

Examen cours AMAL (Advanced Machine Learning) - Masters DAC et M2A – Sorbonne Université

16/02/2021 – Documents autorisés - Durée 1h30

Exercice 1

Introduire des contraintes sur les paramètres permet de contrôler la complexité des modèles d'apprentissage. Nous considérons dans la suite une approche Bayésienne pour des problèmes de régression ou de classification, implémentée avec des réseaux de neurones. On note w une variable aléatoire associée au vecteur de poids d'un réseau de neurones.

1. On veut maximiser le maximum a posteriori (MAP) : $p(w|X, Y) \propto p(Y|x, w)p(w)$

où $D = (X, Y)$ correspond à l'ensemble d'apprentissage. On considère la fonction de coût suivante :

$$L_R(w) = -\ln \ln p(X, w) - \ln \ln p(w) = L(w) + R(w)$$

On remarque que $L_R(w) \propto -\ln \ln p(X, Y)$

On suppose que la probabilité à priori sur les poids suit une densité Gaussienne,

$p(w; 0, \sigma) = N(w; 0, \sigma^2 I)$ où 0 est le vecteur moyenne, σ un scalaire positif et I la matrice identité.

- 1.1. Montrer que $R(w)$ se met sous la forme $\lambda w^T w$, $\lambda \in \mathbb{R}^+$
- 1.2. Quel est l'effet de ce terme sur les poids, en quoi permet-il de contrôler la complexité du modèle utilisé ?
2. On considère un réseau de neurones à une couche cachée, les activations de la couche cachée sont non linéaires, celle de la couche de sortie sont linéaires. On souhaiterait que la pénalité de régularisation offre des qualités de consistance pour des transformations élémentaires des données, dans le sens où ces transformations élémentaires ne changent pas le résultat de l'apprentissage. Ainsi si on considère une transformation linéaire sur les entrées $x' = ax$, une simple transformation $w'^{(1)} = \frac{1}{a}w^{(1)}$ où $w^{(1)}$ est le vecteur de poids de la première couche

donnera un résultat inchangé, de même sur les sorties $\hat{y}' = by$ sera compensé par $w'^{(2)} = bw^{(2)}$ où $w^{(2)}$ est le vecteur de poids de la deuxième couche.

- 2.1. Est-ce que la pénalisation introduite à la question 1 est consistante par rapport à ces deux transformations ? On pourra regarder si le terme de pénalisation $\lambda w^T w$ est inchangé quand on change les poids par les transformations linéaires décrites ci-dessus.
- 2.2. On considère le prior $p(w) = p(w^{(1)}, w^{(2)}) = p(w^{(1)})p(w^{(2)})$ avec $p(w^{(1)}) = N(w^{(1)}; 0, \sigma_1^2 I)$ et $p(w^{(2)}) = N(w^{(2)}; 0, \sigma_2^2 I)$. Donner l'expression du terme de régularisation $R(w) = -\ln \ln p(w)$ correspondant et montrer qu'il peut s'écrire $\lambda_1 w^{(1)T} w^{(1)} + \lambda_2 w^{(2)T} w^{(2)}$

2.3. Est-ce que ce prior est consistant avec les transformations introduites ci-dessus ?

Comme précédemment on pourra regarder si ce prior change avec les transformations de poids introduites.

3. On étudie maintenant une méthode qui consiste à introduire une distribution a priori sur les poids qui prend la forme d'un mélange de gaussiennes. La densité de probabilité de la variable

w est $p(w) = \prod_i p(w_i)$, avec $p(w_i) = \sum_{j=1}^m \pi_j N(w_i; \mu_j, \sigma_j^2)$, où m est le nombre de

composantes du mélange, $N(w_i; \mu_j, \sigma_j^2)$ définit une distribution gaussienne

unidimensionnelle, de moyenne μ_j et de variance σ_j^2 , les π_j sont les coefficients du mélange

qui satisfont la contrainte $\sum_{j=1}^m \pi_j = 1$. On va considérer un terme de régularisation

correspondant à cette distribution :

$$R(w) = - \sum_i \ln \left(\sum_{j=1}^m \pi_j N(w_i; \mu_j, \sigma_j^2) \right)$$

et la fonction de coût régularisée :

$$L_R(w) = L(w) + \lambda R(w)$$

où $L(w)$ est une fonction de coût classique définie sur l'ensemble d'apprentissage (moindres carrés ou entropie croisée) et $\lambda \in \mathbb{R}^+$ le paramètre de régularisation.

3.1. Dessiner un mélange de 3 gaussiennes unidimensionnelles en fixant des valeurs des coefficients de mélange.

3.2. Supposons que tous les paramètres $\mu_j, \sigma_j^2, \pi_j, j = 1 \dots m$ soient fixés, on minimise $L_R(w)$ en optimisant les poids, quel est l'effet de la contrainte sur les poids ?

3.3. L'apprentissage porte sur l'ensemble des paramètres du modèle : les poids w et les paramètres du mélange $\mu_j, \sigma_j^2, \pi_j, j = 1 \dots m$. On introduit la probabilité a posteriori suivante, $p(w) = \gamma_j(w) = \frac{\pi_j N(w; \mu_j, \sigma_j^2)}{\sum_{k=1}^m \pi_k N(w; \mu_k, \sigma_k^2)}$, attention, ici w est une variable scalaire.

3.3.1. Donner l'expression de $\frac{\partial R}{\partial w_i}$ en fonction des $\gamma_j(w)$. Quel est l'effet du terme de régularisation sur les poids ?

3.3.2. Donner l'expression de $\frac{\partial R}{\partial \mu_j}$ en fonction des $\gamma_j(w)$. Quel est l'effet du terme de régularisation sur les μ_j ?

3.3.3. Donner l'expression de $\frac{\partial R}{\partial \sigma_j}$ en fonction des $\gamma_j(w)$. Quel est l'effet du terme de régularisation sur les σ_j ? On veut garder les coefficients σ_j positifs, comment peut-on faire ?

Exercice 2

On considère une problématique d'apprentissage supervisé où les données sont recueillies sur différents environnements $E = \{e_1, \dots, e_m\}$. Chacun de ces environnements a une distribution $P_e(x, y)$ qui lui est propre et un ensemble de données D_e est associé à chaque environnement. L'objectif est d'apprendre une fonction f_θ qui ait de bonnes performances sur tous les environnements. On note $l(f_\theta(x), y)$ la fonction de coût individuelle associée à un couple (x, y) avec x la donnée d'entrée et y la sortie désirée, et $R_e \triangleq E_{(x,y) \sim P_e} l(f_\theta(x), y)$, le risque pour l'environnement e .

L'approche classique de l'apprentissage ignore les différents environnements et optimise un critère global sur l'ensemble E :

$R_{ERM} \triangleq E_{(x,y) \sim P_E} l(f_\theta(x), y)$, où P_E est la distribution des données sur l'ensemble E

1. Quel problème peut-on rencontrer si les distributions P_e diffèrent significativement ?
2. Une autre approche, dite optimisation robuste, propose d'optimiser la pire performance sur un ensemble d'environnements :

$$R_{ROB} \triangleq R_e$$

- 2.1 Montrer que minimiser R_{ROB} par rapport à θ est équivalent à minimiser la quantité suivante

$$R_{INT} \triangleq \sum_e \lambda_e R_e$$

Indication : on peut partir pour θ fixé de $R_e = R_1 - \delta_e$, avec R_1 le risque le plus grand et $\delta_e \geq 0, \delta_1 = 0$ et regarder l'expression du max ci-dessus.

- 2.2 Montrer $\sum_e \lambda_e R_e = E_{\sum_e \lambda_e P_e(x,y)} [l(f_\theta(x), y)]$

3. On considère une relaxation du problème ci-dessus qui est la suivante

$$R_{REX} \triangleq \sum_{e=1}^m \lambda_e R_e$$

La différence par rapport à R_{INT} est que l'on permet maintenant à certains coefficients λ_e d'être négatifs.

- 3.1 Montrer que $R_{REX} = (1 - m\lambda_{min})R_e + \lambda_{min} \sum_{e=1}^m R_e$ où m est le nombre d'environnements considérés.
- 3.2 Montrer que R_{ERM} et R_{INT} sont des cas particulier de R_{REL}
- 3.3 Quel est à votre avis le comportement de ce critère quand on fixe $\lambda_{min} < 0$
- 3.4 Que fait ce critère quand $\lambda_{min} \rightarrow -\infty$

3.5 Comment peut-on implémenter pratiquement la minimisation de ce critère et est-ce un critère facile à optimiser ?

4. Un critère plus simple que le précédent est :

$$R_{Var-REX} = \beta Var(\{R_1, ..., R_m\}) + \sum_{e=1}^m R_e$$

Où m est le nombre d'environnements, $\beta \in R^+$ et

$$Var(\{R_1, ..., R_m\}) = \frac{1}{m} \sum_{e=1}^m R_e^2 - \left(\frac{1}{m} \sum_{e=1}^m R_e \right)^2 = \frac{1}{2m^2} \sum_{e=1}^m \sum_{e'=1}^m (R_e - R_{e'})^2$$

4.1 Interpréter le rôle du terme β , que se passe-t-il si $\beta \rightarrow \infty$ et si $\beta \rightarrow 0$