III - Réseaux Multicouches (Partie 1)

PMC: Perceptron Multi-couche
(MLP: Multi Layer Perceptron)

3ème séance :

6) Estimation de l'Espérance
- 6.1) Approche théorique
- 6.2) Cas de la classification

7) Estimation de la Variance
- 7.1) Maximum de vraisemblance (rappel)
- 7.2) Application à la régression par PMC
- 7.3) Calcul de la vraisemblance

Préambule

On se focalise dans ce cours sur les performances attendues lorsque l'on met au point un système de diagnostique automatique. On veut donc connaître les erreurs commises par celui-ci dans tous les cas de fonctionnement possibles.
On s'intéresse donc à étudier l'approximation obtenue quand on remplace une réalité (fonction réelle sous jacente aux données) par une modélisation mathématique (fonction mathématique choisie par le modélisateur).
Le but de ce cours est de présenter les résultats théoriques des réseaux de neurones en se plaçant dans ce contexte. Ces résultats établissent la capacité des PMC à approximer, à partir de l'ensemble d'apprentissage, espérances et variances conditionnelles qui permettent de caractériser la nature du phénomène étudié.

6 - Estimation de l'Espérance

6.1 - Approche théorique (1)

- Un système réel est décrit comme un vecteur aléatoire $\boldsymbol{x}$ régi par une densité de probabilité $P(\boldsymbol{x})$.
- A chaque vecteur $\boldsymbol{x}$ est associée une valeur $\boldsymbol{d}$ suivant la loi conditionnelle $P(\boldsymbol{d/x})$.
- Généralement, l'apprentissage consiste à déterminer les paramètres W^* pour une famille de fonctions $\{F(x,W)\}$ qui minimisent l'**erreur d'apprentissage** sur un ensemble de N exemples : $App = \{(\boldsymbol{x}_1, \boldsymbol{d}_1), (\boldsymbol{x}_2, \boldsymbol{d}_2), \ldots, (\boldsymbol{x}_N, \boldsymbol{d}_N)\}$

$$E_{App} = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{d}_i - F\left(\boldsymbol{x}_i, W\right)\right)^2$$

Mais la fonction $F(x,w^*)$ ne minimise pas nécessairement l'**erreur en généralisation** :

$$E_{géné} = \iint \left(\boldsymbol{d} - F\left(\boldsymbol{x}, W^*\right)\right)^2 p\left(\boldsymbol{x}\right).p\left(\boldsymbol{d/x}\right)d\boldsymbol{x}d\boldsymbol{d}$$

commentaire

6.1 - Approche théorique (2)

Soit : $E(\boldsymbol{d/x}) = \int \boldsymbol{d}.p(\boldsymbol{d/x})d\boldsymbol{d}$ on démontre le résultat suivant :

$$Min\, E_{géné} \Leftrightarrow Min \int \left(E(\mathrm{d}/\boldsymbol{x}) - F(\boldsymbol{x},\boldsymbol{W})\right)^2 p(\boldsymbol{x})d\boldsymbol{x}$$

Remarque :

La meilleure fonction en généralisation est celle qui approxime au mieux $E(\boldsymbol{d/x})$

Démonstration :

$$E_{géné} = \iint \left(\boldsymbol{d} - E(\boldsymbol{d}/\boldsymbol{x}) + E(\boldsymbol{d}/\boldsymbol{x}) - F(\boldsymbol{x},W)\right)^2 p(\boldsymbol{x},\boldsymbol{d})d\boldsymbol{x}d\boldsymbol{d}$$

$$= \underbrace{\iint \left(\boldsymbol{d} - E(\boldsymbol{d}/\boldsymbol{x})\right)^2 p(\boldsymbol{x},\boldsymbol{d})d\boldsymbol{x}d\boldsymbol{d}}_{(1)} + \underbrace{\int \left(E(\boldsymbol{d}/\boldsymbol{x}) - F(\boldsymbol{x},W)\right)^2 p(\boldsymbol{x})d\boldsymbol{x}}_{(2)}$$

$$+ 2\underbrace{\iint \left(\boldsymbol{d} - E(\boldsymbol{d}/\boldsymbol{x})\right)\left(E(\boldsymbol{d}/\boldsymbol{x}) - F(\boldsymbol{x},W)\right) p(\boldsymbol{x},\boldsymbol{d})d\boldsymbol{x}d\boldsymbol{d}}_{(3)}$$

$$(3) = 2\int \left(E(\boldsymbol{d}/\boldsymbol{x}) - F(\boldsymbol{x},W)\right)\Big[\underbrace{\int \left(\boldsymbol{d} - E(\boldsymbol{d}/\boldsymbol{x})\right) p(\boldsymbol{d}/\boldsymbol{x})d\boldsymbol{d}}_{=}\Big] p(\boldsymbol{x})d\boldsymbol{x}$$

$$\int \boldsymbol{d}.p(\boldsymbol{d}/\boldsymbol{x})d\boldsymbol{d} - E(\boldsymbol{d}/\boldsymbol{x}) = 0$$

Donc $\boldsymbol{E}_{géné}$ = (1) + (2) et comme (1) est indépendant de W, on tire :

$$Min\, E_{géné} \Leftrightarrow Min\ (2)$$

commentaire

6.1 - Approche théorique (3)

But de l'apprentissage : $F(\boldsymbol{x}, W^*) \cong E(\boldsymbol{d}/\boldsymbol{x})$ … Mais $E_{généré}$ et $E(\boldsymbol{d}/\boldsymbol{x})$ sont inconnus

Dans la pratique, on cherche à estimer $E(\boldsymbol{d}/\boldsymbol{x})$ à partir d'un ensemble d'apprentissage : $App = \{(\boldsymbol{x}_1, \boldsymbol{d}_1), (\boldsymbol{x}_2, \boldsymbol{d}_2), \ldots, (\boldsymbol{x}_N, \boldsymbol{d}_N)\}$

L'apprentissage consiste à minimiser le « risque empirique » :

$$E_{App} = \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{d}_i - F\left(\boldsymbol{x}_i, W\right)\right)^2$$

La solution obtenue dépend alors de l'ensemble d'apprentissage : $F(\boldsymbol{x}, W^*/App)$

Remarque : Ce résultat reste valable dans le cas de l'erreur quadratique pondérée :

$$E_{généré} = \iint \frac{\left(\boldsymbol{d} - F\left(\boldsymbol{x}, W\right)\right)^2}{\sigma^2(\boldsymbol{x})} p(\boldsymbol{x}) p(\boldsymbol{d} / \boldsymbol{x}) d\boldsymbol{x}\, d\boldsymbol{d}$$

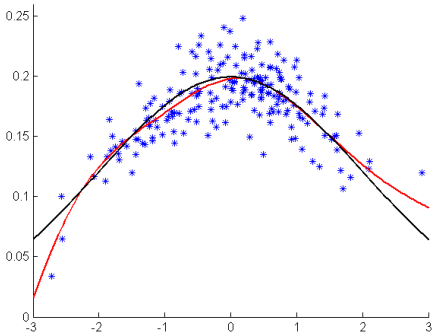
Ainsi,

$$Min\, E_{généré} \Leftrightarrow Min \int \left(E(\boldsymbol{d} / \boldsymbol{x}) - F(\boldsymbol{x}, W)\right)^2 \frac{p(\boldsymbol{x})}{\sigma^2(\boldsymbol{x})} d\boldsymbol{x}$$

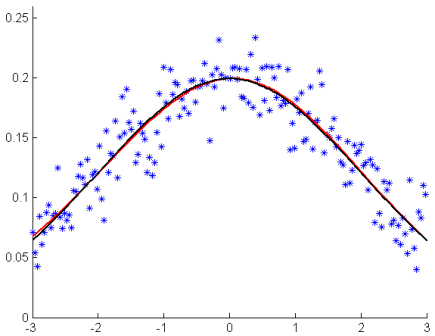
6.1 - Approche théorique (4)

Comparaison de la répartition des données dans la base d'apprentissage :

Répartition naturelle des données



Répartition « équilibrée » des données



6.1 - Approche théorique (5)

Conditions d'approximation :

- Pour les PMC, la précision de l'approximation de $E(\boldsymbol{d}/\boldsymbol{x})$ dépend de l'architecture choisie
- Réalisation d'un codage des données approprié (par exemple une normalisation)
- L'échantillon d'apprentissage doit être de taille N suffisante et représentatif

Remarque :

- La précision de cette approximation pour un point $\boldsymbol{x}$ fixé diffère selon la répartition des exemples $p(\boldsymbol{x})$.

$p(\boldsymbol{x})$ grand ⇔ $F(\boldsymbol{x},W)$ proche de $E(\boldsymbol{d}/\boldsymbol{x})$

$p(\boldsymbol{x})$ petit ⇔ $F(\boldsymbol{x},W)$?

En pratique les régions sous représentées dans l'ensemble d'apprentissage sont plus ou moins ignorées.

commentaire

6.2 - Cas de la classification (1)

Cas de 2 classes

L'ensemble d'apprentissage est réparti en 2 classes. On code les réponses désirées :

$$\begin{cases} \boldsymbol{d} = \boldsymbol{a} & \textit{si} \quad \boldsymbol{x} \in \boldsymbol{C_1} \\ \boldsymbol{d} = \boldsymbol{b} & \textit{si} \quad \boldsymbol{x} \in \boldsymbol{C_2} \end{cases}$$

$F(\boldsymbol{x}, W)$ « *approxime* » : $E(\boldsymbol{d}/\boldsymbol{x}) = a \times p(C_1/\boldsymbol{x}) + b \times p(C_2/\boldsymbol{x})$

Interprétation :

Si $a = 1$ *et* $b = 0$ *(alors)* $E(\boldsymbol{d}/\boldsymbol{x}) = p(C_1/\boldsymbol{x})$

Si $a = 1$ *et* $b = -1$ *(alors)* $E(\boldsymbol{d}/\boldsymbol{x}) = p(C_1/\boldsymbol{x}) - p(C_2/\boldsymbol{x})$

Fonction discriminante de Bayes

Généralisation à Q classes :

Le vecteur qui code la classe k (C_k) a toutes ses composantes égales à b exceptée la *kième* qui prend la valeur a.

$$E(\boldsymbol{d}/\boldsymbol{x}) = axp(C_k/\boldsymbol{x}) + \sum_{q<>k}^{Q} bxp(C_q/\boldsymbol{x})$$

commentaire

6.2 - Cas de la classification (2)

Approximation des probabilités : cas du codage a=1 b=0 :

- En général les sorties du réseau ne donnent pas de vraies probabilités

$$\sum_{k=1}^{Q} F_k(\boldsymbol{x}, W) \neq 1$$

- Il est cependant possible d'obtenir des probabilités en utilisant la fonction de transfert « *softmax* » pour les cellules de sortie :

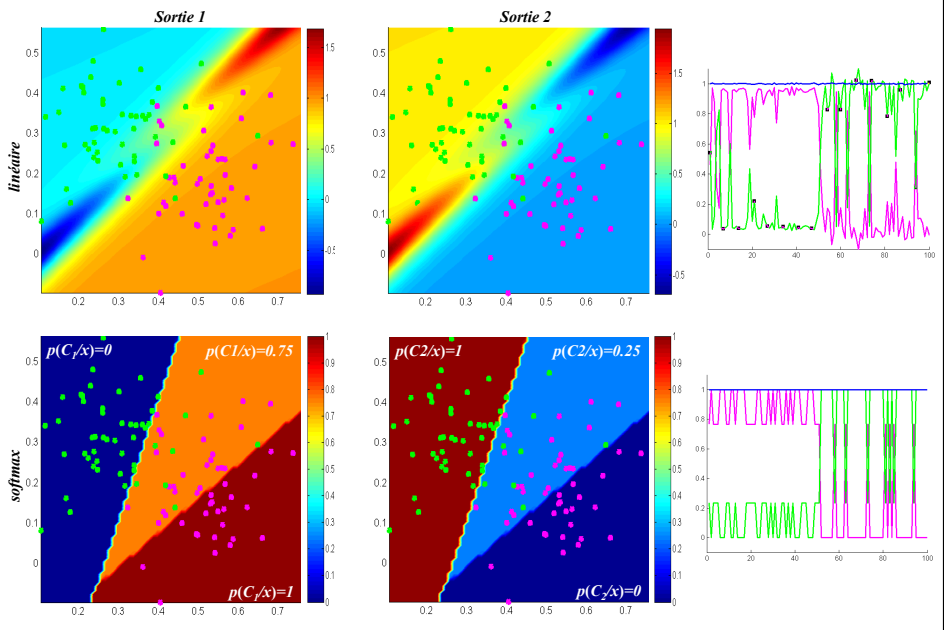
$$y_k = f_k(A_k) = \frac{e^{A_k}}{\sum_{q \in \text{couche de sortie}}^{Q} e^{A_q}}$$

on a bien : $\sum_k y_k = 1$

et : $y_k >= 0 \quad \forall\, k$

commentaire

6.2 - Cas de la classification (3)



commentaire

7 - Estimation de la Variance

7.1 - Maximum de Vraisemblance (rappel)

Soit un ensemble d'observations $\{x_1,...,x_i,...,x_N\}$ réalisations indépendantes d'une v.a. X identiquement distribuée selon une loi de probabilité admettant θ comme paramètre de fonction de densité : $f(x, \theta)$

On appelle « vraisemblance de θ » selon l'échantillon $\{x_1,...,x_i,...,x_n\}$ le nombre :

$$L(x_1,...,x_i,...,x_N,\theta) = f(x_1,\theta) \times f(x_2,\theta), \times ... \times f(x_N,\theta) = \prod_{i=1}^{N} f(x_i,\theta)$$

- => Trouver la valeur du paramètre θ qui rend maximum cette vraisemblance.

$$Max_{\theta} \prod_{i=1}^{N} f(x_i,\theta)$$

- **Estimateur par Maximum de Vraisemblance** noté EMV. L'EMV peut ne pas exister, et peut ne pas être unique.
- => Choisir analytiquement la fonction f.

7.2 - Application à la régression par PMC

$App = \{(x_i, d_i); i = 1, 2, ..., N\}$ $(x_i \in R^p, d_i \in R^q)$ de probabilités $p(x,d)$.

- On suppose que toutes les observations (x_i, d_i) sont indépendantes. Ainsi :

$$P(App) = P((x_1,d_1),(x_2,d_2),....,(x_N,d_N))$$
$$= p(x_1,d_1) \times p(x_2,d_2) \times ... \times p(x_N,d_N)$$

$$P(App) = \prod_{i=1}^{N} p(x_i,d_i) = \prod_{i=1}^{N} p(d_i/x_i)p(x_i)$$

- D'autre part, si on fait l'hypothèse : $x \longrightarrow d = \mu(x) + \varepsilon(x)$ où $\mu(\ )$ est une fonction bien déterminée et $\varepsilon(x)$ une variable aléatoire qui suit une loi normale $N(0, \sigma(x)I)$. Ainsi :

$$p(d/x) = \frac{1}{\sigma(x)\sqrt{2\Pi}} exp\left[-\frac{1}{2}\frac{\|d-\mu(x)\|^2}{\sigma^2(x)}\right]$$

$$P(x_i,d_i) = \frac{1}{\left(\sqrt{2\Pi}\right)^N} \prod_{i=1}^{N} \frac{1}{\sigma(x_i)} \exp\left[-\frac{1}{2}\frac{\|d_i-\mu(x_i)\|^2}{\sigma^2(x_i)}\right] p(x_i)$$

Loi Gaussienne $N(\mu(x), \sigma^2(x))$

7.3 - Calcul de la vraisemblance (1)

- Soit une famille de fonctions : $F = \{ F(\boldsymbol{x}, W) \}$
- On désire approximer μ par une fonction $F(\boldsymbol{x}, \boldsymbol{W})$ particulière de la famille F. Ainsi :

$$P(App,W) = \frac{1}{\left(\sqrt{2\Pi}\right)^N} \prod_{i=1}^{N} \frac{1}{\sigma(\boldsymbol{x}_i)} exp\left[-\frac{1}{2} \frac{\|\boldsymbol{d}_i - F(\boldsymbol{x}_i,W)\|^2}{\sigma^2(\boldsymbol{x}_i)} \right] p(\boldsymbol{x}_i)$$

Posons : $Err(W) = - ln\ P(App,W)$

$$Err(W) = \frac{1}{2} \sum_{i=1}^{N} \frac{\|\boldsymbol{d}_i - F(\boldsymbol{x}_i,W)\|^2}{\sigma^2(\boldsymbol{x}_i)} + \frac{1}{2} \sum_{i=1}^{N} ln\sigma^2(\boldsymbol{x}_i) + \frac{N}{2} ln(2\pi) - \sum_{i=1}^{N} lnp(\boldsymbol{x}_i)$$

(4) (5) (6) (Const)

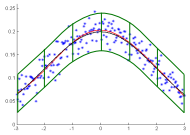
Méthode de maximum de vraisemblance.

Déterminer : $W^* = Arg\ Max_W\ P(App,W)$

$= Arg\ Min_W\ Err(W)$

7.3 - Calcul de la vraisemblance (2)

7.3.1 - Cas où $\sigma^2(x) = \sigma^2$ = constante :



Dans ce cas, le terme (5) ne dépend pas de W.

On peut prendre : $Err(W) = \frac{1}{2} \sum_{i=1}^{N} \frac{\|\boldsymbol{d}_i - F(\boldsymbol{x}_i,W)\|^2}{\sigma^2}$

Ainsi : $Min_W Err(W) = Min_W \left(\sum_{i=1}^{N} \|\boldsymbol{d}_i - F(\boldsymbol{x}_i,W)\|^2 \right)$

On retrouve la somme des erreurs quadratiques.

Soit $\boldsymbol{W}^*$ un système de poids qui minimise cette erreur.

Estimation de σ^2 :

En retenant W^*, l'erreur Err devient :

$$Err(W^*) = \frac{1}{2} \sum_{i=1}^{N} \frac{\|\boldsymbol{d}_i - F(\boldsymbol{x}_i,W^*)\|^2}{\sigma^2} + N \ln \sigma$$

La valeur de σ^2 qui minimise $Err(W^*)$ est :

Erreur moyenne quadratique

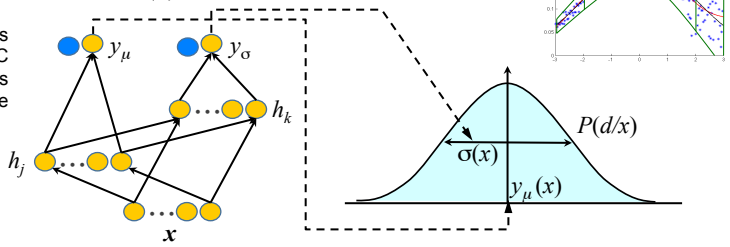
$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{d}_i - F(\boldsymbol{x}_i,W^*)\|^2$$

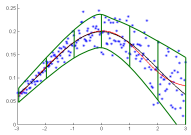
7.3 - Calcul de la vraisemblance (3)

7.3.2 - Cas où $\sigma^2(x)$ dépend de x

⇒ Estimer en plus $\sigma(x)$

On peut alors considérer un PMC avec deux neurones sur la couche de sortie :





- **1er neurone:** entrée : A_μ fonction d'activation : Id sortie : $y_\mu = F_\mu(x,W)$

$$y_\mu = A_\mu \Rightarrow \text{estime } \mu(x)$$

- **2ème neurone:** entrée : A_σ fonction d'activation : exp sortie : $y_\sigma = F_\sigma(x,W)$

$$y_\sigma = e^{A_\sigma} \Rightarrow \text{estime } \sigma^2(x)$$

- *La fonction d'erreur à minimiser par rapport à W devient :*

$$Err(W) = \frac{1}{2}\sum_{i=1}^{N}\frac{\|\boldsymbol{d}_i - F_\mu(\boldsymbol{x}_i,W)\|^2}{\sigma^2(\boldsymbol{x}_i)} + \frac{1}{2}\sum_{i=1}^{N} ln\sigma^2(\boldsymbol{x}_i) \Rightarrow Err(W) = \frac{1}{2}\sum_{i=1}^{N}\frac{(d_i - A_\mu)^2}{e^{A_\sigma}} + \frac{1}{2}\sum_{i=1}^{N} A_\sigma$$

7.3 - Calcul de la vraisemblance (4)

7.3.3 - Algorithme neuronal

On pose :

$$Err_i(W) = \frac{1}{2}\frac{(d_i - A_\mu)^2}{e^{A_\sigma}} + \frac{1}{2}A_\sigma$$

Initialisation de la rétro-propagation du gradient :

- pour le neurone **1** :

$$\frac{\partial Err_i}{\partial A_\mu} = -\frac{(d_i - A_\mu)}{e^{A_\sigma}} \qquad (7)$$

- pour le neurone **2** :

$$\frac{\partial Err_i}{\partial A_\sigma} = -\frac{(d_i - A_\mu)^2}{e^{A_\sigma}} + \frac{1}{2} \qquad (8)$$

L'algorithme se présente alors en 3 phases ...

7.3 - Calcul de la vraisemblance (5)

7.3.3 - Algorithme neuronal (suite)

Phase 1 :

(Initialisation de la moyenne conditionnelle).

On suppose que $\sigma^2(x)$ est constante et on apprend l'erreur quadratique classique. On rétropropage alors uniquement à partir du neurone **1**.

(7) $$\frac{\partial Err_i}{\partial A_\mu} = -\frac{(d_i - A_\mu)}{e^{A_\sigma}}$$ dans ce cas.

Soit W_1 les poids déterminés à la fin de cette phase.

Phase 2 :

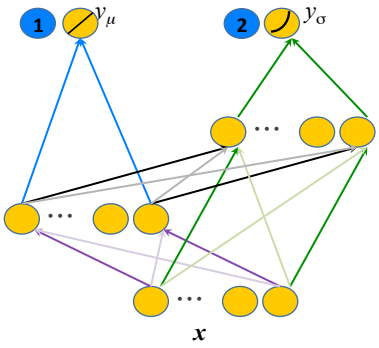
On gèle la sortie du neurone **1** à :

$y_\mu = F_\mu(x_i, W_1)$ pour tout x_i.

On rétropropage alors uniquement à partir du neurone **2** en prenant :

(8) $$\frac{\partial Err_i}{\partial A_\sigma} = -\frac{(d_i - A_\mu)^2}{e^{A_\sigma}} + \frac{1}{2}$$

Soit W_2 les poids déterminés à la fin de cette phase.



Phase 3 :

Minimiser la fonction d'erreur complète en rétropropageant à partir des deux neurones **1** et **2**, en appliquant les formules **(7)** et **(8)**.

7.3 - Calcul de la vraisemblance (6)

7.3.4 - Exemple