer des variables. L'avantage de cette méthode est qu'il n'est plus nécessaire de recalculer le Hessien à chaque élimination d'un poids, mais à chaque suppression d'une variable.

Remarque 4.4 *Quelque soit la méthode utilisée OBD ou OBS, la pertinence d'une connexion (ou d'une variable) est calculée à partir des mêmes données que celles utilisées pour l'apprentissage.*

4.6.5 Une autre variante : N-ECD

L'hypothèse de base de OBD et de OBS est que le réseau a atteint un minimum local. En pratique, l'apprentissage du réseau de neurones est arrêté par l'arrêt prématuré, avant que le minimum local ne soit atteint. (Tresp, V et al.,1997). propose donc deux nouvelles variantes de OBD et OBS : EBD (Early Brain Damage) et EBS (Early Brain Surgeon). À partir de considérations heuristiques, il ajoute deux nouveaux termes dans le calcul de la pertinence des poids pour prendre en compte le fait que la dérivée de la fonction de coût n'est pas nulle à la fin de l'apprentissage. Soit N-ECD (ECD pour Early Cell Damage) la méthode ainsi obtenue :

$$S_i = \sum_{j \in fan-out(i)} \frac{\partial^2 EQM}{2\partial w_j^2} w_j^2 - \frac{\partial EQM}{\partial w_j} w_j + \frac{1}{2} . \frac{\left(\frac{\partial EQM}{\partial w_j}\right)^2}{\frac{\partial^2 EQM}{\partial w_j^2}} \tag{4.14}$$

Algorithme N-ECD

0. Pour $p = k$ jusqu'à 1 (nombre de variables)
1. Atteindre un minimum local (déterminé grâce à un ensemble de validation)

2. Estimer l'erreur $EQM(p)$ (sur un ensemble de test)
3. Calculer la pertinence de chaque entrée grâce à (4.14)
4. Supprimer la variable la moins pertinente
5. Retourner en 0.
6. $M^* = min(EQM(p))$
7. $p_i = p/EQM(p) \approx M^*$ au sens de Fisher
8. $p_0^* = min(p_i)$

4.6.6 Un autre aspect : Variance nullity measure : VNM

L'algorithme proposé par Engelbrecht repose sur la « variance nullity measure » (VNM). L'idée de base est de tester si la variance de la sensibilité d'une entrée ou de la sortie d'un neurone caché pour différentes données est significativement différente de 0. Si cette variance n'est pas significativement différente de zéro et si la sensibilité moyenne est petite, alors cela indique que l'entrée ou le neurone caché correspondant a un faible impact ou pas d'impact du tout sur la sortie du réseau considéré. Un test d'hypothèse peut donc utiliser cette VNM pour déterminer statistiquement si un neurone caché ou une entrée doit être éliminé en utilisant une distribution du χ^2. Pour tester s'il faut supprimer un neurone caché, nous devons déterminer la VNM du poids $w_i^2, (i = 1, ..., n)$ connectant ce neurone caché au neurone de sortie. Pour cela, il est nécessaire de connaître la sensibilité de la sortie y par rapport au paramètre w_i^2 et cette sensibilité correspond à la contribution de ce paramètre sur l'erreur totale faite en sortie. Cette contribution est classiquement déterminée par la dérivée partielle de la sortie du réseau y par rapport au paramètre w_i^2 considéré.
Nous pouvons noter la sensibilité de la sortie par rapport à un neurone caché ou à une entrée par une notation commune S_θ, le paramétre θ peut être (entrée poids, ou bien un neurone caché).
La VNM correspond à la variance $\sigma_{S_\theta}^2$ du paramètre θ considéré :

$$\sigma_{S_\theta}^2 = \frac{\sum_{i=1}^{N}(S_{\theta i} - \mu_{S_\theta})^2}{N} \tag{4.15}$$

Où μ_{S_θ} est la moyenne de la sensibilité pour N exemples , l'equation (4.15) peut être simplifié par :

$$\sigma_{S_\theta}^2 = \frac{\sum_{i=1}^{N}(S_{\theta i^2} - 2S_{\theta i}\mu_{S_\theta} + \mu_{S_\theta}^2)}{N} = \mu_{S_\theta^2} - \mu_{S\theta}^2. \tag{4.16}$$

Ceci rapporte une expression pour la valeur prévue de la sensibilité par rapport aux paramétres θ et $\mu_{S\theta}$ sous cette forme :

$$\mu_{S_\theta^2} = \mu_{S\theta}^2 + \sigma_{S_\theta}^2 \tag{4.17}$$

Il reste à tester l'hypothèse que la variance de la sensibilité pour le neurone caché ou pour l'entrée considéré est approximativement nulle. Pour cela, le test d'hypothèse suivant est construit :

$$\begin{cases} H_0 : \mu_{S\theta}^2 = 0 \\ H_0 : \sigma_{S_\theta}^2 = 0. \end{cases} \tag{4.18}$$

Si la première hypothèse est rejeté, on ne peut pas supprimer les paramètres. puis la deuxième hypothése doit être tester. Un paramètre γ est définit en termes de nullité de la variance. C'est la nullité statistique de la variance.

$$\gamma_{S\theta} = \frac{(N-1)\sigma_{S\theta}^2}{\sigma_0^2} \tag{4.19}$$

Où σ_0^2 est proche de 0 et $\sigma_{S\theta}^2$ peut être éstimé de :

$$\sigma_{S\theta}^2 = \frac{\sum_{i=1}^{N}(S_{\theta i} - \bar{S}_\theta)^2}{N-1} \tag{4.20}$$

$\bar{S}_\theta$ est la moyenne de la sensibilité de paramètre :

$$\bar{S}_\theta = \frac{\sum_{i=1}^{N} S_{\theta i}}{N} \tag{4.21}$$

L'hypothèse que la variance est proche de zéro est évaluée pour chaque paramètre θ avec H_0 :

$$H_0 : \sigma_{S\theta}^2 = \sigma_0^2 \tag{4.22}$$

Contre :

$$H_1 : \sigma_{S\theta}^2 < \sigma_0^2 \tag{4.23}$$

Sous la distrubtion de χ^2, on a :

$$\gamma_c = \chi_{N-1,(1-\alpha/2)}^2$$

Où α est le seuil de signification. Si $\gamma_{S\theta} \leq \gamma_c$, H_1 est accepté et le paramètre est enlevé du réseau.

Conclusion : Étude comparative

Les études comparatives portent sur différents problèmes de classification, même si la plupart des méthodes présentées ici (les méthodes N-OCD et N-ECD) s'appliquent aussi dans le cadre de la régression. Ces comparaisons permettent de tirer quelques conclusions sur les méthodes de sélection de variables. Tout d'abord, il n'existe pas de méthode de sélection qui soit meilleure que les autres. Les méthodes dérivées des techniques d'élagage comme OBD ne sont pas meilleures que les autres. Par contre, les résultats vont dépendre de la politique choisie par rapport aux différents critères utilisés :

- Les critères de pertinence basés sur des hypothèses de linéarité ou de distribution unimodale sont mal adaptés aux autres problèmes.
- L'évaluation du critère de pertinence est liée aux paramètres du modèle. La suppression d'une variable dans le réseau de neurones change automatiquement la valeur de ses paramètres optimaux. Ne pas ré-estimer les paramètres du modèle signifie que l'on considère toutes les variables indépendantes. Un ré-apprentissage est nécessaire si l'on désire prendre en compte la corrélation entre les variables
- Le rôle du critère d'arrêt est déterminant : un critère basé uniquement sur les variations de performances peut s'avérer trop brutal et stopper trop tôt la sélection (ou l'élimination) des variables
- Pour les problèmes de taille "raisonnable", il semble intéressant de faire à la fois la sélection de variables et l'apprentissage du réseau de neurones, l'observation de ces différents phénomènes nous a mené à proposer deux règles permettant d'améliorer à un moindre coût les méthodes dérivées d'OBD comme la plupart des méthodes de sélection de variables neuronales existantes :
  - Il faut réapprendre le réseau à chaque étape, avant de ré-estimer la pertinence des variables
  - Le choix du meilleur ensemble de variables peut se faire grâce à l'estimation des performances sur un ensemble de test et à l'utilisation d'un test statistique pour déterminer l'ensemble de variables minimal.

CHAPITRE 5

ANALOGIES ENTRE ESTIMATEURS BIAISÉS ET TECHNIQUES NEURONALES

5.1 Préambule

Nous avons souligné précédemment que l'une parmi les questions les plus importantes qui continuent de se poser avec acuité dans le domaine de la régression et de la modélisation, linéaire ou non linéaire, est celle concernant la qualité de la généralisation. Cette question reste d'une grande actualité, tout aussi bien dans le cadre des techniques de la statistique traditionnelle, que dans le cadre des techniques récentes telles que les réseaux de neurones. Si la question de la qualité de la généralisation est commune à tous les modèles, linéaire ou non linéaire, d'autres questions auxiliaires se posent et qui sont plus visibles chacune dans son contexte approprié. Nous distinguons le problème de l'instabilité des estimateurs, qui se pose surtout dans le cadre des méthodes statistiques traditionnelles, de celui des minimums locaux, qui se pose dans le cadre des méthodes neuronales.
Dans le modèle linéaire, lorsqu'il existe une forte multicollinéarité entre les variables explicatives, l'estimateur des moindres carrés ordinaires MCO, bien qu'il soit le meilleur parmi ceux sans biais, souffre d'une difficulté énorme qui est celle de l'instabilité. C'est cet écueil qui a conduit à l'élaboration d'autres estimateurs, nécessairement biaisés, mais dont l'erreur quadratique moyenne (en terme de distances des paramètres estimés par rapport aux vrais para-

mètres) est inférieure à celle de l'estimateur des MCO. L'intérêt est d'aboutir à une bonne qualité de généralisation même s'il faut pour cela concéder un petit biais.
Ce même type de problèmes surgit dans le cadre de la modélisation par le moyen des réseaux de neurones. Mais ici, les problèmes de généralisation se posent surtout en termes de surajustement. En cherchant à "trop" minimiser la fonction de coût nous risquons, en plus de dégrader la qualité de la généralisation, de nous piéger dans des minimas locaux.
Exposés de cette manière, les deux problématiques semblent être différentes. Cependant les solutions préconisées pour les deux cas présentent certaines similitudes dignes d'intérêt et qui demandent d'être analysées en profondeur.
D'un côté, les estimateurs conçus dans le cadre des techniques de la statistique traditionnelle, à savoir :

- L'estimateur Ridge de Hoerl et Kennard
- L'estimateur par inverse généralisée de Marquardt
- L'estimateur de James-Stein
- La méthode Lasso

Sont tous des estimateurs raccourcis, dans le sens où la norme du vecteur des paramètres est plus courte que celle des MCO.
De leur côté également, les techniques mises à contribution dans le cadre des réseaux de neurones, à savoir :

- La régularisation
- L'arrêt prématuré

Conduisent aussi à un raccourcissement du vecteur des paramètres.
Par ailleurs, sur un autre point, l'estimateur de Marquardt basé sur l'idée d'une troncature touchant la matrice carré des corrélations, la transformant en une matrice rectangulaire et aboutissant ainsi à l'usage d'inverse généralisée, supprime tacitement quelques "liaisons" entre les variables explicatives. Cette suppression se retrouve, mais explicite, dans les techniques de l'OBD, OCD, OBS, N-OCD, N-ECD, VNM dans le cadre des réseaux de neurones.
La question qui se pose est de voir s'il existe des liens formels, même restreints au modèle linéaire, entre ces techniques. Notre recherche actuelle se développe autour des moyens et des pistes qui permettent d'établir expressément ces similitudes. Différentes hypothèses et différentes directions apparaissent.
L'objet de ce présent chapitre est de discuter, dans le cadre du modèle linéaire, les résultats qui vont dans ce sens, et d'envisager ensuite leurs extensions aux modèles non linéaires.

5.2 Introduction

Nous avons rappelé dans le deuxième chapitre que l'estimateur des moindres carrés est le meilleur estimateur linéaire sans biais (théorème de Gauss Markov) dans le sens où il a la plus petite variance. Plus encore, on peut facilement montrer que l'estimateur des moindres carrés donne l'erreur de généralisation minimale parmi les estimateurs linéaires sans biais. Donc, dans le cadre des estimateurs sans biais, il n'y a pas un estimateur meilleur que celui des moindres carrés pour l'erreur de généralisation. Néanmoins, ceci n'est pas vrai pour les estimateurs biaisés. Il y a eu de multiples tentatives pour chercher un estimateur meilleur que celui des MCO parmi les estimateurs biaisés. Le critère n'est plus la minimisation de la variance mais devient la minimisation de l'erreur quadratique moyenne.
Considérons le modèle standard pour la régression linéaire multiple

$$Y = X\beta + \varepsilon$$

Où il est supposé que X est une matrice $(n \times p)$ et de rang p, β est $(p \times 1)$ et est inconnu, avec $\mathbb{E}[\varepsilon] = 0$ et $\mathbb{E}[\varepsilon\varepsilon'] = \sigma^2 I_n$.
L'estimateur usuel pour le paramètre β est l'estimateur linéaire de Gauss-Markov qui est sans biais et qui a la variance minimale. Cette procédure est bonne lorsque $X'X$ posée sous la forme d'une matrice de corrélation, est proche de la matrice unité. Mais lorsque $X'X$ est loin de la matrice unité, l'estimation par la méthode des moindres carrés est sujette à un certain nombre de faiblesses.

Propriétés de la meilleure estimation linéaire non biaisée

En utilisant l'estimation linéaire non biaisée ayant la variance minimale ou en utilisant l'estimation par la méthode du maximum de vraisemblance quand le vecteur aléatoire ε est normal, nous donne :

$$\hat{\beta} = (X'X)^{-1}X'Y$$

Comme estimation de β, qui donne la plus petite somme des carrés des résidus :

$$\phi(\hat{\beta}) = (Y - X\hat{\beta})'(Y - X\hat{\beta}) \tag{5.1}$$

Pour démontrer les effets de $X'X$, lorsqu'elle n'est pas proche de la matrice unité, considérons les deux propriétés suivantes :
Sa matrice de variance-covariance et sa distance par rapport à son espérance :

$$Var(\hat{\beta}) = \sigma^2 (X'X)^{-1} \tag{5.2}$$

$$L_1^2 = (\hat{\beta} - \beta)'(\hat{\beta} - \beta) \tag{5.3}$$

Nous avons

$$\mathbb{E}[L_1^2] = \sigma^2 Tr(X'X)^{-1} \tag{5.4}$$

Quand l'erreur est normalement distribuée, alors

$$Var[L_1^2] = \sigma^2 Tr(X'X)^{-2} \tag{5.5}$$

Ces propriétés montrent l'incertitude qui touche $\hat{\beta}$ quand $X'X$ s'éloigne de la matrice unité, si on note Les valeurs propres de $X'X$ par

$$\lambda_{max} = \lambda_1 \geq \lambda_2 ... \geq \lambda_p = \lambda_{min} > 0 \tag{5.6}$$

Alors nous avons (Comme espérance du carré de la distance entre $\hat{\beta}$ et β) :

$$\mathbb{E}[L_1^2] = \sigma^2 \sum_{i=1}^{p} \left(1/\lambda_i\right) \tag{5.7}$$

Et, quand l'erreur est normale, sa variance est

$$Var[L_1^2] = 2\sigma^4 \sum_{i=1}^{p} \left(1/\lambda_i\right)^2 \tag{5.8}$$

Les bornes inférieures, respectivement pour l'espérance et la variance sont : σ^2/λ_{min} et σ^4/λ_{min}^2.
Ainsi, si $X'X$ possède une ou plusieurs valeurs propres qui sont petites, la distance entre $\hat{\beta}$ et β aura tendance à être grande. Ce qui est tout à fait remarqué, dans le cas de données non orthogonales, est que les coefficients estimés, $\hat{\beta}_i$ sont grands en valeurs absolue.

5.3 Considérations heuristiques :

Les propositions avancées pour pallier aux défaillances des méthodes usuelles (classiques et neuronales) s'appuient, sans que cela soit tout à fait explicite, sur la combinaison de deux considérations majeures :

– La nature de la fonction de coût.
– Le raccourcissement du vecteur estimé des paramètres.

Nous allons dans ce qui suit proposer une **approche unifiée** qui met en lumière le fait que toutes ces diverses propositions peuvent être perçues comme découlant d'un même et unique principe.
D'abord au sujet de la fonction de coût : le critère le plus communément utilisé tout aussi bien dans les méthodes classiques que dans les méthodes neuronales est la minimisation de la somme des carrés des résidus. Ceci étant un point commun important pour la discussion que nous allons développer.
Maintenant au sujet des méthodes palliatives aux contre performances des méthodes originelles (moindres carrés pour les méthodes standards et méthodes du gradient pour les RN), la quasi-totalité des solutions a abouti à un raccourcissement du vecteur des paramètres.
Le raccourcissement est nécessairement conçu de manière comparative : un vecteur (des paramètres) n'est forcément raccourci que par rapport à un autre.
Les vecteurs de références sont ceux produits par les méthodes usuelles, appelées à être améliorées. Pour revenir à la source de l'idée originelle de raccourcissement, le vecteur de référence est celui produit par la méthode des moindres carrés dans le cadre du modèle linéaire.
La question est de savoir s'il faut nécessairement le raccourcir.
La réponse est oui. Les arguments pour cela seront développés dans ce qui suit.

Proposition 5.1 *Le vecteur des paramètres obtenu par la méthode des moindres carrés possède, en termes d'espérance, une norme plus grande que celle du vecteur des vrais paramètres.*

Démonstration :

En effet, en utilisant (5.3) et (5.4), nous aboutissons immédiatement à :

$$\mathbb{E}(\hat{\beta}'\hat{\beta}) = \beta'\beta + \sigma^2 Tr(X'X)^{-1} \tag{5.9}$$

$$\mathbb{E}(\hat{\beta}'\hat{\beta}) > \beta'\beta + \frac{\sigma^2}{\lambda_{min}} \tag{5.10}$$

Ceci appelle plusieurs remarques :

– Le vecteur des paramètres obtenu par la méthode des moindres carrés est plus allongé (en espérance) que celui du vecteur des vrais paramètres.

– Plus forte est la multicollinéarité entre les variables explicatives et plus grand sera l'allongement (du fait de l'affaiblissement de quelques valeurs propres).
– À cause du fait que l'estimateur des MCO ne soit pas biaisé, (i.e., $\mathbb{E}(\hat{\beta}) = \beta$), et au regard de (5.9) et (5.10), il n'est pas forcément colinéaire au vecteur des vrais paramètres β. Pour le rapprocher il faut donc, en plus de le raccourcir, le faire pivoter (dans la bonne direction).
– L'amélioration apportée par l'estimateur de James-Stein n'est que partielle. Se limitant uniquement au raccourcissement de l'estimateur des MCO $\hat{\beta}^* = c\beta$, cet estimateur ne permet pas de tirer un quelconque avantage que donnerait la manipulation de l'angle entre le vecteur obtenu par la méthode des MCO et celui des vrais paramètres.

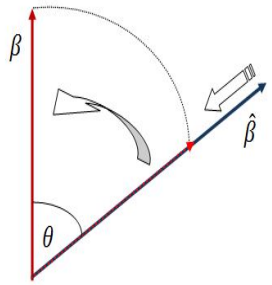


FIGURE 5.1 – Raccourcissement et pivotement de l'estimateur des MCO vers la bonne direction

Origine du problème

Les différentes méthodes de régression sont pratiquement toujours comparées à la méthode des moindres carrés. Il arrive parfois qu'elles soient comparées entre elles, mais presque toujours d'abord en référence à la méthode des moindres carrés. Or, l'estimateur des moindres carrés, qui présente plusieurs bonnes propriétés, n'est pas l'estimateur idéal dans la mesure où de par sa construction même il évite la bonne solution. En effet, du fait que la somme des carrés des résidus soit la plus petite somme des carrés des écarts, elle est nécessairement inférieure à la somme des carrés des vraies erreurs.

$$R_{min} = \sum r_i^2 \leq \sum \varepsilon_i^2$$

Ainsi, un modèle construit sur la base de ces résidus s'écarte nécessairement du VRAI modèle. Même si elle n'est pas suffisante, la condition :

$$\sum r_i^2 = \sum \varepsilon_i^2$$

Est une condition nécessaire pour espérer que le modèle estimé ait une chance d'être le modèle VRAI. Le modèle VRAI est celui dont les résidus sont égaux aux vraies erreurs. Il se trouve parmi ceux qui réalisent la condition :

$$\sum r_i^2 = \sum \varepsilon_i^2$$

Or, nous ignorons les valeurs exactes des erreurs ε_i et de ce fait nous ignorons la valeur exacte de $\sum \varepsilon_i^2$. Nous savons seulement qu'elle ne peut être que supérieure à $\sum r_i^2$.
Pour cela, la fonction de coût, qui doit être minimisée, ne doit pas atteindre $\sum r_i^2$, mais s'arrêter à une valeur qui lui est supérieure $(\sum r_i^2 + Q)$, $Q > 0$.
Ceci indépendamment de la méthode d'estimation, car cette borne inférieure $(\sum r_i^2 + Q)$ devrait vraisemblablement être égale à $\sum \varepsilon_i^2$ (ou être la plus proche possible). Cette quantité est sensée représenter les vraies erreurs dans le cas où le modèle estimé doit s'identifier au modèle VRAI.
Donc, dans son essence même, la méthode des moindres carrés introduit un écart. Dans les cas les plus simples cela ne pose pas un problème, car la somme des carrés des résidus et la somme des carrés des erreurs sont tellement voisines que le modèle estimé se rapproche beaucoup du modèle VRAI. C'est lorsque apparaissent des conditions spéciales (grand nombre de variables explicatives, forte multicollinéarité,...) que R_{min} devient "trop" petit par rapport à $E = \sum \varepsilon_i^2$. Le modèle résultant s'éloigne de ce fait un peu trop de la réalité.
E est résultat des vraies erreurs et il est immuable. Il s'agit donc, dans le processus de minimisation, d'empêcher la fonction de coût de descendre trop bas. Autrement dit, instaurer une contrainte. C'est l'idée de base de tous les estimateurs concurrents des MCO.
R ne doit pas atteindre son minimum R_{min} mais s'arrêter à $R_{min} + Q$.
Ainsi, autant de manières nous avons pour choisir Q autant nous pouvons construire d'estimateurs différents (Estimateur Ridge, estimateur de Marquardt, Estimateur de James-Stein, Lasso, Régularisation, Early stopping, ... etc).

Reste maintenant la question fondamentale :

Si nous sommes tenu d'ajouter une quantité Q, autant choisir celle qui nous guide vers la bonne direction (au sens de la norme du vecteur estimé des paramètres et de son angle par rapport au vecteur des vrais paramètres). Dans le cadre du modèle linéaire, la fonction de coût possède un minimum unique et global, car elle est quadratique par rapport aux paramètres. Il y a donc unicité de la solution des MCO. Mais quand nous nous écartons de cette valeur minimale, le nombre des solutions est infini dès que le nombre des paramètres est supérieur à 1. C'est une infinité de valeurs possibles pour le vecteur des paramètres qui aboutissent à la même valeur de :

$$R_{min} + Q$$

Lequel choisir ?

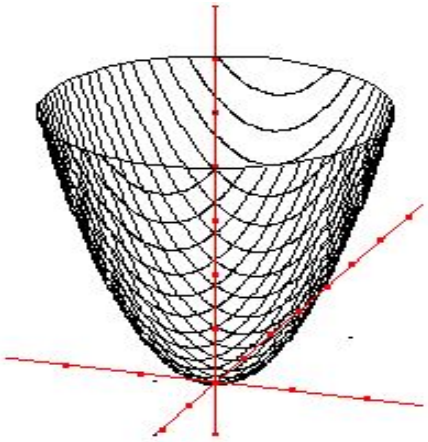


FIGURE 5.2 – Figure représentant une paraboloïde à trois dimensions

La différentiation entre les différentes méthodes (de correction) se fait par le moyen d'évaluer une valeur pour Q. Chaque technique d'évaluation (et implicitement d'interprétation) devient une nouvelle méthode d'estimation. Nous avons :

– L'estimateur Ridge ainsi que l'estimateur de Marquardt peuvent se traduire par la définition :

$$argmin_{\beta \in R^p} \parallel Y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_2^2$$

– La méthode Lasso correspond à la minimisation d'un critère des moindres carrés avec une pénalité :

$$Min_{\beta \in R^p} \| Y - X\beta \|^2 + \lambda_1 \| \beta \|_1$$

– La méthode de régularisation dans les réseaux de neurones

$$argmin_{\beta \in R^p} \| Y - X\beta \|_2^2 + \lambda \| \beta \|_2^2$$

– La méthode de l'arrêt prématuré : Cette méthode ne pose pas explicitement une contrainte sur la norme du vecteur des paramètres. Mais par le fait d'empêcher la fonction de coût d'atteindre son minimum. Elle contraint implicitement les paramètres à ne pas avoir des valeurs trop élevées. Ainsi, même si cela n'est pas formulé directement, il peut être perçu comme un problème de minimisation sous contraintes.
L'estimateur obtenu produit un vecteur des paramètres plus court que celui des moindres carrés. En effet, par le fait que les paramètres changent en vertu de :

$$w(t+1) = w(t) - \eta (X'X)^{-1} \frac{\partial E(w(t))}{\partial w(t)}$$

Où $\eta > 0$ est le taux d'apprentissage et $E(w(t))$ est la fonction d'erreur
Le vecteur résultant au temps t est :

$$w(t) = (1 - \alpha)\hat{w}_{MCO} + \alpha w(0)$$

Où $\alpha = \alpha(t) = e^{-2\eta t}$ L'estimateur est ainsi raccourci par rapport à celui des moindres carrés.

En conclusion, vu sous l'angle des résidus, toutes ces méthodes découlent du même principe.

Remarque 5.1 *Tous les estimateurs évoqués aboutissent au raccourcissement du vecteur des paramètres et à un changement de directions. Néanmoins, dans la plupart des cas, ces actions n'ont pas été visées initialement lors de l'élaboration de ces estimateurs. La conception des divers estimateurs a été faite pour leur donner certaines propriétés, principalement une meilleure qualité de généralisation et une meilleure stabilité.*
Le raccourcissement et le changement de directions ont été des conséquences indirectes mais qui se retrouvent dans tous les estimateurs (sauf dans celui de James-Stein où le raccourcissement a été explicitement cherché).

Hoerl et Kennard ont constaté l'éloignement du modèle (vecteur des coefficients) estimé par rapport au modèle vrai. Le but essentiel de leur travail est d'empêcher que cet éloignement ne soit trop grand. En faisant ceci (c'est-à-dire en pensant à réduire $L_1^2 = (\hat{\beta}-\beta)'(\hat{\beta}-\beta)$), ils ont construit un estimateur qui se trouve être raccourcissant par rapport à celui des moindres carrés. Le raccourcissement est une propriété obtenue indirectement en essayant de répondre à une autre préoccupation.
Le travail de Marquardt répond lui aussi à une autre préoccupation que celle de raccourcir directement l'estimateur.
Si Hoerl et Kennard ont choisi de 'doper' les plus petites valeurs propres, Marquardt à choisi carrément de les éliminer et de travailler avec une matrice rectangulaire. Il a été ainsi conduit à l'utilisation des inverses généralisées. Une propriété obtenue, mais non directement visée par son travail, est le raccourcissement de l'estimateur.
Le but commun de ces deux travaux est le rapprochement (ou la minimisation de l'éloignement) du vecteur estimé par rapport au vecteur vrai.
L'estimateur Lasso introduit expressément la contrainte sur le vecteur des paramètres. Il se différentie principalement des autres estimateurs par le fait de contraindre la somme des valeurs absolues des paramètres plutôt que leurs carrés, comme cela s'est fait pour les autres estimateurs.
Pour les méthodes neuronales, tous les algorithmes d'entraînement se basent sur la notion du gradient de l'erreur. De ce fait, ils héritent tous du même défaut : celui d'allonger le vecteur des paramètres au fil de l'entraînement.
Deux moyens se présentent pour contrecarrer ce défaut : Ou bien imposer une contrainte sur la norme du vecteur des paramètres lors du processus de la minimisation de la fonction de coût. C'est la méthode de régularisation, ou bien arrêter l'entraînement avant terme. De cette manière le vecteur des paramètres n'aura pas la possibilité de s'allonger excessivement. C'est la méthode de l'arrêt prématuré, qui équivaut donc à une contrainte sur la norme du vecteur des paramètres.

Cas particulier de la méthode de l'Extreme Learning Machine

La toute dernière théorie qui a vu le jour sous le nom de "Extreme Learning Machine", et que nous avons décrit au chapitre précédent présente

l'avantage de ne pas recourir à un ajustement itératif des paramètres.
Ce concept a permis de concevoir des algorithmes extrêmement rapides. Ces algorithmes sont basés sur le choix de valeurs aléatoires pour les poids de la couche cachée puis d'une solution analytique pour déterminer les poids de la couche de sortie. Tous les avantages des réseaux de neurones sont préservés, essentiellement l'approximation universelle et la bonne qualité de généralisation.
Bien que la forme du modèle reste la même que ceux des réseaux à propagation avant, un certains nombre de problèmes sont esquivés grâce au fait que les poids des neurones cachés n'ont pas à être ajustés. Ils sont de ce fait considérés comme des constantes, et les écueils des transformations non linéaires provoquées par les fonctions d'activations des neurones cachés ne se répercutent pas sur la fonction de coût. Dans cette fonction de coût, ne sont considérés variables que les poids de la couche de sortie. Dans la mesure où les fonctions d'activations de la couche de sortie sont linéaires, la fonction de coût est quadratique par rapport aux poids de la couche de sortie. Elle n'admet donc qu'un seul minimum. C'est ce qui fait que les problèmes de minima locaux sont levés par l'utilisation de cette méthode. Un autre avantage, qui a été présenté comme secondaire dans la littérature mais qui, sous l'optique que nous développons revêt une grande importance, est celui de la minimisation de la norme du vecteur des paramètres.
En effet, L'Extreme Learning Machine présente trois propriétés importantes :
1. La minimisation de l'erreur d'entraînement (que nous interprétons comme la minimisation des résidus) :
La solution $\hat{\beta} = H^{\dagger}T$ est l'une des solutions des moindres carrés du système linéaire général $H\beta = T$, et de ce fait l'erreur d'entraînement minimale peut être atteinte par cette solution :

$$\| H\hat{\beta} - T \| = \| HH^{\dagger}T - T \| = min_{\beta} \| H\beta - T \|$$

2. La plus petite norme du vecteur des poids :
En effet, la solution $\hat{\beta} = H^{\dagger}T$ possède la plus petite norme parmi toutes les solutions des moindres carrés du système $H\beta = T$:

$$\| \hat{\beta} \| = \| H^{\dagger}T \| \leq \| \beta \|$$

$$\forall \beta \in \beta : \| H\beta - T \| \leq \| Hz - T \|, \forall z \in \tilde{\mathbb{N}} \times \mathbb{N}$$

3. La solution $\hat{\beta} = H^{\dagger}T$ des moindres carrés de norme minimale du système $H\beta = T$ est unique.
Ces propriétés sont spécifiques de l'inverse généralisée de Moore-Penrose, qui a été utilisée dans l'élaboration de cet estimateur.
Ainsi, bien que conçu d'une manière foncièrement différente par rapport aux méthodes neuronales consacrées, l'Extreme Learning Machine rejoint la démarche générale du raccourcissement du vecteur des paramètres établie pour les autres estimateurs.

5.4 Différentiation des estimateurs concurrents

Malgré leurs très nombreuses similitudes, les estimateurs concurrents à celui des moindres carrés ne constituent pas un même estimateur. Ce sont des estimateurs différents. Nous allons nous restreindre à deux d'entre-eux, l'estimateur de Hoerl et Kennard et l'estimateur de Marquardt, et montrer qu'ils possèdent les mêmes propriétés alors qu'ils sont en fait différents.

L'estimateur Ridge de Hoerl et Kennard :

L'idée fondamentale dans la conception de cet estimateur est d'ajouter une petite valeur k à la diagonale de la matrice de corrélation $X'X$ L'estimateur est :

$$\hat{\beta}^* = (X'X + kI)^{-1}X'Y$$

Théorème 5.4.1 *Soit $k \geq 0$ et soit $\hat{\beta}^*$ satisfaisant :*

$$\hat{\beta}^* = (X'X + kI)^{-1}X'Y$$

Alors $\hat{\beta}^*$ minimise la somme des carrés des résidus :

$$\phi(\hat{\beta}^*) = (Y - X\hat{\beta}^*)'(Y - X\hat{\beta}^*)$$

Sur, et dans, la sphère centrée à l'origine et de rayon $\parallel \hat{\beta}^* \parallel$.
En outre, $\phi(\hat{\beta}^*)$ est une fonction croissante en k.
Ainsi, la solution ridge requiert une ***augmentation*** *de la somme des carrés des résidus par rapport à celle des moindres carrés.*

Théorème 5.4.2 *Si $\hat{\beta}^*$ est la solution de $(X'X + kI)\hat{\beta}^* = X'Y$ pour une certaine valeur de k, alors $\parallel \hat{\beta}^* \parallel$ est une fonction monotone décroissante de k telle que $\lim_{x\to\infty} \parallel \hat{\beta}^* \parallel = 0$.*

Théorème 5.4.3 *$g = X'Y$ étant le vecteur gradient de $\phi(\hat{\beta})$.*
Soit γ_k l'angle entre $\hat{\beta}^*$ et g. Alors γ_k est une fonction monotone décroissante de k telle que $\lim_{k\to\infty} \gamma_k = 0$
Du fait que g est indépendant de k, il découle que $\hat{\beta}^*$ pivote vers g quand $k \to \infty$.

Théorème 5.4.4 *L'estimation $\hat{\beta}^*$ est une transformation linéaire de $\hat{\beta}$, et la transformation ne dépend que de X et de k.*

Théorème 5.4.5 *La variance de $\hat{\beta}^*$ est :*

$$Var(\hat{\beta}^*) = \sigma^2 Z_k (X'X)^{-1} Z_k'$$

$$Var(\hat{\beta}^*) = \sigma^2 [(X'X) + kI]^{-1} [(X'X) + kI]^{-1}$$

Théorème 5.4.6 *L'erreur quadratique moyenne de $\hat{\beta}^*$ est*

$$EQM(L_1^2) = Tr[Var(\hat{\beta}^*)] + \beta'^{(Z_k - I)'(Z_k - I)}\beta \tag{5.11}$$

$$EQM(L_1^2) = Variance + (Biais)^2 \tag{5.12}$$

Le second terme à droite est le carré du biais introduit en utilisant $\hat{\beta}^*$ à la place de $\hat{\beta}$ Il sera égal à zéro quand $k = 0$. Le terme variance dans (5.12) est une fonction monotone décroissante de k.
Le terme biais dans (5.12) est une fonction monotone croissante de k. Sa valeur limite quand $k \to \infty$ est $\beta'\beta$.

Théorème 5.4.7 *Si $\beta'\beta$ est borné, alors il existe $k > 0$ tel que l'erreur quadratique moyenne de $\hat{\beta}^*$ est plus petite que l'erreur quadratique moyenne de l'estimateur des moindres carrés.*
*En conséquence, si l'erreur quadratique moyenne est adoptée comme critère, alors l'estimateur des moindres carrés est **inadmissible**, spécialement pour les données **non orthogonales**.*

Théorème 5.4.8 *L'estimateur ridge est équivalent à l'estimateur des moindres carrés lorsqu'il est ajouté, à l'ensemble des données existantes, un ensemble fictif de points pris conformément à une expérience orthogonale H_k, la réponse Y sera fixée à zéro pour chacun de ces points supplémentaires.*

Estimateur par inverse généralisé de Marquardt

L'estimateur est

$$\hat{\beta}^+ = A_r^+ X'Y$$

Où

$$A_r^+ = S_r D_r^{-1} S_r' = \sum_{i=1}^{r} \frac{1}{\lambda_i} S_i S_i'$$

Pour un rang assigné r, et où S_i est le vecteur propre de $X'X$ correspondant à λ_i.

Théorème 5.4.9 *Soit $\hat{\beta}^+$ la solution des équations normales $X'X\hat{\beta} = X'Y$ obtenu en assignant le rang r à la matrice $A = X'X$, et utilisant l'inverse généralisée A_r^+. Alors $\hat{\beta}^+$ minimise la somme des carrés des résidus :*

$$\phi(\hat{\beta}) = (Y - X\hat{\beta})'(Y - X\hat{\beta})$$

Pour tout $\hat{\beta}$ dans le sous-espace généré par S_r.

Théorème 5.4.10 *$\| \hat{\beta}^+ \|^2$ est une fonction croissante pas à pas de r.*

Théorème 5.4.11 *Soit γ_r l'angle entre $\hat{\beta}^+$ et g. Alors $\gamma_r \geq \gamma_{r-1}$ (r est un naturel) ,tant que*
γ_r et γ_{r-1} satisfont aux inégalités $(0 < \gamma_r)$,$(\gamma_r/\gamma_{r-1} << 1)$,$(\gamma_{r-1}/\gamma_{r-2} << 1)$
*Du fait que g est indépendant de $\hat{\beta}^+$, il découle que $\hat{\beta}^*$ **pivote** vers g quand r décroît.*

Théorème 5.4.12 *L'estimation $\hat{\beta}^+$ est une transformation linéaire de $\hat{\beta}$, et la transformation ne dépend que de X et de r.*

Théorème 5.4.13 *La variance de $\hat{\beta}^+$ est*

$$Var(\hat{\beta}^+) = \sigma^2 [S_r D_r^{-1} S_r'] X'X [S_r D_r^{-1} S_r']$$

Théorème 5.4.14 *L'erreur quadratique moyenne de $\hat{\beta}^+$ est*

$$EQM(L_1^2) = Tr[Var(\hat{\beta}^+)] + \beta'^{(Z_k - I)'(Z_k - I)}\beta \tag{5.13}$$

$$EQM(L_1^2) = Variance + (Biais)^2 \tag{5.14}$$

Le second terme à droite est le carré du biais introduit en utilisant $\hat{\beta}^+$ à la place de $\hat{\beta}$ Il sera égal à zéro quand $r = p$. Le terme variance dans (5.14) est une fonction monotone croissante de r. Le terme biais dans (5.14) est une fonction monotone croissante de r.

Théorème 5.4.15 *Une condition suffisante pour que l'erreur quadratique moyenne $EQM(L_1^2)$ soit inférieure à la variance de l'estimateur des moindres carrés est*

$$\sum_{i=r+1}^{p} \frac{1}{\lambda_i} > \frac{1}{\sigma^2}(\beta'\beta)$$

Ainsi, l'estimateur Ridge et l'estimateur par inverse généralisée ont en commun plusieurs propriétés :

- Tous les deux sont meilleurs que l'estimateur des moindres carrés dans le cas où les données ne sont pas orthogonales.
- Pour les deux estimateurs, le degré du biais (choix de k ou de r) peut être confiné dans un intervalle raisonnable dans chaque problème pratique.
- La solution par inverse généralisée est spécialement adaptée dans le cas où des valeurs propres valent précisément zéro.
- La solution Ridge est adaptée au cas où des valeurs propres sont très petites, mais non nulles.

Néanmoins, l'estimateur de Hoerl et Kennard est différent de celui de Marquardt. L'estimateur Ridge n'est pas un estimateur par inverse généralisée. Il suffit de constater que l'inverse du Ridge ne réalise pas

$$AA^+A = A$$

Remarque 5.2 *Ces deux estimateurs sont différents de celui de* ***James Stein****,* $\beta_{JS} = c.\beta_{MCO}$ *où* $0 < c \leq 1$.

5.5 Généralisation des résultats de Hagiwara (2002) :

Dans son article : "Regularization learning, early stopping and biased estimator" (2002), Hagiwara présente une interprétation statistique unifiée pour la méthode de l'apprentissage par régularisation et la méthode de l'arrêt prématuré pour les réseaux linéaires, dans le cadre du modèle linéaire.
Il démontre que ces deux concepts sont équivalents à l'utilisation d'estimateurs biaisés qui résulteraient de réseaux conçus pour avoir une erreur de généralisation plus faible que celle de l'estimateur des moindres carrés.
Il a également démontré que cet estimateur biaisé est un estimateur raccourci.
Il a démontré en outre que le paramètre optimal de régularisation ainsi que le temps optimal d'arrêt, au sens de la meilleure qualité de généralisation, sont obtenus en résolvant le dilemme biais-variance.
L'auteur a finalement donné les estimations, à partir des données d'entraînement, du paramètre optimal de régularisation et le temps optimal d'arrêt.
L'article s'achève avec des simulations numériques pour tenter de montrer que ces estimations sont susceptibles d'améliorer la qualité de généralisation comparativement à la méthode des moindres carrés.
Le travail élaboré tout le long de cette thèse démontre que tous ces résultats peuvent être substantiellement étendus et améliorés.
Tout d'abord, en ce qui concerne l'interprétation statistique unifiée, elle n'est plus seulement relative aux deux méthodes neuronales évoquées ci-dessus, mais elle s'étend aussi à bon nombre d'estimateurs biaisés, connus comme concurrents de l'estimateur des moindres carrés.
En outre, notre interprétation ne se restreint pas uniquement au cadre du modèle linéaire, mais elle s'étend à toute forme de modélisation (non nécessairement linéaire) ayant été investie auparavant par la méthode des moindres carrés.
Nous avons souligné, d'autre part, que le principe du raccourcissement est commun à tous les estimateurs biaisés.
Au sujet du paramètre optimal de régularisation et du temps optimal d'arrêt, dans la mesure où la notion d'optimalité est comprise au sens de la moindre erreur de généralisation, nous avons démontré au chapitre 1 en toute généralité que, quelle que soit la méthode d'estimation, la réalisation du minimum de l'erreur de généralisation ne peut pas être autre que par la résolution du

dilemme biais-variance.
Le surplus de généralisation des résultats réside dans le fait que l'espérance est prise par rapport à tout l'ensemble d'entraînement (variables explicatives et variable expliquée). Dans les travaux de Hagiwara, elle n'est prise que par rapport à la variable expliquée. Elle ne tient donc compte que de l'erreur aléatoire incluse dans le modèle sans tenir compte de la fluctuation d'échantillonnage dans les valeurs des variables explicatives ayant contribué à la construction de cet estimateur.

CHAPITRE 6

MISE EN APPLICATION

6.1 Résumé

Dans ce travail nous proposons une nouvelle méthode pour l'estimation du délai postmortem. Nous utilisons en cela des réseaux de neurones artificiels à propagation avant et à apprentissage supervisé. Nous avons mené une étude comparative sur un échantillon de 257 individus pour souligner, par rapport aux méthodes traditionnelles, l'avantage considérable qu'apporte cette nouvelle technique dans la précision des estimations.

6.2 Introduction

Un problème important qui se pose en médecine légale est celui de l'estimation du délai postmortem. Une recherche a été développée sur de longues décades et a produit des méthodes très diverses, élaborées dans le but d'aboutir à des estimations acceptables (Naumenko VG, 1984 ; Kaliszan M et al., 2009). Différentes voies ont été explorées, aussi diverses que l'entomologie médicolégale ou les analyses des liquides organiques (Amendt J et al., 2011 ; Lin X et al., 2011). La méthode thermométrique, qui est l'un des principaux outils pour l'évaluation de l'intervalle postmortem dans les premières heures qui suivent la mort, a été l'objet d'un intérêt soutenu (Berent J, 2006 ; C.Henssge et al., 2002). Son avantage repose sur le fait qu'elle soit construite sur la base de mesures quantitatives. Des d'hypothèses successives telles que :

- Le refroidissement cadavérique est proportionnel au temps.
- Le flux thermique est proportionnel à la différence de température entre le corps et l'air ambiant ont conduit à l'évolution de la méthode thermométrique.

La méthode de Henssge (Henssge C et al., 2002), bien que meilleure que celles qui l'ont précédé, ne résout pas tout à fait le problème, l'intervalle d'estimation (plus de trois heures) reste trop large pour répondre correctement à l'exigence des considérations pratiques.
Le problème est que la loi sous jacente au phénomène doit être beaucoup trop complexe pour être saisie par cette formule. Le modèle est manifestement hautement non linéaire, et il n'existe pas encore de connaissances théoriques suffisantes pour déterminer sa structure. Il semble qu'il n'y ait pas encore d'avancées décisives, bien que plusieurs tentatives de modélisation mathématique du phénomène de refroidissement corporelle soient entreprises (Henssge C et al., 1984 ; Biermann FM et Potente S, 2011).
Malheureusement, les observations réelles montrent que les modèles qui en ont résulté, linéaire pour la première hypothèse et exponentiel pour la seconde, demeurent très imprécis. Le modèle le plus récent et le plus utilisé actuellement est bâti sur la formule proposée par (C.Henssge, 2002). Cette formule tient compte de trois facteurs : La température ambiante, la température du corps et la masse corporelle.
Dans ce travail, nous proposons une nouvelle méthode pour estimer délai postmortem. Cette méthode est basée sur les réseaux de neurones artificiels. Elle sera comparé à celle de Henssge, adoptant les mêmes variables prédictives

6.3 Méthode sélectionnée

L'estimation du délai postmortem se pose naturellement en un problème de modélisation. Il s'agit de concevoir une formule qui établit quantitativement la relation précise liant les différentes variables prédictives à la variable à prédire.
Les variables prédictives sont les différentes caractéristiques mesurées sur le corps et le milieu environnant. La variable à prédire est le laps de temps séparant le moment de la mort et le moment de la prise des mesures.
Cela se ramène ainsi à la recherche d'une fonction mathématique qui est d'abord susceptible d'expliquer les observations en notre possession, et qui

continue surtout de se vérifier en dehors de l'ensemble de données qui a servi à son élaboration. Cette fonction mathématique pourrait découler d'un modèle de connaissance arrivé à maturité.
Il semble, cependant, que ce n'est pas encore le cas dans le domaine de l'estimation du délai postmortem. Des modèles différents se sont succédé, apportant de plus en plus d'amélioration dans la précision des estimations, mais néanmoins sans aboutir à des réponses satisfaisantes. La marge des erreurs dans les estimations, même avec les méthodes les plus récentes, reste trop grande. Il est possible que les formules proposées à ce jour ne rendent pas tout à fait compte de la loi sous jacente au phénomène. Si telle est la situation, imposer à la relation une forme pré-établie est une restriction qui ne peut se concevoir, tant qu'il n'y a pas de bases théoriques qui pourraient le justifier. Si, par contre, nous admettons que le but véritable est finalement la recherche de la qualité de la prédiction plutôt que la formulation explicite de la loi, alors il serait peut être plus avantageux de se tourner vers des techniques, récentes plus efficaces pour répondre à cet objectif. Ce sont les réseaux de neurones artificiels, qui ont la capacité d'apprendre directement et automatiquement des données, sans besoin de formulation a priori. Ils ont la capacité d'approximer le mieux possible la loi en question et ont un pouvoir de généralisation fiable (Simon Haykin 1999 ; Hornik et al., 1989).
Si nous pouvons admettre que la véritable loi qui gouverne le phénomène laisse nécessairement sa trace dans les observations, résultats de nos mesures, l'enjeu est de trouver le moyen d'en extraire l'information utile. Cela, bien malgré que les données soient sans doute parasitées et entachées d'erreurs.
Or, dans le cadre de l'approximation des fonctions, il est prouvé dans la théorie des réseaux de neurones artificiels (Hornik et al., 1990 ; Terrence 1999) que cet objectif pourrait être atteint d'une manière hautement satisfaisante. Il suffit pour cela de réunir un certain nombre de conditions pour garantir d'approcher au mieux la fonction enfouie dans les données. La meilleure approximation de cette fonction, une fois obtenue, fournirait des résultats dans l'estimation du délai postmortem plus satisfaisants que ceux donnés par les méthodes actuelles.
La condition première est déjà l'existence d'une relation déterministe entre les variables prédictives et la variable à prédire. **La deuxième condition** est la disponibilité d'un nombre suffisant de cas, pour chacun desquels nous connaissons d'une part, les valeurs des variables prédictives et d'autre part, la valeur vraie de la variable à prédire. **La troisième condition** est la mise en œuvre d'un réseau correctement construit et entraîné d'une manière

adéquate.
Le travail que nous proposons ici ne se fixe pas comme objectif l'aboutissement à une solution idéale, son premier but est de prouver que les méthodes neuronales sont susceptibles de conduire à des solutions beaucoup plus satisfaisantes que celles produites par les méthodes actuellement existantes. Ce qui, de fait, ouvre un champ d'investigation très prometteur.
A cette fin, nous avons choisi de mener une étude comparative opposant la méthode neuronale à la méthode de Henssge (Henssge C et Brinkmann B, 1984).
Nous avons fixé notre choix sur la méthode de Henssge, la plus récente des méthodes thermométriques, car elle est reconnue comme étant la plus précise parmi toutes les méthodes existantes, thermométriques ou autres.
Comme il s'agit d'une comparaison, les conditions doivent être rigoureusement les mêmes pour les deux méthodes, tant au niveau des variables qu'au niveau des observations. La méthode neuronale adoptera les mêmes variables prédictives que celles utilisées dans la formule de Henssge. Le délai postmortem est la variable à prédire par les deux méthodes.
Les données utilisées (257 cas) ont été récoltées avec beaucoup de soin par un médecin légiste. Cette récolte de données s'est faite sur des années, bien préalablement à notre étude comparative. Elle ne pouvait pas être menée pour avantager une méthode par rapport à une autre.
La comparaison entre les deux méthodes s'est faite sur la base de l'étendue des erreurs d'estimation, produites par chacune, sur l'ensemble des 257 observations.
Le critère que nous avons adopté est l'erreur quadratique moyenne, mais pour une illustration plus immédiate nous avons aussi noté l'erreur absolue moyenne.

6.4 Aspect technique

L'une des différences fondamentales qui existe entre les deux méthodes est que la formule de Henssge est définitivement établie et n'est plus sujette à variation. Les valeurs de ses coefficients ont été fixées avec l'échantillon qui a servi à son élaboration. Elle reste figée et ses performances demeurent constantes.
Par contre, le modèle construit sur les réseaux de neurones reste perfectible tant qu'il y aurait de nouvelles données à ajouter à l'ensemble d'entraîne-

ment. Ce nouvel apport de données peut s'ajouter à ce qui a déjà servi afin de constituer un nouvel ensemble d'entraînement. Cela se traduit toujours par un gain en performance.
L'architecture du réseau n'est pas nécessairement définitive, elle peut évoluer pour permettre de tirer le meilleur profit de tout nouvel ensemble d'entraînement.
Sous cet angle, la méthode neuronale peut être vue plutôt comme un procédé que comme une formule. Des chercheurs différents peuvent mettre en commun leurs données afin de construire un réseau plus performant que ceux qu'ils auraient construits séparément, en utilisant chacun ses seules données.
Deux questions majeures doivent être prises en considération lors de la comparaison entre les deux méthodes : le niveau de la performance et la qualité de généralisation. Une méthode sera considérée meilleure que l'autre si elle a, sur l'échantillon de comparaison, une erreur quadratique moyenne plus petite que celle de l'autre. Mais ce critère à lui seul n'est pas suffisant. Il faut surtout s'assurer que la capacité de généralisation est bonne, c'est-à-dire que la même qualité des résultats devrait s'observer sur tout autre éventuel échantillon.
Pour la méthode de Henssge, la qualité de généralisation ne dépend pas de l'ensemble utilisé pour la comparaison, c'est une propriété intrinsèque à la formulation de cette méthode. Par contre, pour la méthode neuronale, la qualité de généralisation dépend de l'ensemble d'entraînement, et il faut veiller à ce qu'elle soit à un niveau satisfaisant.
En effet, lorsque le réseau est correctement construit et lorsque la méthode d'entraînement est adéquate, nous risquons d'aboutir à un ajustement très poussé.
Ce serait la solution parfaite, mais qui ne peut être valide que dans le cas où il n'y a pas d'erreurs qui entachent les données, et où toutes les variables prédictives ont été prises en considération. Or, ce n'est sans doute pas le cas dans le problème que nous considérons. Ainsi, cette force de la méthode neuronale risque de devenir un désavantage par le fait que le réseau risque d'inclure les erreurs dans le modèle. Les résultats seraient trop bons pour l'échantillon mais, en contre partie, la généralisation sur d'autres données serait faible.
Afin d'éviter cette situation, l'ensemble des données est subdivisé en trois sous ensembles : un ensemble d'entraînement, un ensemble de validation et un ensemble de test. De cette manière, l'entraînement du réseau se fait de sorte à améliorer le plus possible la performance, mais sans que cela soit au

détriment de la capacité de généralisation.
La construction et l'entraînement d'un réseau de neurones artificiels sont tout à fait aisés. Il existe de nombreux logiciels conçus à cette fin. Il est donc possible, pour chacun ayant suffisamment de données, de créer et d'entraîner son propre réseau.
Pour ceux n'ayant pas suffisamment de données, il suffit de contacter les auteurs pour obtenir le réseau entraîné par eux, ou obtenir un programme simple conçu sous Excel et qui émule un réseau entraîné près à l'usage.
Il faut souligner que nous avons fait notre comparaison, et tiré nos conclusions sur la base des données à notre disposition. Nous sommes convaincus qu'avec plus de données les résultats seront meilleurs, principalement en ce qui touche toutes les possibilités pour la température ambiante, pour laquelle nous n'avons disposé que de valeurs comprises entre $4.5\overset{\circ}{C}$ et $18\overset{\circ}{C}$. Il reste à étendre cet intervalle vers les plus grandes et les plus petites valeurs.
Néanmoins, nous n'avons pas aimé attendre encore trop longtemps pour communiquer notre méthode. Il se peut qu'il y ait des chercheurs ou des praticiens qui possèdent déjà ce genre de données et qui peuvent donc entraîner des réseaux sur ce qu'ils possèdent. Nous pouvons également leur proposer nos données afin qu'ils aient un ensemble plus fourni.

6.5 Étude comparative

6.5.1 Description des données

La collecte des données s'est étendue sur plusieurs années. Le lieu de la collecte des données est le Centre Hospitalo-universitaire de Constantine. Toutes les mesures ont été effectuées par un médecin légiste qualifié qui a d'abord enregistré le moment de la mort puis, en des temps plus ou moins éloignés selon le cas, les valeurs des trois variables (poids, température corporelle et température ambiante). Les cas répertoriés sont des malades décédés au sein même de l'hôpital. Pour chacun de ces cas, le médecin légiste est aussitôt appelé pour constater le décès effectif du malade. Le moment du décès est noté avec le plus de précision possible.
Notre échantillon initial est constitué de tous les cas de décès constatés par le même médecin légiste. De cet ensemble nous avons exclu les cas où la mort s'est accompagnée d'une forte fièvre et les cas où l'heure de la mort n'est pas très sûre. L'ensemble restant, composé de 257 cas, constitue notre échantillon

d'étude. Les cadavres ont été déplacés vers la morgue aussitôt que la constatation de la mort est enregistrée par le médecin légiste. Pour chacun de ces cadavres, un moment précis a été choisi pour la mesure du poids du corps et de la température rectale. En ce même moment la température ambiante est notée, et c'est la température à l'intérieur de la morgue durant le séjour du cadavre. Elle est toujours mesurée avec le même dispositif. Nous avons ainsi, d'une part trois variables prédictives : le poids (P), la température rectale (T_r°) et la température ambiante (T_α°), et d'autre part, une variable à prédire qui est le délai postmortem (DPM), et qui est le laps de temps séparant le moment de la mort et le moment de la prise des mesures. Pour chacun de ces 257 cadavres nous connaissons les valeurs exactes des trois variables (P), (T_r°) et (T_α°) ainsi que la vraie valeur de la variable (DPM). Les mesures sur l'ensemble des individus ont été faîtes dans les mêmes conditions :

- Air sec sans mouvements
- Corps totalement nu depuis le moment de la mort.

La prise de la température rectale s'est faite dans tous les cas avec un matériel de même sensibilité, ainsi que la pesée des corps.
Toutes les mesures sur les trois variables ont été faites avec le plus de soin possible, et si éventuellement il y aurait de mauvaises mesures, elles le seraient au détriment des deux méthodes d'estimation à la fois, et ne seraient pas à l'avantage de l'une par rapport à l'autre.

6.5.2 Estimation par la méthode de Henssge

Le modèle de C.Henssge (Henssge C et al., 2002) est bâti sur la formule :

$$\frac{T_r^{\circ} - T_\alpha^{\circ}}{37.2^{\circ} - T_\alpha^{\circ}} = 1.25e^{-kt} - 0.25e^{-5kt} \tag{6.1}$$

Où k est un paramètre dépendant du poids P de l'individu :

$$k = \frac{1.2815}{P^{0.625}} - 0.0284$$

Cette formule tient compte de trois facteurs : la masse corporelle P (en kilogrammes), la température ambiante T_α° et la température du corps T_r° (en degré Celsius). Le temps (t) marquant le délai entre la survenue de la mort et le moment des mesures des températures et du poids, est déduit de cette formule. Ce temps (t) est l'estimation du délai postmortem. Nous avons

calculé ce temps pour les 257 observations. Voir (Schweitzer, W) et (Smart JL et Kaliszan M) pour un calcul automatisé. Pour chaque observation, l'écart entre ce temps (t) et la vraie valeur du DPM constitue l'erreur d'estimation. La moyenne des carrés de ces écarts est l'erreur quadratique moyenne (EQM).

6.5.3 Estimation par la méthode neuronale

6.5.3.1 Construction du réseau

Le réseau utilisé est un réseau à propagation avant. Il est doté de :
- Une couche cachée de 10 neurones ayant chacun la tangente hyperbolique comme fonction d'activation.
- Un neurone de sortie dont la fonction d'activation est linéaire. Le choix de cette architecture est motivé par ce qui suit :

• Le travail remarquable de Hornik (Hornik et al., 1990 ; Terrence., 1999) garantie que s'il existe une relation déterministe entre les variables d'entrée et la variable de sortie, nous pouvons être en mesure de construire un réseau capable de l'approximer au mieux. Ce réseau comporte une seule couche de neurones cachés ayant tous la même fonction de transfert et une couche de sortie comportant un seul neurone dont la fonction de transfert est linéaire.
• Le seul souci reste alors la détermination du nombre adéquat de neurones dans la couche cachée. Ce nombre est lié à la quantité de données à notre disposition. Autant ce nombre de neurones est grand autant le réseau est flexible et donc autant l'approximation est bonne (Hagan M.T et M. Menhaj 1994). Cependant cette optimisation ne peut pas être conduite à ses limites car, dans la mesure où les variables prédictives sont entachées d'erreurs, il faut veiller aussi à la qualité de généralisation. L'équilibre biais-variance doit être assuré. Un nombre trop réduit de neurones cachés provoque un grand biais du modèle, tandis qu'un nombre trop grand conduit à une grande variance. Dans les deux cas, c'est une mauvaise qualité de généralisation qui est obtenue. Il n'y a pas de méthodes systématiques qui permettent d'obtenir le nombre optimal de neurones cachés. Ce nombre reste tributaire de l'échantillon utilisé pour l'entraînement du réseau. Dans notre présente étude, nous avons trouvé que 10 neurones cachés assurent le mieux l'équilibre entre le bais et la variance. Cependant ce nombre n'est pas strictement le seul nombre optimal, nous pouvons ajouter ou retrancher jusqu'à deux neurones sans que la qualité ne se dégrade sensiblement.
• Nous avons préféré utiliser la tangente hyperbolique comme fonction d'ac-

tivation pour les neurones cachés à cause du fait qu'elle soit antisymétrique. C'est une propriété qui permet au réseau à propagation avant, lorsqu'il est entraîné avec l'algorithme de rétro-propagation (ou ses variantes), d'apprendre plus vite, en termes de nombre d'itérations d'entraînement requis. Les autres fonctions sigmoïdes ne jouissant pas de cette propriété, conduisent à un apprentissage plus lent.

6.5.3.2 Méthode d'entraînement

L'algorithme de Levenberg Marquardt est la méthode la plus appropriée pour l'entraînement de notre réseau. En effet, ce réseau n'étant pas très gros, il n'y a pas lieu d'utiliser la méthode du gradient simple ou ses variantes dont les temps de convergence sont supérieurs de plusieurs ordres de grandeur à ceux des méthodes du second ordre (Friedlander). En outre, parmi ces dernières, la méthode de Levenberg Marquardt se distingue par le fait qu'elle est spécifiquement conçue pour minimiser l'erreur quadratique moyenne, qui est la fonction de coût que nous avons adopté dans notre étude.
L'ensemble des observations à notre disposition, composé de 257 observations, a été subdivisé par une procédure aléatoire en trois sous ensembles :

- Un ensemble d'entraînement composé de 60% des observations. Ce sont 155 cas qui sont présentés au réseau durant l'entraînement. Le réseau est ajusté en fonction des erreurs enregistrées sur ces cas.
- Un ensemble de validation composé de 20% des observations. Ce sont 51 cas qui sont utilisés pour mesurer la capacité de généralisation du réseau.
  L'entraînement s'arrête automatiquement lorsque la performance cesse de s'améliorer sur cet ensemble.
- Un ensemble de test composé de 20% des observations. Ce sont 51 cas qui n'ont pas un effet sur l'entraînement et qui fournissent ainsi une mesure indépendante de la performance du réseau pendant et après l'entraînement.

La qualité de l'entraînement du réseau dépend de la subdivision de l'ensemble des données en ces trois sous ensembles. Chacune des subdivisions possibles conduit à un réseau entraîné avec ses propres valeurs des paramètres et ses propres performances.
L'entraînement du réseau peut être reconduit autant de fois avec de nouvelles subdivisions de l'ensemble des données jusqu'à ce que deux critères soient simultanément satisfaits le mieux possible :

– La meilleure performance globale du réseau.
– Des valeurs de performances voisines sur les trois sous ensembles : d'entraînement, de validation et de test.

6.5.3.3 Estimation

Le réseau qui résulte du meilleur entraînement est celui qui sera adopté pour les estimations ultérieures. Ses paramètres étant fixés, la procédure d'estimation consiste à fournir à ce réseau, pour chaque cas, les valeurs des variables prédictives et à noter sa réponse. Cette réponse est l'estimation de la valeur du délai post mortem pour le cas considéré.

6.6 Résultats et discussion

6.6.1 Résultats

Notre ensemble de données est composé de 257 observations. Pour chacune de ces observations, nous connaissons la vraie valeur du délai postmortem (tv).
La formule de Henssge est utilisée pour estimer ce délai postmortem (tv) à partir des valeurs du poids (P), de la température rectale (T_r°) et de la température ambiante (T_α°) .
Nous avons utilisé le réseau de neurones entraîné pour effectuer cette même tâche d'estimation, en utilisant les mêmes valeurs.
Notre but est de comparer la performance de la méthode de Henssge et celle des réseaux de neurones, en termes d'écart de l'estimation du délai postmortem (ev), par rapport à sa vraie valeur (tv).
Le critère de comparaison entre les deux méthodes est construit sur cet écart sous la forme d'erreur quadratique moyenne (EQM).

$$EQM = \frac{1}{257}\sum_{i=1}^{257}(tv_i - ev_i)^2 \tag{6.2}$$

Où $i = 1, 2, ..., 257$ est le numéro de l'observation. Les résultats ont été les suivants :

EQM	Valeurs	Intervalle de confiance (niveau 95%)
Méthode de Henssge	$EQM_H = 20.83$	$(17.34 - 24.32)$
Méthode neuronale	$EQM_{NN} = 5.69$	$(4.54 - 6.84)$

TABLE 6.1 – Les valeurs de l'EQM, Méthode de Henssge vs Méthode neuronale

À noter que, dans cette formule, l'unité de temps se trouve élevée au carré.
Pour cette raison, nous avons calculé aussi l'erreur absolue moyenne (EAM) pour chacune des deux méthodes :

$$EAM = \frac{1}{257} \sum_{i=1}^{257} |tv_i - ev_i| \tag{6.3}$$

On obtient les résultats suivants :

EAM	Valeurs	Intervalle de confiance (niveau 95%)
Méthode de Henssge	$EAM_H = 3.52$ (une erreur moyenne d'environ 3 heures et demi)	$(3.17 - 3.88)$
Méthode neuronale	$EAM_{NN} = 1.85$ (une erreur moyenne d'environ une heure et 50 minutes)	$(1.66 - 2.03)$

TABLE 6.2 – Les valeurs de l'EAM, Méthode de Henssge vs Méthode neuronale

6.6.2 Discussion

6.6.2.1 Premier point : Inférieur à 7 heures

La méthode neuronale apporte ainsi un gain substantiel dans la précision de l'estimation du délai post mortem. Le résultat est appréciable, cela même avec un ensemble d'apprentissage de taille assez réduite. L'avantage de cette

méthode est qu'il suffit d'apporter un peu plus de données pour que les résultats soient encore meilleurs.
Dans l'ensemble des 257 observations qui a servi à notre étude, les valeurs du vrai délai postmortem s'étalent de 20 minutes à 18 heures et 20 minutes. Nous avons ainsi une moyenne d'environ 15 observations par unité de temps. Nous avons repris notre étude comparative exactement de la même manière, mais en la restreignant aux observations dont le délai postmortem ne dépasse pas 7 heures. Cet ensemble est constitué de 184 observations. La moyenne est d'environ 32 observations par unité de temps. Les résultats ont été les suivants :
Erreur quadratique moyenne par la méthode de Henssge :

EQM	Valeurs	Intervalle de confiance (niveau 95%)
Méthode de Henssge	$EQM_H = 21.14$	$(16.84 - 25.45)$
Méthode neuronale	$EQM_{NN} = 1.21$	$(0.95 - 1.46)$

TABLE 6.3 – Les valeurs de l'EQM, Méthode de Henssge vs Méthode neuronale

EAM	Valeurs	Intervalle de confiance (niveau 95%)
Méthode de Henssge	$EAM_H = 3.51$ (une erreur moyenne d'environ 3 heures et demi)	$(3.08 - 3.94)$
Méthode neuronale	$EAM_{NN} = 0.86$ (une erreur moyenne d'environ 52 minutes)	$(0.76 - 0.96)$

TABLE 6.4 – Les valeurs de l'EAM, Méthode de Henssge vs Méthode neuronale

Cette qualité d'estimation s'accompagne d'une qualité de généralisation du même ordre. Au vu des deux études, sur l'ensemble entier des données et sur le sous ensemble restreint aux sept premières heures de la mort, nous pourrions suspecter qu'il y ait de multiples raisons qui font que la performance est moins bonne lorsque l'étendue de la variable à prédire est plus large. Ceci mérite des investigations plus poussées. Néanmoins, dans la mesure où nous avons maintenu les mêmes conditions pour les deux cas, la raison

qui nous semble la plus vraisemblable est la densité des données par unité de temps.
Le choix précis de la barre des 7 heures n'est pas obligatoire, ni tout à fait arbitraire. En fait, toutes les valeurs qui lui sont voisines peuvent être prises comme frontières de séparation. En outre, ce choix n'a pas été dicté uniquement par la question de la densité des observations par unité de temps, Il y a d'autres raisons qui nous ont poussés à focaliser sur les premières heures qui suivent la mort.
La première est le fait, qui est admis, que le refroidissement corporel connaît d'abord un palier dans lequel la descente de la température du corps est lente et rend difficile l'estimation du délai postmortem (Smart JL et Kaliszan M., 2012).
La deuxième raison est que nous avons observé, sur nos données, que la méthode de Henssge a tendance à surestimer le délai post mortem dans les premières heures. L'erreur moyenne sur les sept premières heures est de 2.86 heures. En contre partie, elle a tendance à le sous estimer par la suite, et l'erreur moyenne vaut -1.42 heures. Il y a comme une rupture de modèle, et nous avons pensé qu'il serait intéressant d'entraîner un réseau exclusivement sur ce premier intervalle de temps qui suit la mort. Notre premier réseau, qui est entraîné sur toutes les données, à tendance à sous estimer le délai postmortem dans les premières heures avec une erreur moyenne de -0.64 heures. Mais ce deuxième réseau, qui est spécialisé sur cet intervalle, n'enregistre aucun biais systématique et l'erreur moyenne est inférieure à 0.07 heures.

6.6.2.2 Deuxième point : Coefficients correctifs

La formule de Henssge s'applique sans changements lorsque le cadavre se trouve dans un endroit où l'air est sec et sans mouvements, et lorsque le corps est totalement nu depuis le moment de la mort. Mais lorsque ces conditions ne sont pas respectées, un coefficient correctif est appliqué pour tenir compte des diverses variantes, la valeur de l'estimation est multipliée selon le cas :

- Par un nombre compris entre 0.35 et 0.95 en présence de facteurs qui accélèrent le refroidissement du corps, tels que mobilité de l'air ou présence de l'eau.
- Et par un nombre compris entre 1.1 et 2.4 en présence de facteurs qui ralentissent le refroidissement, tels que couvertures ou habillement chaud.

Tout comme pour la méthode de Henssge, la méthode neuronale a également besoin de coefficients correctifs pour pouvoir tenir compte des situations diverses rencontrées sur le terrain.
Les coefficients adoptés pour la méthode de Henssge peuvent tout aussi bien être utilisés, et de la même manière, pour la méthode neuronale. La raison en cela est que si l'une des méthodes produit une erreur d'estimation plus forte que l'autre, le coefficient correctif, qui ne fait qu'amplifier ou atténuer ces erreurs, ne provoquera pas un changement dans le sens de l'inégalité.

6.6.2.3 Troisième point : Éviter le surapprentissage

Les réseaux multicouches à propagation avant peuvent produire des solutions parfaites parce qu'ils sont des approximateurs universels. Ils sont capables d'approximer n'importe quelle fonction mesurable d'un espace de dimension finie à un autre avec un degré de précision désiré s'il y'a suffisamment des couches cachées (Hornik K et autres., 1989 ; Hornik K., 1991). Cependant, cela ne vaut que si les variables prédictives sont correctement identifier d'une part, et les valeurs de la fonction sont exactes et non affectées par des erreurs d'autre part. Ceci est sans doute pas le cas dans le problème que nous considérons. Ainsi, nous devons être prudents lors du traitement de ce problème, car la recherche d'une très bonne performance peut conduire à une surajustement. Il est donc nécessaire d'assurer la qualité de la généralisation, tout en cherchant une bonne performance. Pour cette raison, l'ensemble de validation et l'ensemble de test ne devraient pas être de trop petite taille (moins de 20% des observations).

6.7 Conclusion

L'étude expérimentale comparative que nous avons faite prouve qu'avec même un échantillon de taille assez réduite nous arrivons à des améliorations substantielles dans l'estimation du délai postmortem par rapport aux estimations fournies par la formule de Henssge. Elles le sont particulièrement pour le premier palier de refroidissement (les 7 premières heures après la mort). C'est ce palier qui posait justement problème pour les méthodes traditionnelles du fait de la lenteur de la variation thermique. En outre, une extension de l'étude est tout à fait concevable. En effet, lorsque le nombre des observations augmente, les réseaux de neurones améliorent la précision des estimations et la

capacité de la généralisation. Par rapport aux méthodes traditionnelles qui produisent des formules fixes et dont les performances restent constantes, les réseaux de neurones ont la capacité d'être continuellement amélioré.
En raison de la mise en œuvre pratique et assez facile, l'entrainement peut être reprise chaque fois que nous avons des données supplémentaires.
Les nouvelles techniques, tels que les réseaux de neurones artificiels ou les systèmes neuro-flous, basées sur l'apprentissage par l'exemple plutôt que sur la détermination des paramètres d'un modèle pré établi, pourraient nous permettre de faire une prospection très large. Les données recèlent nécessairement l'information inhérente au phénomène. L'enjeu est de pouvoir l'extraire de ces données. Ainsi, toutes les variables mesurables sont candidates à étude.
La méthode que nous avons proposé, ne nécessitant pas un modèle de connaissance a priori, ouvre des champs d'investigations dans plusieurs directions et qui peuvent même sortir du cadre strict de la thermométrie. Nous pouvons envisager de :

- Renforcer les résultats sur les mêmes domaines d'apprentissage.
- Etendre ces domaines d'apprentissage, particulièrement celui de la température ambiante (au-delà de $18\overset{\circ}{C}$).
- Inclure des variables impliquant, d'une manière directe ou indirecte, la surface du corps et son volume.
- Inclure des variables quantifiables autres que la température corporelle (tels que les concentrations de certains d'éléments chimiques dans différentes parties du corps).
- Vérifier l'influence de l'âge et du sexe.
- Quantifier et intégrer les caractéristiques du milieu.
- Vérifier l'influence de la forme médico-légale de la mort (violente et non violente).

Remerciements

Nous sommes très reconnaissants au Professeur A. Belloum, unité de thanatologie, Département de médecine légale du CHU Ben Baddis Constantine Algérie, qui nous a fourni de toutes les données et sans lui, ce travail ne serait pas achevé.

Bibliographie et Références

[1] Açikgöz HN. (2010). Forensic entomology. *Turkiye Parazitol Derg,* 34(3) :216-21.

[2] Al-Alousi, L. M., Anderson, R. A., Worster, D. M., & Land, D. V. (2002). Factors influencing the precision of estimating the postmortem interval using the triple-exponential formulae (TEF) : Part II. A study of the effect of body temperature at the moment of death on the postmortem brain, liver and rectal cooling in 117 forensic cases. *Forensic science international,* 125(2), 231-236.

[3] Al-Alousi, L. M. (2002). A study of the shape of the post-mortem cooling curve in 117 forensic cases. *Forensic science international,* 125(2), 237-244.

[4] Amendt, J., Richards, C. S., Campobasso, C. P., Zehner, R., & Hall, M. J. (2011). Forensic entomology : applications and limitations. *Forensic science, medicine, and pathology,* 7(4), 379-392.

[5] Amendt, J., Krettek, R., & Zehner, R. (2004). Forensic entomology. *Naturwissenschaften,* 91(2), 51-65.

[6] Barbero, A., & Sra, S. (2011). Fast Newton-type methods for total variation regularization. *In Proceedings of the 28th International Conference on Machine Learning*(ICML-11) (pp. 313-320).

[7] Bartlett, P. L. (1997). For valid generalization, the size of the weights is more important than the size. *Advances in neural information processing systems (NIPS),* 9, 134.

[8] BenoíT, F., Van Heeswijk, M., Miche, Y., Verleysen, M., & Lendasse, A. (2013). Feature selection for nonlinear models with extreme learning machines. *Neurocomputing,*102, 111-124.

[9] Berent, J. (2004). Determining the post mortem interval based on temperature measurements. Part I : From the first 19th century studies to Marshall & Hoare's double exponential model. *Archiwum medycyny sadowej i kryminologii,* 55(3), 209-214.

[10] Berent, J. (2006). Determining post mortem interval by temperature data. Part II : research results from the 1970s to the end of the 20th century. *Archiwum medycyny sadowej i kryminologii,* 56(2), 103.

[11] Biermann, F. M., & Potente, S. (2011). The deployment of conditional probability distributions for death time estimation. *Forensic science international,* 210(1), 82-86.

[12] Chibat, A., Zerdazi, D., & Rahmani, F. L. (2016). Estimation of post-mortem period by means of artificial neural networks. *Electronic Journal of Applied Statistical Analysis,* 9(2), 326-339.

[13] Cibas, T. ; Fogelman Soulié, F. ; Gallinari, P. and Raudys, S. (1994). Variable Selection with Optimal Cell Damage. *In Proceedings of ICANN'94.*

[14] Coolen, A. C., Kühn, R., & Sollich, P. (2005). Theory of neural information processing systems. *OUP Oxford.*

[15] Cox, D. R. (2006) Principles of statistical inference. *Cambridge University Press.*

[16] Cox, D. R., & Barndorff-Nielsen, O. E. (1994). Inference and asymptotics. *CRC Press.* (Vol. 52).

[17] Cornelius T.(1998). Neural Network Systems, Techniques and Applications, Volumes 1, 2, 3, 4, 5, 6 & 7,. *Leondes ; Academic Press, California USA.*

[18] Cornelius T.(2002). Database and data Communication Network Systems , Volumes 1, 2 & 3. *Leondes ; Academic Press, California USA.*

[19] Cornelius T. (2003). Intelligent Systems, Technology and Applications, Volumes 1, 2, 3, 4, 5 & 6. *Cornelius T. Leondes ; Academic Press, California USA.*

[20] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems,* 2(4), 303-314.

[21] Czernichow, T. (1996.) Architecture Selection through Statistical Sensitivity Analysis. *In Proceedings of ICANN'96, Bochum, Germany.*

[22] Den Hartog, E. A., & Lotens, W. A. (2004). Postmortem time estimation using body temperature and a finite-element computer model. *European journal of applied physiology,* 92(6), 734-737.

[23] Dias, F. M., Antunes, A., & Mota, A. M. (2003). Regularization versus early stopping : A case study with a real system. *In 2nd IFAC Conference Control Systems Design, Bratislava, República Eslovaca.*

[24] Doan, C. D., & Liong, S. Y. (2004, July). Generalization for multilayer neural network Bayesian regularization or early stopping. *In Proceedings of Asia Pacific Association of Hydrology and Water Resources 2nd Conference*(pp. 5-8).

[25] Dreyfus, G., Martinez, J. M., Samuelides, M., Gordon, M. B., Badran, F., Thiria, S., & Hérault, L. (2002). Réseaux de neurones-Méthodologie et applications. *Réseaux de neurones-Méthodologie et applications.*

[26] Dunne, R. A. (2007). A statistical approach to neural networks for pattern recognition *John Wiley & Sons.* (Vol. 702).

[27] Efron, B. (1998). RA Fisher in the 21st century. *Statistical Science,* Vol. 13, No. 2, 95-122.

[28] Engelbrecht, A. P. (2001). A new pruning heuristic based on variance analysis of sensitivity information. *IEEE transactions on Neural Networks,* 12(6), 1386-1399.

[29] Evgeniou, T., Poggio, T., Pontil, M., & Verri, A. (2002). Regularization and statistical learning theory for data analysis. *Computational Statistics Data Analysis,* 38(4), 421-432.

[30] Fine, T. L. (2006). Feedforward neural network methodology. *Springer Science & Business Media.*

[31] Fournier Martin. (2007-2008). La multicolinéarité. *Licence Econométrie / MASS. Econométrie II.*

[32] Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. *Springer, Berlin : Springer series in statistics.* (Vol. 1).

[33] Funahashi, K. I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural networks,* 2(3), 183-192.

[34] Gallinari, P., & Cibas, T. (1999). Practical complexity control in multilayer perceptrons. *Signal Processing,* 74(1), 29-46.

[35] Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation,* 4(1), 1-58.

[36] Giles, R. C. S. L. L. (2001). Overfitting in Neural Nets : Backpropagation, Conjugate Gradient, and Early Stopping. *In Advances in Neural Information Processing Systems 13 : Proceedings of the 2000 Conference* (Vol. 13, p. 402). MIT Press.

[37] Green, M. A., & Wright, J. C. (1985). The theoretical aspects of the time dependent Z equation as a means of postmortem interval estimation using body temperature data only. *Forensic science international,* 28(1), 53-62.

[38] Guo, P., Lyu, M. R., & Chen, C. L. P. (2003). Regularization parameter estimation for feedforward neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics),* 33(1), 35-44.

[39] Gustafson and Hajlmarsson. (1995). 21 maximum likelihood estimators for model selection. *Automatica.*

[40] Hassibi, B.; Stork, D.G. and Wolf, G. (1994). Optimal Brain Surgeon : Extensions and Performance Comparisons. *Neural Information Processing Systems* 6 :263-270.

[41] Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE transactions on Neural Networks,* 5(6), 989-993.

[42] Hagiwara, K., & Kuno, K. (2000). Regularization learning and early stopping in linear networks. *In Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference,* (Vol. 4, pp. 511-516). IEEE.

[43] Hagiwara, K. (2002). Regularization learning, early stopping and biased estimator. *Neurocomputing,* 48(1), 937-955.

[44] Haykin, S., & Network, N. (2004). A comprehensive foundation. *Neural Networks,* 2(2004).

[45] Henssge, C., & Brinkmann, B. (1984). Determination of time of death by rectal temperature. Mathematical analysis of empirical material versus thermodynamic modeling. *A critical case presentation. Archiv für Kriminologie,* 174(3-4), 96.

[46] Hestenes, M. R., & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *NBS,* (Vol. 49, p. 1).

[47] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics,* 12(1), 55-67.

[48] Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression : applications to nonorthogonal problems. *Technometrics,* 12(1), 69-82.

[49] Hoerl, A. E., & Kennard, R. W. (1976). Ridge regression iterative estimation of the biasing parameter. *Communications in Statistics-Theory and Methods,* 5(1), 77-88.

[50] Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks,* 2(5), 359-366.

[51] Hornik, K., Stinchcombe, M., & White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks,* 3(5), 551-560.

[52] Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks,* 4(2), 251-257.

[53] Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine : theory and applications. *Neurocomputing,* 70(1), 489-501.

[54] Huang, G., Huang, G. B., Song, S., & You, K.(2015). Trends in extreme learning machines : A review. *Neural Networks,* 61, 32-48.

[55] Jashnani, K. D., Kale, S. A., & Rupani, A. B. (2010). Vitreous humor : biochemical constituents in estimation of postmortem interval. *Journal of forensic sciences,* 55(6), 1523-1527.

[56] James, W., & Stein, C. (1961, June). Estimation with quadratic loss. *In Proceedings of the fourth Berkeley symposium on mathematical statistics and probability,* (Vol. 1, No. 1961, pp. 361-379).

[57] Kaliszan, M., Hauser, R., & Kernbach-Wighton, G. (2009). Estimation of the time of death based on the assessment of post mortem processes with emphasis on body cooling. *Legal Medicine,* 11(3), 111-117.

[58] Kil'diushov, E. M., & Kil'diushov, M. S. (2001). Determination of the time of death according to rectal thermometry data using computer calculations. *Sudebno-meditsinskaia ekspertiza,* 45(5), 3-5.

[59] Knight, B. (1988). The evolution of methods for estimating the time of death from body temperature. *Forensic Science International,* 36(1), 47-55.

[60] Le Cun, Y., Denker, J., Solla, S., Howard, R. E., & Jackel, L. D. (1990). Optimal Brain Damage-Advances in Neural Information Processing Systems II. *Morgan Kauffman, San Mafeo, CA.*

[61] LeCun, Y. ; Denker, J.S. and Solla, S.A. (1990). Optimal Brain Damage. *Neural Information Processing Systems,* 2 :598-605.

[62] Leray, P. and Gallinari, P. (1997). Report on Variable Selection. *Neurosat Project, Environment and Climate DG III, Science, Research and Development,* ENV4-CTP96-0314, D1-1-1.

[63] Leray, P. and Gallinari, P. (1999). Feature Selection with Neural Networks. *Behaviormetrika (special Issue on Analysis of Knowledge Representation in Neural Network Models.* 26(1) :145-166.

[64] Leray, P., & Gallinari, P. De l'utilisation d'OBD pour la sélection de variables. *Tech. Rep.* (No. 01-001).

[65] Lin, X., Yin, Y. S., & Ji, Q. (2011). Progress on DNA quantification in estimation of postmortem interval. *Fa yi xue za zhi,* 27(1), 47-9.

[66] Marc Parizeau. (2006). Réseaux de neurones. *GIF-21140 et GIF-64326,* Université Laval. Canada.

[67] Mao, J. ; Mohiuddin, K. and Jain, A.K. (1994). Parsimonious Network Design and Feature Selection Through Node Pruning. *In Proceedings of the 12th International Conference on Pattern Recognition.* 622-624.

[68] Marquaridt, D. W. (1970). Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation. *Technometrics,* 12(3), 591-612.

[69] Matzner-Løber, É. (2007). Régression : Théorie et applications. *Springer Science & Business Media.*

[70] MacKay, D. J. (1992). Bayesian interpolation. *Neural computation,* 4(3), 415-447.

[71] Moody, J. (1991). Note on generlization, regularization and architecture selection in non linear learning systems. *Proceedings of the first IEEE Workshop on Neural Networks for Signal Processing. 1-10.*

[72] Muñoz Barús, J. I., Febrero-Bande, M., & Cadarso-Suárez, C. (2008). Flexible regression models for estimating postmortem interval (PMI) in forensic medicine. *Statistics in medicine,* 27(24), 5026-5038.

[73] Naumenko, V. G. (1983). Current state and perspectives of the solution of the problem of determining the time of death. *Sudebno-meditsinskaia ekspertiza,* 27(2), 9-12.

[74] Neter, J., Kutner, M. H., Nachtsheim, C. J., & Wasserman, W. (1996). Applied linear statistical models. (Vol. 4, p. 318). Chicago : Irwin.

[75] Nelson, E. L. (2000). Estimation of short-term postmortem interval utilizing core body temperature : a new algorithm. *Forensic science international,* 109(1), 31-38.

[76] Neville, J., & Jensen, D. (2008). A bias/variance decomposition for models using collective inference. *Machine Learning,* 73(1), 87-106.

[77] Nokes, L. D. M., Henssge, C., Knight, B. H., Madea, B., & Krompecher, T. (2002). The estimation of the time since death in the early postmortem period. *Hodder Arnold.*

[78] Rao, C. R., & Toutenburg, H. (1995). Linear models. In Linear models. *Springer New York.* (pp. 3-18).

[79] Plutowski, M. (1994). Selecting training exemplars for neural network learning. *(Doctoral dissertation, University of California, San Diego).*

[80] Pigolkin, I., Bogomolov, D. V., & Korovin, A. A. (1998). The current methods for determining the time of death. *Sudebno-meditsinskaia ekspertiza,* 42(3), 31-33.

[81] Plutowski, M. (1994). Selecting training exemplars for neural network learning. *Doctoral dissertation, University of California, San Diego.*

[82] Rosin, P. L., & Fierens, F. (1995, July). Improving neural network generalisation. *In Geoscience and Remote Sensing Symposium. IGARSS'95.'Quantitative Remote Sensing for Science and Applications',* (Vol. 2, pp. 1255-1257). IEEE.

[83] Rumerhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagation errors. *Nature,* 323, 533-536.

[84] Rumelhart, D. E., McClelland, J. L., & PDP Research Group. (1988). Parallel distributed processing. *IEEE.* (Vol. 1, pp. 354-362).

[85] Samarasinghe, S. (2006). Neural networks for applied sciences and engineering : from fundamentals to complex pattern recognition. *CRC Press.*

[86] Sigurdsson, S., Larsen, J., & Hansen, L. K. (2000). On comparison of adaptive regularization methods. *In Neural Networks for Signal Processing*

X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop IEEE. (Vol. 1, pp. 221-230).

[87] Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P. Y., ... & Juditsky, A. (1995). Nonlinear black-box modeling in system identification : a unified overview. *Automatica,* 31(12), 1691-1724.

[88] Smart, J. L., & Kaliszan, M. (2012). The post mortem temperature plateau and its role in the estimation of time of death. *A review. Legal Medicine,* 14(2), 55-62.

[89] Stahlberger, A. and Riedmiller, M. (1997). Fast Network Pruning and Feature Extraction Using the Unit-OBS Algorithm. *Neural Information Processing Systems,* 9 :655-661.

[90] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological),* 111-147.

[91] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological),* 267-288.

[92] Tiffanie Guillot, Antoine Denoyer. (2013). Méthodes de type LASSO pour modélisation et sélection de variables en très grande dimension. *M1 SITN, Université Claude Bernard.*

[93] Tokpavi Sessi. Le modèle de régression linéaire multiple et la méthode des moindres carrés ordinaires. *EconomiX-CNRS,Université Paris Ouest, 200 avenue de la République, 92001 Nanterre Cedex, France. Bureau : Bâtiment G-515.*

[94] Toukourou, M. S., Johannet, A., & Dreyfus, G. (2009). Flash flood forecasting by statistical learning in the absence of rainfall forecast : a case study. *In International Conference on Engineering Applications of Neural Networks* (pp. 98-107). Springer Berlin Heidelberg.

[95] Tresp, V. ; Neuneier, R. and Zimmermann, G. (1997). Early Brain Damage. *Neural Information Processing Systems,* 9 :669-675.

[96] Van de Laar, P. ; Gielen, S. and Heskes, T. (1997). Input Selection with Partial Retraining. *In Proceedings of ICANN'97.*

[97] Vavilov, A., & Viter, V. I. (2007). Using some modern mathematical models of postmortem cooling of the human body for the time of death determination. *Sudebno-meditsinskaia ekspertiza,* 50(5), 9.

[98] Vapnik, V. (2013). The nature of statistical learning theory. *Springer Science & Business Media.*

[99] Vapnik, V. N., & Vapnik, V. (1998). Statistical learning theory *New York : Wiley.* (Vol. 1).

[100] Verica, P., Janeska, B., Gutevska, A., & Duma, A. (2007). Post mortem cooling of the body and estimation of time since death. *Soudni lekarstvi/casopis Sekce soudniho lekarstvi Cs. lekarske spolecnosti J. Ev. Purkyne,* 52(4), 50-56.

[101] Viter, V. I., & Vavilov, A. (2007). State-of-the-art of mathematical modeling of postmortal thermodynamics for the time of death determination. *Sudebno-meditsinskaia ekspertiza,* 51(1), 15-18.

[102] Young, G. A., & Smith, R. L. (2005). Essentials of statistical inference. *Cambridge University Press.* (Vol. 16).

ملخص

هذه الأطروحة هى محاولة متواضعة، لإدراج نظرية الشبكات العصبونية فى إطار الإحصاء التطبيقى. تم فيها دراسة السؤال الجوهري المتعلق بالإستقراء الإحصائي في ضوء المقاربة العصبونية.

مفهوم التعميم كان محل اهتمام بالغ بهدف تصميم مقاربة موحدة تضم الطرق الإحصائية التقليدية وتلك الناتجة عن نظرية الشبكات العصبونية، وقدّمت على أنها ناتجة من نفس المفهوم.

المقدرات البديلة لطريقة أقل التربيعات قدّمت على التوالي موازاة للطرق العصبونية المختلفة المنشأة للتطرق لمسائل التقهقر و التنبؤ. وقد أجريت دراسة مقارنة بهدف إظهار أن المفهوم الأساسى، في الأصل لجميع هذه الطرق المختلفة يمكن أن ينظر إليه على أنه مفهوم موحد.

تم تقديم حالة تطبيقية لتوضيح القدرة التنبؤية للشبكات العصبونية مقارنة بالأساليب التقليدية.

Abstract

This thesis is an attempt to contribute, even slightly, in situating the neural networks theory into the framework of applied statistics. The central issue of statistical inference was studied under the light of neural approach. A lot of attention was payed to the notion of generalization, with the aim to conceive an unified approach, ghattering toghether traditional statistical methods with those resulting from neural networks theory, and presenting them as emerging from the same principle.
The competitor estimators to the least squares one are surveyed, this is also done for the different neural techniques conceived for the needs of regression and prediction. A comparative study was done with the aim to show that the fondamental concept, at the level of the roots, of these different methods can be seen as unique.
An application case is presented to illustrate the predictive power of neural nets versus the classical methods.

Keywords :

Linear regression ; Ridge regression ; Shrinkage estimators ; Generalized linear models ; Inference under constraints ; Neural nets and related approaches ; Learning and adaptive systems.

Résumé

Cette thèse est une tentative pour contribuer un tant soit peu à situer encore plus la théorie des réseaux de neurones artificiels dans le cadre de la statistique appliquée. La question centrale de l'inférence statistique y est étudiée sous l'éclairage de cette approche neuronale.
La notion de généralisation a été l'objet d'une grande attention avec le but de concevoir une approche unifiée englobant les méthodes de la statistique traditionnelle et celles qui résultent de la théorie des réseaux de neurones et présentées comme découlant d'un même et unique principe.
Les estimateurs concurrents à celui des moindres carrés ordinaires sont passé en revue, parallèlement aux diverses techniques neuronales conçues pour la régression et la prédiction. Une étude comparative a été entreprise afin de montrer que le concept fondamental à l'origine de toutes ces diverses méthodes peut être perçu comme unique.
Un cas d'application a été présenté pour illustrer le pouvoir prédictif des réseaux de neurones par rapport à celui des méthodes classiques.

Mots clés :

Régression linéaire ; Régession ridge ; Estimateurs concurrents ; Modèles linéaires généralisés ; Inférence sous contraintes ; Réseaux de neurones et approches connexes ; Apprentissage et systèmes adaptatifs.

Classification AMS 2010 : 62J05 ... 62J07 ... 62J12 ... 62M45 ... 68T05 ... 82C32 .