9 Réseau de neurones

9.1 Un problème bayésien

Soit (θ, X) un couple aléatoire à valeurs dans $\mathbb{R} \times \mathbb{R}^d$ et soit $g : \mathbb{R} \to \mathbb{R}$ une fonction. A partir de l'observation de X, on cherche à estimer au mieux $g(\theta)$ dans le sens suivant : on veut trouver la fonction $\hat{\theta} : \mathbb{R}^d \to \mathbb{R}$ pour laquelle

$$\mathbb{E}\Big[\big(\hat{\theta}(X) - g(\theta)\big)^2\Big] \tag{1}$$

est minimale. C'est le cas si et seulement si $\hat{\theta}(X) = \mathbb{E}\big[g(\theta) \mid X\big]$ presque sûrement, mais cette espérance conditionnelle n'est pas calculable si on ne connaît pas la loi du couple (θ, X). Même si la loi du couple (θ, X) est connue, cette espérance conditionnelle peut être difficile à approcher numériquement.

9.2 Réseau de neurones

On suppose que l'on dispose d'une suite de couples aléatoires (θ_k, X_k), $1 \leq k \leq K$, i.i.d. et de même loi que (θ, X). On se donne une famille de fonctions $F_\alpha : \mathbb{R}^d \to \mathbb{R}$, $\alpha \in \mathbb{R}^N$, et on cherche à minimiser

$$\alpha \to \mathbb{E}\Big[\big(F_\alpha(X) - g(\theta)\big)^2\Big]$$

Pour cela on peut utiliser un algorithme de gradient stochastique : on construit une suite $(\alpha_k)_{0 \leq k \leq K}$ de la manière suivante : on se donne un pas $\gamma > 0$, $\alpha_0 \in \mathbb{R}^N$ et on pose

$$\alpha_{k+1} = \alpha_k - \gamma \frac{\partial}{\partial \alpha_k} \big(F_{\alpha_k}(X_{k+1}) - g(\theta_k)\big)^2 \quad k = 0, \ldots, K-1$$

Un exemple de réseau de neurones correspond à la famille F_α suivante. On se donne :

— des entiers $N_\ell \in \mathbb{N}^*$, $\ell = 0, \ldots, L$ avec $N_0 = d$ et $N_L = 1$.

— des vecteurs $b^{(\ell)} \in \mathbb{R}^{N_{\ell+1}}$, des matrices $a^\ell \in \mathbb{R}^{N_{\ell+1} \otimes N_\ell}$ pour $0 \leq \ell \leq L-1$.

— des fonctions d'activation $h_\ell : \mathbb{R} \to \mathbb{R}$ pour $0 \leq \ell \leq L-1$.

Pour $\alpha = (b^{(\ell)}, a^{(\ell)})_{0 \leq \ell \leq L-1}$ on définit $F(\alpha, X)$ comme suit :

— On pose $X^{(0)} = X$

— Pour $\ell = 0, \ldots, L-1$ on pose

$$X_i^{(\ell+1)} = h_\ell\left(b_i^{(\ell)} + \sum_{j=1}^{N_\ell} a_{ij}^{(\ell)} X_j^{(\ell)}\right), \quad 1 \leq i \leq N_{\ell+1}$$

— On pose $F(\alpha, X) = X^{(L)}$

On choisit $h_{L-1}(x) = x$ et $h_\ell(x) = x^+$ pour $\ell \leq L-2$.

9.3 Calcul du gradient

Comment calcule-t-on le gradient $\frac{\partial}{\partial\alpha}H$ où $H=\big(F_\alpha(X)-g(\theta)\big)^2$? On le fait par récurrence arrière :

1. $F_\alpha(X)=X_1^{(L)}$ donc
$$\frac{\partial}{\partial X_1^{(L)}}H=2(F_\alpha(X)-g(\theta))$$
Comme
$$X_1^{(L)}=h_{L-1}\Big(b_1^{(L-1)}+\sum_{j=1}^{N_{L-1}}a_{1j}^{(L-1)}X_j^{(L-1)}\Big)$$
il vient, avec $Y_1^{(L-1)}=b_1^{(L-1)}+\sum_{j=1}^{N_{L-1}}a_{1j}^{(L-1)}X_j^{(L-1)}$
$$\frac{\partial}{\partial b_1^{(L-1)}}H=h'_{L-1}\big(Y_1^{(L-1)}\big)\frac{\partial}{\partial X_1^{(L)}}H\qquad\frac{\partial}{\partial a_{1j}^{(L-1)}}H=h'_{L-1}\big(Y_1^{(L-1)}\big)X_j^{(L-1)}\frac{\partial}{\partial X_1^{(L)}}H$$
De plus on a
$$\frac{\partial}{\partial X_j^{(L-1)}}H=\frac{\partial X_1^{(L)}}{\partial X_j^{(L-1)}}\frac{\partial}{\partial X_1^{(L)}}H=h'_{L-1}\big(Y_1^{(L-1)}\big)a_{1j}^{(L-1)}\frac{\partial}{\partial X_1^{(L)}}H$$
2. Pour $1\le\ell\le L$, étant donné les $\frac{\partial}{\partial X_i^{(\ell)}}H$, $1\le i\le N_\ell$, on calcule
$$\frac{\partial}{\partial b_i^{(\ell-1)}}H=\frac{\partial X_i^{(\ell)}}{\partial b_i^{(\ell-1)}}\frac{\partial}{\partial X_i^{(\ell)}}H=h'_{\ell-1}\big(Y_i^{(\ell-1)}\big)\frac{\partial}{\partial X_i^{(\ell)}}H$$
où $Y_i^{(\ell-1)}=b_i^{(\ell-1)}+\sum_{j=1}^{N_{\ell-1}}a_{ij}^{(\ell-1)}X_j^{(\ell-1)}$,
$$\frac{\partial}{\partial a_{ij}^{(\ell-1)}}H=X_j^{(\ell-1)}h'_{\ell-1}\big(Y_i^{(\ell-1)}\big)\frac{\partial}{\partial X_i^{(\ell)}}H$$
De plus
$$\frac{\partial}{\partial X_j^{(\ell-1)}}H=\sum_i\frac{\partial X_i^{(\ell)}}{\partial X_j^{(\ell-1)}}\frac{\partial}{\partial X_i^{(\ell)}}H=\sum_i a_{ij}^{(\ell-1)}h'_{\ell-1}(Y_i^{(\ell-1)})\frac{\partial}{\partial X_i^{(\ell)}}H$$

9.4 Premier modèle

Le script `rneurones.py` met en œuvre cet algorithme pour le modèle où θ est de loi uniforme sur $[-\pi,\pi]$ et où, sachant θ, $X=(X_1,\dots,X_n)$ est une famille i.i.d. de variables aléatoires gaussiennes de loi $\mathcal{N}(\theta,1)$. On cherche à estimer $g(\theta)=\cos(\theta)$.
Au cours de l'entraînement, on compare la performance du réseau neurones avec celle de l'estimateur $\cos\big(\frac{1}{n}\sum X_i\big)$.

1. Augmenter le nombre de couches pour essayer de construire un meilleur estimateur.
2. Modifier le programme pour considérer le cas où $g(\theta)$ est la partie entière de θ.

9.5 Second modèle

Soit U et Z_i, $1 \leq i \leq n$, des variables i.i.d. $\mathcal{N}(0,1)$. On pose

$$\theta_i = Z_i + 2U, \quad 1 \leq i \leq n$$

et considère $X = (X_1, \ldots, X_n)$ tel que, conditionnellement à $\theta = (\theta_1, \ldots, \theta_n)$, les X_i sont indépendantes et $X_i \sim \mathcal{N}(\theta_i, 1)$. On choisit

$$g(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{\theta_i > 0\}}$$

Comparer un réseau de neurones avec l'estimateur

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{X_i > 0\}}$$