FONDEMENTS DE L'APPRENTISSAGE MACHINE (IFT3395/6390)

Professeur: Pascal Vincent

Examen Final

Vendredi 24 avril 2009

Durée: 2h45

Toute documentation papier est permise (livre, notes de cours, ...)

Prénom:

Nom:

Code permanent:

IFT 3395 ou 6390?

Le total de l'examen est sur 100 pts.

Veuillez répondre aux questions dans les zones de blanc laissées à cet effet.

Notations

Les notations suivantes sont définies pour tout l'examen, là où elles ont un sens:
On suppose qu'on dispose d'un ensemble de données de n exemples: $D_n = \{z^{(1)}, ..., z^{(n)}\}$. Dans le cas supervisé chaque exemple $z^{(i)}$ est constitué d'une paire *observation, cible:* $z^{(i)} = (x^{(i)}, t^{(i)})$, alors que dans le cas non-supervisé, on n'a pas de notion de cible explicite donc juste un vecteur d'observation: $z^{(i)} = x^{(i)}$.
On suppose que chaque observation est constituée de d traits caractéristiques (composantes): $x^{(i)} \in \mathbb{R}^d$: $x^{(i)} = (x_1^{(i)}, ..., x_d^{(i)})$

ATTENTION: on vous demande de respecter *scrupuleusement* la notation définie dans cet énoncé. Donc avant de recopier directement des formules de vos notes de cours, assurez-vous d'y substituer la bonne notation! Sans quoi on pourrait conclure que vous ne comprenez pas à quoi correspond ce que vous écrivez... donc prenez le temps de bien assimiler la notation ci-dessus. Par exemple souvenez-vous qu'ici la cible est notée t (ou $t^{(i)}$ s'il s'agit du ième exemple).

1 Algorithmes d'apprentissage appropriés pour différents types de problèmes (8 pts)

1.1 Apprentissage supervisé

Indiquez les *différents types de problèmes* d'apprentissage que vous connaissez que l'on considère comme des problèmes d'apprentissage « supervisé ».
Pour *chaque* type de problème: a) expliquez d'abord dans vos mots de quoi il s'agit, ce qui le caractérise. b) nommez tous les *algorithmes d'apprentissage* que nous avons vus ou que vous connaissez qu'on peut appliquer **directement** pour ce type de problème (une liste de noms suffit).

1.2 Apprentissage non supervisé

Indiquez les *différents types de problèmes* d'apprentissage que vous connaissez que l'on considère comme des problèmes d'apprentissage « **non supervisé** ».
Pour *chaque* type de problème: a) expliquez dans vos mots en quoi il consiste. **b)** indiquez à quoi cela peut servir en pratique. **c)** nommez tous les *algorithmes d'apprentissage* que nous avons vus ou que vous connaissez qu'on peut appliquer **directement** pour ce type de problème (une liste de noms suffit).

2 Sélection de modèle (9 pts)

Vous êtes embauché dans une entreprise qui réalise des systèmes de vérification d'identité et qui travaille sur un nouveau système de reconnaissance de visages pour un important client. Le système doit être capable de distinguer une dizaine de personnes autorisées et les différencier de toute autre personne non autorisée. L'entreprise dispose d'une base de donnée comportant 200 000 images de visages étiquetés (identifiés comme autorisé ou non autorisé). Un collègue vient vous voir et vous dit qu'il a essayé 3 variantes d'algorithme de classification (par exemple des réseaux de neurones avec un nombre différent de neurones cachés), qu'il a entraîné sur ces 200 000 images. Le premier obtenait 4% d'erreur, le second 2%, et le 3ème 0.3% d'erreur sur les 200 000 images. Puisque son expérience montre clairement que le 3ème a une performance bien meilleure, c'est donc celui-là qu'il veut utiliser dans le nouveau système.

2.1 Êtes-vous d'accord avec lui? Expliquez justifiez votre réponse.

2.2 Si vous n'êtes pas d'accord, comment proposeriez-vous à votre collègue de procéder pour décider laquelle des variantes utiliser? Expliquez en détail.

2.3 Le client vous demande une estimation fiable de la performance à laquelle il pourra s'attendre du système sur le terrain. Comment vous y prendriez-vous pour la lui fournir?

3 Concepts graphiques (21 pts)

Considérons l'apprentissage d'une fonction paramétrée f, avec un ensemble de paramètres $\theta = \{\theta_1, ..., \theta_M\}$, des hyper-paramètres $\lambda = \{\lambda_1, ..., \lambda_K\}$, et une fonction de perte correspondante (pour un exemple z): $L(z, f)$. Nous avons vu un certain nombre de notions qui correspondent à un concept géométrique ou graphique. Je fais référence ici aux notions suivantes: **1.** Région de décision. **2.** Courbes d'apprentissage. **3.** Paysage de coût (d'erreur) ou surface de de coût (d'erreur).
Pour **chacune** de ces notions, tour à tour, on vous demande:

a) D'expliquer brièvement la notion dans vos propres mots.

b) De dire pour quel(s) problème(s) d'apprentissage elle s'applique (pour lesquels elle a un sens): clas20sification, régression, estimation de densité?

c) Donnez une expression mathématique de l'objet graphique ou géométrique (l'ensemble de points) que cette notion représente. Utilisez pour cela les notations définies au début de l'examen et plus haut dans cette question, en expliquant à quoi correspond toute nouvelle notation que vous introduisez.

d) Illustrez le concept par une figure, en indiquant clairement quelle quantité est représentée sur chaque axe. Expliquez en une phrase quelles sont les limitations de cette illustration en ce qui a trait à la dimensionalité des quantités impliquées.

e) Écrivez, sous forme d'un bref pseudo-code de haut niveau (sans vous perdre dans les détails), l'algorithme qui permettrait de tracer un tel graphique.

Soyez bref mais précis, et numérotez clairement vos réponses: a) b) c) d) e)

3.1 Concept de région de décision

3.2 Concept de courbes d'apprentissage

3.3 Concept de paysage (ou surface) de coût (ou d'erreur)

4 Réseaux de neurones (25 pts)

On considère un réseau de neurones, paramétré par un ensemble de paramètres θ, comme une fonction $f_\theta(x)$. Pour une entrée $x \in \mathbb{R}^d$, il produit une sortie $y = f_\theta(x)$.

4.1 Questions générales

4.1.1 Pour chacun des problèmes suivants, si vous vouliez utiliser un réseau de neurones, on vous demande de préciser:

- quelle *non-linéarité de sortie* vous utiliseriez (si besoin): nommez-là et écrivez la formule de son calcul. Expliquez ce que représente alors y.
- quelle *fonction de perte* vous utiliseriez: nommez-là et donnez son expression mathématique $L(y, t)$

a) pour un problème de régression

b) pour un problème de classificaiton à 2 classes

c) pour un problème de classification à m classes, avec $m > 2$

4.1.2 Dans un modèle de réseau de neurones on dispose de plusieurs leviers (hyper-paramètres) et techniques pour *contrôler la capacité* du modèle et ainsi gérer le risque de sur-apprentissage. Quels sont ces leviers et techniques? Nommez-les, expliquez précisément ce qu'ils représentent et, pour chacun, indiquez le sens de l'effet du levier, c.a.d. si le fait de l'**augmenter** va *augmenter la capacité* du modèle (et le risque de sur-apprentissage) ou au contraire *diminuer la capacité* (et augmenter le risque de sous-apprentissage).

4.2 Réseaux de neurones de type Radial Basis Function

Pour les réseaux de type Perceptron Multicouche (MLP) vus en cours, un neurone N_k de la première couche cachée reçoit une entrée x et a un vecteur de poids synaptiques $w^{(k)} \in \mathbb{R}^d$ et un biais $b^{(k)} \in \mathbb{R}$. Il calcule sa sortie h_k avec la formule $h_k = \text{sigmoid}(< w^{(k)}, x > + b^{(k)})$, où $< w^{(k)}, x >$ dénote le produit scalaire usuel.
On s'intéresse pour cette partie à un type de réseaux de neurones différent, nommé RBF (Radial Basis Function). Ces réseaux à une couche cachée sont très similaires aux MLP. La différene est qu'un neurone RBF N_k de la couche cachée, ayant un vecteur de poids w_k calcule sa sortie h_k ainsi:

$$\begin{aligned} h_k &= \exp(-\beta \|x - w^{(k)}\|^2) \\ &= \exp\left(-\beta \sum_{j=1}^{d} (x_j - w_j^{(k)})^2\right) \end{aligned}$$

où exp désigne l'exponentielle et β est un hyper-paramètre (le même pour tous les neurones de la première couche cachée. Remarquez aussi qu'il n'y a **pas de biais**. Une unique couche cachée de m neurones RBF ayant des sorties $(h_1, ..., h_m) = h$ est typiquement suivie d'une couche de sortie linéaire avec des poids $(a_1, ..., a_m) = a$ pour donner une sortie $y = f_\theta(x) = < a, h > = \sum_{k=1}^{m} a_k h_k$.

4.2.1 Quel est l'ensemble θ des paramètres (excluant les hyper-paramètres) d'un tel réseau RBF?

$\theta = \{$ $\}$

A combien de nombre réels ajustables cela correspond-t-il?

4.2.2 Le coût pour un exemple x pour lequel le réseau prédit $f_\theta(x)$ alors que la vraie cible est t est donné par une fonction de coût différentiable $L(f_\theta(x), t)$. On cherche les valeurs des paramètres qui vont minimiser le coût empirique moyen sur un ensemble d'apprentissage $D_n = \{(x^{(1)}, t^{(1)}), ..., (x^{(n)}, t^{(n)})\}$. Exprimez ce problème de minimisation, puis nommez la technique qu'on va typiquement utiliser pour trouver une solution, et détaillez-là brièvement sous la forme d'un pseudo-code de haut niveau.

4.2.3 On s'intéresse au gradient, c.a.d la dérivée partielle du coût L par rapport aux paramètres. **on suppose qu'on a déjà calculé** $\frac{\partial L}{\partial y}$, et on va rétropropager le gradient. Exprimez et calculez (en fonction de a_k, h_k, et $\frac{\partial L}{\partial y}$):

$$\frac{\partial L}{\partial a_k} =$$

$$\frac{\partial L}{\partial h_k} =$$

puis, en fonction (entre autres) de $\frac{\partial L}{\partial h_i}$
Rappel de la formule pour dérriver une exponentielle: $\exp(u)' = u' \exp(u)$, *ou encore:* $\frac{\partial \exp(u)}{\partial \theta} = \frac{\partial u}{\partial \theta} \exp(u)$

$$\frac{\partial L}{\partial w_j^{(k)}} =$$

4.2.4 Vérificaiton du gradient par différence finie

a) Expliquez brièvement dans vos mots la technique de vérificaiton du calcul du gradient par différences finies. **b)** Pour quoi est-elle utile? **c)** Ecrivez un pseudo-code de haut niveau pour calculer le gradient par différences finies.

5 Mélange de Gaussiennes (20 pts)

On considère, en dimension d, un mélange de k Gaussiennes avec des matrices de covariance **diagonales**.

5.1 A quoi sert un mélange de Gaussienne: pour quel type de problème d'apprentissage s'en sert-on? Dans quels cas un mélange de Gaussiennes est-il plus approprié qu'une unique Gaussienne?

5.2 Chacune des Gaussiennes **diagonales** de ce mélange a des paramètres: nommez-les et indiquez leur dimension. A combien de nombre réels ajustables cela correspond-t-il?

5.3 En plus des paramètres de chaque Gaussienne, quels autres paramètres y a-t-il dans un tel mélange? En tout combien y a-t-il de nombre réels ajustables dans les paramètres d'un tel mélange de Gaussiennes?

5.4 Écrivez la formule permettant de calculer la densité donnée par le mélange de Gaussiennes en un point $x \in \mathbb{R}^d$ (avec les notations que vous avez utilisé ci-haut pour représenter les paramètres).

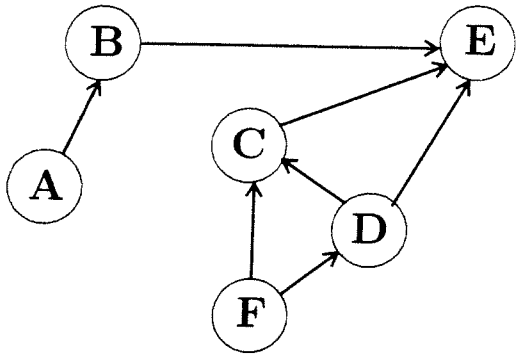
5.5 On suppose qu'on dispose de deux procédures informatiques fournies par une librairie logicielle:

- une fonction `tirageGaussienne` reçoit en paramètre les paramètres d'une Gaussienne et retourne un point de $\mathbb{R}^d$ tiré aléatoirement selon cette distribution Gaussienne.
- une fonction `tirageDiscret` reçoit en paramètre un vecteur de probabilités sommant à 1, effectue un tirage selon ces probabilités discrètes et retourne l'indice entier correspondant. Ainsi par ex., si on lui passe en paramètre le vecteur $(0.30, 0.50, 0.20)$, la procédure retournera la valeur 1 dans 30% des appels, la valeur 2 dans 50% des appels et la valeur 3 dans 20% des appels.

Écrivez, sous forme de pseudo-code, la procédure permettant de générer un points $x \in \mathbb{R}^d$ selon la distribution de mélange de Gaussiennes paramétrée telle que vous l'avez définie plus haut.

6 Modèle graphique probabiliste (8 pts)

Soit le modèle graphique probabiliste suivant (réseau Bayesien).
On rappelle que chaque noeud du graphe représente une variable aléatoire.



a) Écrivez la probabilité jointe de tous les noeuds du graphe sous la forme du produit de probabilités (conditionnelles) qu'implique ce modèle graphique.

b) On suppose qu'on sait effectuer des tirages aléatoires (générer des valeurs) selon les lois $P(F)$ et $P(A)$ et selon *chacune* des probabilités conditionnelles que vous avez indiquées dans votre décomposition ci-dessus. Écrivez sous forme d'un bref algorithme (pseudo-code), comment on pourrait générer une matrice de donnée $\boldsymbol{X}$ comportant n rangées de 6 éléments, où chaque rangée correspondrait à un vecteur d'observations $\boldsymbol{X}_i = (a_i, b_i, c_i, d_i, e_i, f_i)$ distribué selon la loi $P(A, B, C, D, E, F)$ ci-dessus. Utilisez la notation $a \sim P(A|B = b, C = c, \text{etc...})$ pour représenter le tirage aléatoire d'une valeur a selon une loi conditionnelle.

7 Bagging (9 pts)

Expliquez brièvement le principe du bagging appliqué à un ensemble de données D, avec un algorithme d'apprentissage $\mathcal{A}$. On suppose que $\mathcal{A}$ retourne, pour tout ensemble de données D', une fonction de prédiction $f_{D'} = \mathcal{A}(D')$. (ex: si $\mathcal{A}$ est l'algorithme du perceptron, alors $\mathcal{A}(D')$ retournerait une fonction de décision linéaire dont les paramètres sont ajustés à l'ensemble D').

a) Décrivez, sous forme d'un pseudo-code de haut niveau, la procédure d'entraînement

b) Décrivez comment prendre une décision pour un point de test x.

c) Avec quel type d'algorithme d'apprentissage $\mathcal{A}$ est-il particulièrement recommandé d'utiliser une méthode d'ensemble telle que le Bagging? Pourquoi? (Quel problème de cet algorithme cela permet-il d'atténuer?)