Contrôle de Connaissance
Master Recherche Information, Apprentissage, Cognition - Université Paris-Sud 11
OPT 2, Apprentissage Statistique et Optimisation Approfondis

Mercredi 19 février 2014
Durée : 3h00
Documents autorisés : supports et notes de cours

Les exercices sont indépendants.
Rédigez les différentes parties sur des copies différentes pour faciliter la correction.

Partie I

1 Réseaux neuronaux (4 points)

Q. 1.1 *Définissez un perceptron à d entrées et une sortie. Combien a-t-il de paramètres ? Etant donné une base d'apprentissage*

$$\mathcal{E} = \{(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}, i = 1 \dots n\}$$

écrivez la fonction à minimiser pour apprendre ces paramètres.

Q. 1.2 *Définissez un réseau multi-couches. Peut-on approcher la fonction XOR avec un réseau multi-couches linéaire ?*

Q. 1.3 *Définissez la fonction sigmoide et écrivez sa dérivée.*

Q. 1.4 *Peut-on apprendre les poids d'un réseau neuronal où chaque neurone implémente une fonction sigmoide dans un contexte non-supervisé ? Quelle est la fonction à optimiser (plusieurs réponses possibles) ?*

Q. 1.5 *Est-ce que l'initialisation d'un réseau neuronal a une influence sur le résultat de l'apprentissage ? Pourquoi ?*

Q. 1.6 *Quels sont les hyper-paramètres d'un réseau neuronal à une couche cachée ? Comment les ajuster en fonction du problème ?*

2 Réseau ESN (4 points)

Un Echo State Network (ESN) est un réseau neuronal récurrent défini de manière stochastique. Etant donné d entrées, on choisit : m, nombre de neurones internes ; ρ, probabilité que le neurone i soit connecté au neurone j, avec un poids $w_{i,j}$ de loi Gaussienne ($\mathcal{N}(0, 1)$).

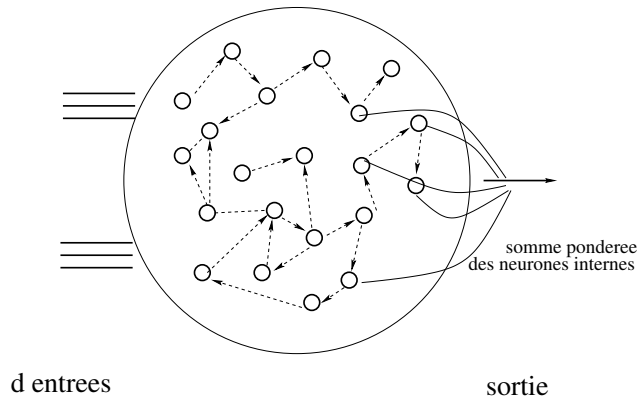

FIGURE 1 – Echo State Network

Toutes les entrées sont connectées aux m neurones internes (avec un poids lui aussi de loi Gaussienne). Tous les neurones internes sont connectés à la sortie. On se limite à apprendre les poids de sortie (la sortie est une somme pondérée des valeurs des neurones internes).

Q. 2.1 *Ecrire le critère à minimiser pour apprendre les poids de sortie. Discutez l'intérêt d'un ESN par rapport à un perceptron.*

Q. 2.2 *Comment valider le résultat d'un ESN ?*

Q. 2.3 *Considérez la matrice $\mathcal{W} = (w_{i,j})$: faut-il prendre des précautions particulières ? (indication, pensez aux valeurs propres / vecteurs propres de $\mathcal{W}$).*

Partie II

3 Optimisation (8 points)

Dans cette partie nous considérons la minimisation de fonctions $f : \mathbf{x} \in \mathbb{R}^n \mapsto f(\mathbf{x}) \in \mathbb{R}$. Les coordonnées d'un vecteur $\mathbf{x}$ de $\mathbb{R}^n$ sont notées $\mathbf{x} = (x_1, \ldots, x_n)$, i.e. pour tout i, $x_i \in \mathbb{R}$. Nous avons vu en cours que des algorithmes évolutionnaires pour optimiser de telles fonctions sont les strategies d'évolutions (ES) qui génèrent à chaque itération t, λ nouveaux points de $\mathbb{R}^n$ en échantillonnant un vecteur Gaussien centré autour de X_t (que l'on suppose unique) :

$$X_t^i = X_t + \sigma \mathcal{N}_i(0, C) \text{ pour } i = 1, \ldots \lambda \ , \tag{1}$$

où $\sigma > 0$ et C est une matrice symétrique définie positive (matrice de covariance du vecteur Gaussien). L'indice i dans $\mathcal{N}_i(0, C)$ souligne le fait que les vecteurs Gaussiens sont tirés indépendamment pour chaque enfant.

Q. 3.1 *Donner la forme géométrique des lignes d'iso-densité d'un vecteur Gaussien $\mathcal{N}(0, C)$. Relier cette forme géométrique aux vecteurs propres et valeurs propres de C.*

Nous considérons les fonctions tests suivantes :
- $f_1(\mathbf{x}) = x_1^2 + 10^6 \sum_{i=2}^n x_i^2$
- $f_2(\mathbf{x}) = 10^6 x_1^2 + \sum_{i=2}^n x_i^2$
- $f_3(\mathbf{x}) = f_1(\mathbf{R}\mathbf{x})$
- $f_4(\mathbf{x}) = f_2(\mathbf{R}\mathbf{x})$ où pour les fonctions f_3 et f_4 la matrice $\mathbf{R} \in \mathcal{M}_n(\mathbb{R})$ est une matrice de rotation tirée aléatoirement uniformément dans l'ensemble des matrices de rotations.

Q. 3.2 *Quelles sont les propriétés de ces fonctions tests ?*

L'algorithme CMA-ES est utilisé pour minimiser les fonctions f_1, f_2, f_3 et f_4 en dimension $n = 5$. L'algorithme est utilisé avec ses paramètres par défaut. Le vecteur moyenne initial est égal à $(1, 1, 1, 1, 1)$ et le step-size initial égal à 10, la matrice de covariance initiale est égale à la matrice identité. La visualisation d'un run de l'algorithme minimisant chacune de ces 4 fonctions est présenté à la Figure 2.

Q. 3.3 *Expliquer ce qui est présenté sur chacun des 4 graphiques de visualisation. Pour le graphique en haut à gauche on ne s'intéressera qu'à la courbe bleue et la courbe verte.*

Q. 3.4 *Identifier les fonctions correspondant aux runs présentés en Figure 2 (a), (b), (c) et (d). Les réponses doivent être soigneusement justifiées.*

Q. 3.5 *Décrire en détail ce qui se passe pour le run associé au graphique Figure 2 (c).*

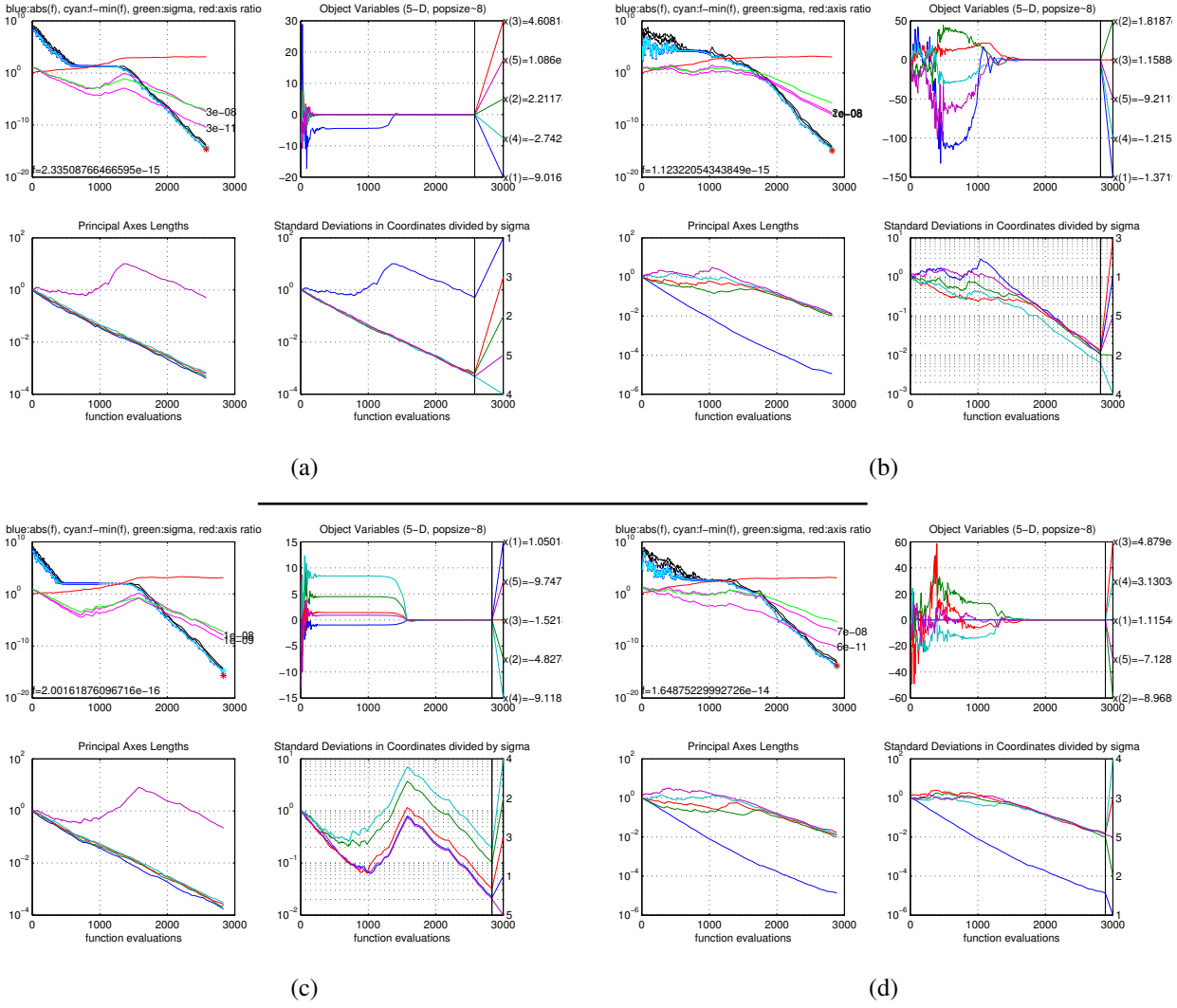

FIGURE 2 – Single runs of the CMA-ES algorithm on the functions f_1, f_2, f_3 and f_4. Identify the function corresponding to each run.

4 Partie III (10 points)

4.1 Choix multiples (5 points)

Chaque réponse correcte vaut 1 point. Chaque réponse incorrecte vaut -0.25 points.

Q. 4.1 *Lequel des énoncés suivants est faux.*

A Le sur-apprentissage (overfitting) est dangereux parce qu'il est impossible à détecter.

B En général, le sur-apprentissage augmente avec la dimensionnalité de l'espace d'entrées.

C En général, le sur-apprentissage diminue avec la taille des données d'entraînement.

D En général, le sur-apprentissage augmente avec la complexité de la classe de fonctions.

Q. 4.2 *Quel est le temps d'execution asymptotique d'AdaBoost avec des stumps de décision (T : nombre d'itérations ; n : nombre de points d'entraînement ; d : nombre de traits (attributs, features))*

A $O\big(ndT \log n\big)$

B $O\big(nd(nT + \log n)\big)$

C $O\big(n^2 dT\big)$

D $O\big(nd(T + \log n)\big)$

Q. 4.3 *Lequel des énoncés suivants est vrai.*

A AdaBoost trouve toujours un classifieur avec une erreur d'entrainement zéro.

B AdaBoost ne trouve jamais un classifieur avec une erreur d'entrainement zéro.

C AdaBoost trouve toujours un classifieur avec une erreur d'entrainement zéro après $O(\ln n)$ itérations ou n est le nombre de points d'entrainement.

D AdaBoost trouve un classifieur avec une erreur d'entrainement zéro après $O(\ln n)$ itérations s'il esiste $\delta > 0$ tel que l'erreur de chaque classifieur de base est au plus $1/2 - \delta$.

Q. 4.4 *Le théorème de Bayes permet de calculer la loi à posterioi $p(\theta \mid x)$ à partir de la vraisemblance $p(x \mid \theta)$, la loi à priori $p(\theta)$, et la preuve $p(x)$:*

$$p(\theta \mid x) = \frac{p(x \mid \theta)p(\theta)}{p(x)}.$$

L'algorithme de Metropolis-Hastings est utile pour échantillonner $p(\theta \mid x)$ parce que souvent

A $p(x \mid \theta)$ est incalculable,

B $p(\theta)$ est incalculable,

C $p(x)$ est incalculable,

D $p(x \mid \theta)$ et $p(x)$ sont incalculables mais $p(x \mid \theta)/p(x)$ est calculable.

Q. 4.5 *Lequel des énoncés suivants est vrai.*

A L'algorithme de Metropolis-Hastings génère une chaine iid (des points indépendants et identiquement distribués).

B Pour un importance sampling *efficace, la distribution instrumentale q doit être proche a la distribution de cible π, et q doit avoir une queue plus lourde que π*

C L'importance sampling est inéfficace parce qu'il n'accepte pas tout les points.

D L'algorithme Metropolis adaptatif génère une chaine markovienne.

4.2 Algorithmique (5 points)

Un "stump de regression" est une fonction réelle de la forme

$$h(x) = \begin{cases} a & \text{si } x \geq \theta, \\ b & \text{sinon.} \end{cases}$$

On a un ensemble de données d'apprentissage $D_n = \big((x_1, y_1), \ldots, (x_n, y_n)\big)$ ou les x_i's sont des observations de valeur réelle, $x_1 < x_2 < \ldots < x_n$, et les y_i's sont des cibles de valeur réelle. L'objectif de la régression est de minimiser l'erreur carée

$$R(h) = \frac{1}{n} \sum_{i=1}^{n} \big(h(x_i) - y_i\big)^2.$$

Proposez un algorithme qui trouve le meilleur stump de regression (c.à.d. a, b et θ) en temps $O(n)$.