Réseaux de neurones

Partie 2

Fonction erreur

• Un système est décrit comme un vecteur aléatoire ***x*** dont les valeurs sont régies par la densité de probabilité $p(\mathbf{x})$.

• A chaque vecteur ***x*** est associée un vesteur ***y*** qui se réalise suivant la loi conditionnelle $p(\mathbf{y}/\mathbf{x})$.

• L'apprentissage consiste à ajuster les paramètres **W,** pour une famille de fonctions $\{F(\mathbf{x},\mathbf{W}\}_w$, en minimisant l'erreur d'apprentissage calculée sur un ensemble de *n* exemples:

$D = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$

$$E_D = \frac{1}{n}\sum_{i=1}^{n} \left\| y_i - F\left(x_i, w\right)\right\|^2$$

Mais la fonction $F(\mathbf{x},\mathbf{W}^*)$ ne minimise pas nécessairement l'erreur en *généralisation.*

$$E_{gene} = \int \left\| y_i - F(x_i, w)\right\|^2 p(x).p(y/x)dxdy$$

Soit :

$$\mu(x) = \int y p(y/x) dy = E(y/x)$$

•On démontre le résultat suivant :

$$Min E_{généré} \Leftrightarrow Min \int \|\mu(x) - F(x,w)\|^2 p(x) dx$$

Remarque

•La meilleure fonction en généralisation est celle qui approxime au mieux μ(x)

Démonstration (cas où $y \in \Re$)

$$E_{généré} = \int (y - \mu(x) + \mu(x) - F(x,w))^2 p(x,y) dxdy$$

$$= \int (y - \mu(x))^2 p(x,y) dxdy + \int (\mu(x) - F(x,w))^2 p(x,y) dxdy$$

(1) (2)

$$+2 \int (y - \mu(x))(\mu(x) - F(x,w)) p(x,y) dxdy$$

(3)

$$(3) \quad = 2 \int (\mu(x) - F(x,w)) \left[\int (y - \mu(x)) p(y/x) dy \right] p(x) dx$$

$$= \int y p(y/x) dy - \mu(x) = 0$$

Donc $\mathbf{E}_{\text{généré}}$ = (1) + (2) et comme (1) est indépendant de w

On tire Min $\mathbf{E}_{\text{généré}}$ ⇔ Min (2)

• But de l'apprentissage : $F^*(\boldsymbol{x},\boldsymbol{W}) \cong \mu(\boldsymbol{x})$

• $E_{géné}$ et $\mu(\boldsymbol{x})$ sont inconnus

• Dans la pratique, on cherche à estimer $\mu(\boldsymbol{x})$ à partir d 'un ensemble d'apprentissage :

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$$

• L'apprentissage consiste à minimiser la fonction erreur $\boldsymbol{E}_D$:

$$E_D = \frac{1}{N} \sum_{i=1}^{N} \left\| y_i - F\left(x_i, w\right) \right\|^2$$

• La solution obtenue dépend alors de l'ensemble d 'apprentissage D : $F(x, w^*/D)$

Remarque

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

$$E_{géné} = \int \frac{\left\| \mu(x) - F(x,W) \right\|^2}{\sigma^2(x)} p(x) p(y/x) dx\, dy$$

Ainsi, $$Min\, E_{géné} \Leftrightarrow Min \int \left\| \mu(x) - F(x,w) \right\|^2 \frac{p(x)}{\sigma(x)^2} dx$$

Cas de la Classification

Exemple : 2 classes

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

$$\begin{cases} y = a & si \quad x \in C_1 \\ y = b & si \quad x \in C_2 \end{cases} \quad avec\ a \neq b$$

Problème : déterminer $f(\bullet, \mathbf{W})$ qui minimise :

$$E = \int (f(x,w) - y)^2 \, p(x,y) dx dy$$

$$Min_w E \Leftrightarrow Min_w \int (f(x,w) - E(y/x))^2 \, p(x) \, dx$$

$f(\mathbf{x}, \mathbf{W})$ « approxime »

$$\mu(x) = E(y/x) = a \times p((C_1/x) + b \times p(C_2/x))$$

•**Interprétation :**

Si ***a = 1 et b = 0*** ***d(x) = p(C$_1$ / x)***

Si ***a = 1 et b = -1*** ***d(x) = p(C$_1$ / x) - p(C$_2$ / x)***

Fonction discriminante de Bayes

Problème : Déterminer aux moindres carrés f (•, **W**) qui minimise :

$$E = \frac{1}{N}\left[\sum_{x \in C_1}\left(f(x,w)-a\right)^2 + \sum_{x \in C_2}\left(f(x,w)-b\right)^2\right]$$

•**Hypothèses :**

N grand

$$p(C_1) \approx \frac{n_1}{N} \qquad n_1 = Card(C_1)$$

$$p(C_2) \approx \frac{n_2}{N} \qquad n_2 = Card(C_2)$$

Remarque :

• La précision des approximations de $\mu(x)$ dépend de l'architecture choisie.

• La précision ponctuelle de cette approximation diffère selon la répartition des exemples $p(x)$.

***p(x)* grand** ⇔ ***f(x,w)* proche de $\mu(x)$**

***p(x)* petit** ⇔ ***f(x,w)* ?**

• Les régions sous représentées sont plus ou moins ignorées.

Tous ces résultats se généralisent à *M* classes.

Remarques :

• L'erreur quadratique tient compte des erreurs absolues.

• Il peut y avoir de grandes erreurs relatives pour les « petites » probabilités. On pourra utiliser la distance de Kullback-Leibler. *** Ajouter la formule

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

$$\sum_{i=1}^{M} f_i(x, w) \neq 1$$

Il est possible d'utiliser alors pour les neurones de sorties des fonctions **softmax :**

Chaque neurone i de la couche de sortie calcule son état par :

$$S_i = \exp(A_i) \Big/ \sum_{l \in \text{sortie}} \exp(A_l)$$

A_i est la somme pondérée des entrées du neurone ***i***.

Maximum de vraisemblance

$D = \{(x_i, y_i);\ i = 1, 2, \ldots, N\}$ avec $y_i \in \mathcal{R}$

• Les données sont générées suivant une distribution de probabilités $p(x,y)$.

• On suppose que toutes les observations sont indépendantes.

Ainsi :

$$P(D) = P\left((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\right)$$
$$= p(x_1, y_1) \times p(x_2, y_2) \times \ldots \times p(x_N, y_N)$$

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

• D'autre part, si on fait l'hypothèse :

$$x \longrightarrow y = \mu(x) + \varepsilon(x)$$

•Où $\mu(\ )$ est une fonction bien déterminée et $\varepsilon(x)$ une variable aléatoire qui suit une loi normale $N(0, \sigma(x)\mathbf{I})$.

Ainsi :

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

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

$P(x_i)$

- Soit une famille de fonctions :

$$F=\{f(x,W),W\}$$

- On désire approximer μ par une fonction ***f(x, W)*** particulière de la famille *F*.

Ainsi :

$$P(D,W)=\frac{1}{\left(\sqrt{2\Pi}\right)^N}\prod_{i=1}^{N}\frac{1}{\sigma(x_i)}\exp\left[-\frac{1}{2}\frac{\|y_i-f(x_i,w)\|^2}{\sigma^2(x_i)}\right]$$

Posons :

$$E(W)=-\ln P(D,W)$$

$$=\frac{1}{2}\sum_{i=1}^{N}\frac{\|y_i-f(x_i,w)\|^2}{\sigma^2(x_i)}+\frac{1}{2}\sum_{i=1}^{N}\ln\sigma^2(x_i)+Cont$$

1 2 3

Méthode de maximum de vraisemblance.

Déterminer : $w^*=Arg\,Max_w\ P(D,w)$

$$=Arg\,Min_w\ E(w)$$

Cas où σ (x) = σ constante :

Dans ce cas, l'expression de 2 ne dépend pas de ***W***.

On peut prendre :

$$E(W)=\frac{1}{2}\sum_{i=1}^{N}\frac{\|y_i-f(x_i,w)\|^2}{\sigma^2}$$

Ainsi : $$Min_W E(W)=Min_W\left(\sum_{i=1}^{N}\|y_i-f(x_i,W)\|^2\right)$$

On retrouve la somme des erreurs quadratiques.

Soit ***W**** un système de poids qui minimise cette erreur.

Estimation de σ :

En retenant *w**, l'erreur E devient :

$$E(w^*)=\frac{1}{2}\sum_{i=1}^{N}\frac{\|y_i-f(x_i,W^*)\|^2}{\sigma^2}+N\ln\sigma+\frac{N}{2}\ln 2\Pi$$

La valeur de σ^2 qui minimise *E (w*)* est :

$$\sigma^2=\frac{1}{N}\sum_{i=1}^{N}\|y_i-f(x_i,W^*)\|^2$$

Erreur moyenne quadratique

Cas où σ (x) dépend de x. (cas où $y \in \Re$)

Il faut alors estimer en plus σ (***x***).

On peut alors considérer un réseau de neurones multicouche avec deux neurones sur la couche de sortie.

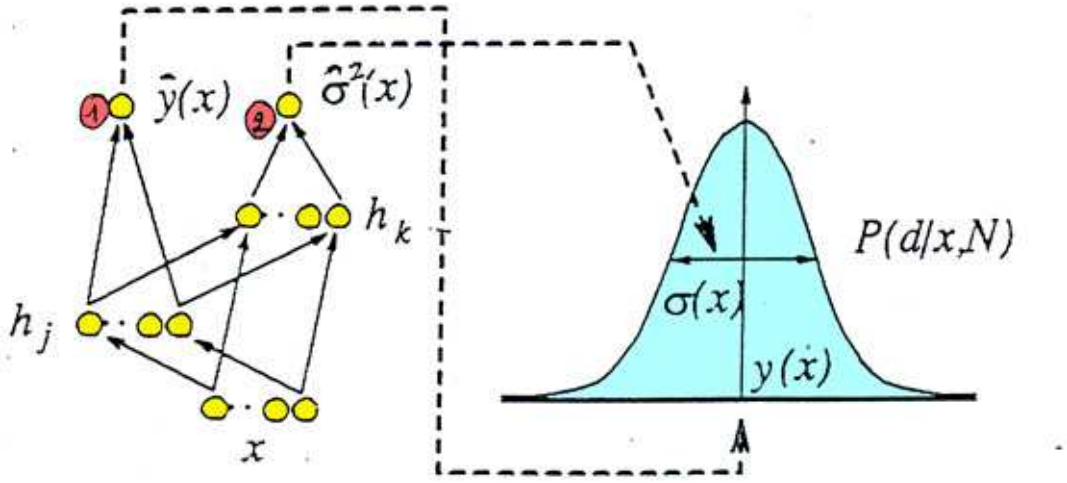
• Le premièr neurone est linéaire et calcule *f(**x**, **W**)* qui estime $\mu(x)$.

• Le second neurone comporte la fonction de transfert e^x et calcule $\sigma^2(x, W)$ qui estime $\sigma^2(x)$.

La fonction d'erreur s'exprime :

$$E(W)=\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i-f(x_i,W))^2}{\sigma^2(x,W)}+\frac{1}{2}\sum_{i=1}^{N}\ln\sigma^2(x_i,W)+\frac{N}{2}\ln(2\Pi)$$

Qu'il faut minimiser par rapport à **W**.



On pose :

$$E_i(W) = \frac{(y_i - f(x_i, W))^2}{\sigma^2(x_i, W)} + \ln\sigma^2(x_i, W)$$

L'initialisation de la rétro-propagation du gradient :

- pour le neurone **1** : $\frac{\partial E_i}{\partial \hat{y}} = -\frac{(y_i - f(x_i, w))}{\sigma^2(x_i, w)}$ **(*)**
- pour le neurone **2** : $\frac{\partial E_i}{\partial e} = -\frac{(y_i - f(x_i, w))^2}{\sigma^2(x_i, w)} + 1$ **(* *)**

L'algorithme se présente alors en 3 phases

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** .

(*) $\frac{\partial E_i}{\partial \hat{y}} = -\left(y_i - f\left(x_i, w\right)\right)$ dans ce cas.

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

Phase 2 : On gèle la sortie du neurone **1** à :

$$\hat{y}_i = f\left(x_i, W_1\right) \quad \text{pour tout } x_i.$$

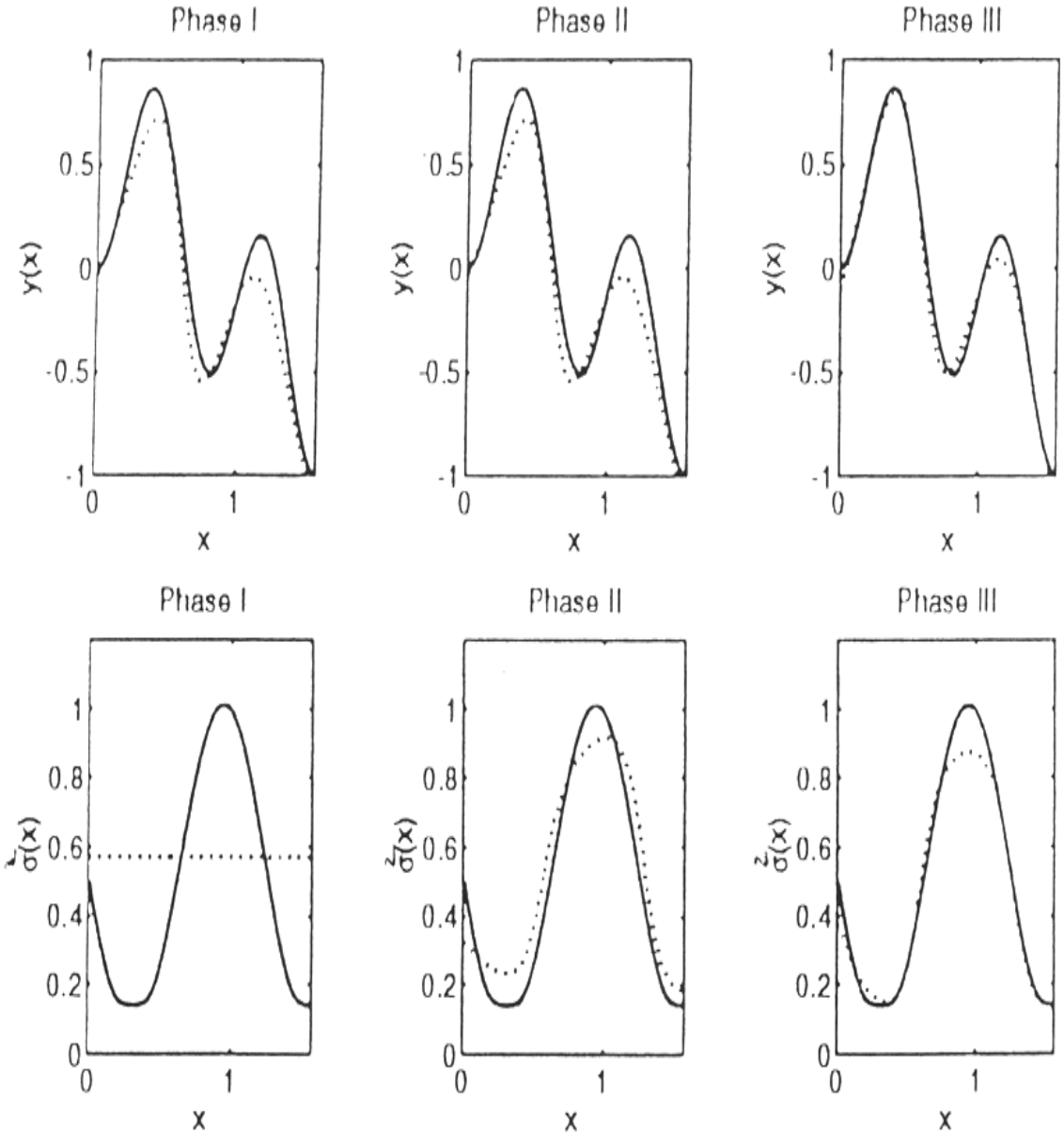
On rétropropage alors uniquement à partir du neurone **2** en prenant :

(**) $$\frac{\partial E_i}{\partial \hat{y}} = -\frac{\left(y_i - f\left(x_i, w\right)\right)^2}{\sigma^2(x,w)} + 1$$

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

Phase 3 : Minimiser la fonction d'erreur complète en rétropropageant à partir de deux neurones **1** et **2**, en appliquant les formules (*) et (**).

Exemple :


Phase I
Phase II
Phase III
y(x)
x
Phase I
Phase II
Phase III
$\hat{\sigma}^2(x)$
x