

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEURE
ET DE LA RECHERCHE SCIENTIFIQUE**



**Université hadj lakhdar Batna
Faculté de Technologie
Département d'électronique
*Mémoire***

Présenté en vue de l'obtention du diplôme de
Magister en Electronique
Option : **Micro-ondes pour Télécommunications**

Présenté par
Ould Ahmed
Val khyarkoum
Ingénieur d'Etat en Electronique

Thème

***Classification des régresseurs non linéaires
en utilisant un classificateur SVM***

Devant le jury

Dr. BENATIA Djamel
Dr. BENOUDJIT Nabil
Dr. ARAR Djemai
Dr.SGAIA Djamel

Prof. Univ. Batna
Prof. Univ. Batna
M.C.A Univ. Batna
M.C.A Univ. M'sila

President
Rapporteur
Examineur
Examineur

Année universitaire : 2011/2012

Dédicaces

A :

Mes chers parents

Mes frères et mes sœurs

Toute ma famille

Tous mes amis

Je dédie ce mémoire

Ould Ahmed Val khyarkoum

Remerciements

*Je voudrais remercier Mr. **Benoudjit Nabil** professeur à l'Université de Batna d'avoir accepté de diriger ce mémoire et qui, avec beaucoup de disponibilité, a suivi ce travail. Je le remercie vivement pour ses conseils toujours judicieux et pour ses hautes qualités scientifiques et pédagogiques.*

J'adresse ma profonde reconnaissance à Mr. Benatia Djamel professeur à l'Université de Batna de me faire l'honneur de présider le jury de soutenance.

Je tiens à associer à ces remerciements, Mr. Arar Djemai, Maître de conférences à l'Université de Batna et Mr. Saigaa Djamel, Maître de conférences à l'Université de M'sila d'avoir accepté de participer à ce jury et d'examiner ce travail.

Mes pensées iront également à mes parents qu'ils trouvent dans ce manuscrit toute ma profonde reconnaissance pour leurs aides et leurs encouragements.

Sans oublier les personnes qui m'ont partagé les années d'études en post de graduation dans le département d'électronique à l'université de Batna que je souhaite bon courage.

Ould Ahmed ual khyarkoum

Table de matière

<i>Table de matière</i>	4
<i>Liste des figures</i>	8
<i>Liste des tableaux</i>	11
<i>Introduction général</i>	13
<i>Chapitre1 : Interprétation des bases de données</i>	
<i>1-1. Introduction</i>	16
<i>1-2. Création des sous ensembles</i>	16
<i>a) Sous-ensemble (apprentissage)</i>	16
<i>b) Sous-ensemble (validation)</i>	16
<i>c) Sous-ensemble (test)</i>	17
<i>1-3. Les bases de données</i>	17
<i>1-3-1. base de données DIESEL</i>	17
<i>a) Sous-ensembles d'apprentissage(DIESEL)</i>	18
<i>b) Sous-ensembles de validation(DIESEL)</i>	18
<i>c) Sous-ensemble de test(DIESEL)</i>	19
<i>1-3-2. La base de données JUS</i>	19
<i>a) Sous-ensembles d'apprentissage(JUS)</i>	20
<i>b) Sous-ensembles de validation(JUS)</i>	20
<i>c) Sous-ensemble de test(JUS)</i>	21
<i>1-3-3. La base de données VIANDE</i>	21
<i>a) Sous-ensembles d'apprentissage(VIANDE)</i>	22
<i>b) Sous-ensembles de validation(VIANDE)</i>	22
<i>c) Sous-ensemble de test(VIANDE)</i>	23

1-3-2. <i>La base de données WINE</i>	23
a) <i>Sous-ensembles d'apprentissage(WINE)</i>	24
b) <i>Sous-ensembles de validation(WINE)</i>	24
c) <i>Sous-ensemble de test(WINE)</i>	25
1-4. <i>Conclusion</i>	25
Chapitre 2 : Régressions non linéaires à base de réseaux de neurones	
2-1. <i>Introduction</i>	27
2-2. <i>Régressions à base du réseau RBF</i>	27
2-2-1. <i>Modélisation mathématique du réseau RBF</i>	27
2-2-2. <i>Représentation connexionniste d'un réseau RBF</i>	28
2-2-3. <i>Apprentissage des modèles RBF</i>	29
2-2-4. <i>Calcul des poids</i>	29
2-2-5. <i>Apprentissage par Descente de gradient</i>	30
2-2-6. <i>Erreur quadratique moyenne normalisée (fonction coût)</i> ..	31
2-2-7. <i>Résultats sur la base DIESEL du modèle RBF</i>	32
2-2-8. <i>Résultats sur la base JUS du modèle RBF</i>	34
2-2-9. <i>Résultats sur la base VIANDE du modèle RBF</i>	36
2-2-10. <i>Résultats sur la base WINE du modèle RBF</i>	39
2-2-11. <i>Résultats globaux du modèle RBF par base de données</i> ...	41
2-3. <i>Régressions à base du réseau MLP</i>	42
2-3-1. <i>Le neurone formel</i>	43
2-3-2. <i>Représentation connexionniste d'un réseau MLP</i>	43

2-3-3. Définition du réseau MLP	44
2-3-4. Initialisation des poids des connexions	44
2-3-5. Propagation des entrées.....	44
2-3-6. retropropagation de l'erreur.....	44
2-3-7. Correction des poids des connexions.....	45
2-3-8. problèmes du dimensionnement.....	45
2-3-9. problèmes du sur-apprentissage.....	46
2-3-10. problèmes des minima locaux.....	46
2-3-11. Résultats sur la base DIESEL du modèle MLP.....	46
2-3-12. Résultats sur la base JUS du modèle MLP.....	48
2-3-13. Résultats sur la base VIANDE du modèle MLP.....	49
2-3-14. Résultats sur la base WINE du model MLP.....	51
2-3-15. Résultats globaux du modèle MLP par base de données...52	
2-4. Conclusion.....	53

Chapitre 3 : Régressions non linéaires à base de noyau (SVM)

3-1. Introduction.....	54
3-2. Formalisation.....	54
3-3. Notion de marge.....	55
3-3-1. Quantification de la marge.....	56
3-3-2. Quelques noyaux utilisables.....	56
3-4. Les régresseurs SVM.....	57

<i>3-5. Résultats sur la base DIESEL du modèle SVM (regresseur).....</i>	<i>58</i>
<i>3-6. Résultats sur la base JUS du modèle SVM (regresseur).....</i>	<i>60</i>
<i>3-7. Résultats sur la base VIANDE du modèle SVM (regresseur)..</i>	<i>62</i>
<i>3-8. Résultats sur la base WINE du modèle SVM (regresseur).....</i>	<i>64</i>
<i>3-9. Résultats globaux du modèle MLP par base de données.....</i>	<i>66</i>
<i>3-10. Conclusion.....</i>	<i>67</i>

Chapitre 4 : Classification des régresseurs en utilisant un SVM

<i>4-1. Introduction.....</i>	<i>69</i>
<i>4-2. schéma bloc de la Méthode de classification.....</i>	<i>69</i>
<i>4-3. Résultats de classification (base DIESEL).....</i>	<i>71</i>
<i>4-4. Résultats de classification (base JUS).....</i>	<i>72</i>
<i>4-5. Résultats de classification (base VIANDE).....</i>	<i>73</i>
<i>4-6. Résultats de classification (base WINE).....</i>	<i>73</i>
<i>4-7. Conclusion</i>	<i>74</i>

<i>Conclusion générale et perspectives.....</i>	<i>76</i>
---	-----------

<i>BIBLIOGRAPHIE.....</i>	<i>77</i>
---------------------------	-----------

<i>Résumé</i>	<i>81</i>
---------------------	-----------

<i>Abstract.....</i>	<i>82</i>
----------------------	-----------

Liste des figures

<i>Figure 1.1 La base de données DIESEL (sous-ensemble d'apprentissage).....</i>	<i>18</i>
<i>Figure 1.2 La base de données DIESEL (sous-ensemble de validation).....</i>	<i>18</i>
<i>Figure 1.3 La base de données DIESEL (sous-ensemble de test).....</i>	<i>19</i>
<i>Figure 1.4 La base de données JUS (sous-ensemble d'apprentissage).....</i>	<i>20</i>
<i>Figure 1.5 La base de données JUS (sous-ensemble de validation).....</i>	<i>20</i>
<i>Figure 1.6 La base de données JUS (sous-ensemble de test).....</i>	<i>21</i>
<i>Figure 1.7 La base de données VIANDE (sous-ensemble d'apprentissage).....</i>	<i>22</i>
<i>Figure 1.8 La base de données VIANDE (sous-ensemble de validation).....</i>	<i>22</i>
<i>Figure 1.9 La base de données VIANDE (sous-ensemble de test).....</i>	<i>23</i>
<i>Figure 1.10 La base de données WINE (sous-ensemble d'apprentissage).....</i>	<i>24</i>
<i>Figure 1.11 La base de données WINE (sous-ensemble de validation).....</i>	<i>24</i>
<i>Figure 1.12 La base de données WINE (sous-ensemble de test).....</i>	<i>25</i>
<i>Figure 2.1 Représentation connexionniste d'un réseau RBF.....</i>	<i>28</i>
<i>Figure 2.2 graphes de la sortie désirée et la sortie estimée base de DIESEL (test).....</i>	<i>32</i>
<i>Figure 2.3 graphe NMSE de validation en fonction du paramètre du modèle (DIESEL)...</i>	<i>33</i>

<i>Figure 2.4</i>	<i>graphe NMSE de validation en fonction du nombre de neurone (DIESEL)</i>	<i>.....33</i>
<i>Figure 2.5</i>	<i>graphe NMSE de validation du paramètre (sigma) (DIESEL)</i>	<i>.....34</i>
<i>Figure 2.6</i>	<i>graphes de la sortie désirée et la sortie estimée base de JUS (test)</i>	<i>.....34</i>
<i>Figure 2.7</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle (JUS)</i>	<i>.....35</i>
<i>Figure 2.8</i>	<i>graphe NMSE de validation en fonction du nombre de neurone (JUS)</i>	<i>.....35</i>
<i>Figure 2.9</i>	<i>graphe NMSE de validation en fonction du paramètre (sigma) (JUS)</i>	<i>.....36</i>
<i>Figure 2.10</i>	<i>graphes de la sortie désirée et la sortie estimée base VIANDE (test)</i>	<i>.....37</i>
<i>Figure 2.11</i>	<i>graphe NMSE de validation du paramètre du modèle (VIANDE)</i>	<i>.....37</i>
<i>Figure 2.12</i>	<i>graphe NMSE de validation en fonction du nombre de neurone(VIANDE)</i>	<i>..38</i>
<i>Figure 2.13</i>	<i>graphe NMSE de validation du paramètre (sigma) (VIANDE)</i>	<i>.....38</i>
<i>Figure 2.14</i>	<i>graphes de la sortie désirée et la sortie estimée base de WINE (test)</i>	<i>.....39</i>
<i>Figure 2.15</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle (WINE)</i>	<i>..40</i>
<i>Figure 2.16</i>	<i>graphe de NMSE de validation en fonction du (nombre de neurone) (WINE)</i>	<i>.40</i>
<i>Figure 2.17</i>	<i>graphe NMSE de validation en fonction du paramètre (sigma) (WINE)</i>	<i>.....41</i>
<i>Figure 2.18</i>	<i>Neurone Formel</i>	<i>.....42</i>
<i>Figure 2.19</i>	<i>Représentation connexionniste d'un réseau MLP</i>	<i>.....43</i>
<i>Figure 2.20</i>	<i>graphes de la sortie désirée et la sortie estimée base de DIESEL (test)</i>	<i>.....47</i>
<i>Figure 2.21</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle (DIESEL)</i>	<i>.47</i>
<i>Figure 2.22</i>	<i>graphe NMSE de validation en fonction du nombre de neurones (DIESEL)</i>	<i>.48</i>
<i>Figure 2.23</i>	<i>graphes de la sortie désirée et la sortie estimée base JUS (test)</i>	<i>..... 48</i>
<i>Figure 2.24</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle (JUS)</i>	<i>...49</i>
<i>Figure 2.25</i>	<i>graphe NMSE de validation en fonction du nombre de neurones (JUS)</i>	<i>.....49</i>
<i>Figure 2.26</i>	<i>graphe de la sortie désirée et la sortie estimée base de VIANDE (test)</i>	<i>..... 50</i>
<i>Figure 2.27</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle VIANDE</i>	<i>..50</i>
<i>Figure 2.28</i>	<i>graphes de la sortie désirée et la sortie estimée base de WINE (test)</i>	<i>.....51</i>

<i>Figure 2.29</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle (WINE)..</i>	<i>51</i>
<i>Figure 3.1</i>	<i>classifieur linéaire.....</i>	<i>55</i>
<i>Figure 3.2</i>	<i>choix d'un classifieur.....</i>	<i>55</i>
<i>Figure 3.3</i>	<i>distance à l'hyperplan séparateur.....</i>	<i>56</i>
<i>Figure 3.4.a</i>	<i>fonction quadratique.....</i>	<i>57</i>
<i>Figure 3.4.b</i>	<i>fonction de Laplace.....</i>	<i>57</i>
<i>Figure 3.4.c</i>	<i>fonction de Huber.....</i>	<i>57</i>
<i>Figure 3.4.d</i>	<i>fonction peu sensible.....</i>	<i>57</i>
<i>Figure 3.5</i>	<i>graphes de la sortie désirée et la sortie estimée base de DIESEL (test).....</i>	<i>58</i>
<i>Figure 3.6</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle (DIESEL)..</i>	<i>59</i>
<i>Figure 3.7</i>	<i>graphe NMSE de validation en fonction du paramètre (DIESEL).....</i>	<i>59</i>
<i>Figure 3.8</i>	<i>graphe NMSE de validation du paramètre (sigma) (DIESEL).....</i>	<i>60</i>
<i>Figure 3.9</i>	<i>graphes de la sortie désirée et la sortie estimée base de JUS (test).....</i>	<i>60</i>
<i>Figure 3.10</i>	<i>graphe NMSE de validation en fonction du paramètre du modèle (JUS).....</i>	<i>61</i>
<i>Figure 3.11</i>	<i>graphe NMSE de validation en fonction du paramètre (C)(JUS).....</i>	<i>61</i>
<i>Figure 3.12</i>	<i>graphe NMSE de validation en fonction du paramètre (sigma) (JUS).....</i>	<i>62</i>
<i>Figure 3.13</i>	<i>graphes de la sortie désirée et la sortie estimée base de VIANDE (test).....</i>	<i>62</i>
<i>Figure 3.14</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle VIANDE..</i>	<i>63</i>
<i>Figure 3.15</i>	<i>graphe NMSE de validation en fonction du paramètre (C) (VIANDE).....</i>	<i>63</i>
<i>Figure 3.16</i>	<i>graphe NMSE de validation en fonction du paramètre (sigma) (VIANDE).....</i>	<i>64</i>
<i>Figure 3.17</i>	<i>graphes de la sortie désirée et la sortie estimée base de WINE (test).....</i>	<i>65</i>
<i>Figure 3.18</i>	<i>graphe NMSE de validation en fonction des paramètres du modèle (WINE)....</i>	<i>65</i>
<i>Figure 3.19</i>	<i>graphe NMSE de validation en fonction du paramètre (C) (WINE).....</i>	<i>66</i>
<i>Figure 3.20</i>	<i>graphe NMSE de validation du paramètre (sigma) (WINE).....</i>	<i>66</i>
<i>Fig 4</i>	<i>Schéma bloc de la méthode de classification.....</i>	<i>69</i>

Liste des tableaux

<i>Tableau 2.1 NMSE (test) du modèle RBF pour les quatre bases de données.....</i>	<i>41</i>
<i>Tableau 2.2 NMSE (test) du modèle MLP pour les quatre bases de données</i>	<i>52</i>
<i>Tableau 3 NMSE (test) du modèle SVM pour les quatre bases de données</i>	<i>67</i>
<i>Tableau 4.1 (résultat de classification)DIESEL pour les quatre bases de données</i>	<i>72</i>
<i>Tableau 4.2 (résultat de classification)JUS pour les quatre bases de données</i>	<i>72</i>
<i>Tableau 4.3 (résultat de classification)VIANDE pour les quatre bases de données</i>	<i>73</i>
<i>Tableau 4.4 (résultat de classification)WINE pour les quatre bases de données</i>	<i>73</i>

Introduction générale

Introduction générale

L'amélioration des résultats d'un modèle est un objectif constant de toute recherche scientifique objective, les réseaux de neurones et les machines à vecteur de support sont des modèles très utilisés dans les opérations de régression et de classification.

La régression non linéaire peut se faire avec des modèles à base de réseaux de neurones ou avec des modèles à base des noyaux, on utilise dans ce travail les modèles les plus connus : le réseau à fonctions radiales de base (Radial Basis Functions) RBF, le perceptron multi couches (Multi-Layer Perceptron) MLP et les machines à vecteurs de support (Support Vector Machine) SVM.

Après avoir établi les résultats des trois modèles utilisés dans la régression (RBF, MLP, SVM) sur une base de données, un problème fondamental se pose; il s'agit du choix du meilleur régresseur parmi les régresseurs utilisés.

Les machines à vecteurs de support (SVM) peuvent être utilisées comme des régresseurs mais elles peuvent aussi être utilisées comme des classificateurs.

Dans ce mémoire les SVMs sont utilisées comme des régresseurs avec des modèles à base des Réseaux de neurones (RBF, MLP), en plus, elles sont aussi utilisées comme des classificateurs.

La méthode consiste à faire apprendre à un classificateur SVM par les matrices d'entrée d'une base de données, la sortie représente la meilleure réponse parmi trois régresseurs (RBF, MLP, SVM), exprimée sous forme de classes afin de fixer les paramètres du modèle classificateur.

Une fois le classificateur est validé, il devient capable de faire la classification sur un champ de données qui n'est pas précédemment utilisée ni pour l'apprentissage ni pour la validation.

La classification ici est l'action d'associer à chaque échantillon de la base de données utilisée un régresseur parmi les modèles mis en jeu (RBF, MLP, SVM).

Alors on peut résoudre le problème du choix du régresseur optimal par l'utilisation d'un classificateur SVM qui permet de sélectionner la meilleure régression pour chaque échantillon de la base de données utilisée.

Les bases de données utilisées pour la validation des modèles dans ce travail sont des bases de données réelles [1], [2].

Ce mémoire est constitué de quatre chapitres. Dans le premier chapitre on interprète les quatre bases de données utilisées pour la validation des modèles et des méthodes mise en jeu dans ce travail.

Le deuxième chapitre est consacré à la régression non linéaire à base des réseaux de neurones (RBF, MLP), il comporte l'étude théorique et les résultats de validation sur les différentes bases de données utilisées.

Le troisième chapitre s'intéresse à l'étude théorique et la validation par les différentes bases de données de la régression non linéaire à base de noyau SVM.

Le quatrième chapitre de ce travail concerne par la classification des régresseurs étudiés dans les chapitres précédents par un classificateur SVM il détaille la méthode de classification et donne les résultats de validation.

Le travail est achevé par une conclusion générale qui donne une idée sur le degré de la réalisation de ses objectifs et les perspectives permettant la continuité de ce travail dans le but d'améliorer en plus les résultats de la recherche.

Chapitre 1

Interprétation des bases de données

1-1. Introduction

Dans ce chapitre on interprète les quatre bases de données utilisées, et on montre l'ensemble des traitements et les subdivisions qu'elles doivent subir pour répondre aux exigences des modèles à établir.

Les bases de données utilisées dans ce travail (DIESEL, JUS, VIANDE, WINE) sont des bases réelles utilisées pour la validation de plusieurs travaux et publications dans le domaine de la chimie « chemometrics ».

1-2 Création des sous ensembles

Pour répondre à des contraintes de la méthode d'établissement des modèles des réseaux de neurones et des machines à vecteurs de support chaque base de données utilisée doit subir un prétraitement de données qui consiste à diviser l'ensemble des échantillons en trois sous ensembles (apprentissage, validation, test).

a) Sous-ensemble (apprentissage)

Le sous-ensemble d'apprentissage est utilisé pour établir un modèle. C'est-à-dire pour fixer les paramètres du modèle et déterminer les poids d'un réseau.

Une fois le modèle est établi avec des données d'apprentissage, nous avons besoin de savoir la réponse du modèle avec des données inconnues, le modèle doit

être employé avec un autre sous-ensemble non utilisé dans le processus d'apprentissage.

b) Sous-ensemble (validation)

Nous pouvons essayer des modèles de réseau de neurones avec de diverses architectures (par exemple différents nombres des neurones dans la couche cachée d'un RBFN) puis choisir l'architecture qui donne la meilleure réponse avec le sous-ensemble de validation.

Le modèle final est choisi en tant que gagnant parmi les modèles en concurrence de fait que son exactitude sur le sous-ensemble de validation est la plus haute.

c)Sous-ensemble (test)

Quand un modèle est finalement choisi, sa réponse avec l'ensemble de validation est toujours une évaluation optimiste de la façon dont il s'exécuterait avec des données inconnues.

1-3. Les bases de données

La première base de données utilisée dans ce travail est la base DIESEL elle représente la concentration de cétone (élément chimique) dans des échantillons du diesel (carburant).

La deuxième base de données est la base JUS d'orange qui représente la concentration du sucre (saccharose) dans des échantillons du jus d'orange.

La troisième base de données est dite la base VIANDE, elle consiste à la mesure de la concentration de la graisse dans des échantillons de la viande.

La dernière base de données utilisée est la base WINE elle représente la concentration de l'alcool dans des échantillons de vin.

1-3-1 base de données DIESEL

La base de données DIESEL représente la concentration de cétone dans des échantillons du diesel analysés par la transmittance de l'infrarouge en spectroscopie.

Les sous ensembles d'apprentissage de validation et test contiennent respectivement 20 échantillons, 113 échantillons et 112 échantillons.

La figure (1.1) montre la collection des 20 échantillons utilisés pour l'apprentissage, et la figure (1.2) illustre les 113 échantillons utilisés pour la validation, par contre la figure (1.3) montre les 112 échantillons utilisés pour le test.

a) Sous-ensembles d'apprentissage (DIESEL)

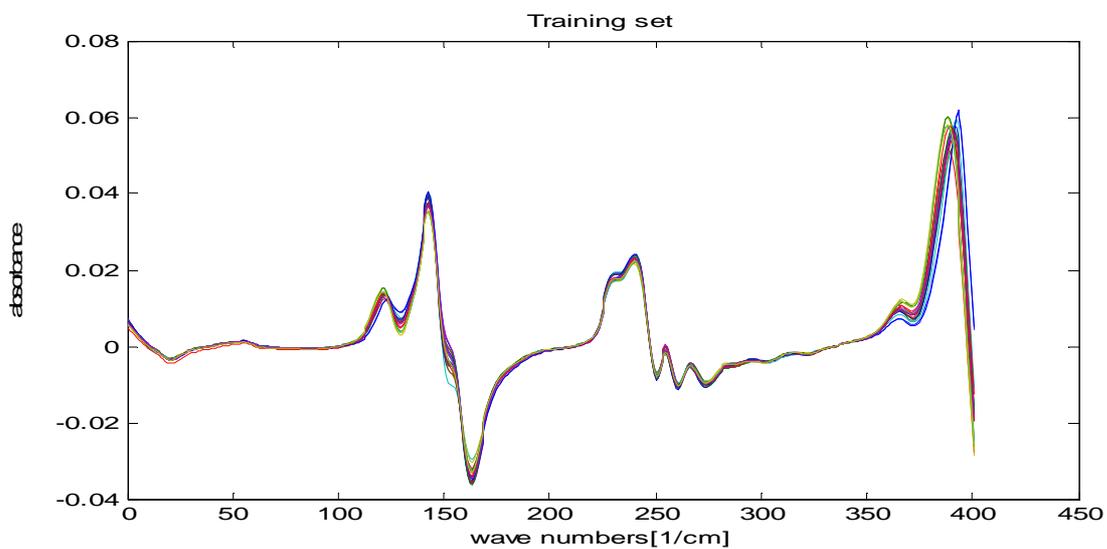


Figure 1.1 La base de données DIESEL (sous -ensemble d'apprentissage)

b) Sous-ensembles de validation (DIESEL)

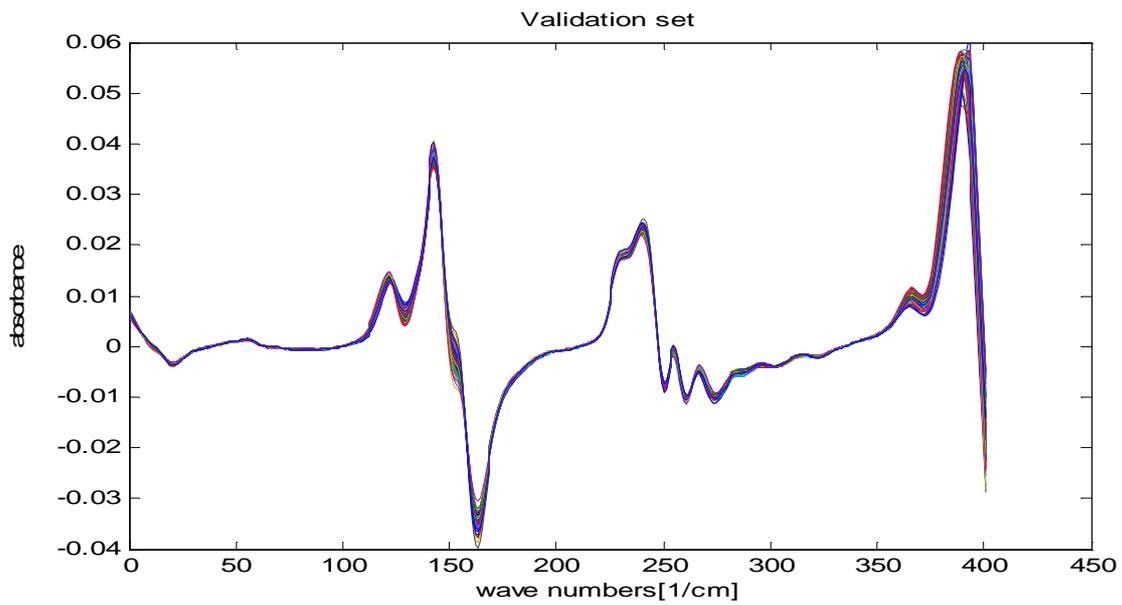


Figure 1.2 La base de données DIESEL (sous-ensemble de validation)

c) Sous-ensembles de test (DIESEL)

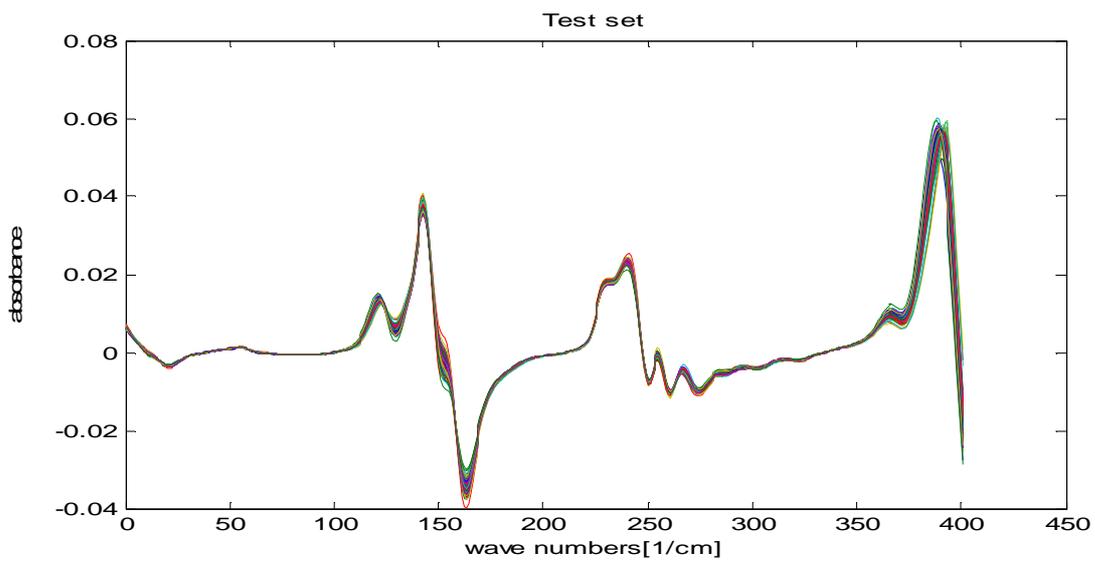


Figure 1.3 La base de données DIESEL (sous-ensemble de test)

1-3-2. La base de données JUS

La deuxième base de données se rapporte à la détermination de la concentration en sucre (saccharose) dans des échantillons de jus d'orange mesurés par la réflectivité de l'infrarouge en spectrométrie.

Chaque spectre contient 700 variables spectrales qui sont l'absorbance ($\log(1/R)$) à 700 longueurs d'onde entre 1100 nanomètres et 2500 nanomètres (où R est la réflectivité par échantillon de surface).

Les sous ensembles d'apprentissage de validation et test contiennent respectivement 100 échantillons, 50 échantillons et 68 échantillons.

La figure (1.4) montre la collection des 100 échantillons utilisés pour l'apprentissage, et sur la figure (1.5) est représenté les 50 échantillons utilisés pour la validation, par contre la figure (1.6) montre les 68 échantillons utilisés pour la phase de test.

a)Sous-ensembles d'Apprentissage (JUS)

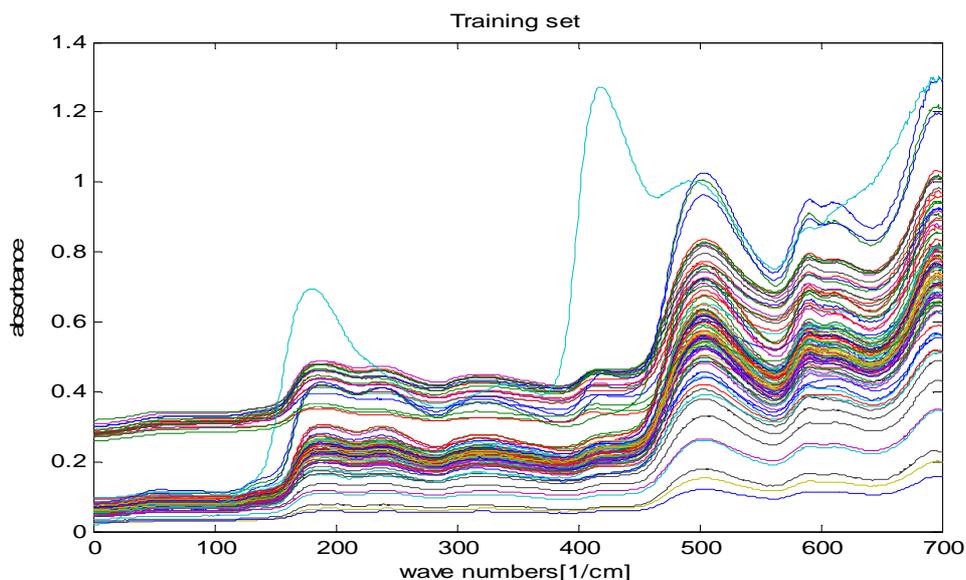


Figure 1.4 La base de données JUS (sous-ensemble d'apprentissage)

b) Sous-ensembles de validation (JUS)

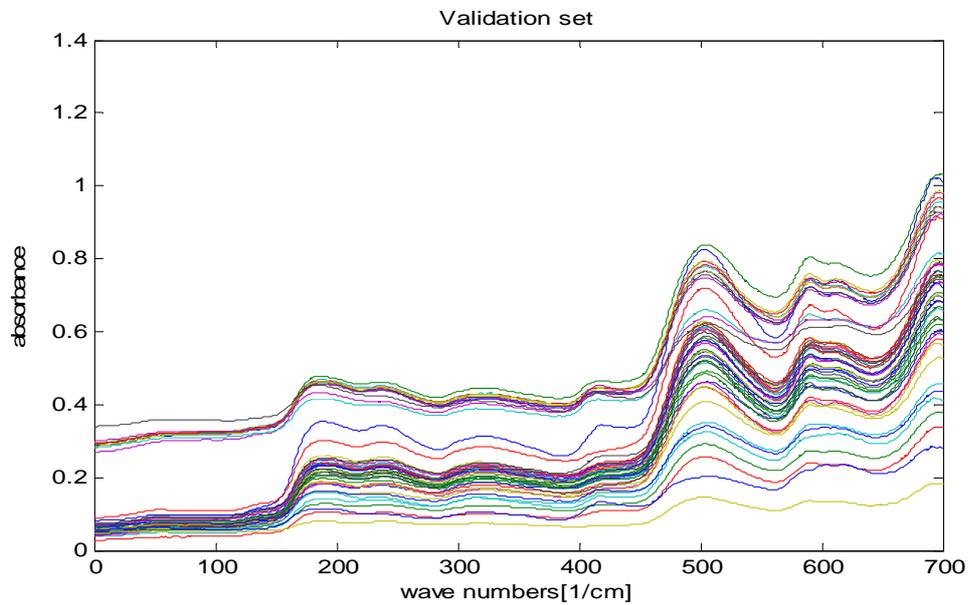


Figure 1.5 La base de données JUS (sous-ensemble de validation)

c) sous-ensembles de test (JUS)

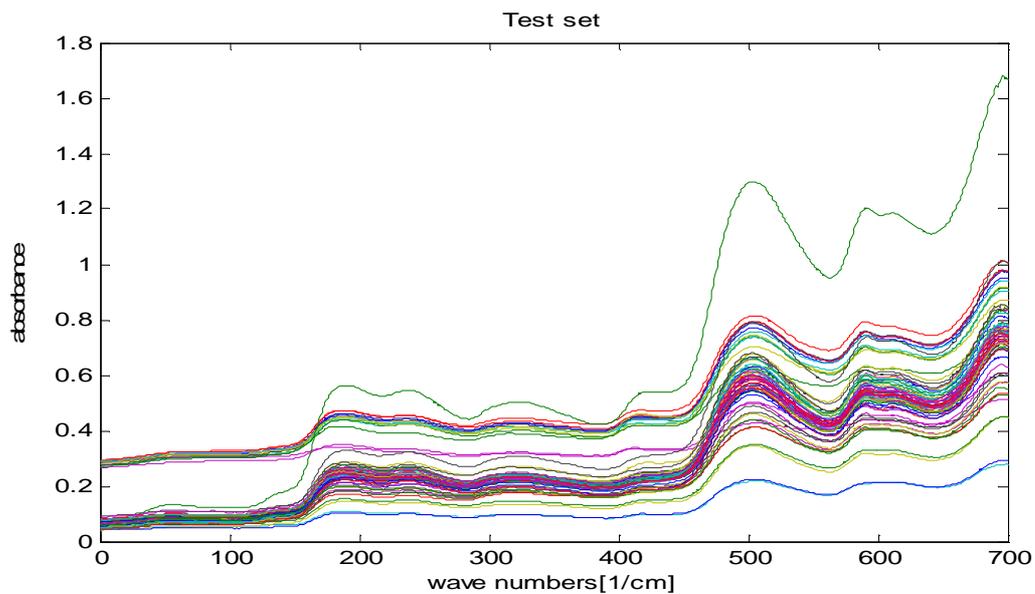


Figure 1.6 La base de données JUS (sous-ensemble de test)

1-3-3. La base de données VIANDE

La troisième base de données se rapporte à la détermination de la teneur en graisse des échantillons de viande analysés par la transmittance de l'infrarouge en spectroscopie.

Les sous ensembles d'apprentissage de validation et test contiennent respectivement 114 échantillons, 58 échantillons et 43 échantillons.

La figure (1.7) montre la collection des 114 échantillons utilisés pour l'apprentissage, et sur la figure (1.8) sont représentées les 58 échantillons utilisés pour la validation, par contre, la figure (1.9) montre les 43 échantillons utilisés pour la phase de test.

a)Sous-ensembles d'Apprentissage (VIANDE)

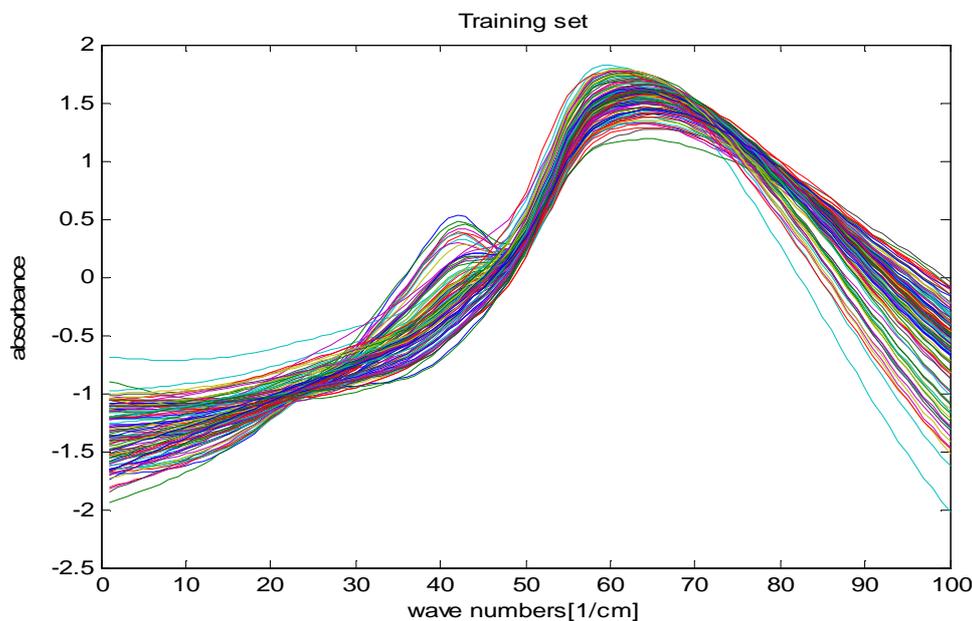


Figure 1.7 La base de données VIANDE (sous-ensemble d'apprentissage)

b) Sous-ensembles de validation (VIANDE)

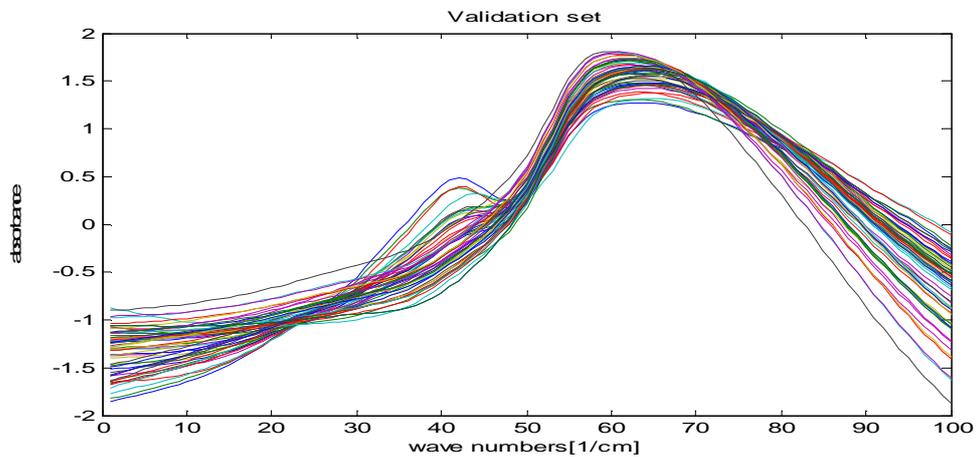


Figure 1.8 La base de données VIANDE (sous-ensemble de validation)

c) Sous-ensembles de test (VIANDE)

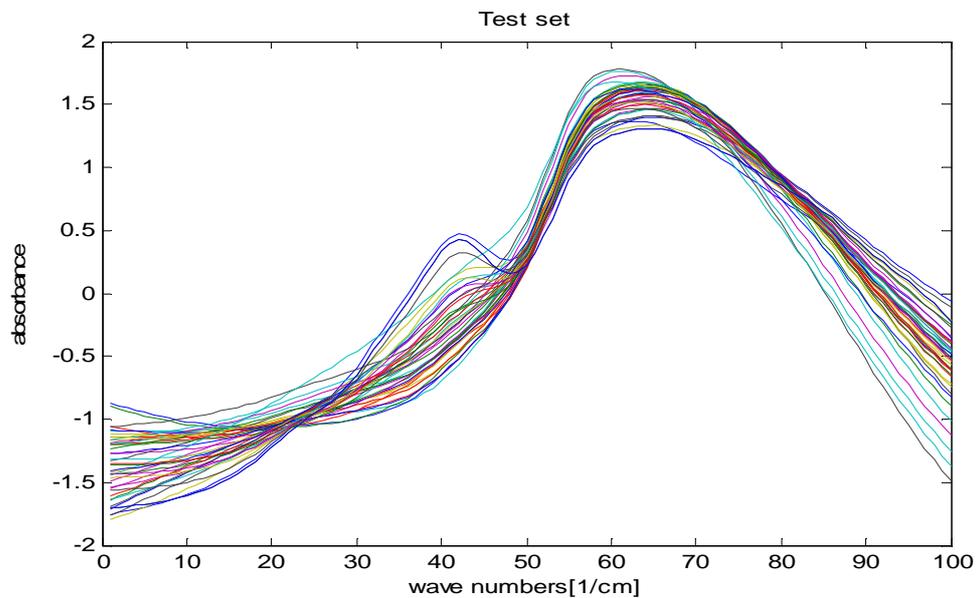


Figure 1.9 La base de données VIANDE (sous-ensemble de test)

1-3-4. La base de données WINE

La dernière base de données se rapporte à la détermination de la concentration d'alcool par l'infrarouge en spectroscopie dans des échantillons de vin, reliant 256 variables spectrales qui sont l'absorbance ($\log \frac{1}{T}$) avec 256 longueur d'onde entre 4000 et 400 cm^{-1} (où T est la transmittance par d'échantillon).

Les sous ensembles d'apprentissage de validation et de test contiennent respectivement 60 échantillons, 34 échantillons et 30 échantillons.

La figure (1.10) montre la collection des 60 échantillons utilisés pour l'apprentissage, et sur la figure (1.11) sont montrés les 34 échantillons utilisés pour la validation, par contre, la figure (1.12) montre les 30 échantillons utilisés pour le test.

a) Sous-ensembles d'Apprentissage (wine)

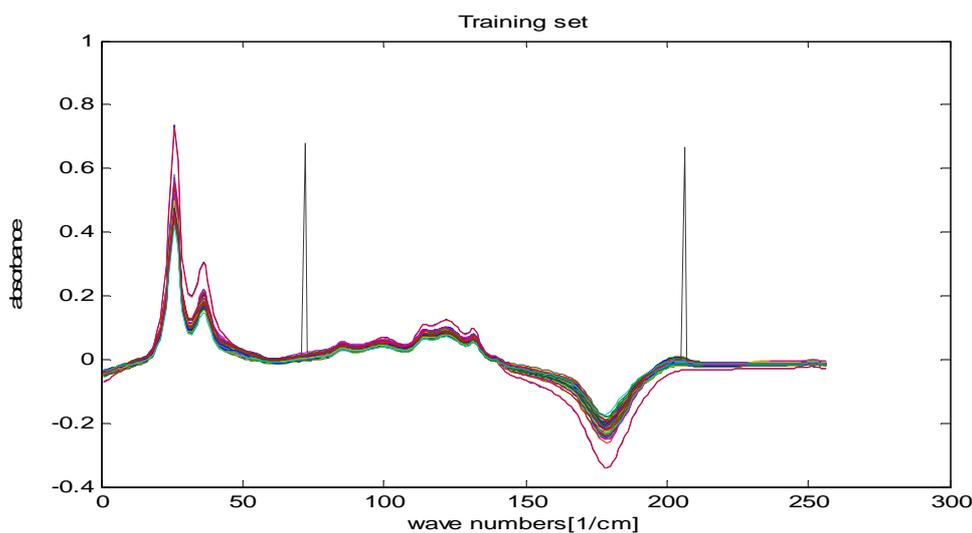


Figure 1.10 La base de données wine (sous-ensemble d'apprentissage)

b) Sous-ensembles de validation (wine)

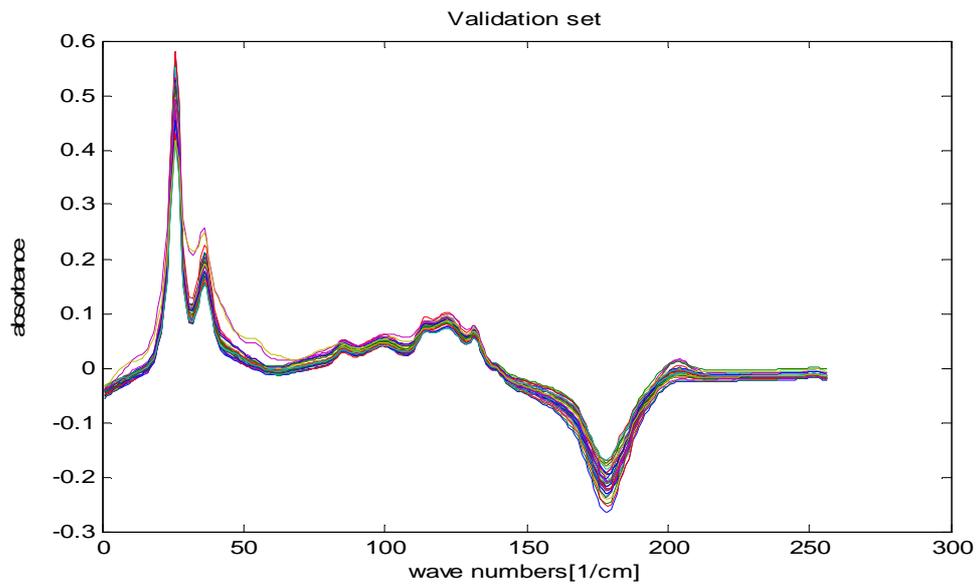


Figure 1.11 La base de données wine (sous-ensemble de validation)

c) Sous-ensembles de test (wine)

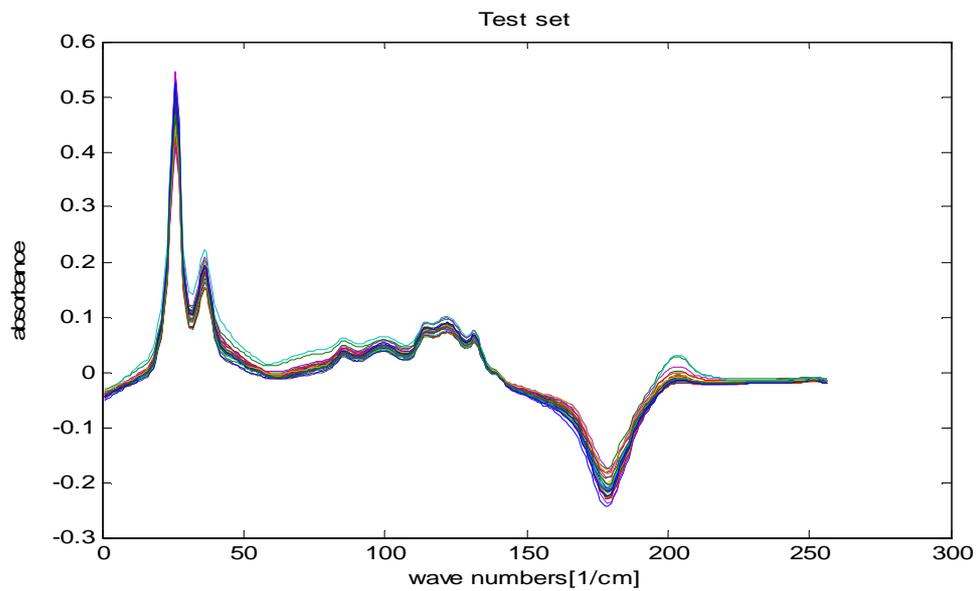


Figure 1.12 La base de données wine (sous-ensemble de test)

1-4. Conclusion

Dans ce chapitre nous avons interprété les bases de données utilisées pour la validation de ce projet de mémoire; les bases sont divisées en sous ensembles pour qu'elles puissent répondre aux exigences de la méthode.

Le travail avec des bases de données réelles établies par des sources avérées donne sans doute une fiabilité supplémentaire aux résultats de validation des méthodes mises en jeu dans le processus engagé.

Chapitre 2

Régressions non linéaires à base

de réseaux de neurones

2-1. Introduction

Dans ce chapitre on étudie deux types de réseaux de neurone RBF et MLP afin d'établir leurs résultats régressifs sur toutes les bases de données utilisées.

Un réseau de neurones est un outil d'analyse statistique permettant de construire un modèle de comportement à partir de données qui sont des exemples de ce comportement.

Les réseaux de neurones peuvent être représentés par une boîte noire à l'entrée de laquelle on présente un vecteur à « n » dimensions, représentant les données d'un problème, et à la sortie de laquelle on récupère un vecteur de « m » dimension qui représente la solution déterminée par le système.

2-2. Régressions à base du réseau RBF

Les réseaux à fonctions de base radiales (RBF) sont des modèles connexionnistes simples à mettre en œuvre et assez intelligibles, et ils sont très utilisés pour la régression, ils possèdent une fonction d'activation de type

gaussienne, leurs propriétés théoriques et pratiques ont été étudiées en détail depuis la fin des années 80; il s'agit certainement avec le Perceptron multicouche, du modèle connexionniste le mieux connu.

2-2-1. Modélisation mathématique du réseau RBF

Une fonction de base radiale (**RBF**) est une fonction ϕ symétrique autour d'un Centre μ_j :

$$\phi_j(x) = \phi (\|x - \mu_j\|) \quad (2. 1)$$

Où $\|.\|$ est une norme [03]. Par exemple, La Fonction gaussienne est un **RBF** avec la norme euclidienne.

$$\phi (r) = e^{-r^2/2\sigma^2} \quad (2. 2)$$

En général, les **RBF** sont paramétrées par σ qui correspond à la « largeur » de la fonction :

$$\phi_j(x) = \phi (\|x - \mu_j\|, \sigma_j) \quad (2. 3)$$

Un modèle ou *réseau* **RBF** calcule une combinaison linéaire de fonctions radiales de centres μ_j :

$$y(x) = \sum_{j=1}^N W_j \phi (\|x - \mu_j\|, \sigma_j) \quad (2. 4)$$

2-2-2. Représentation connexionniste d'un réseau RBF

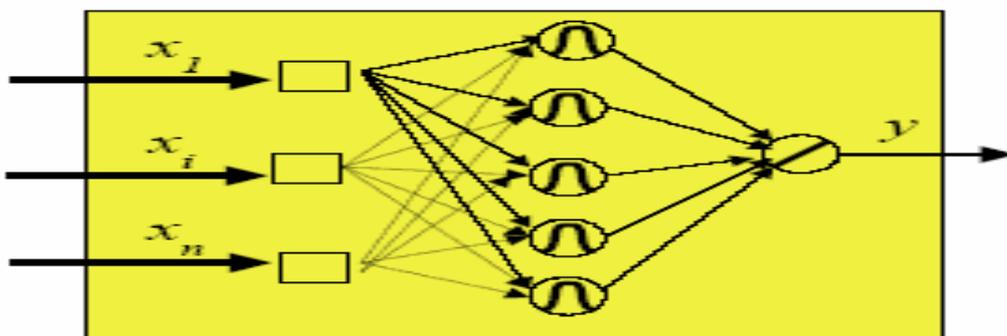


Figure 2.1 Représentation connexionniste d'un réseau RBF.

On distingue trois couches (**figure 2.1**) : entrée \mathbf{x} , fonctions radiales (couche cachée), sortie \mathbf{y} , et trois jeux de paramètres : les centres μ_j , les largeurs σ_j et les poids w_j .

Les combinaisons linéaires de gaussiennes sont utilisées depuis les années **60** pour construire des interpolations ou approximations de fonctions [04].

A la fin des années **80**, la présentation de ces modèles comme des réseaux connexionnistes a suscité un regain d'intérêt [05] [06] motivé en grande partie par la possibilité d'utiliser un algorithme d'apprentissage très rapide (sans recourir à des techniques d'optimisation non linéaire comme dans le cas du Perceptron multicouche), donnant en général des résultats voisins des meilleurs modèles connexionnistes.

L'apprentissage des modèles **RBF** est supervisé néanmoins il faut disposer d'un échantillon de l exemples $(x_i; y_i)$.

2-2-3. Apprentissage des modèles RBF

L'apprentissage d'un modèle **RBF** consiste à déterminer son architecture (le nombre N de fonctions radiales) et à fixer les valeurs des paramètres.

L'apprentissage d'un réseau **RBF** est de type supervisé, on dispose alors d'un ensemble d'apprentissage constitué de l couples (vecteur d'entrée, valeur cible) :

$$(x_1; y_1); \dots ; (x_m; y_m); \quad x_i \in R^d, y_i \in R \quad (2. 5)$$

Et de la fonction coût associé à chaque exemple :

$$E_i = \frac{1}{2} (y_i - F(x_i))^2 \quad (2. 6)$$

Une caractéristique intéressante des modèles **RBF** est que l'on peut diviser les Paramètres en trois groupes : les centres μ , les largeurs σ et les poids w .

L'interprétation de chaque groupe permet de proposer un algorithme d'apprentissage séquentiel, simple et performant [06].

2-2-4. Calcul des poids

Si l'on suppose les centres et largeurs connus, les poids w optimaux se calculent aisément :

$$y(x) = \sum_{j=1}^N W_j \phi(\|x - \mu_j\|, \sigma_j) = \sum_{j=1}^N W_j h_j(x) \quad (2. 7)$$

On cherche la solution w qui minimise la différence e entre la sortie estimée et la sortie désirée. On a donc un système d'équations linéaires qui s'écrit :

$$y = HW + e \quad (2. 8)$$

La matrice H , de taille $l \times N$, donne les réponses des N centres **RBF** sur les l exemples, y est un vecteur regroupant les l sorties y_i sur l'ensemble d'apprentissage, et e est le vecteur d'erreur. Le critère à optimiser est :

$$E = e^T e \quad (2. 9)$$

Si l'on ajoute un terme de régularisation de type *ridge regression* [07], qui pénalise les solutions avec de grandes valeurs des poids, on écrit :

$$E = e^T e + \alpha W^T w \quad (2. 10)$$

La solution s'obtient par un calcul classique de pseudo-inverse, et s'écrit :

$$\mathbf{w} = (\mathbf{H}^T \mathbf{H} + \alpha \mathbf{I})^{-1} \mathbf{H}^T \mathbf{y} \quad (2. 11)$$

Où \mathbf{I} est la matrice identité de taille l .

2-2-5. Apprentissage par Descente de gradient

Une alternative à l'apprentissage séquentiel décrit dans la section précédente consiste à optimiser les paramètres du modèles RBF par descente de gradient, comme on le fait pour d'autres modèles connexionnistes. Il faut pour cela calculer les dérivées du coût (éventuellement régularisé) par rapport aux différents paramètres.

Pour une fonction gaussienne:

$$\phi_{ij} = \exp\left(-\frac{\|x_i - \mu_j\|^2}{2\sigma_j^2}\right) \quad (2. 12)$$

Et la fonction coût (erreur)

$$E_i = \frac{1}{2} \left(y_i - y(x_i) \right)^2 \quad (2. 13)$$

On peut mettre en œuvre un algorithme d'apprentissage standard de minimisation de l'erreur, en version *batch* (calcul de l'erreur sur l'ensemble des exemples avant mise à jour des paramètres) ou en ligne (mises à jour après chaque exemple, approche qui en général offre de meilleures performances).

Il s'agit cependant d'un problème non linéaire, et l'algorithme d'optimisation à de grandes chances de rester bloqué dans un minimum local de la fonction coût. La réussite de l'optimisation dépend donc beaucoup des conditions initiales.

Il est donc recommandé de n'utiliser l'optimisation globale des paramètres par descente de gradient qu'après un apprentissage séquentiel classique.

La descente de gradient permet alors d'effectuer un réglage fin des paramètres qui améliore les performances [07].

Notons que les solutions **RBF** obtenues avec un apprentissage par descente de gradient sont souvent assez différentes de celles obtenues par apprentissage séquentiel. En particulier, rien ne garantit plus la localité de chaque fonction de base (autrement dit, les largeurs peuvent prendre des valeurs élevées).

2-2-6. Erreur quadratique moyenne normalisée(fonction coût)

Comme mentionné dans le chapitre 1, selon chaque méthode employée, l'erreur de plusieurs modèles doit être évaluée par des données différentes de celles utilisées pour l'apprentissage, ceci est réalisé par l'utilisation d'un ensemble de validation V qui contient N_V échantillons.

$$V = \{(\mathbf{x}_q, \mathbf{y}_q) \in \mathbf{R}^d \times \mathbf{R}, 1 \leq q \leq N_V : \mathbf{y}_q = f(\mathbf{x}_q)\} \quad (2.14)$$

Le critère d'erreur peut être choisi comme l'erreur quadratique moyenne normalisée définie par :

$$NMSE_V = \frac{\frac{1}{N_V} \sum_{q=1}^{N_V} (\hat{\mathbf{y}}_q - \mathbf{y}_q)^2}{\frac{1}{N_V + N_T} \sum_{j=1}^{N_V + N_T} (\mathbf{y}_j - \bar{\mathbf{y}})^2} \quad (2.15)$$

Où N_V, N_T sont respectivement le nombre d'échantillons dans l'ensemble d'apprentissage et le nombre d'échantillons dans l'ensemble validation, $\hat{\mathbf{y}}_q$ est la valeur prévue par le modèle et \mathbf{y}_q la valeur actuelle correspondante au spectre \mathbf{q} .

Noter que l'équation (2.16) normalise les erreurs en ce qui concerne l'écart type des valeurs de \mathbf{y} dans les ensembles combinés d'apprentissage et de validation.

Ce critère d'erreur défini par l'erreur quadratique moyenne normalisée (Normalized Mean Square Error) NMSE est utilisé comme critère de précision pour tous les modèles employés dans ce mémoire.

2-2-7. Résultats sur la base DIESEL du model RBF

Sur la figure suivante nous illustrons la superposition de la sortie désirée et la sortie estimée par le régresseur **RBF** optimal (base DIESEL).

On peut remarquer que la sortie estimée poursuit d'une manière claire la sortie désirée.

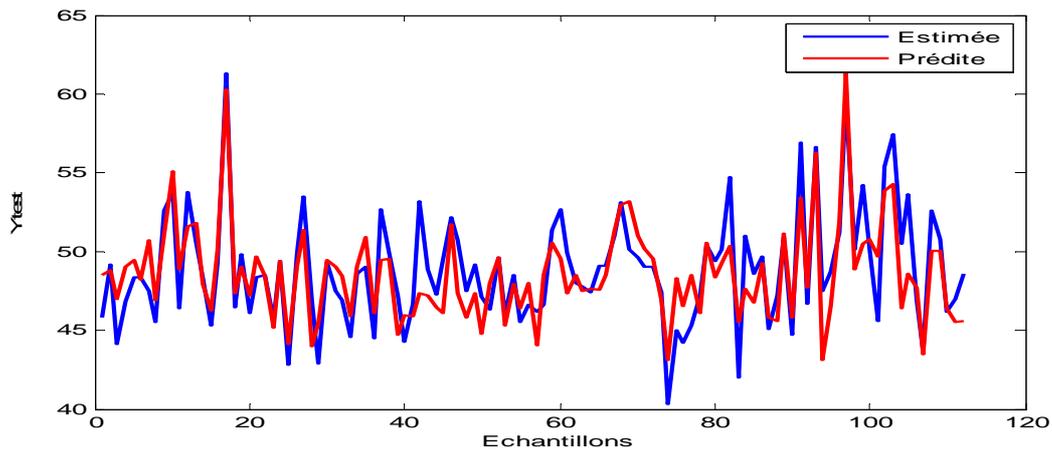


Figure 2.2 graphes de la sortie désirée et la sortie estimée base de DIESEL (test)

Le graphe de la figure 2.3 montre l'évolution de l'erreur de validation en fonction des paramètres du modèle (nombre de neurones, largeur de la fonction gaussienne) pour la base de DIESEL.

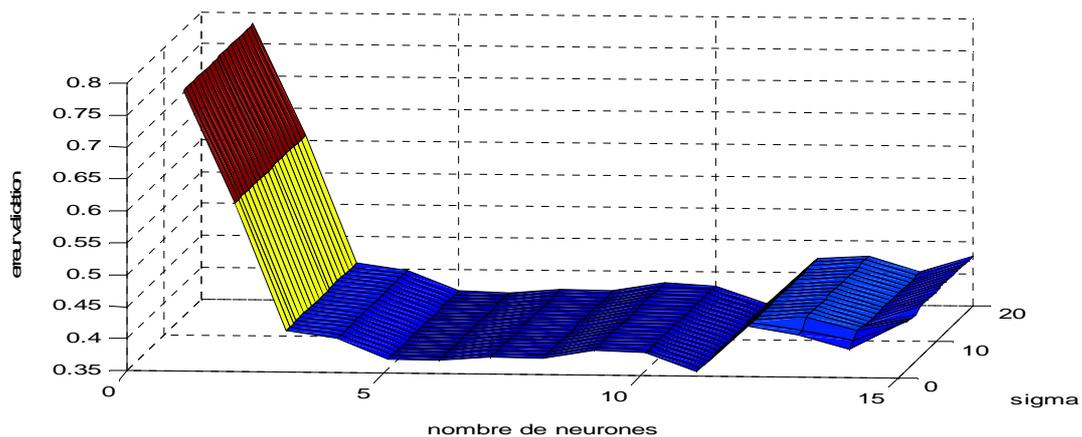


Figure 2.3 graphe NMSE de validation en fonction de (nombre des paramètres du modèle (DIESEL))

Sur la figure suivante nous montrons l'évolution de l'erreur de validation en fonction seulement du nombre de neurone (base DIESEL)

On remarque que la valeur minimale correspond à (nombre de neurone = 11)

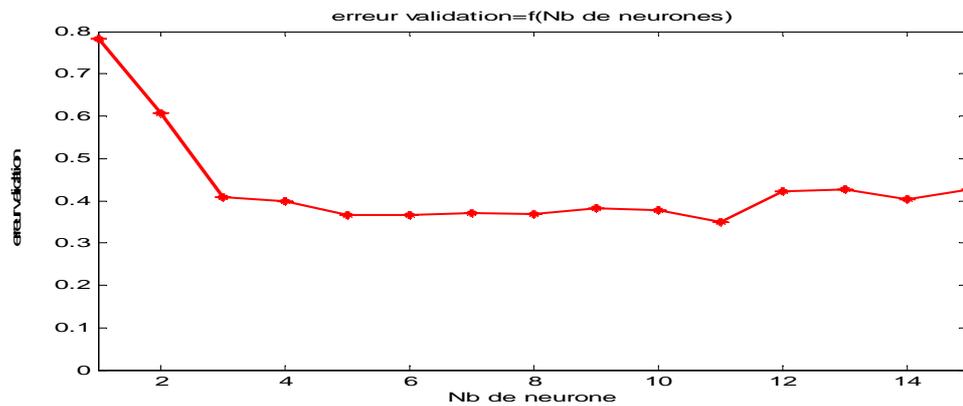


Figure 2.4 graphe NMSE de validation en fonction du nombre de neurones (DIESEL)

La figure suivante représente l'évolution de l'erreur de validation en fonction seulement de la largeur de la fonction gaussienne (base DIESEL).

L'erreur de validation a partir de la valeur $\sigma=6$ décroît afin d'atteindre une valeur minimale quasi constante.

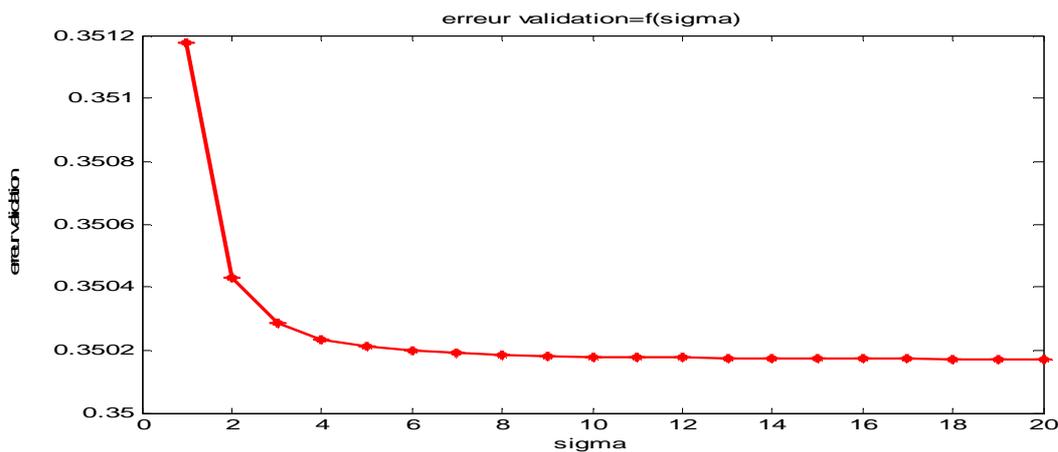


Figure 2.5 graphe NMSE de validation en fonction de (sigma) (DIESEL)

2-2-8. Résultats sur la base JUS du model RBF

La figure suivante illustre la superposition de la sortie désirée et la sortie estimée par le régresseur **RBF** optimal (base JUS).

La même remarque concernant la base DIESEL est aussi valable pour la base JUS.

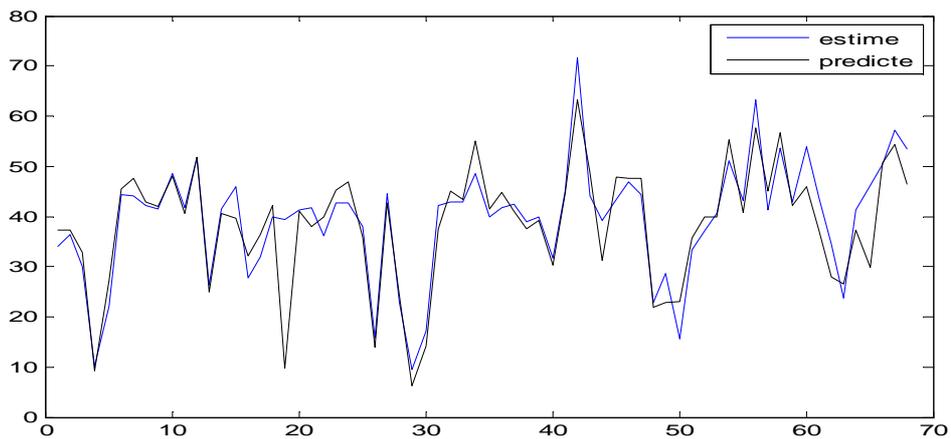


Figure 2.6 graphes de la sortie désirée et la sortie estimée base de JUS (test)

Sur la figure suivante est illustrée l'évolution de l'erreur de validation en fonction des paramètres du modèle (nombre de neurones, largeur de la fonction gaussienne) concernant la base de données JUS.

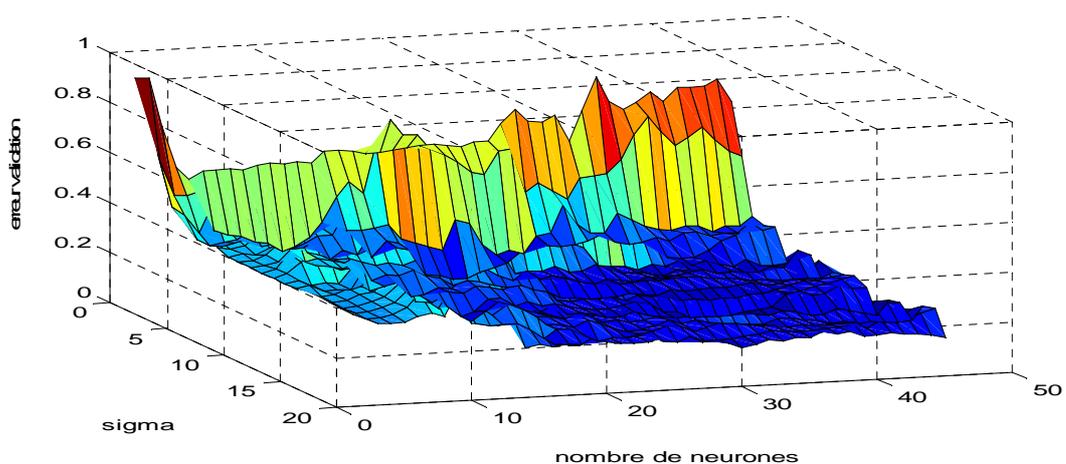


Figure 2.7 graphe NMSE de validation en fonction des paramètres du modèle (JUS)

Sur le graphe suivant nous enregistrons l'évolution de l'erreur de validation en fonction seulement du nombre de neurone (base JUS).

On peut remarquer que l'erreur de validation décroît globalement pour atteindre une valeur minimale correspondant à la valeur de 45 neurones dans la couche cachée.

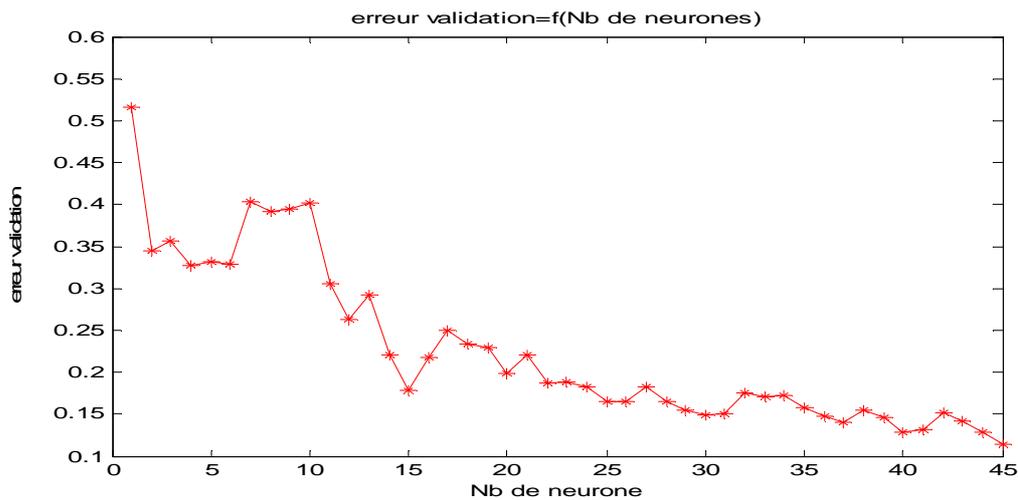


Figure 2.8 graphe NMSE de validation en fonction du nombre de neurones (JUS)

La figure suivante montre l'évolution de l'erreur de validation en fonction seulement de la largeur de la fonction gaussienne (base JUS).

On remarque dans ce cas que l'erreur de validation atteint rapidement l'ordre de valeur minimale qui correspond à la valeur de la largeur sigma = 14.

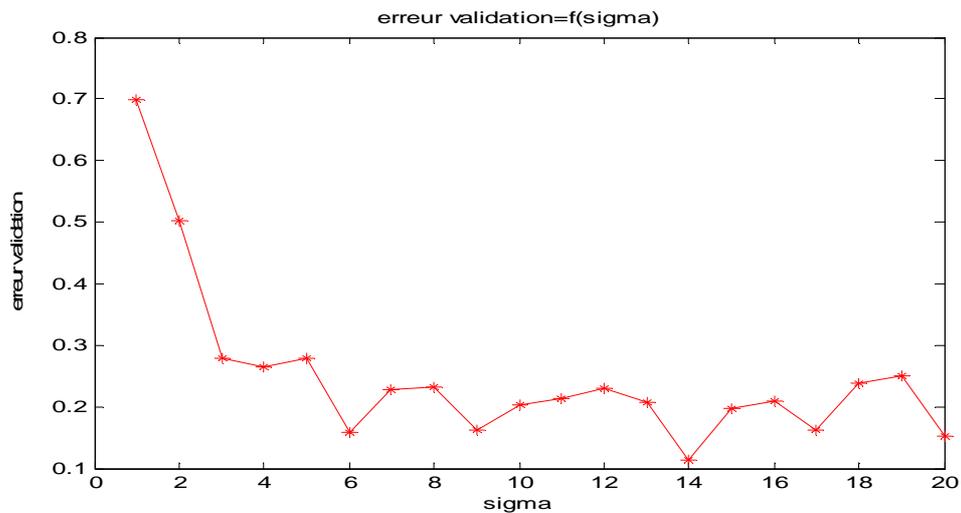


Figure 2.9 graphe NMSE de validation en fonction de sigma (JUS)

2-2-9. Résultats sur la base VIANDE du model RBF

Sur la figure suivante est montrée la superposition de la sortie désirée et la sortie estimée par le régresseur **RBF** optimal (base VIANDE).

La même remarque concernant les bases DIESEL et JUS est valable pour la base VIANDE.

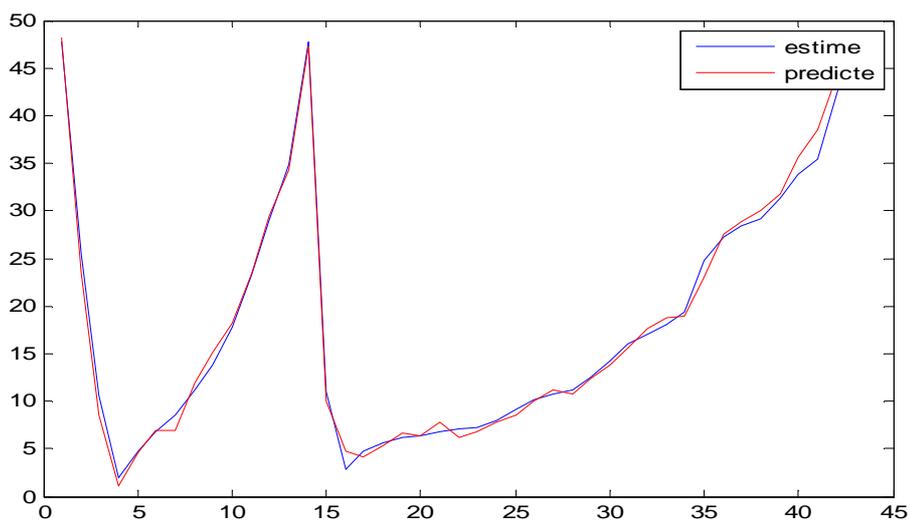


Figure 2.10 graphes de la sortie désirée et la sortie estimée base de VIANDE (test)

La figure suivante reflète l'évolution de l'erreur de validation en fonction des paramètres du modèle (nombre de neurones, largeur de la fonction gaussienne) (base VIANDE).

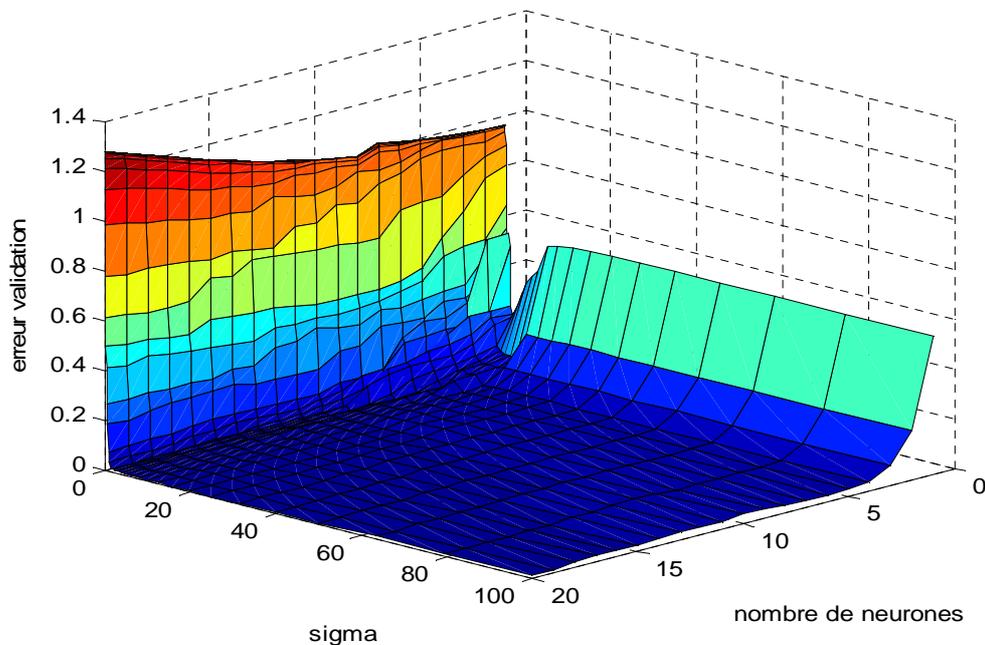


Figure 2.11 graphe NMSE de validation en fonction des paramètres du modèle (VIANDE)

Sur la figure suivante l'évolution de l'erreur de validation apparait en fonction seulement du nombre de neurone (base JUS VIANDE).

Dans ce cas l'erreur de validation chute rapidement pour tendre vers une valeur minimale lorsque (nombre de neurone = 20).

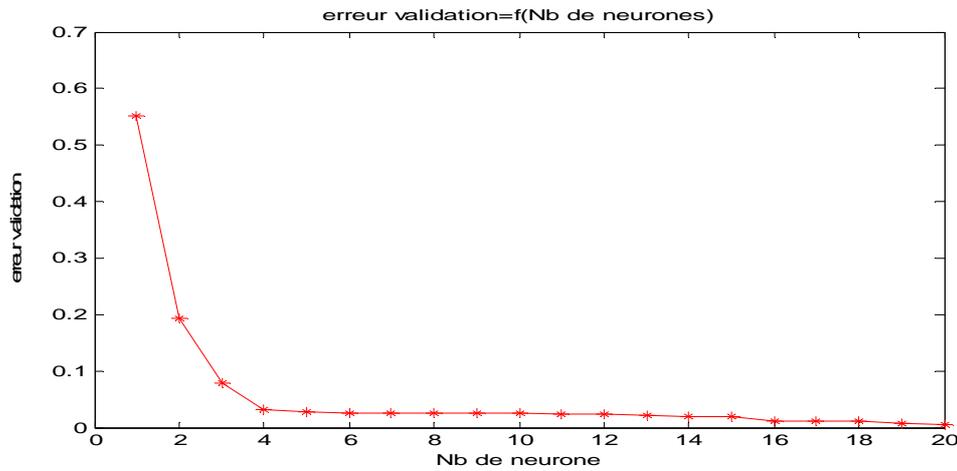


Figure 2.12 graphe NMSE de validation en fonction du nombre de neurones (VIANDE)

L'illustration de l'évolution de l'erreur de validation en fonction seulement de la largeur de la fonction gaussienne (VIANDE) est montrée sur la figure suivante.

L'erreur de validation décroît rapidement vers des valeurs très petites dont la plus petite valeur dans cet intervalle correspond à (sigma = 14).

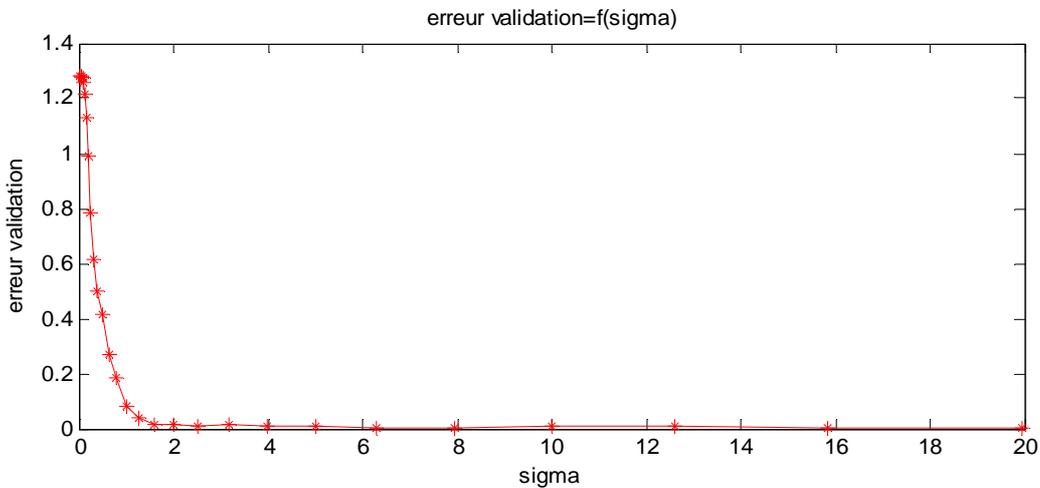


Figure 2.13 graphe NMSE de validation en fonction de sigma (VIANDE)

2-2-10. Résultats sur la base WINE du modèle RBF

Sur la figure suivante est représentée la superposition de la sortie désirée et la sortie estimée par le régresseur **RBF** optimal (base WINE).

On peut dire que la remarque concernant les bases de données précédentes est valable aussi pour la base WINE, il s'agit de dire que la sortie estimée poursuit l'évolution de la sortie désirée.

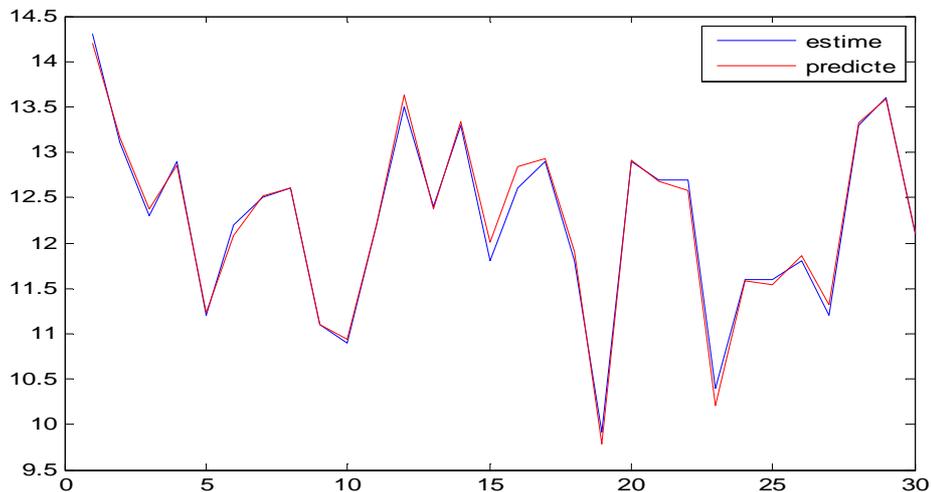


Figure 2.14 graphes de la sortie désirée et la sortie estimée base de WINE (test)

Sur la figure suivante est montrée l'évolution de l'erreur de validation en fonction des paramètres du modèle (nombre de neurones, largeur de la fonction gaussienne) (base WINE).

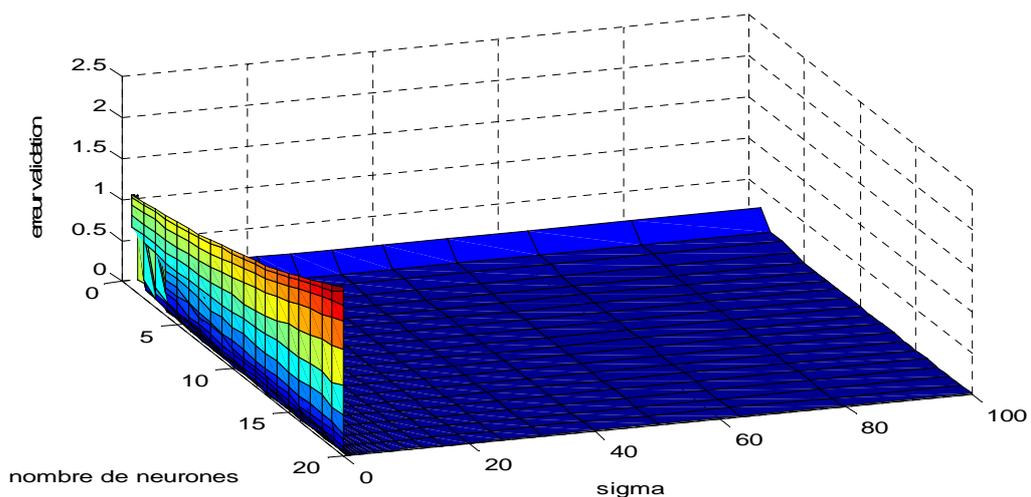


Figure 2.15 Le graphe NMSE de validation en fonction des paramètres du modèle(WINE)

Sur la figure suivante nous illustrons l'évolution de l'erreur de validation en fonction seulement du nombre de neurones pour la base WINE.

Le modèle tend rapidement vers des erreurs de validation optimales dont la plus petite est trouvée pour (nombre de neurone = 14).

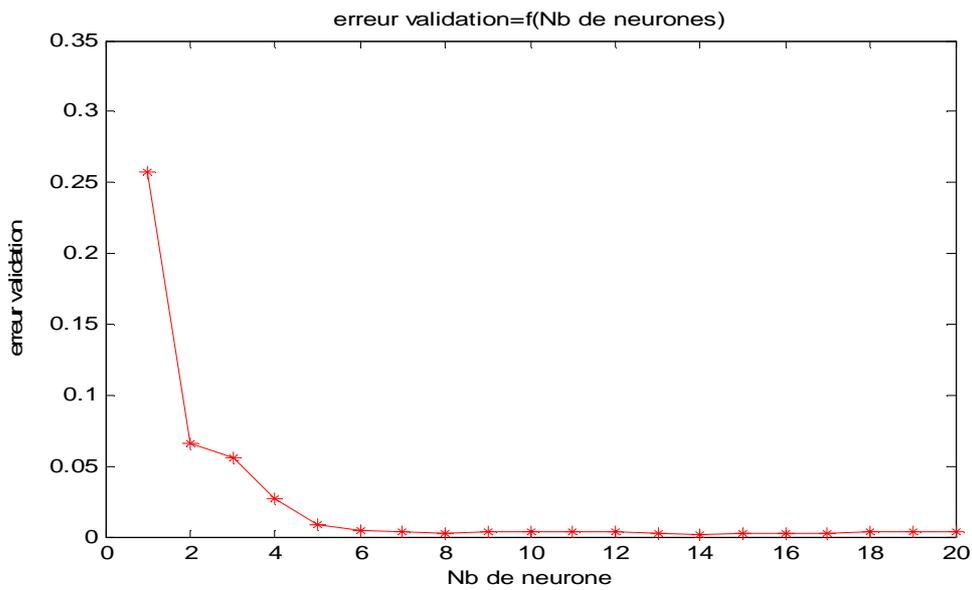


Figure 2.16 graphe de NMSE de validation en fonction du nombre de neurones (WINE)

La figure suivante donne l'évolution de l'erreur de validation en fonction seulement de la largeur de la fonction gaussienne (base WINE).

L'erreur de validation minimale correspond à la largeur de la gaussienne du RBF ($\sigma = 0.32$).

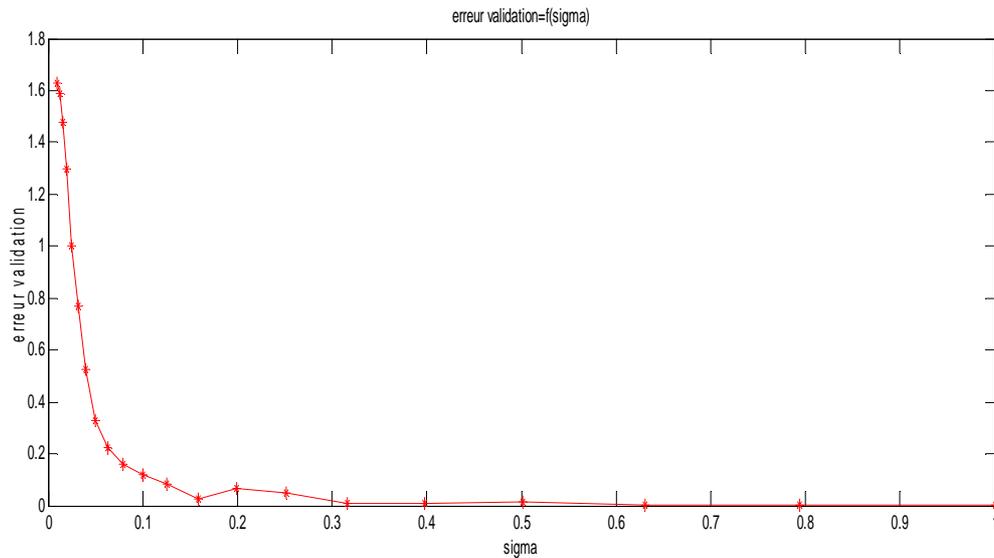


Figure 2.17 graphe NMSE de validation en fonction de sigma (WINE)

2-2-11. Résultats globaux du modèle RBF par base de données

Dans le tableau (2.1) est donnée l'erreur de test du modèle RBF selon les bases des données utilisées.

Le modèle RBF a donné des réponses différentes selon la base de données. La base de données WINE correspond à l'erreur de test minimale, par contre l'erreur de test maximale correspond à la base de données DIESEL.

Base de données	DIESEL	JUS	VIANDE	WINE
NMSEtest	0.3345	0.1712	0.0065	0.0045

Tableau 2.1 NMSE (test) du modèle RBF pour les quatre bases de données

2-3. Régression à base du réseau MLP

Les Perceptrons multicouches appartiennent aux réseaux multicouches ils ne possèdent donc pas de boucle de retour, ils possèdent une fonction d'activation de type sigmoïde.

Le MLP est une extension multicouche du perceptron, qui est un réseau assez limité à un seul neurone.

2-3-1. Le neurone formel

Le neurone formel est une unité élémentaire. Il effectue la somme pondérée de ses entrées, et la soumet à une fonction non linéaire dérivable, pour un neurone formel possédant n entrées, le neurone effectue la somme pondérée:

$$y = \sum_{i=1}^n w_i x_i \quad (2. 16)$$

Sa sortie est activée grâce à une fonction non linéaire :

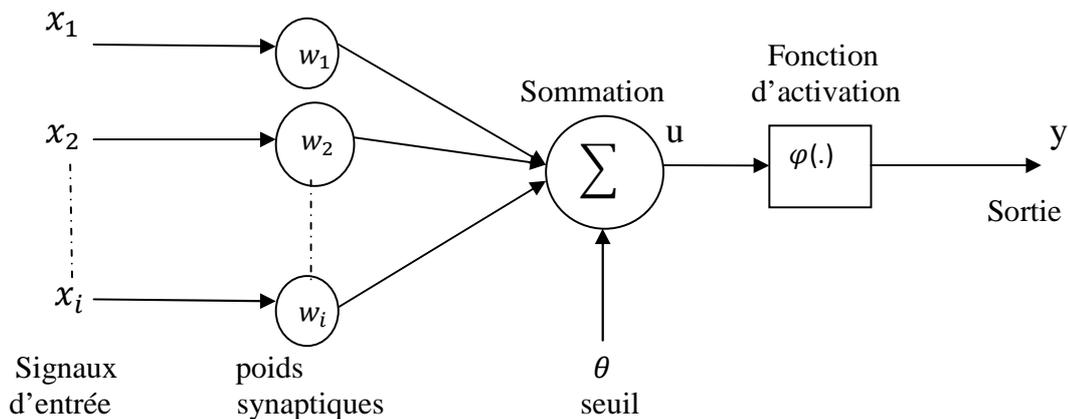


Figure (2.18) Neurone Formel

$$y = \varphi \left(\sum_{i=1}^n w_i x_i \right) \quad (2. 17)$$

Plusieurs fonctions sont utilisées pour l'activation :

La fonction sigmoïde :

$$g(a) = \frac{1}{1 + \exp(-a)} \quad (2. 18)$$

La fonction de Heaviside:

$$g(a) = \begin{cases} 0 & \text{si } a < 0 \\ 1 & \text{ailleurs} \end{cases} \quad (2. 19)$$

2-3-2. Représentation connexionniste d'un réseau MLP

La figure montre un réseau MLP de deux couches cachées, plus la couche d'entrée et la couche de sortie, les fonctions de transferts utilisées dans les couches cachées sont des sigmoïdes, par contre dans la couche de sortie la fonction de transfert utilisée est une fonction linéaire.

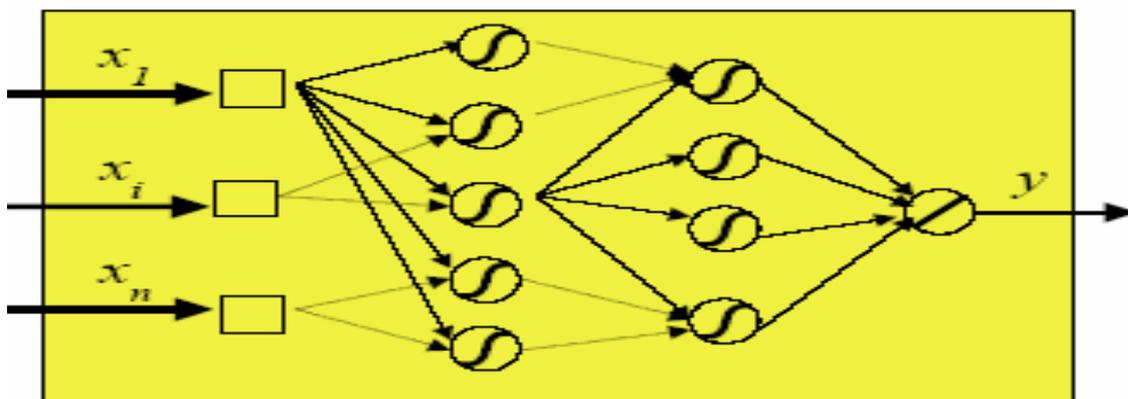


Figure 2.19 Représentation connexionniste d'un réseau MLP.

2-3-3. Définition du réseau MLP

Considérons un réseau à une couche cachée. Le réseau possède :

- Une couche d'entrée à « m » cellules d'entrées $x_i = e_i$ (Il ne s'agit pas de neurones, ces cellules présentent simplement les entrées e_i au réseau).
- Une couche cachée à « n » neurones d'activation y_i .
- Une couche de sortie a p neurones d'activation z_k .
- « n × m » connexions entre la couche d'entrée et la couche cachée, chacune pondérée par v_{ji} .
- « m × p » connexions entre la couche cachée et la couche de sortie, chacune pondérée par w_{kj} .

2-3-4. Initialisation des poids des connexions

Ces poids sont choisis au hasard

2-3-5. Propagation des entrées

Les e_i étant présentées à la couche d'entrée : $x_i = e_i$ on propage vers la couche cachée :

$$y_i = \varphi \left(\sum_{i=1}^m x_i v_{ij} + x_0 \right) \quad (2. 20)$$

Puis de la couche cachée vers la couche de sortie :

$$z_k = \varphi \left(\sum_{j=1}^n y_j w_{kj} + y_0 \right) \quad (2. 21)$$

Les valeurs x_0 et y_0 sont des biais : des scalaires et non des sorties de la couches précédente.

2-3-6. retropropagation de l'erreur

Pour chaque exemple de la base d'apprentissage appliqué en entrée du réseau, on calcule son erreur sur les couches de sortie, c'est à dire la différence entre la sortie désirée s_k et la sortie réelle z_k :

$$E_k = z_k(1 - z_k)(s_k - z_k) \quad (2. 22)$$

On propage cette erreur sur la couche cachée ; l'erreur de chaque neurone de la couche cachée est donnée par :

$$F_j = y_j(1 - y_j) \sum_{k=1}^p w_{kj} E_k \quad (2. 23)$$

2-3-7. Correction des poids des connexions

Il reste à modifier les poids des connexions entre la couche d'entrée et la couche cachée :

$$\begin{cases} \Delta w_{kj} = \mu y_j E_k \\ \Delta x_0 = \mu E_k \end{cases} \quad (2. 24)$$

Entre la couche cachée et la couche de sortie :

$$\begin{cases} \Delta v_{ij} = \mu x_i F_j \\ \Delta y_0 = \mu F_j \end{cases} \quad (2. 25)$$

μ Étant un paramètre qu'il reste à déterminer.

La propagation des entrées doit être bouclée jusqu'à un critère d'arrêt à définir.

2-3-8. problèmes du dimensionnement

Le gros problème des MLP est qu'on ne peut pas connaître a priori les dimensions du réseau pour un problème donné. L'expérience montre qu'il n'est pas nécessaire d'avoir plus d'une couche cachée ,d'autres références montrent par exemple qu'on obtient de meilleurs résultats avec un réseau à une couche cachée de 150 neurones plutôt qu'avec deux couches cachées de 65 puis 39 neurones.

En revanche, on ne peut pas déterminer à priori le nombre de neurones de la couche cachée nécessaire à un problème donné.

Le μ permettant d'ajuster les poids des connexions est également délicat à optimiser, il est nécessaire de régler ces paramètres en tâtonnant, c'est à dire qu'on

regarde le comportement du réseau sur la base d'apprentissage, et on choisit la meilleure configuration.

2-3-9. problèmes du sur-apprentissage

Un autre paramètre doit être trouvé empiriquement : le nombre d'itérations lors de la phase d'apprentissage.

Celui ci est primordial puisque il apparaît au bout d'un certain nombre d'itération le phénomène bien connu du sur-apprentissage, durant lequel le MLP commence à apprendre par cœur les exemples de la base d'apprentissage et perd sa capacité à généraliser.

En utilisant une base de validation, on peut calculer l'erreur de généralisation du réseau en fonction du nombre d'itération. On choisira naturellement le nombre d'itérations qui minimise l'erreur de généralisation.

2-3-10. problèmes des minima locaux

Nous avons vu qu'il était possible de tomber dans des minima locaux, cela est caractéristique des méthodes à descente de gradient.

En lançant l'apprentissage plusieurs fois avec des initialisations différentes des poids des connexions, on vérifie que ce n'est pas le cas, on peut également ajouter une composante aléatoire dans les poids pour explorer des coins de l'espace des paramètres que l'algorithme n'explore pas (principe du recuit simulé).

2-3-11. Résultats sur la base DIESEL du modèle MLP

Sur la figure suivante nous enregistrons la superposition de la sortie désirée et la sortie estimée par le régresseur **MLP** optimal pour la base DIESEL.

On peut remarquer comme pour le modèle RBF que la sortie estimée par un régresseur MLP poursuit régulièrement la sortie désirée.

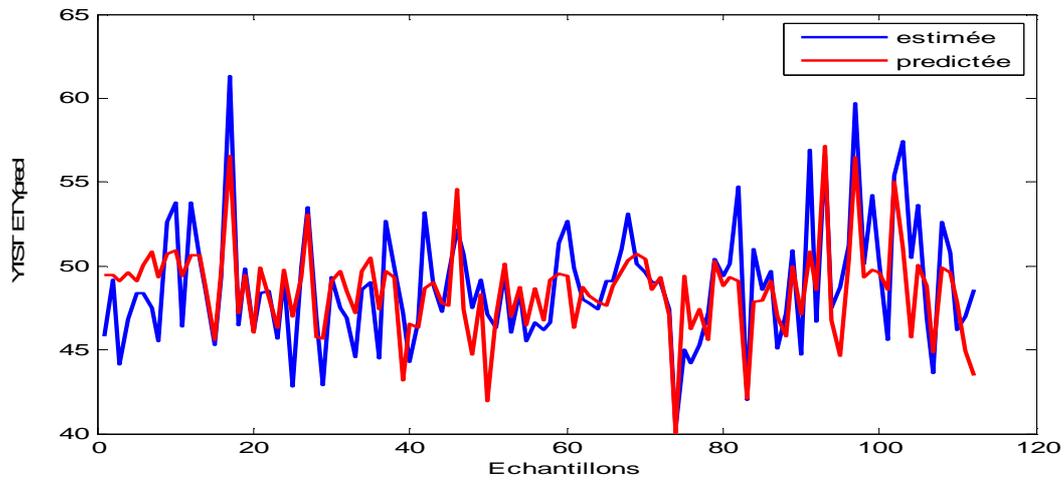


Figure 2.20 graphes de la sortie désirée et la sortie estimée base de DIESEL (test)

La figure 2.21 montre l'évolution de l'erreur de validation en fonction du (nombre de neurones dans la première couche cachée, nombre de neurones dans la deuxième couche cachée) (base DIESEL).

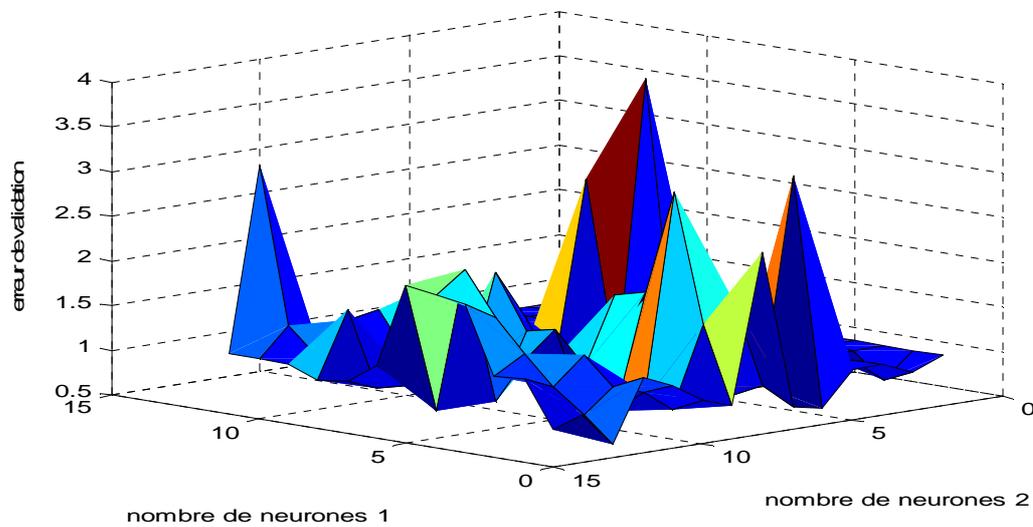


Figure 2.21 graphe NMSE de validation en fonction des paramètres du modèle(DIESEL)

Sur la figure suivante est illustrée l'évolution de l'erreur de validation en fonction du (nombre de neurones maximal) (base DIESEL).

L'erreur de validation décroît globalement pour atteindre la valeur minimale pour un nombre de neurones ($C_{max}=16$).

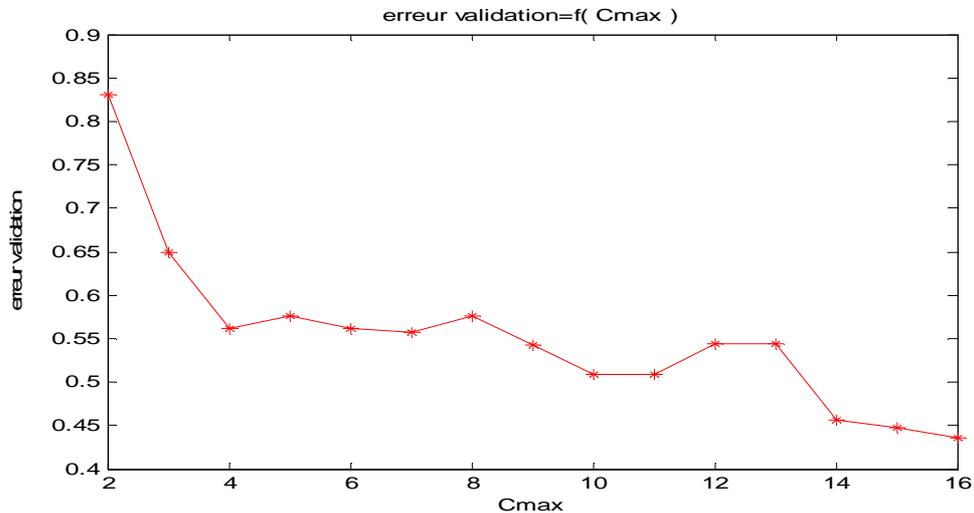


Figure 2.22 graphe NMSE de validation en fonction du nombre de neurones (DIESEL)

2-3-12. Résultats sur la base JUS du modèle MLP

Sur la figure suivante on voit la superposition de la sortie désirée et la sortie estimée par le régresseur **MLP** optimal pour la base JUS.

On remarque comme le cas de tous les modèles étudiés que la sortie estimée décrit approximativement la sortie désirée.

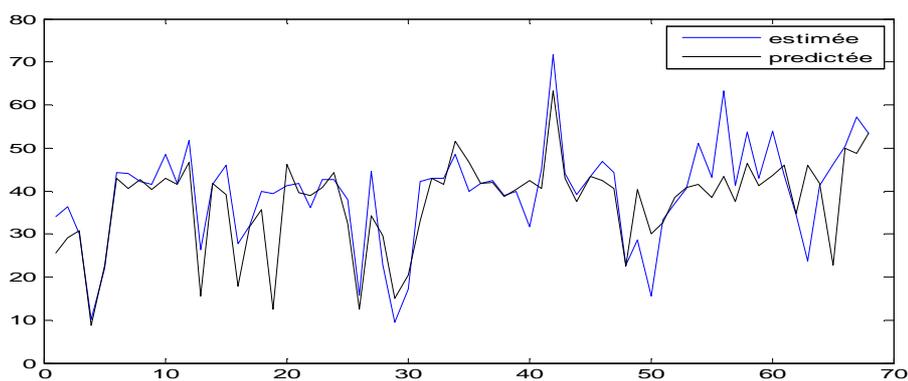


Figure 2.23 graphes de la sortie désirée et la sortie estimée base de JUS (test)

On montre sur la figure suivante l'évolution de l'erreur de validation en fonction du (nombre de neurones dans la première couche cachée, nombre de neurones dans la deuxième couche cachée) (base JUS).

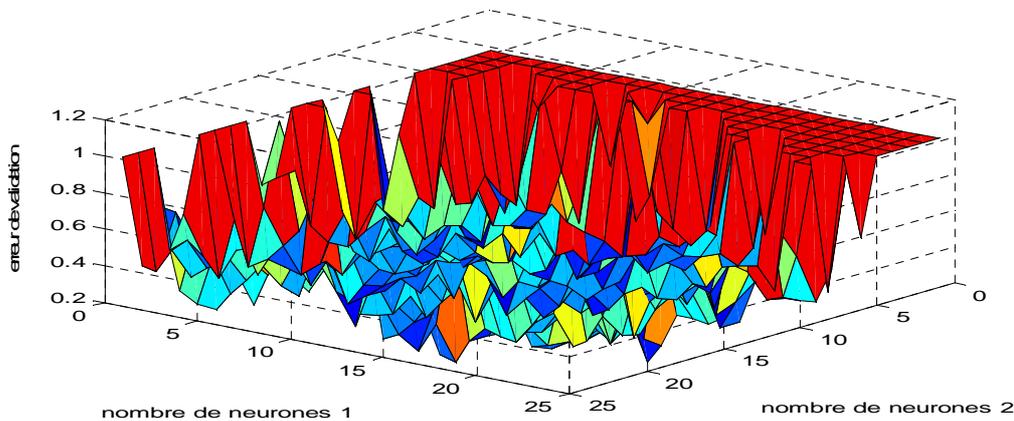


Figure 2.24 graphe NMSE de validation en fonction des paramètres du modèle (JUS)

La figure suivante enregistre l'évolution de l'erreur de validation en fonction du (nombre de neurones maximal) (base JUS)

L'erreur de validation minimale est trouvée pour un nombre de neurones ($C_{max}=26$).

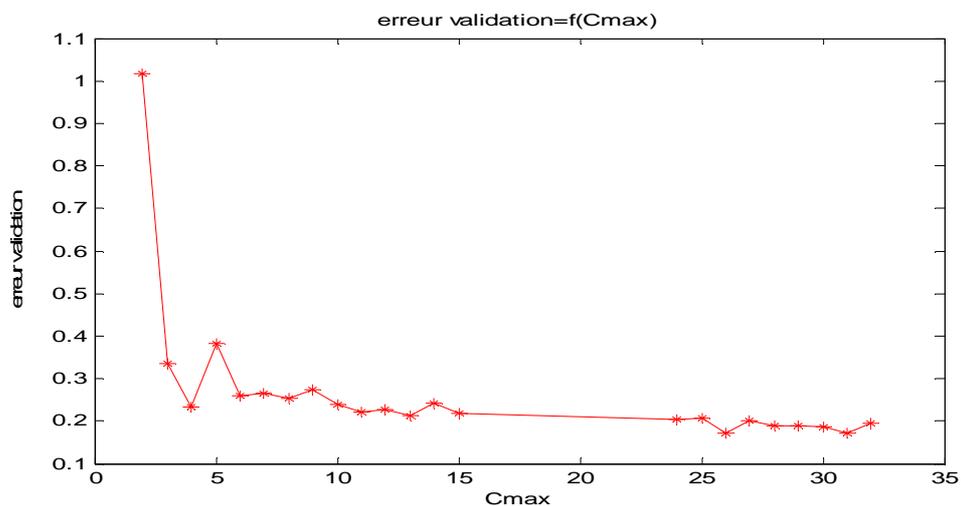


Figure 2.25 graphe NMSE de validation en fonction du nombre de neurones (JUS)

2-3-13. Résultats sur la base VIANDE du model MLP

Sur la figure suivante est enregistrée la superposition de la sortie désirée et la sortie estimée par le régresseur **MLP** optimal (base **VIANDE**).

Les deux sorties estimée et désirée possèdent approximativement les mêmes allures.

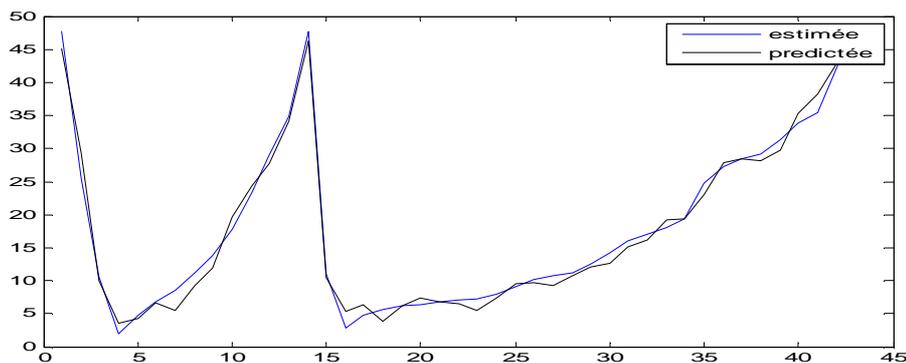


Figure 2.26 graphes de la sortie désirée et la sortie estimée base **VIANDE (test)**

Sur la figure suivante nous montrons l'évolution de l'erreur de validation en fonction du (nombre de neurones dans la première couche cachée, nombre de neurones dans la deuxième couche cachée) (base **VIANDE**).

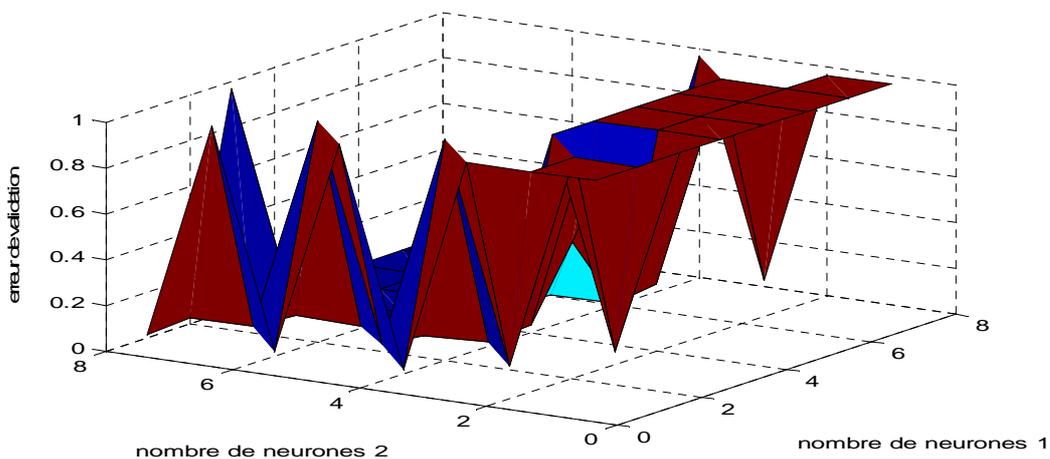


Figure 2.27 graphe NMSE de validation en fonction des paramètres du modèle(VIANDE)

2-3-14. Résultats sur la base WINE du model MLP

Sur la figure suivante sont données la superposition de la sortie désirée et la sortie estimée par le régresseur **MLP** optimal (base WINE).

La sortie estimée décrit presque la sortie désirée.

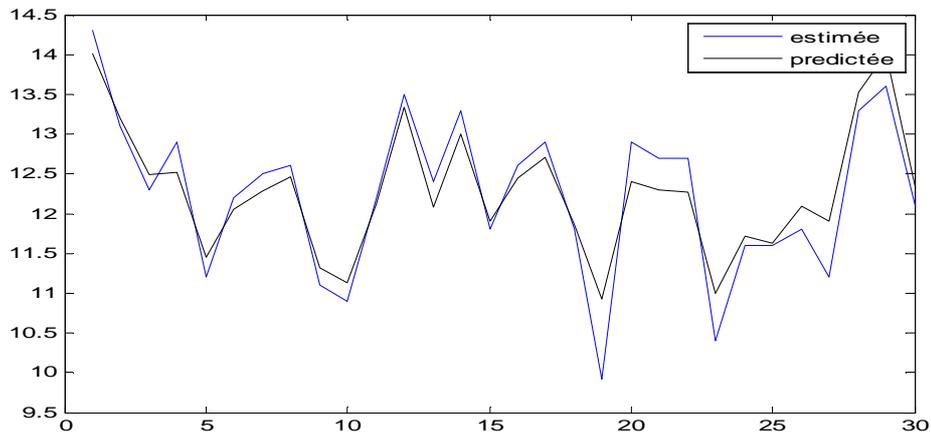


Figure 2.28 graphes de la sortie désirée et la sortie estimée base de WINE (test)

Nous enregistrerons sur la figure suivante l'évolution de l'erreur de validation en fonction du (nombre de neurones dans la première couche cachée, nombre de neurones dans la deuxième couche cachée) (base WINE).

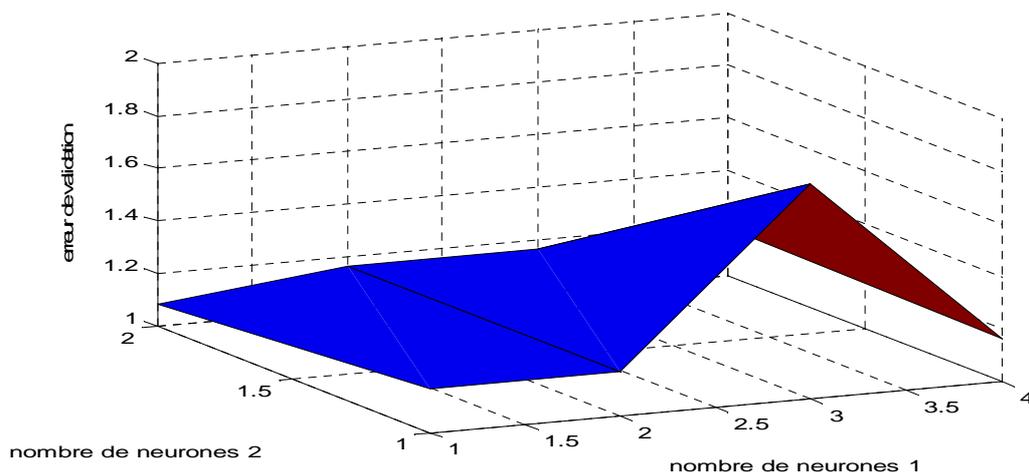


Figure 2.29 graphe NMSE de validation en fonction des paramètres du modèle (WINE)

2-3-15. Résultats globaux du modèle MLP par base de données

Dans le tableau 2.2 sont enregistrées les erreurs de test du modèle **MLP** selon la base des données utilisée.

Le régresseur MLP répond différemment selon la base de données, l'erreur de test minimale correspond à la base de données **VIANDE** et l'erreur de test maximale correspond à la base de données **DIESEL**.

Base de données	DIESEL	JUS	VIANDE	WINE
NMSEtest	0.3564	0.3188	0.0134	0.0641

Tableau 2.2 NMSE (test) du modèle MLP pour les quatre bases de données

2-4. Conclusion

Dans ce chapitre nous avons interprété les résultats de la régression non linéaire à base des réseaux de neurones (RBF, MLP).

Les variations de l'erreur de validation en fonction des paramètres des modèles (RBF, MLP) sont illustrées sur des figures.

Le modèle RBF a donné le meilleur résultat pour la base de données **WINE** par contre le modèle MLP donne son meilleur résultat pour la base de données **VIANDE**.

L'établissement d'une méthode qui permet de choisir le meilleur régresseur pour chaque échantillon de la même base de données est indispensable pour améliorer les résultats de régressions.

Chapitre 3
Régressions non linéaires a base
de noyaux (SVM)

3-1. Introduction

Le principe d'un SVM est de transformer l'espace dans lequel on travaille en un autre espace dit de grande dimensions et à l'intérieur duquel nous établirons une séparation en différentes classes.

Le passage d'un espace à l'autre s'effectue suivant quelques transformations mathématiques utilisant des fonctions non linéaires, encore appelées noyaux (fonctions de kernel)

3-2. Formalisation

Soit un ensemble d'apprentissage $S = \{(x_i, y_i)\}$ avec $i=1 \dots n$ dont les éléments obéissent à la loi jointe

$$P(x, y) = P(x) P\left(\frac{y}{x}\right) \quad (3-1)$$

On cherche à approcher une loi sous-jacente $f(x)$ telle que $y_i = f(x_i)$ par une hypothèse $h_\alpha(x)$ aussi proche que possible. Les α sont les paramètres du système d'apprentissage.

Si $f(\cdot)$ est discrète on parle de classification

Si $f(\cdot)$ est une fonction continue on parle alors de régression

Il existe un hyperplan qui sépare les deux classes les points de l'hyper plan sont décrits par l'équation

$$w \cdot x + b = 0 \quad (3-2)$$

Ou w est la normale au plan et $\frac{|b|}{\|w\|}$ la distance entre l'hyperplan et l'origine

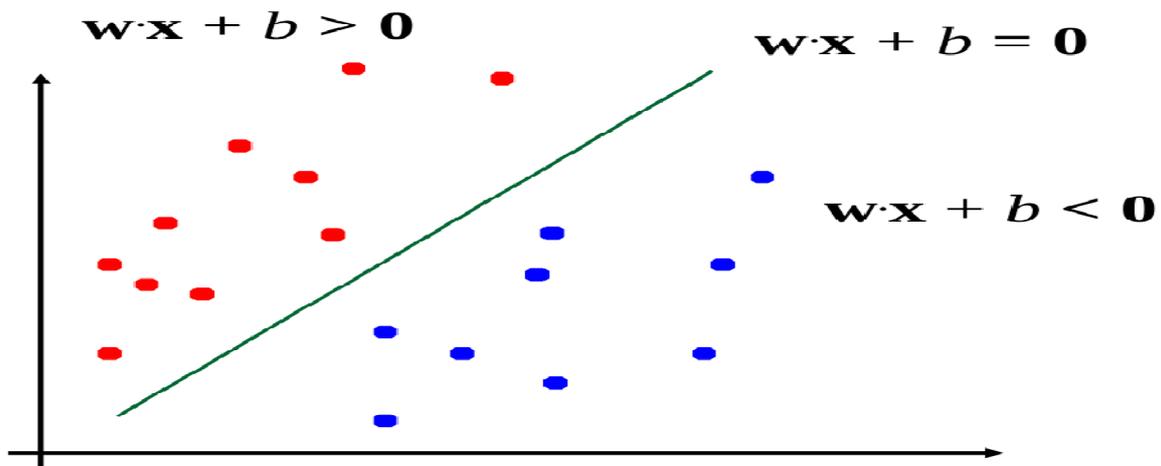


Figure 3.1 classifieur linéaire

Pour trouver le meilleur classifieur Il existe de nombreux choix possibles pour w et b :

$$y(x) = \text{signe}(w \cdot x + b) = \text{signe}(k w \cdot x + k \cdot b) \quad (3-3)$$

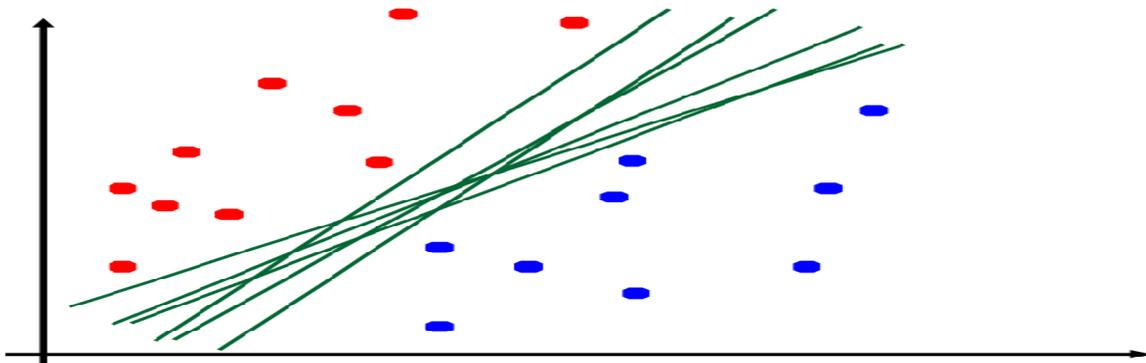


Figure 3.2 choix d'un classifieur

3-3. Notion de marge

Dans le cas séparable, on va considérer les points les plus près de l'hyperplan séparateur : vecteurs supports (support vectors).

Pour tout point de l'espace des exemples, la distance à l'hyperplan séparateur est donnée par :

$$r = \frac{|w \cdot x + b|}{\|w\|} \quad (3-4)$$

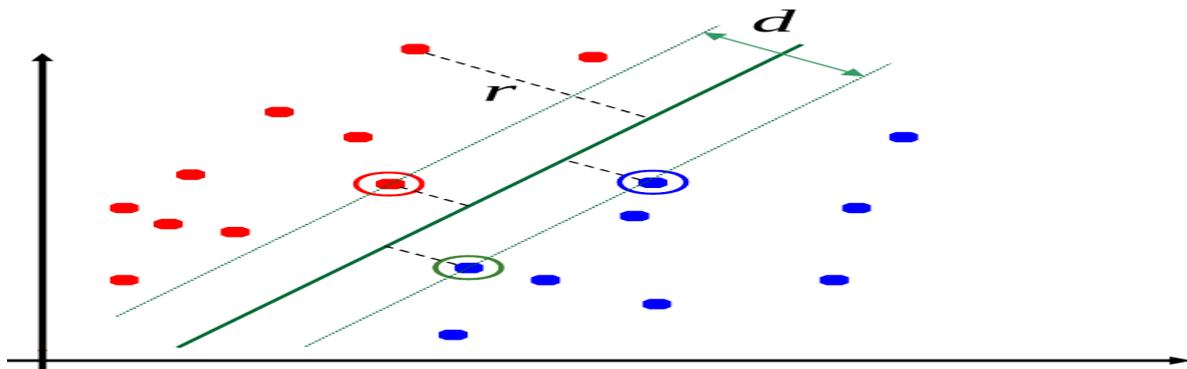


Figure 3.3 distance à l'hyperplan séparateur

On appelle marge d la distance entre les 2 classes, c'est cette distance d qu'on souhaiterait maximiser

3-3-1. Quantification de la marge

On considère que les points les plus proches sont situés sur les hyperplans canoniques données par :

$$w \cdot x + b = \pm 1 \quad (4-8)$$

Dans ce cas, la marge est :

$$d = \frac{2}{\|w\|}$$

Les conditions d'une bonne classification sont :

$$\begin{cases} w \cdot x + b \geq 1, & \text{si } y_i = 1 \\ w \cdot x + b < -1, & \text{si } y_i = -1 \end{cases} \quad (4-9)$$

3-3-2. Quelques noyaux utilisables

Noyau linéaire

$$K(x_i, x_j) = x_i \cdot x_j$$

Noyau polynomial de degré p

$$K(x_i, x_j) = (1 + x_i \cdot x_j)^P$$

Noyau Gaussien

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

Cette formulation est équivalente aux réseaux de neurones à bases radiales avec l'avantage supplémentaire que les centres des fonctions à base radiale (qui sont les vecteurs supports) sont optimisés.

3-4. Les régresseurs SVM

Les SVMs peuvent être appliqués aux problèmes de régression par l'introduction d'une fonction de perte alternative cette fonction de perte doit être modifiée pour inclure une mesure de distance, la figure 3.1 illustre les quatre fonctions de perte possibles.

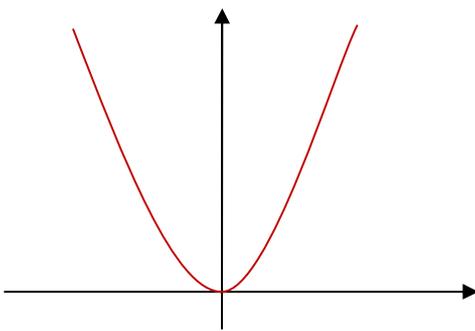


Figure 3.4.a fonction quadratique

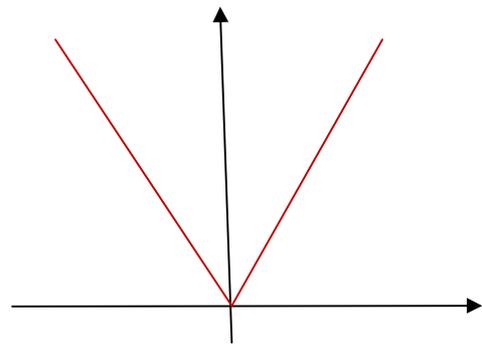


Figure 3.4.b fonction de Laplace

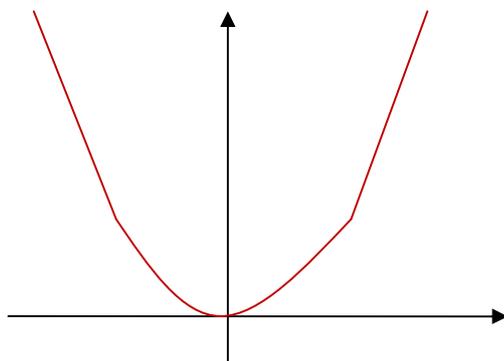


Figure 3.4.c fonction de Huber

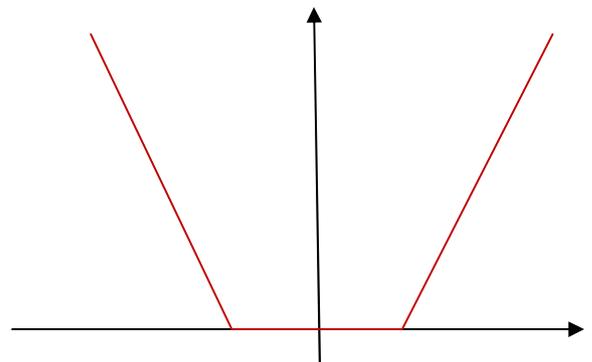


Figure 3.4.d fonction peu sensible

La fonction de perte dans la figure 3.1.a correspond au critère conventionnel d'erreur des moindres carrés, la fonction de perte dans la figure 3.1.b est la fonction de Laplace qui est moins sensible aux annexes que la fonction de perte quadratique, Huber a proposé la fonction de perte dans la figure 3.1.c comme fonction de perte robuste qui a les propriétés optimales quand la distribution fondamentale des données est inconnue

Pour aborder cette issue, Vapnik a proposé la fonction de perte dans la figure 3.2.d comme approximation à la fonction de perte du Huber

3-5. Résultats sur la base DIESEL du modèle SVM (régresseur)

Sur la figure suivante montre la superposition de la sortie désirée et la sortie estimée par le régresseur **SVM** optimal (base DIESEL).

On peut remarquer que la sortie estimée poursuit la sortie désirée comme pour les régresseurs précédemment étudiés.

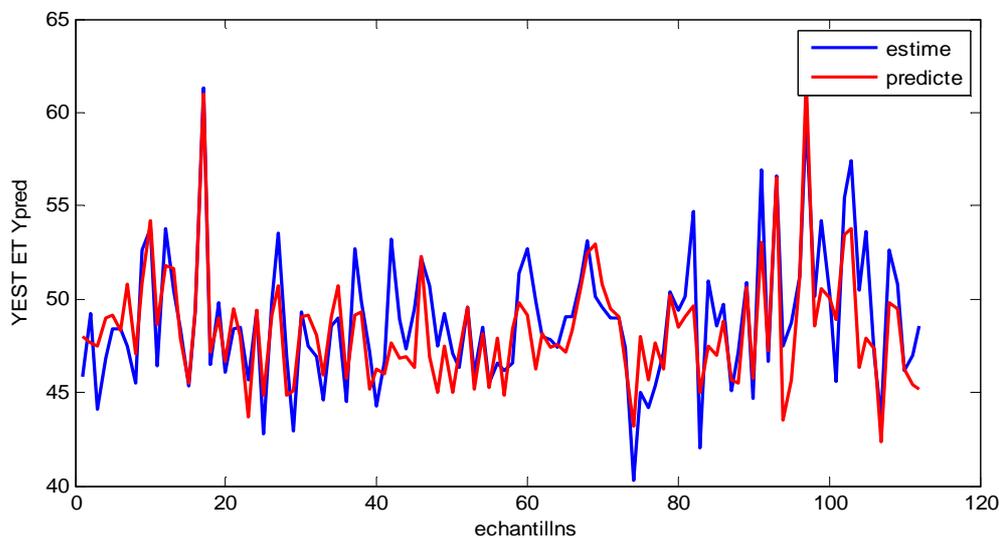


Figure 3.5 graphes de la sortie désirée et la sortie estimée base de DIESEL (test)

Sur la figure suivante nous illustrons l'évolution de l'erreur de validation en fonction des paramètres du régresseur SVM (c , σ) (base DIESEL).

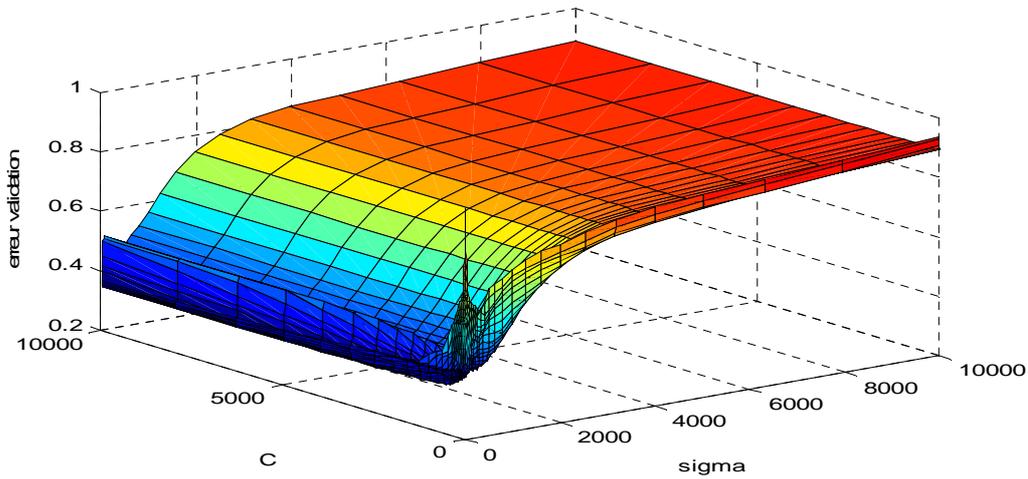


Figure 3.6 graphe NMSE de validation en fonction des paramètres du modèle (DIESEL)

Sur la figure suivante nous montrons l'évolution de l'erreur de validation en fonction seulement du paramètre (c) du régresseur SVM (base DIESEL).

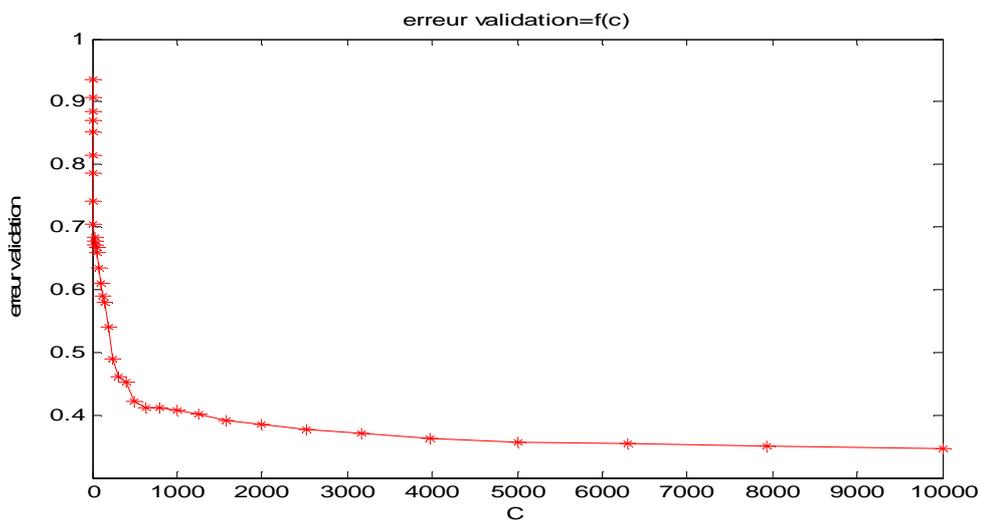


Figure 3.7 graphe NMSE de validation en fonction du paramètre (C) (DIESEL)

On montre sur la figure suivante l'évolution de l'erreur de validation en fonction seulement du paramètre (sigma) du régresseur SVM (base DIESEL).

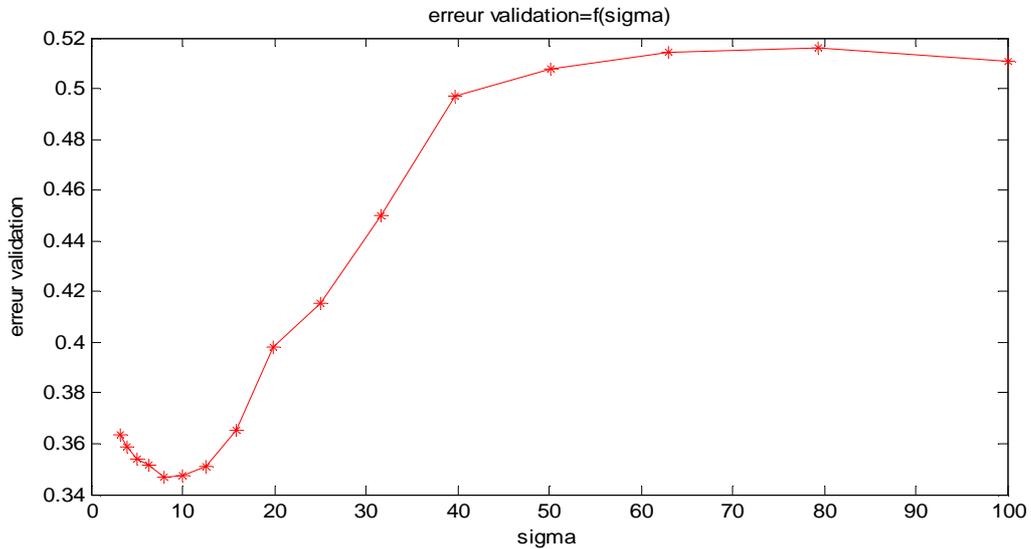


Figure 3.8 graphe NMSE de validation en fonction du paramètre (sigma) (DIESEL)

3-6. Résultats sur la base JUS du model SVM (régresseur)

Sur la figure suivante nous enregistrons la superposition de la sortie désirée et la sortie estimée par le régresseur SVM optimal (base JUS).

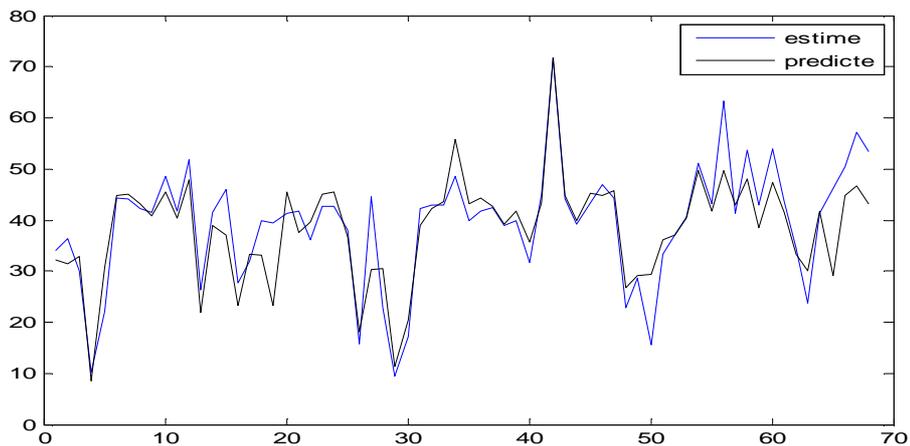


Figure 3.9 graphes de la sortie désirée et la sortie estimée base de JUS (test)

Sur la figure suivante est donnée l'évolution de l'erreur de validation en fonction des paramètres du régresseur **SVM** (c , σ) (base JUS).

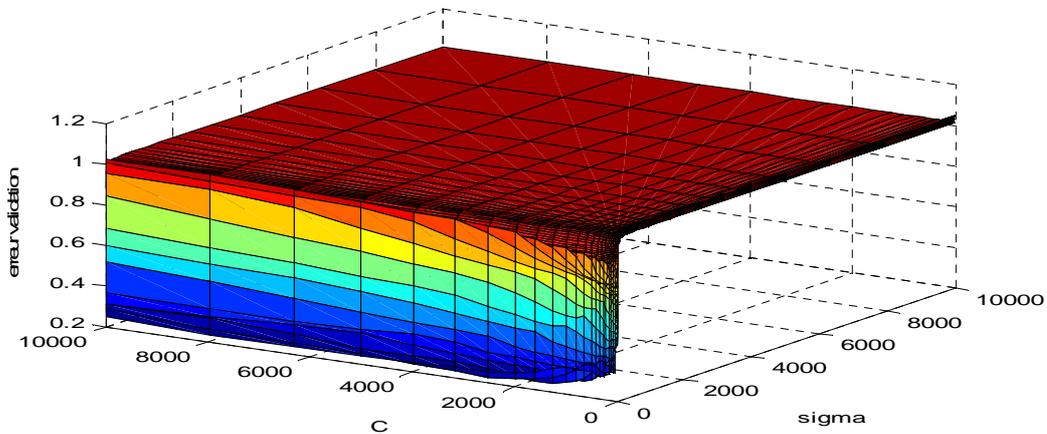


Figure 3.10 Le graphe NMSE en fonction des paramètres du modèle pour la base (JUS)

Sur la figure suivante nous enregistrons l'évolution de l'erreur de validation en fonction seulement du paramètre (c) du régresseur **SVM** (base JUS).

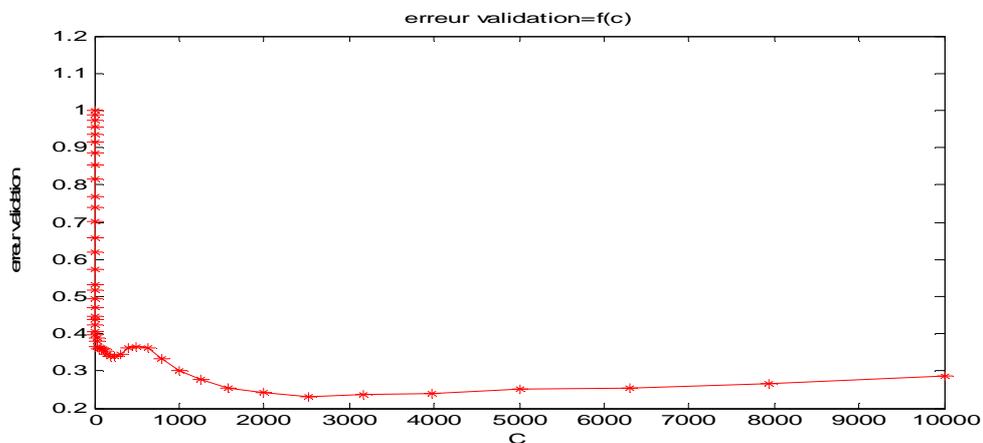


Figure 3.11 graphe NMSE de validation en fonction du paramètre (C) pour la base (JUS)

Sur la figure suivante est enregistrée l'évolution de l'erreur de validation en fonction seulement du paramètre (σ) du régresseur **SVM** (base JUS).

L'erreur de validation décroît pour atteindre une valeur minimale puis elle croît sur le reste de l'intervalle.

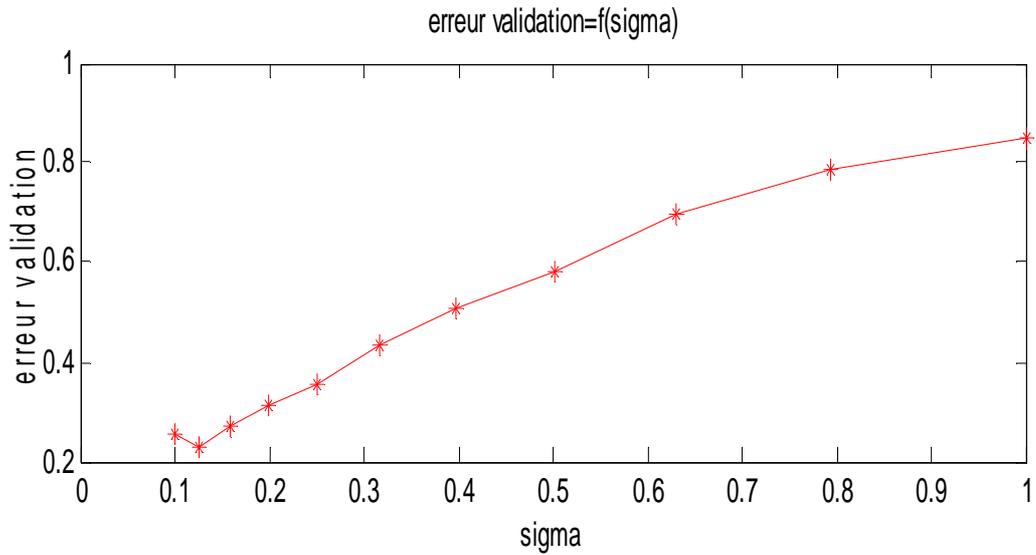


Figure 3.12 graphe NMSE de validation en fonction du paramètre (sigma) (JUS)

3-7. Résultats sur la base VIANDE du modèle SVM (régresseur)

La figure suivante illustre la superposition de la sortie désirée et la sortie estimée par le régresseur SVM optimal (base VIANDE).

On remarque comme d'un des cas précédents que la sortie estimée poursuit régulièrement la sortie désirée.

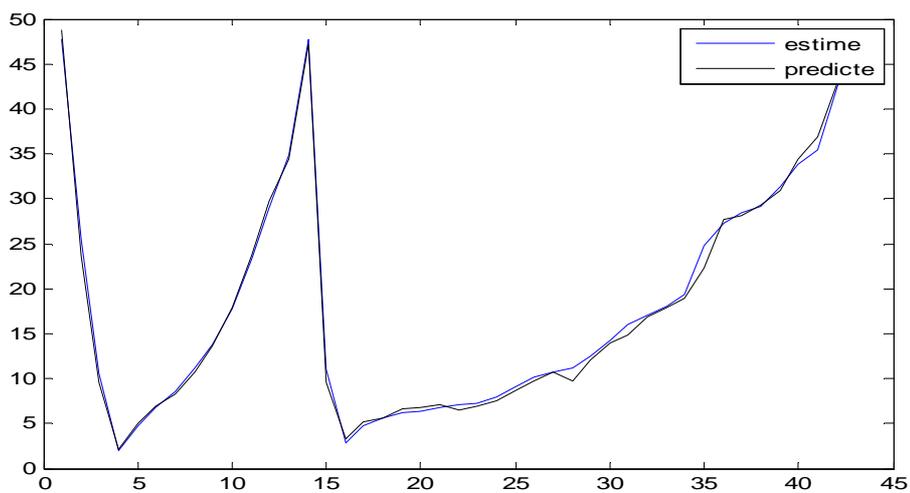


Figure 3.13 graphes de la sortie désirée et la sortie estimée base de VIANDE (test)

La figure de trois dimensions suivante montre l'évolution de l'erreur de validation en fonction des paramètres du régresseur **SVM** (c, sigma) (base VIANDE).

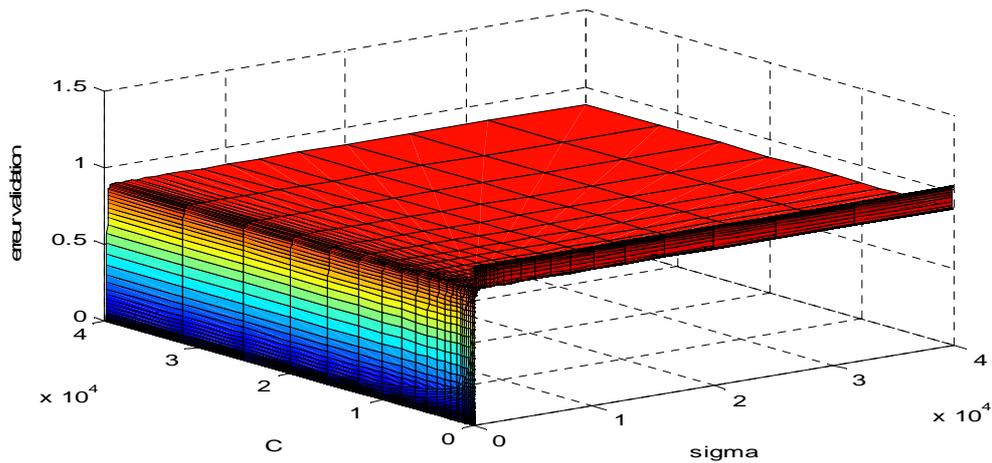


Figure 3.14 graph NMSE de validation en fonction des paramètres du modèle(VIANDE)

La figure suivante illustre l'évolution de l'erreur de validation en fonction seulement du paramètre (c) du régresseur **SVM** (VIANDE).

L'erreur de validation décroît en fonction des valeurs du paramètre c sur cet intervalle.

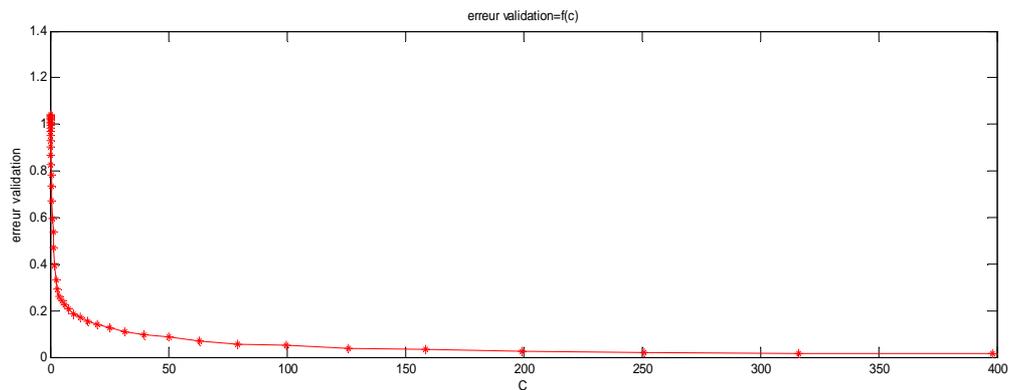


Figure 3.15 graphe NMSE de validation en fonction du paramètre (C) (VIANDE)

Sur la figure suivante on donne l'évolution de l'erreur de validation en fonction seulement du paramètre (sigma) du régresseur **SVM** (base **VIANDE**).

L'erreur de validation tend rapidement vers des petites valeurs puis elle augmente lentement sur le reste de cet intervalle.

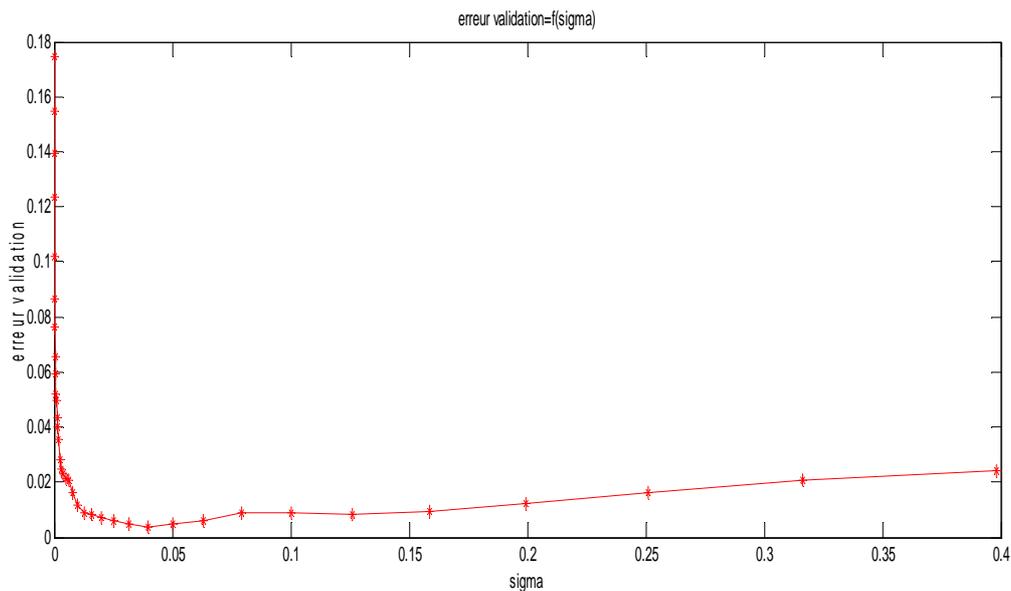


Figure 3.16 graphe NMSE de validation en fonction du paramètre (sigma) (**VIANDE**)

3-8. Résultats sur la base WINE du modèle SVM (régresseur)

Sur la figure suivante on montre la superposition de la sortie désirée et de la sortie estimée par le régresseur **SVM** optimal (base **WINE**).

On remarque que la sortie estimée et la sortie désirée ont presque les mêmes allures.

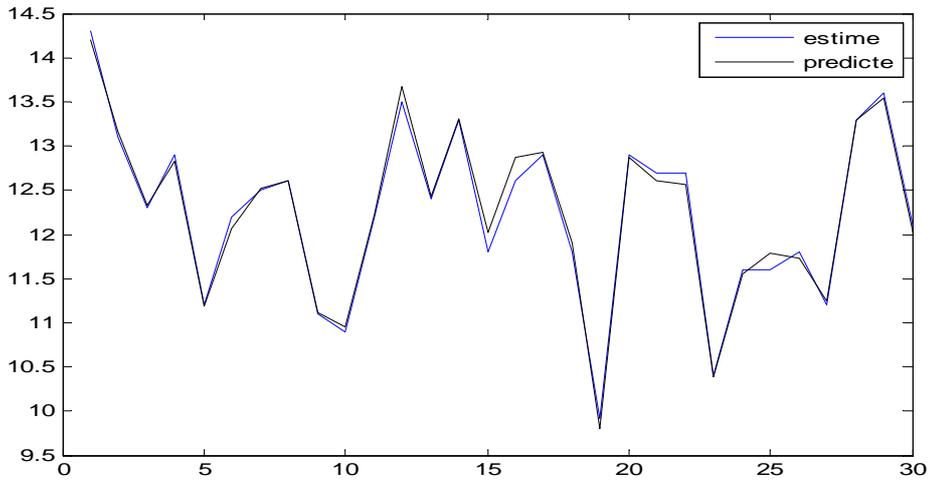


Figure 3.17 graphes de la sortie désirée et la sortie estimée base de WINE (test)

Sur la figure suivante on voit l'évolution de l'erreur de validation en fonction des paramètres du régresseur SVM (c, sigma) (base WINE).

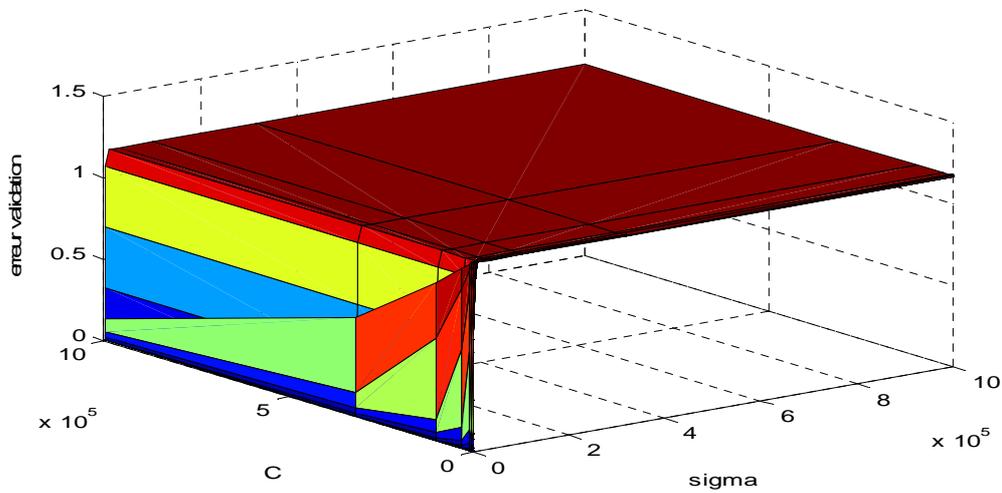


Figure 3.18 graphe NMSE de validation en fonction des paramètres du modèle (WINE)

Sur la figure suivante on montre l'évolution de l'erreur de validation en fonction seulement du paramètre (c) du régresseur SVM (WINE).

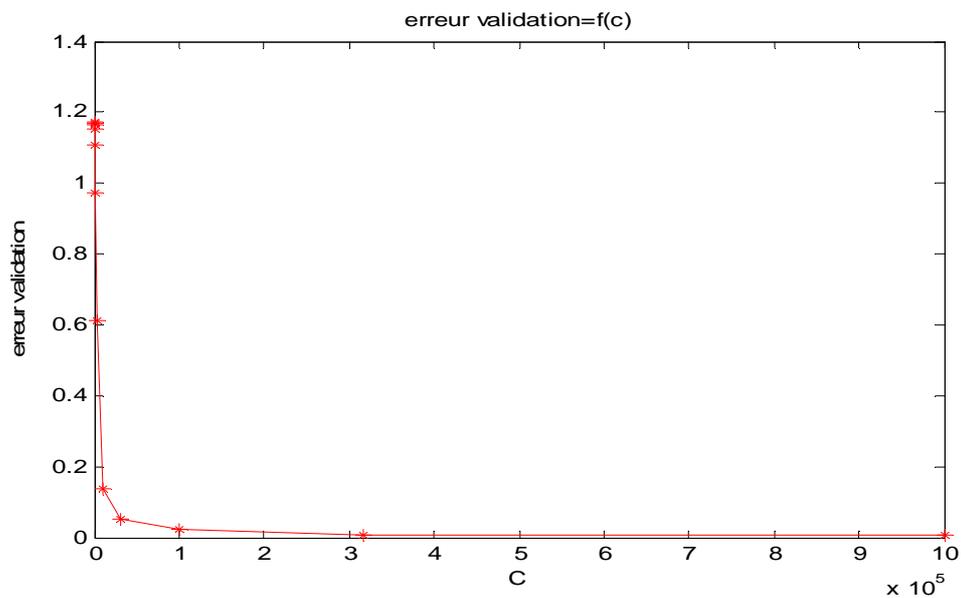


Figure 3.19 graphe NMSE de validation en fonction du paramètre (C) (WINE)

Sur la figure suivante on voit l'évolution de l'erreur de validation en fonction seulement du paramètre (sigma) du régresseur **SVM** (base WINE).

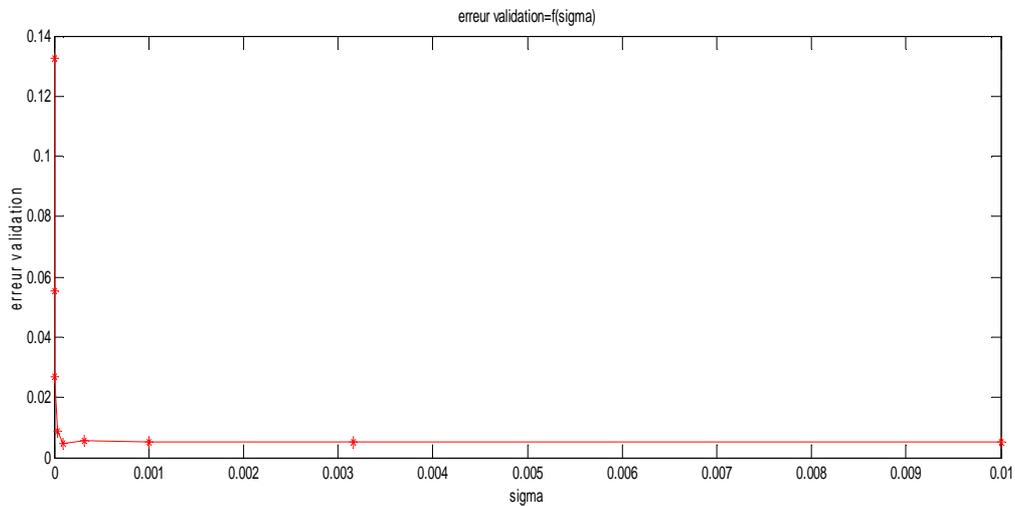


Figure 3.20 graphe NMSE de validation en fonction du paramètre (sigma) (WINE)

3-9. Résultats globaux du modèle MLP par base de données

Dans le tableau 3 on montre l'erreur de test du modèle **SVM** selon les bases de données utilisées.

On remarque que le régresseur SVM donne son meilleur résultat avec la base de données VIANDE.

Base de données	DIESEL	JUS	VIANDE	WINE
NMSEtest	0.3409	0.1732	0.0036	0.0054

Tableau 3 NMSE (test) du modèle SVM pour les quatre bases de données

3-10. Conclusion

Dans ce chapitre nous avons interprété les résultats des modèles à base de noyaux (SVM), nous avons montré l'évolution de l'erreur de validation en fonction des paramètres du modèle.

Le modèle SVM donne globalement des bons résultats comparativement avec les régresseurs à base de réseaux de neurones, mais dire que le régresseur SVM répond mieux que les modèles à bases de réseaux de neurones n'est pas une règle car le régresseur RBF donne des réponses plus précises que le SVM sur trois bases de données (DIESEL, JUS, WINE).

Il est indispensable donc de faire une méthode de classification des régresseurs pour qu'on puisse aboutir à une amélioration des résultats, c'est-à-dire une meilleure performance

Chapitre 4
Classification des régresseurs
en utilisant un SVM

4-1. Introduction

Dans ce chapitre nous allons interpréter et expliquer la partie essentielle de ce projet de mémoire, il s'agit de la méthode de classification des régresseurs précédemment établis par un classificateur SVM.

Nous allons expliquer et détailler particulièrement le schéma bloc de la méthode afin de comprendre la démarche et le fonctionnement du classificateur.

Les résultats de classification doivent être interprétés dans des tableaux pour qu'on puisse faire des comparaisons selon le type du régresseur mis en vigueur d'une part et d'autre par selon la base de données utilisée.

4-2. schéma bloc de la Méthode de classification

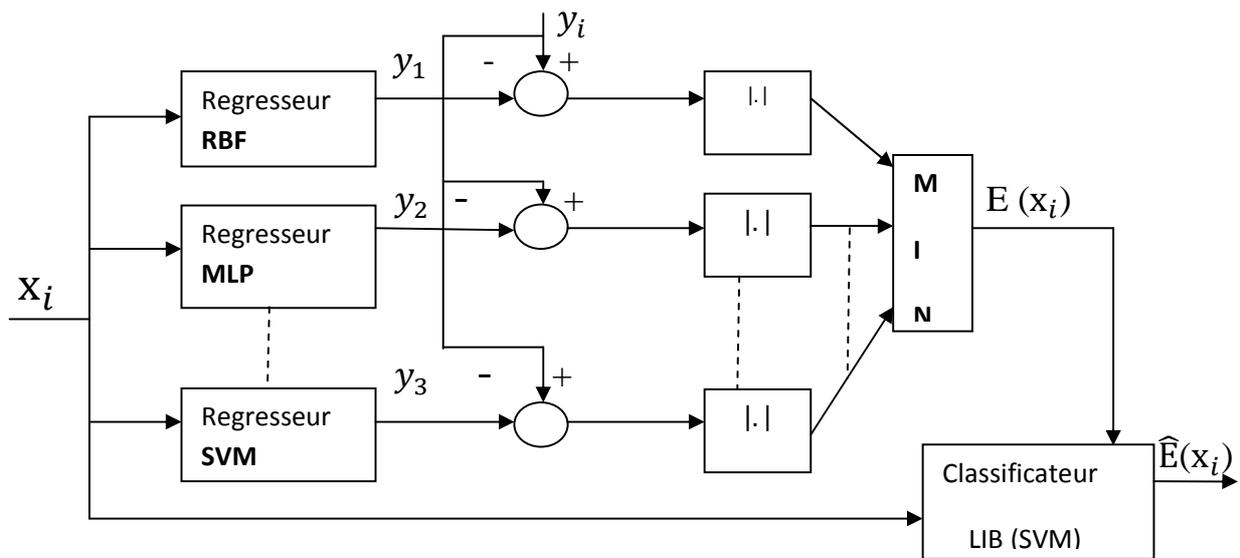


Fig 4 Schéma bloc de la méthode de classification

La méthode de classification [01] commence par l'établissement des résultats des trois régresseurs à base des réseaux de neurones (RBF, MLP) et à base des noyaux (SVM) sur l'ensemble d'apprentissage pour fixer les paramètres des modèles et définir l'architecture des régresseurs.

Les réponses y_1 du régresseur RBF et y_2 du régresseur MLP et y_3 du régresseur SVM établies sur le sous ensemble d'apprentissage sont comparées respectivement avec la sortie désirée disponible pour la phase d'apprentissage et validation.

La comparaison est faite par un critère de valeur absolue :

$E_1 = |y_i \text{ desiree} - y_1|$ Représente l'erreur absolue Pour le modèle RBF

$E_2 = |y_i \text{ desiree} - y_2|$ Représente l'erreur absolue Pour le modèle MLP

$E_3 = |y_i \text{ desiree} - y_3|$ Représente l'erreur absolue Pour le modèle SVM

Les erreurs absolues $E_1 E_2 E_3$ doivent être comparées par le critère de valeur minimale.

Le meilleur régresseur parmi l'ensemble des régresseurs est identifié selon la valeur minimum de l'erreur absolue.

$$E = \min (E_1, E_2, E_3)$$

Après avoir déterminé l'erreur absolue minimale, le meilleur régresseur pour cet échantillon correspond à cette erreur.

Pour qu'on puisse faire un vecteur des classes on doit associer à chaque régresseur (MLP, RBF, SVM) une classe (1 ou 2 ou 3).

Le régresseur RBF correspond à la classe 1

Le régresseur MLP correspond à la classe 2

Le régresseur SVM correspond à la classe 3

Pour l'apprentissage du classificateur SVM les entrées sont les X_i (sous ensemble d'apprentissage) et les sorties désirées sont les classes correspondantes précédemment établies.

Le classificateur doit être validé sur le sous ensemble de validation, les entrées sont les X_i (sous ensemble de validation) et les sorties sont les classes correspondantes.

Le modèle de classification maintenant établi est capable d'associer un vecteur des classes à des matrices d'entrées X_i (sous ensemble de test), alors on peut identifier le meilleur régresseur pour chaque échantillon de l'entrée.

L'erreur quadratique moyenne normalisée globale doit être évidemment inférieure à l'erreur de chaque régresseur individuel ce qui amène forcément à une amélioration des résultats de régression.

4-3. Résultats de classification (base DIESEL)

Dans le tableau 4.1 nous illustrons les résultats globaux des trois régresseurs utilisés avec la base DIESEL, le tableau est constitué de quatre lignes.

La première ligne représente les trois régresseurs utilisés plus une case globale pour illustrer les résultats de classification.

La deuxième ligne représente l'erreur quadratique moyenne normalisée correspondante à chaque modèle plus l'erreur quadratique moyenne normalisée globale, la méthode de calcul de cette erreur est détaillée dans le chapitre 2.

La troisième ligne représente le pourcentage du gain en erreur pour chaque régresseur par rapport à l'erreur globale calculée par la formule suivante.

$$(\text{Gain en erreur}) \% = \left(\frac{\text{NMSE}_{\text{test du modèle}} - \text{NMSE}_{\text{globale}}}{\text{NMSE}_{\text{test du modèle}}} \right) \times 100$$

Ce gain donne une idée sur l'amélioration de l'erreur d'un modèle par rapport à l'erreur globale de classification.

La quatrième ligne représente le taux de classification pour chaque régresseur estimé par le classificateur SVM.

Dans le cas de la base DIESEL 15.17% des échantillons de la base de données doivent être utilisés avec le régresseur RBF, 51.78% des échantillons de la base de données doivent être utilisés avec le régresseur MLP, 33.03% des échantillons de la base de données doivent être utilisés avec le régresseur SVM.

Remarque : cette interprétation des résultats de la base DIESEL peut suffire pour interpréter les résultats des autres bases de données (JUS, VIANDE, WINE) à cause de la ressemblance des tableaux.

model	RBF	MLP	SVM	global
NMSEtest	0.3345	0.3564	0.3409	0.3075
(Gain en erreur) %	8.07	13.72	9.79	-
(Taux de classification) %	15.17	51.78	33.03	-

Tableau 4.1 (resultats de classification)DIESEL

4-4. Résultats de classification (base JUS)

Dans le tableau suivant nous illustrons les résultats globaux des trois régresseurs utilisés avec la base JUS.

Pour l'interprétation des résultats (voir Résultats de classification (base DIESEL))

model	RBF	MLP	SVM	global
NMSEtest	0.1712	0.3188	0.1732	0.1375
(Gain en erreur) %	19.68	56.86	20.61	-
(Taux de classification) %	41.17	10.29	48.52	-

Tableau 4.2 (resultats de classification)JUS

4-5. Résultats de classification (base VIANDE)

Dans le tableau suivant nous illustrons les résultats globaux des trois régresseurs utilisés avec la base VIANDE.

Pour l'interprétation des résultats (voir Résultats de classification (base DIESEL))

model	RBF	MLP	SVM	global
NMSEtest	0.0065	0.0134	0.0036	0.0035
(Gain en erreur) %	46.15	78.88	2.77	-
(Taux de classification) %	32.55	18.60	48.83	-

Tableau 4.3 (resultats de classification)VIANDE

4-6. Résultats de classification (base WINE)

Dans le tableau suivant nous illustrons les résultats globaux des trois régresseurs utilisés avec la base WINE.

Pour l'interprétation des résultats (voir Résultats de classification (base DIESEL))

model	RBF	MLP	SVM	global
NMSEtest	0.0045	0.0641	0.0054	0.0038
(Gain en erreur) %	15.55	94.07	29.62	-
(Taux de classification) %	55.66	13.33	30.00	-

Tableau 4.4 (resultats de classification)WINE

4-7. Conclusion

Dans ce chapitre nous avons traité le problème fondamental de ce mémoire, il s'agit de choisir le meilleur régresseur parmi un ensemble des régresseurs dont les résultats sont déjà établis.

Cette méthode de classification améliore les résultats et minimise sensiblement l'erreur quadratique moyenne normalisée globale sur toutes les bases de données utilisées pour la validation de cette méthode.

Le degré d'amélioration des résultats dépend d'une part de la distribution des bases de données et d'autre part de la structure et des paramètres de l'ensemble des régresseurs.

Conclusion générale et
perspectives

Conclusion générale et perspectives

Dans l'introduction de ce travail nous avons posé un problème lié au choix du meilleur régresseur parmi l'ensemble des régresseurs utilisés, dans le but d'améliorer les résultats de la régression en utilisant une méthode de classification.

Par exemple les résultats de l'un des modèles (RBF, MLP, SVM) sont établis sur l'une des bases de données (DIESEL, JUS, VIANDE, WINE), des échantillons de la base utilisés peuvent donner leurs meilleures réponses avec l'un des régresseur, par contre les autres échantillons de la même base donnent leurs meilleurs résultats avec un autre régresseur, donc la classification des différents régresseurs était indispensable, la méthode de classification a montré une grande utilité vu l'amélioration sensible des résultats de régression par minimisation de l'erreur quadratique moyenne normalisée.

Alors nous avons abouti relativement à résoudre le problème de classification, le degré de résolution d'un tel problème dépend d'une part du modèle en vigueur et d'autre part de la base de données utilisée.

Une perspective de ce travail consiste à chercher dans l'amélioration des résultats avec l'utilisation de la sélection des variables par l'information mutuelle, afin de réduire la redondance et la colinéarité entre les variables d'entrée, ou aussi par l'augmentation de nombre et de types des régresseurs utilisés.

BIBLIOGRAPHIE

- [01] Lorenzo. Bruzzone ,Farid .Melgani (Robust Multiple Estimator Systems FOR The Analysis Of Biophysical Parameters From Remotely Sended Data).2005
- [02] N. Benoudjit, D. Francoi, M. Meurens,M. Verleysen (Spectrophotometric Variable Selection By Mutuel Information).2004
- [03] BUHMANN M. D., (*Radial Basis Functions*), Cambridge University press, Cambridge, 2003
- [04] BROOMHEAD D., LOWE D., (Multivariate functional interpolation and adaptative networks), *Complex Systems*, vol. 2,1988.
- [05] HOERL A., (Application of ridge analysis to regression problems), *Chemical Engineering Progress*, vol. 58, p. 54-59, 1962.
- [06] ORR M. J. L., (Regularization in the Selection of Radial Basis Function Centers), *Neural Computation*, vol. 7, p. 606-623, 1995.
- [07] ORR M. J. L., (Regularization in the Selection of Radial Basis Function Centers), *Neural Computation*, vol. 7, p. 606-623, 1995.
- [08] Cortes, C. and Vapnik, V. (Support vector networks. Machine Learning). 1995
- [09] M.T. Hagan, H.B. Demuth, M. Beale, (Neural Network Design), PWS Publishing Company, 1995.
- [10] J.C. Principe, N.R. Euliano, W.C. Lefebvre, (Neural and Adaptive Systems Fundamentals through Simulations), Wiley, 2000.
- [11] Simon Haykin, (Neural Networks: A Comprehensive Foundation), IEEE Press, 1994.
- [12] J.A. Freeman, D.M. Skapura, (Neural Networks: Algorithms, Applications, and Programming Techniques), Addison- Wesley, 1992.

- [13] R.P. Lippmann, (An Introduction to Computing with Neural Nets), *IEEEASSP Magazine*, pp. 4-22, avril 1987.
- [14] R. Krishnapuram, J.M. Keller, (A Possibilistic Approach to Clustering), *IEEE Transactionson Fuzzy Systems*, vol. no. 2, p. 98-110, 1993.
- [15] Bernd Fritzke, (A Growing Neural Gas Network Learns Topologies), *Advances in Neural Information Processing Systems 7*, G. Tesauro, D.S. Touretzky et T.K. Leen (editeurs), MIT Press, Cambridge MA, 1995.
- [16] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, D.B. Rosen, (Fuzzy ARTMAP: Neural network architecture for incremental supervised learning of analog multidimensional Maps), *IEEE Transactions on Neural Networks*, vol. 3, no. 5, p. 698-713, 1992.
- .
- [17] POWELL M. J. D., (Radial basis functions for multivariable interpolation: a review), *Algorithms for approximation*, New York, NY, USA, Clarendon Press, p. 143-167, 1987.
- [18] BROOMHEAD D., LOWE D., (Multivariate functional interpolation and adaptative networks), *Complex Systems*, vol. 2, 1988
- [19] MOODY J., DARKEN C., (Fast Learning in Networks of Locally-tuned Processing Units), *Neural Computation*, vol. 1, p. 281-294, 1989
- [20] CANU, S. (Machines à noyaux pour l'apprentissage statistique). *Techniques de l'ingénieur*.2007
- [21] Guerneur, Y. and Paugam-Moisy, H. (Apprentissage Automatique), chapitre Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines, pages 109–138.1999
- [22] POWELL M. J. D., (Radial basis functions for interpolation: a review), *Algorithms for approximation*, New York, NY, USA, Clarendon Press, p. 143-167, 1987.

- [23] MOODY J., DARKEN C., (Fast Learning in Networks of Locally-tuned Processing Units), *Neural Computation* V ol. 1, p. 281-294, 1989.
- [24] HOERL A., (Application of ridge analysis to regression problems), *Chemical Engineering Progress*, vol. 58, p. 54-59, 1962.
- [25] FOGELMAN SOULIÉ F., LAMY B., VIENNET E., (Multi-Modular Neural Networks Architectures for Pattern Recognition : Applications in Optical Characters Recognition and Human Face Recognition), *Int. J. Pattern Recognition and Artificial Intelligence*, vol. 7, n° 4, p. 721-755, 1993, Extended version as Tech. Report 827, LRI 1993.
- [26] COLLOBERT R., BENGIO S., MARIÉTHOZ J., Torch : a modular machine learning software library, Rapport n° 02-46, IDIAP, 2002.
- [27] ORR M., HALLAM J., MURRAY A., LEONARD T, (Assessing RBF networks using DELVE), *International Journal of Neural Systems*, vol. 10, p. 397-415, 2000.
- [28] LeCun, Y. Modèles connexionnistes de l'apprentissage (connectionist Learning models). PhD thesis, University P. et M. Curie (Paris 6), June 1987.
- [29] M.D. Richard, R.P. Lippman, (Neural network classifiers estimates Bayesian a posteriori probabilities), *Neural Computation*, 1991.
- [30] J.S.Bridle, (Probabilistic interpretation of feed forward classification network outputs, with relationships to statistical pattern recognition), *Neuro-computing: Algorithms, Architectures and Applications*, 1989.
- [31] C.L.Liu, K.Nakashima, H.Sako H.Fujisawa, (Handwritten digit recognition using state-of-the-art techniques), IEEE, 2002.
- [32] J.Liu, P.Gader, (Neural networks with enhanced outlier rejection ability for off -line handwritten word recognition), *Pattern Recognition*, 2002.
- [33] V. N. Vapnik, (Statistical Learning Theory), New York, Wiley, 1998

- [34] A. Smola. and B. Scholkopf, (A Tutorial on support vector regression), NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, university of London, 1998
- [35] Oliver. Bousquet, (introduction au support vector machines SVM), centre De mathématiques appliquées, école polytechnique de Palaiseau, 2001
- [36] F. Moutarde, (introduction aux réseaux de neurones), centre de robotique, École des mines de paris, Avril 2007
- [37] C. Wei Hsu, C.C. Chang and C. Jen Lin, (A Practical Guide to Support Vector Classification). National Taiwan University Taipei, Taiwan.2009

Résumé

La régression non linéaire peut se faire avec des modèles à base de réseaux de neurones ou avec des modèles à base des noyaux, on utilise dans ce travail les modèles les plus connus : le réseau à fonctions radiales de base (Radial Basis Functions) RBF, le perceptron multi couches (Multi-Layer Perceptron) MLP et les machines à vecteurs de support (Support Vector Machine) SVM, après avoir établi les résultats des trois modèles utilisées dans la régression (RBF, MLP, SVM) un problème fondamental se pose, il s'agit du choix du meilleur régresseur.

Pour résoudre ce problème on a établi une méthode qui consiste à faire apprendre à un classificateur SVM par les matrices d'entrée d'une base de données, la sortie représente la meilleure réponse parmi trois régresseurs (RBF, MLP, SVM) exprimée sous forme de classes afin de fixer les paramètres du modèle classificateur, une fois le classificateur est validé, il devient capable de faire la classification sur un ensemble de données qui n'est pas précédemment utilisée ni pour l'apprentissage ni pour la validation, la classification ici est l'action d'associer à chaque échantillon de la base de données utilisée un régresseur parmi les modèles mis en jeu (RBF ou MLP ou SVM).

Mots clefs : réseaux de neurones artificiels, les machines à vecteurs de support, régression non linéaire, classification

Abstract

The nonlinear regression can be done with models containing neural networks or with models containing the cores, we uses in this work the most known models: the Radial Basis Functions (RBF), the perceptron multi layers (MLP), and the Support Vector Machine (SVM), after having established the results of the three models used in regression (RBF, MLP, and SVM) a fundamental problem arise, it acts of the choice of the best regresseur.

To solve this problem we established a method consists in making learn a classifier SVM by the matrices from entry of a database and the exit who represents the best answer among three régresseurs (RBF, MLP, and SVM) expressed in the form of classes in order to fix the parameters of the classifier, once the classifier is validated, it becomes able to make classification on a field of data which is previously used neither for the training nor for the validation, classification here is the action to associate with each sample of the database used a regresseur among the concerned models (RBF, MLP, SVM),

Key words: artificial neural networks, support vector machine, nonlinear regression, classification