ANNEXE A. SURFACE DE COÛT : MINIMA LOCAUX

Résumé

Dans cette annexe, nous étudions l'évolution de la forme de la surface de coût en fonction du nombre d'exemples de l'ensemble d'apprentissage. Comme nous l'avons vu dans le chapitre 4 (Apprentissage des réseaux de neurones), la surface de coût peut comporter un minimum (dit global) ; dans ce cas, la recherche de celui-ci est relativement facile. Cette recherche devient plus difficile lorsque la surface possède plusieurs minima locaux.

Nous reprenons, plus en détail, le problème "maître/élève" (modélisation à deux paramètres), introduit au chapitre 4, qui nous permet d'étudier l'existence et l'influence des minima locaux de la fonction de coût. Nous avons constaté, en étudiant cet exemple, que ces minima locaux apparaissent ou disparaissent en fonction du nombre d'exemples d'apprentissage.

Ainsi, nous montrons que la facilité de l'apprentissage augmente avec le nombre de points d'apprentissage.

A.1 Rappel

Les résultats obtenus avec des modèles non linéaires tels que les réseaux de neurones dépendent de :

- la capacité de ces modèles à approcher n'importe quelle régression,
- l'efficacité de l'algorithme de minimisation de la fonction de coût,
- la qualité (nombre et représentativité) des exemples de l'ensemble d'apprentissage.

Tous ces points sont importants ; le chapitre 3 (Les réseaux de neurones) a présenté la propriété d'approximation universelle des réseaux de neurones, puis le chapitre 4 (Apprentissage des réseaux de neurones) a étudié des algorithmes d'optimisation. Ainsi, les deux premiers points ont été abordé et ne présentent plus d'ambiguïté. Il reste le dernier point.

L'utilisateur des réseaux de neurones est toujours convaincu, avec raison, que le nombre d'exemples est important pour garantir la bonne représentativité de l'ensemble d'apprentissage. Ici, nous allons montrer qu'un grand nombre d'exemples est aussi très important pour la convergence vers le minimum global de la fonction de coût.

Pour pouvoir visualiser sur un plan la surface de coût, nous choisissons un exemple de modélisation à 2 paramètres seulement.

A.2 Modélisation à 2 paramètres

[Antoniadis 92] a proposé un exemple de modélisation à deux paramètres ; nous le traitons ici d'une manière différente, en faisant varier le nombre d'exemples d'apprentissage.

La construction du problème est la suivante. Nous engendrons les points de l'ensemble d'apprentissage à partir d'une fonction (F) à une seule variable et à deux paramètres ; les sorties désirées sont notées y_p, et les sorties du modèle y :

$$y_p = F(x) + \omega$$

où x : entrée distribuée aléatoirement entre -3 et +3 suivant une loi uniforme

ω : bruit gaussien de variance égale à 0.5

avec $F(x) = B\,e^{-Ax}$

$A = 0{,}669$

$B = 0{,}214$

A partir de ces données, nous avons construit une base d'apprentissage ***E***.

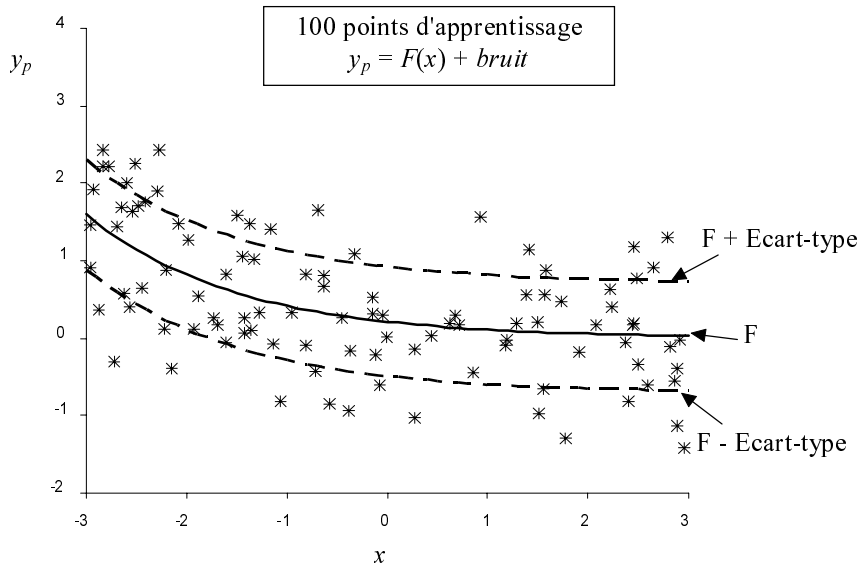

Figure A.1 : Régression et 100 points d'apprentissage
(trait plein : régression F, trait pointillé : F ± écart-type du bruit)

La figure A.1 présente la régression F (fonction génératrice des exemples, inconnue dans la pratique) et un ensemble d'apprentissage (les 100 premiers points de l'ensemble ***E***).

La variance du bruit étant assez importante, la fonction désirée n'est, *a priori*, pas évidente à retrouver. Dans ce problème académique, nous utilisons une famille de fonctions qui contient la fonction de régression.

A.2.1 Famille de fonctions

Nous considérons la famille de fonctions, qui contient la régression F, définie par :

$$f(x;\alpha,\beta) = \beta\,e^{-\alpha x}$$

Cette famille de fonctions, engendrée par 2 paramètres (α et β) et peut se mettre sous la forme du "réseau de neurones" suivant :

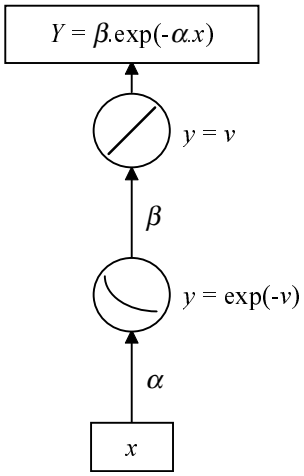

Figure A.2 : "Réseau de neurones" reproduisant la fonction f(x;α,β)

Ce réseau à deux coefficients comporte une entrée (x), un neurone caché (fonction d'activation $y = e^{-v}$) et une sortie linéaire (fonction d'activation $y = v$).

A.2.2 Fonction de coût

Nous utilisons la fonction de coût des moindres carrés, notée $J^A(\alpha,\beta)$:

$$J^A(\alpha,\beta) = \frac{1}{N} \cdot \sum_{i=1}^{N} \left(y^i - y_p^i\right)^2 = \frac{1}{N} \cdot \sum_{i=1}^{N} \left(\beta\, exp\left(-\alpha \cdot x^i\right) - y_p^i\right)^2$$

avec N : Nombre de points d'apprentissage

Cette fonction de coût correspond à l'Écart Quadratique Moyen sur l'ensemble des points d'apprentissage (*EQMA*).

A.2.3 Procédure expérimentale

Dans un premier temps, on construit l'ensemble d'apprentissage en prenant N points de $\boldsymbol{E}$. Puis la recherche du minimum de l'EQMA se déroule de la façon suivante :

- Les paramètres (α et β) sont initialisés aléatoirement entre -1 et +1 suivant une distribution uniforme.
- Le point de départ des paramètres étant choisi, une méthode d'optimisation du deuxième ordre (quasi-Newton) recherche un minimum de l'EQMA.

  Nous répétons cette procédure (initialisation des paramètres et optimisation) 100 fois en changeant les valeurs initiales des paramètres (α et β).

Cette recherche du minimum de la fonction de coût s'effectue avec 5 ensembles d'apprentissage déterminés par :

N = 1000, 100, 10, 4 et 3.

A.2.4 Résultats avec 1000 exemples d'apprentissage

Avec 1000 points d'apprentissage, les résultats sont présentés dans le tableau suivant :

Ensemble d'apprentissage	**Optimisation**			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
1000 Points	100/100	0,470	0,685	0,212
Régression			0,669	0,214

Tableau A.1 : Résultat des estimations avec 1000 points d'apprentissage

- La colonne *Fréquence* donne la fréquence d'obtention du minimum considéré ; ici l'algorithme a toujours atteint le même minimum (100 fois sur 100).
- L'*EQMA* est la valeur de la fonction de coût après l'optimisation.
- De même, *Alpha* et *Bêta* sont les estimations des paramètres après l'optimisation.
- La dernière ligne présente les valeurs des paramètres choisies pour la construction des exemples (fonction génératrice + bruit de variance égale à 0,5).

A chaque initialisation des paramètres, l'algorithme d'optimisation a donc atteint le même minimum, qui correspond à des valeurs des paramètres qui sont très proches de celles des paramètres du modèle.

La figure A.3 montre les courbes de niveau du coût, qui ne dépend que de α et β. On constate que la surface ne présente qu'un minimum global ; toutes les initialisations différentes des paramètres conduisent au même point. Le minimum global ($*$) et l'estimation des paramètres (+) sont confondus.

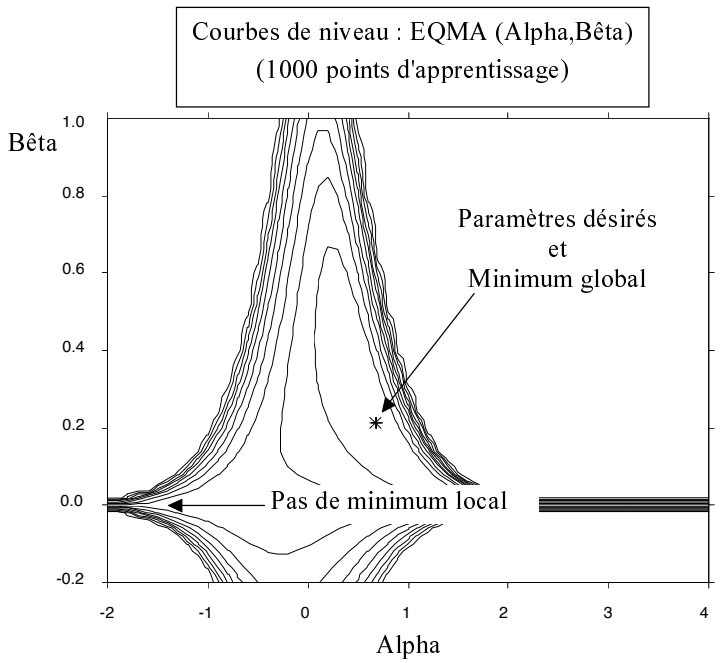

Figure A.3 : Courbes de niveau de la fonction de coût (1000 points d'apprentissage)

Afin de mieux apprécier la forme de cette surface, nous décomposons la fonction de coût de la façon suivante :

$$\begin{aligned} EQMA(\alpha,\beta) &= \frac{1}{N}\sum_{i=1}^{N}\left(y^i - y_p^i\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\beta\cdot\exp\left(-\alpha\cdot x^i\right) - y_p^i\right)^2 \\ &= \frac{1}{N}\left[\beta^2\sum_{i=1}^{N}\exp\left(-2\alpha\cdot x^i\right) - 2\beta\sum_{i=1}^{N} y_p^i\cdot\exp\left(-\alpha\cdot x^i\right) + \sum_{i=1}^{N}\left(y_p^i\right)^2\right] \end{aligned}$$

Pour une valeur de α donnée, cette décomposition fait apparaître la fonction de coût comme une parabole en β.

En notant :

$$\beta_{\min}(\alpha) = \frac{\sum_{i=1}^{N} y_p^i\cdot\exp\left(-\alpha\cdot x^i\right)}{\sum_{i=1}^{N}\exp\left(-2\cdot\alpha\cdot x^i\right)}$$

On obtient le minimum de la fonction de coût $EQMA_{min}$ pour une valeur de α donnée par la relation :

$$EQMA_{\min}(\alpha) = Min\left[EQMA(\alpha,\beta\in R)\right] = EQMA\left(\alpha,\beta_{\min}(\alpha)\right)$$

Ainsi, à chaque valeur de α, nous pouvons calculer le minimum de la fonction de coût. La figure A.4 présente la fonction $EQMA_{min}(\alpha)$ avec 1000 points d'apprentissage. Elle possède un minimum global :

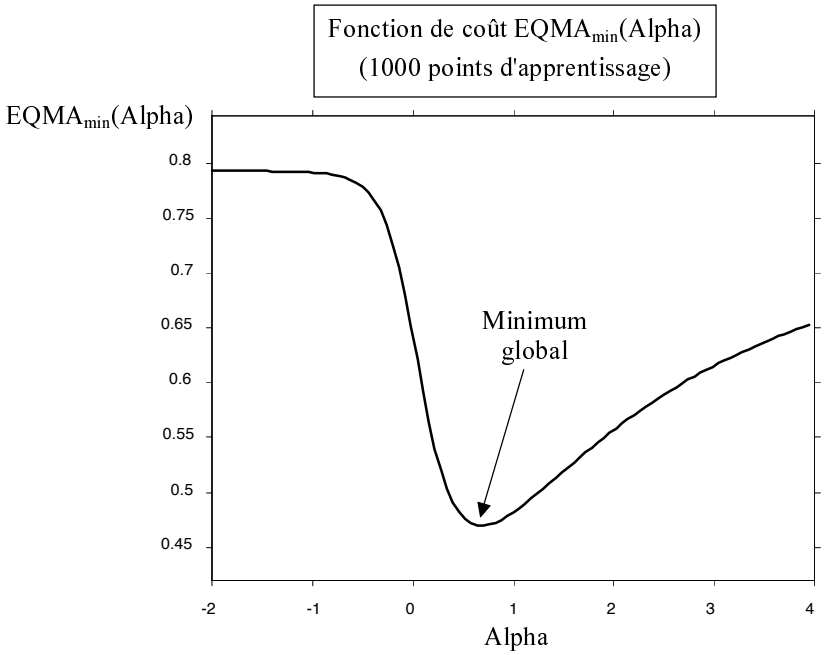

Figure A.4 : Forme de la fonction $EQMA_{min}(\alpha)$

Après l'apprentissage, le modèle trouvé est très proche de la fonction désirée (valeurs de α et β proche de A et B, tableau A.1).

Sur la figure A.5, on présente un point d'apprentissage sur 10, la fonction désirée et la fonction trouvée :

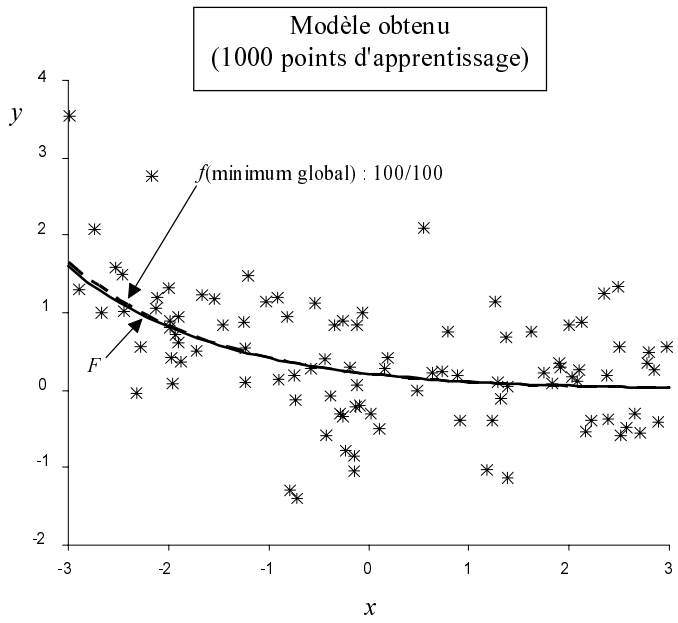

Figure A.5 : Modèle obtenu après apprentissage

En résumé, avec 1000 points d'apprentissage, l'algorithme d'optimisation atteint donc à chaque fois le même point et trouve une fonction $f(x;\alpha,\beta)$ très proche de la régression $F(x)$.

A.2.5 Résultats avec 100 exemples d'apprentissage

Avec 100 points d'apprentissage, les résultats des 100 apprentissages sont donnés dans le tableau A.2 :

Ensemble d'apprentissage	**Optimisation**			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
100 Points	98/100	0,517	0,768	0,175
	2/100	0,976	-12,3	≈ 0 (< 0)
Régression			0,669	0,214

Tableau A.2 : Résultat des estimations avec 100 points d'apprentissage

Dans une très grande proportion (98/100), l'algorithme trouve les bonnes valeurs pour α et β qui conduisent à un EQMA proche du bruit. Néanmoins, 2 fois sur 100, il reste bloqué dans un minimum local. Dans ce cas, l'EQMA est beaucoup plus grand que la variance du bruit.

La figure A.6 montre le point obtenu dans 98% des cas (minimum global) et la direction du minimum local :

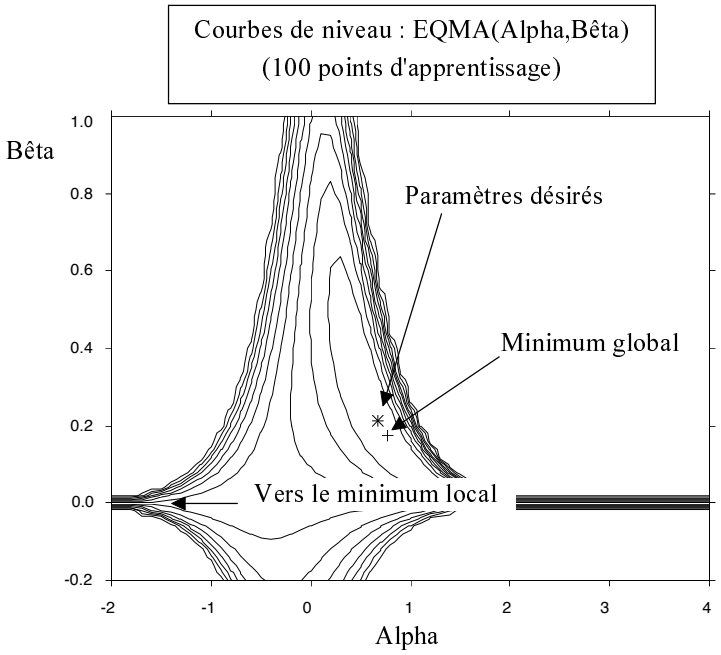

Figure A.6 : Courbes de niveau de la fonction de coût (100 points d'apprentissage)

Le tracé de la fonction $EQMA_{min}(\alpha)$ montre que la fonction décroît pour α tendant vers moins l'infini :

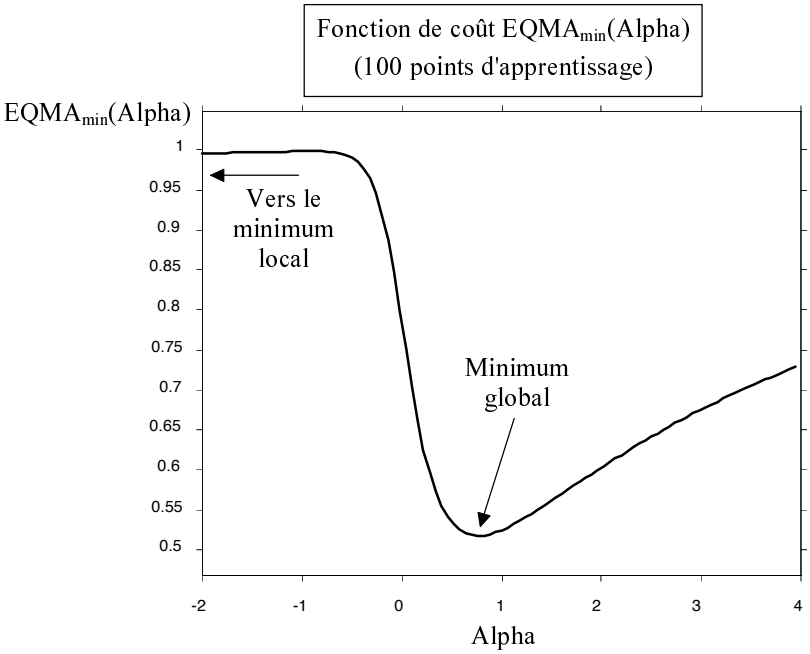

Figure A.7 : Forme de la fonction $EQMA_{min}(\alpha)$

Nous traçons les fonctions $f(x;\alpha,\beta)$ avec les valeurs atteintes par l'algorithme d'optimisation.

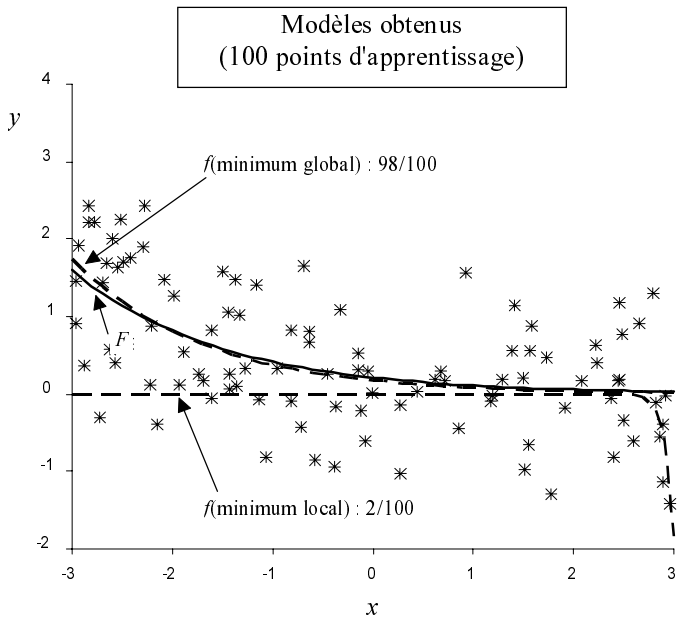

Figure A.8 : Modèles obtenus après apprentissage

Nous constatons que 2 fois sur 100, la fonction trouvée se bloque du côté négatif et se contente de passer par le point le plus à droite en s'annulant partout ailleurs.

A.2.6 Résultats avec 10 exemples d'apprentissage

Avec 10 points d'apprentissage, les résultats des 100 procédures d'optimisation sont donnés dans le tableau A.3 :

Ensemble d'apprentissage	**Optimisation**			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
10 Points	84/100	0,483	3,23	$2{,}39\ 10^{-4}$
	16/100	0,821	-10,2	$\approx 0\ (< 0)$
Régression			0,669	0,214

Tableau A.3 : Résultat des estimations avec 10 points d'apprentissage

Ici, la probabilité d'atteindre le minimum local n'est plus du tout négligeable (16%).

On remarque également que l'estimation des paramètres correspondants au minimum global est complètement erronée ($\alpha = 3{,}23$ et $\beta = 2{,}39\ 10^{-4}$). L'ensemble d'apprentissage n'est plus représentatif du phénomène.

A.2.7 Résultats avec 4 exemples d'apprentissage

Avec 4 points d'apprentissage, les résultats des 100 procédures d'optimisation sont donnés dans le tableau A.4 :

Ensemble d'apprentissage	Optimisation			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
4 Points	91/100	0,664	0,657	0,349
	9/100	1,504	-10,3	≈ 0 (< 0)
Régression			0,669	0,214

Tableau A.4 : Résultat des estimations avec 4 points d'apprentissage

Même avec 4 points, la probabilité d'atteindre le minimum local n'est pas nulle. Les tracés des courbes de niveau et des modèles obtenus avec 4 points d'apprentissage sont semblables aux précédents (avec 100 et 10 points d'apprentissage).

A.2.8 Résultats avec 3 exemples d'apprentissage

Avec 3 points d'apprentissage, les résultats sont donnés dans le tableau A.5 :

Ensemble d'apprentissage	Optimisation			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
3 points	100/100	0,101	0,331	0,831
Régression			0,669	0,214

Tableau A.5 : Résultat des estimations avec 3 points d'apprentissage

C'est seulement avec 3 points d'apprentissage que l'on retrouve les résultats obtenus avec 1000 points. En effet, le tracé de la fonction $EQMA_{min}(\alpha)$ montre un seul minimum global (voir figure A.9). Néanmoins, il faut noter que le minimum global correspond à une très mauvaise estimation des paramètres :

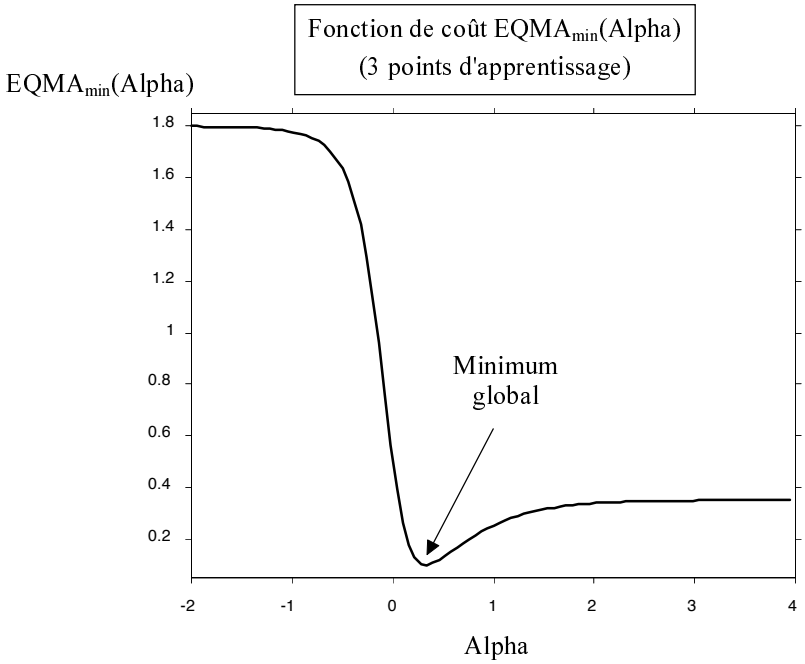

Figure A.9 : Forme de la fonction EQMA$_{min}$(α)

A.2.9 Tableau récapitulatif

Le tableau A.6 regroupe les résultats des différentes estimations :

Ensemble d'apprentissage	**Optimisation**			
	Fréquence	**EQMA**	**Alpha**	**Bêta**
1000 Points	100/100	0,470	0,685	0,212
100 Points	98/100	0,517	0,768	0,175
	2/100	0,976	-12,3	≈ 0 (< 0)
10 Points	84/100	0,483	3,23	$2{,}39\ 10^{-4}$
	16/100	0,821	-10,2	≈ 0 (< 0)
4 Points	91/100	0,664	0,657	0,349
	9/100	1,504	-10,3	≈ 0 (< 0)
3 points	100/100	0,101	0,331	0,831
Régression			0,669	0,214

Tableau A.6 : Tableau récapitulatif

Nous constatons que la probabilité de rester bloqué dans le minimum local n'est pas négligeable avec 10 points d'apprentissage. L'EQMA correspondant est égale à 0,8 ce qui est nettement supérieur à celui trouvé avec 1000 points d'apprentissage. Il est donc plus difficile de faire passer une courbe près de 10 points que près de 1000 points. Ce problème montre qu'il faut toujours posséder un nombre d'exemples d'apprentissage le plus grand possible pour obtenir **facilement** un **bonne** estimation des paramètres du modèle.

A.3 Conclusion

Ce travail sur la forme de la fonction de coût a mis en évidence un phénomène inattendu. En effet, il est évident que la représentativité d'un grand échantillon d'apprentissage (avec beaucoup d'individus) est meilleure que celle d'un petit ; mais, on pourrait penser qu'il est plus facile de réaliser l'apprentissage d'un modèle non-linéaire s'il y a peu d'individus. Cette annexe permet d'affirmer le contraire : un échantillon d'apprentissage avec peu d'individus peut conduire à une surface de coût comportant des minima locaux, et aussi à une mauvaise estimation des paramètres du modèle.

Nous avons donc intérêt à posséder l'échantillon d'apprentissage le plus vaste possible : ainsi la fonction de coût présentera moins de minima locaux et les algorithmes d'optimisation trouveront plus facilement le minimum global.

Nous retrouvons sur cet exemple le fait que le nombre d'éléments de l'ensemble d'apprentissage est une donnée fondamentale car un ensemble d'apprentissage abondant garantit, d'une part, une bonne représentativité de l'échantillon d'apprentissage (estimation des paramètres) et, d'autre part, une forme plus régulière de la surface de coût (optimisation plus facile).