Université du Québec
École
de technologie
supérieure

Application du réseau de neurones RBFN à l'estimation des coûts de logiciels

Samir Mbarki, Ali Idri, Alain Abran

CARI 2004 Tunis 22-25 nov. 2004

Liste des sujets

- Introduction
- Réseau de neurones RBNF
- Application du RBNFà l’estimation de coûts – Cas COCOMO I
- Analyse des résultats
- Discussion

Introduction

- Modèles paramétriques d'estimation nécessitent:
  - La connaissance de la fonction reliant l'effort et les facteurs de coûts
  - Une forme explicite – mathématique – avec variable dépendantes et variables indépendantes
  - Doit être calibré dans un nouveau contexte

Introduction

Modèles non-paramétriques:

- Certains reliés à l'intelligence artificielle, raisonnement à base de cas, arbres de décision
  - Aptitude à représenter n'importe quelle fonction (approximateurs universels)
  - Capacité d'apprentissage automatique à partir de données historiques

Travaux antérieurs

- Réseau de neurones:
  - Perceptron multi-couches
  - Fonction sigmoide pour activation du réseau
  - Algorithme de retro-propagation
  - Validation avec la base de données COCOMO I
- Interprétation avec:
  - Perceptron à 3 couches
  - Réseau transformé en un système à base de règles floues

Travaux en cours

- Réseau RBFN: Radial Basis Function Network
- Étude de l'impact du choix des différents paramètres de l'architecture du réseau RBFN sur la précision des estimation

s_i

Réseau RBNF

- Un réseau ‘feed-forward’ à trois couches:
  - Couche d’entrée
  - Couche cachée
  - Couche de sortie
  - Fonction d’activation = radiale
    - C_i = centre de la fonction gaussienne
    - $Sigma_i$ = rayon de la fonction gaussienne
    - Distance euclidienne entre vecteur d’entére x et vecteur centre c_i

$$y_i(x) = \exp\left(-\|x - c_i\|^2 / s_i^2\right)$$

Réseau RBFN

- Sortie Z du réseau RBFN déterminée par l'équation où:
  - y_i = activation du $i^{ème}$ neurone caché de l'équation d'activation

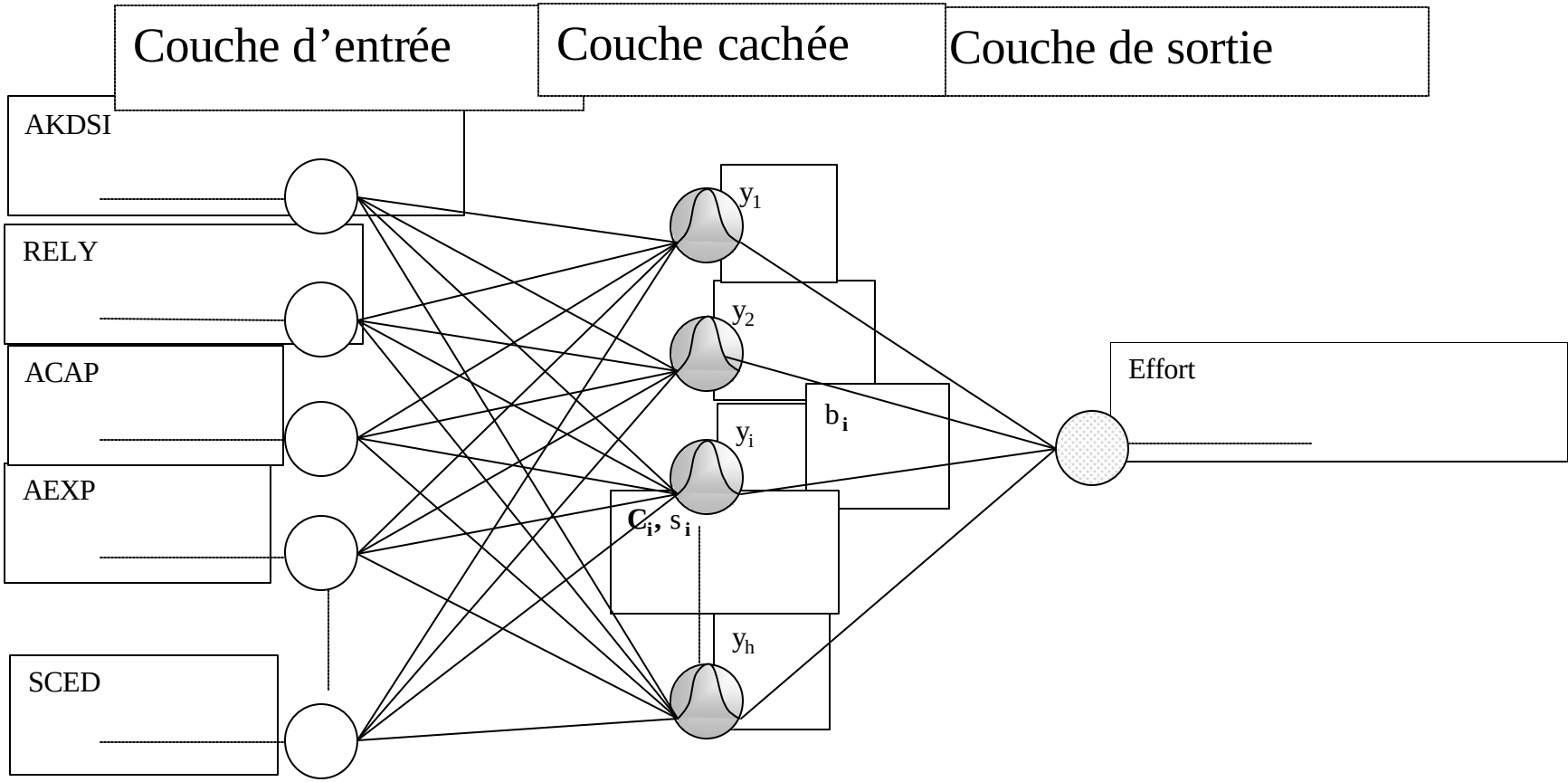
$$Z = \sum_{i=1}^{h} b_i y_i$$

Estimation avec RBFN

- Étape 1: détermination des neurones de la couche cachée et de leurs paramètres

- Étape 2: Détermination des poids entre la couche cachée et la couche de sortie
  - Réalisé par une technique d'apprentissage surpevisé comme la méthode de descente du gradient ou méthode de régularisation

Architecture d'un réseau RBFN



Méthode d'apprentissage

- *Adaptive Pattern Classifier* – APC-III – Hwang et al.
  - Algorithme de type *one-pass* – converge rapidement
  - Adaptive – nombre de cluster non fixé
  - Calcule un Rayon R_0 proportionnel à la moyenne des distance minimales entre les projets logiciels
    - N = nombre de projets dans la base de données
    - Alpha = constante à choisir

$$R_0 = a / N \sum_{i=1}^{N} \min_{i \neq j} \left(\left\| P_i - P_j \right\| \right)$$

Suite du RBFN

- Si R_0 est très petit le nombre de clusters générés sera élevé, et vise versa
- 2ième étape:
  - Algorithme cherche pour chaque P_i le premier cluster C_j dont le centre c_j a une distance euclidienne avec P_i inférieur ou égale au rayon R_0
    - Accompagné d'un déplacement du centre c_j
  - Si P_i n'appartient à aucun cluster, un nouveau cluster est cré dont le centre est P_i

Résultats avec COCOMO I

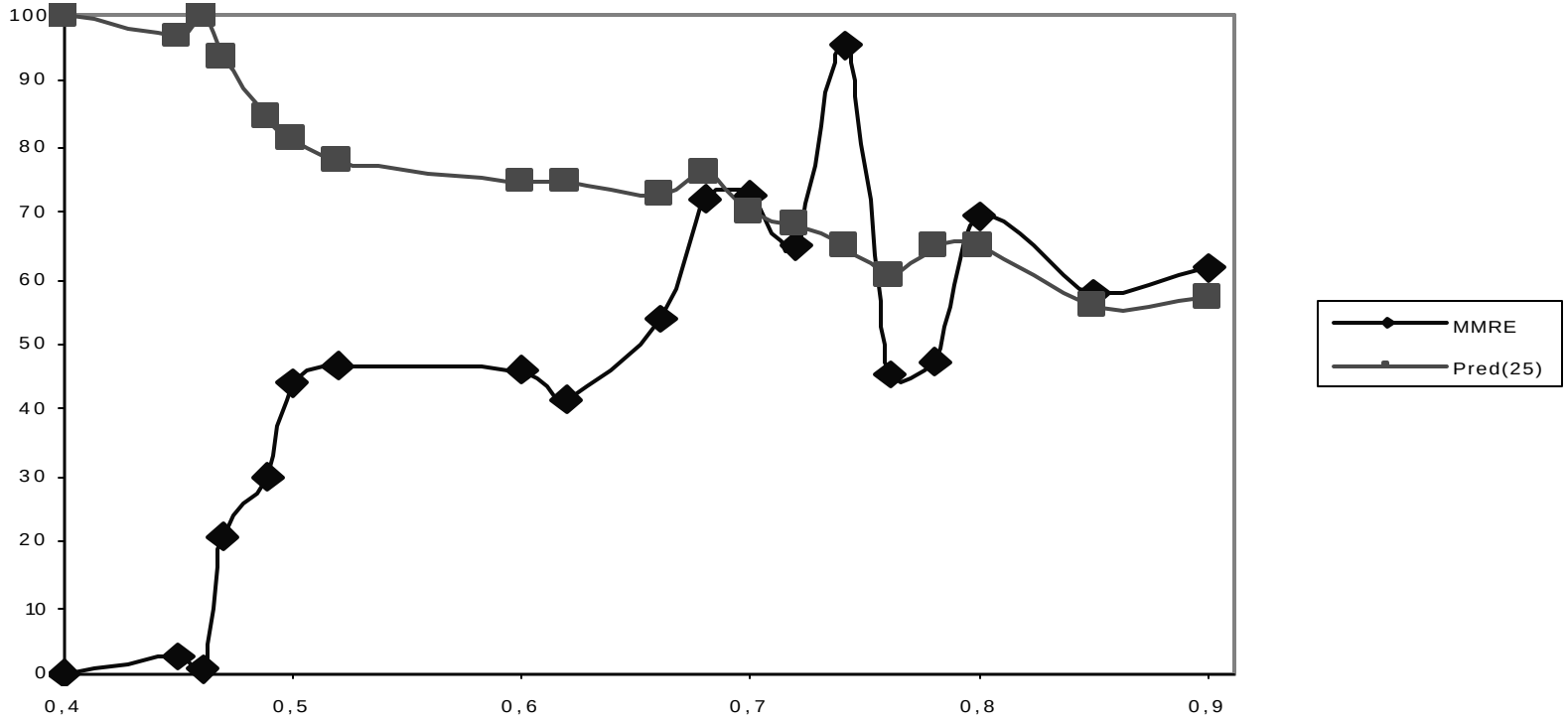
- COCOMO I 1981:
  - 63 projets
  - Décrits avec 13 attributs
  - Donc 13 neurones en entrée et 1 seul en sortie
  - Neurones en couche caché = nombre de clusters générés par APC-III

Résultats avec COCOMO I

- Précision évaluée par:
  - MMRE: mean magnitude of relative error
  - Pred (0,25) = % des projets avec MRE plus petit que 25% où

$$MRE = \left| (Effort_{reel} - Effort_{estimé}) / Effort_{réel} \right|$$

Variation du MMRE et Pred(0,25) en fonction de alpha



Résultats avec COCOMO I

- Précisions des estimations meilleures quand:
  - Alpha est inférieur ou égale à 0,49:
    - MMRE plus petit que 30% et Pred (0,25) plus grand que 70%
  - Quand alpha est supérieure à 0,49:
    - MMRE devient plus grand
    - Mais Pred(0,25) est toujours satisfaisant

Résultats avec COCOMO I

- La précision des estimations varie non seulement en fonction du paramètre alpha (nombre de neurones de la face cachée) mais également de la variance sigma utilisée par la fonction d'activation des neurones de la couche cachée
  - Stratégie adoptée pour le choix de sigma: en fonction du rayon R_0

Résultats avec COCOMO I

- Meilleurs résultats d'estimation si $sigma_i$ est au voisinage de R_0
- Si loin de R_0, les estimations sont moins précises

Sommaire

- Étude de l'application du réseau de neurones RBFN à l'estimation des coûts de logiciel
- Processu d'apprentissage en 2 étapes:
  - Détermination des paramètres de la couche cachée avec l'algorithme de classification non supervisée APC-III
  - ALgorithme de descente de gradient appliqué pour la determination des poides entre couche cachée et celle de sortie
- Étude empirique avec la base de données COCOMI I

Questions

?