Apprentissage et reconnaissance (GIF-4101 / GIF-7005)
Département de génie électrique et de génie informatique
Automne 2011

UNIVERSITÉ LAVAL

EXAMEN FINAL

Instructions : – Une feuille aide-mémoire recto-verso manuscrite est permise ;
– Durée de l'examen : 2 h 50.

Pondération : – Cet examen compte pour 35% de la note finale.
– La note est saturée à 100% si le total des points avec bonus excède cette valeur.

Question 1 (15 points sur 100)

Une matrice de décision $\mathbf{W}$, de taille $K \times L$, permet de combiner les décisions d'un ensemble de L classifieurs à deux classes, pour faire du classement de données à K classes. L'équation de décision basée sur cette matrice est la suivante :

$$\bar{\mathrm{h}}_i(\mathbf{x}) = \sum_{j=1}^{L} w_{i,j} \mathrm{h}_{j,i}(\mathbf{x}),$$

où :
— $\mathrm{h}_{j,i}(\mathbf{x})$ est le j-ème classifieur de base de l'ensemble ;
— $w_{i,j}$ est l'élément à la position (i,j) dans la matrice de décision $\mathbf{W}$;
— $\bar{\mathrm{h}}_i(\mathbf{x})$ est la décision combinée de l'ensemble pour la classe C_i.

(5) (a) Supposons que l'on veut résoudre un problème à $K = 3$ classes à l'aide d'un ensemble de classifieurs à deux classes combinés selon la méthode *un contre tous* (en anglais, *one against all*). Donnez le nombre de classifieurs à deux classes à utiliser ainsi que la matrice de décision $\mathbf{W}$ correspondant à cette configuration.

Solution: Avec un ensemble de classifieurs de type *un contre tous*, le nombre de classifieur à utiliser correspond au nombre de classes traitées, soit $L = K = 3$ classifieurs. La matrice de décision à utiliser pour cette configuration est la suivante :

$$\mathbf{W} = \begin{bmatrix} +1 & -1 & -1 \\ -1 & +1 & -1 \\ -1 & -1 & +1 \end{bmatrix}.$$

(5) (b) Supposons maintenant que l'on veut résoudre ce problème à $K = 3$ classes toujours à l'aide d'un ensemble de classifieurs à deux classes, mais cette fois en combinant les classifieurs selon la méthode de *séparation par paires* (en anglais, *pairwise separation*).

Donnez le nombre de classifieurs à deux classes à utiliser ainsi que la matrice de décision $\mathbf{W}$ correspondant à cette configuration.

Solution: Le nombre de classifieurs à utiliser correspond au nombre de combinaisons de classes possible, en tenant compte de la symmétrie entre les classifieurs (discriminer C_i de C_j correspondant à discriminer C_j de C_i à un signe près), soit $L = K(K-1)/2 = 3$ classifieurs. La matrice de décision correspondante est la suivante :

$$\mathbf{W} = \begin{bmatrix} +1 & +1 & 0 \\ -1 & 0 & +1 \\ 0 & -1 & -1 \end{bmatrix}.$$

(5) (c) Finalement, supposons que l'on veut résoudre ce problème à $K = 3$ classes d'un ensemble redondant de $L = 7$ classifieurs, avec un matrice de décision basée sur un code à correction d'erreur (en anglais, *error code output correction*). Donnez la matrice de décision $\mathbf{W}$ correspondant à cette configuration. Déterminez également le nombre d'erreurs de classement des classifieurs de base que cette configuration de système peut tolérer sans se tromper.

Solution: Une configuration possible de matrice de décision à code de correction d'erreur est la suivante :

$$\mathbf{W} = \begin{bmatrix} -1 & -1 & -1 & +1 & +1 & +1 & +1 \\ -1 & +1 & +1 & -1 & -1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 \end{bmatrix}.$$

La distance de Hamming mininale entre chaque combinaison de lignes est de 4. Ceci indique donc que cette configuration de système peut tolérer une erreur de classement ($\lfloor (4-1)/2 \rfloor = 1$) par les classifieurs de base sans prendre une mauvaise décision de classement.

Question 2 (20 points sur 100)

Soit un réseau de neurones de type RBF pour deux classes, composé d'une couche cachée de R neurones de type gaussien, suivi d'une couche de sortie avec un neurone avec fonction de transfert linéaire. La valeur de la sortie pour un tel réseau de neurones pour une valeur d'entrée $\mathbf{x}$ est donnée par l'équation suivante,

$$\mathrm{h}(\mathbf{x}) = \sum_{i=1}^{R} w_i \phi_i(\mathbf{x}) + w_0 = \sum_{i=1}^{R} w_i \exp\left[-\frac{\|\mathbf{x} - \mathbf{m}_i\|^2}{2s_i^2}\right] + w_0,$$

où :
— $\mathbf{m}_i$ est la valeur du centre du i-ème neurone gaussien de la couche cachée ;
— s_i est l'étalement du i-ème neurone gaussien ;

— w_i est le poids connectant le i-ème neurone gaussien de la couche cachée au neurone de sortie ;
— w_0 est le poids-biais du neurone de sortie.

Supposons que l'on fixe la valeur de l'étalement s_i à une valeur prédéterminée, et que l'on veut apprendre les valeurs w_i, w_0 et $\mathbf{m}_i$ par descente du gradient, en utilisant comme erreur le critère du perceptron,

$$E_{percp} = -\sum_{\mathbf{x}^t \in \mathcal{Y}} r^t \, \mathrm{h}(\mathbf{x}^t),$$

où :

— r^t est la valeur désirée pour le neurone de sortie du réseau, soit $r^t = 1$ si $\mathbf{x}^t$ appartient à la classe C_1 et $r^t = -1$ autrement ;
— $\mathcal{Y} = \{\mathbf{x}^t \in \mathcal{X} \,|\, r^t \, \mathrm{h}(\mathbf{x}^t) < 0\}$ est l'ensemble des données $\mathbf{x}^t$ du jeu $\mathcal{X}$ qui sont mal classée par le réseau.

(10) (a) Développez les équations permettant de mettre à jour les poids w_i et w_0 du neurone de sortie par descente du gradient, en utilisant comme erreur le critère du perceptron.

Solution:

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial(-\sum_{\mathbf{x}^t \in \mathcal{Y}} r^t(\sum_{i=1}^{R} w_i \phi_i(\mathbf{x}^t) + w_0))}{\partial w_i} = -\sum_{\mathbf{x}^t \in \mathcal{Y}} r^t \phi_i(\mathbf{x}^t) \\
\Delta w_i &= -\eta \frac{\partial E}{\partial w_i} = \eta \sum_{\mathbf{x}^t \in \mathcal{Y}} r^t \phi_i(\mathbf{x}^t) \\
\frac{\partial E}{\partial w_0} &= \frac{\partial(-\sum_{\mathbf{x}^t \in \mathcal{Y}} r^t(\sum_{i=1}^{R} w_i \phi_i(\mathbf{x}^t) + w_0))}{\partial w_0} = -\sum_{\mathbf{x}^t \in \mathcal{Y}} r^t \\
\Delta w_0 &= -\eta \frac{\partial E}{\partial w_0} = \eta \sum_{\mathbf{x}^t \in \mathcal{Y}} r^t \\
w_i &= w_i + \Delta w_i, \quad i = 0, \ldots, R
\end{aligned}$$

(10) (b) Développez les équations permettant de mettre à jour les valeurs des centres $\mathbf{m}_i$ des neurones gaussiens de la couche cachée par descente du gradient, en utilisant comme erreur le critère du perceptron.

Solution:

$$
\begin{aligned}
\frac{\partial \phi_i}{\partial m_{i,j}} &= \frac{\partial \exp\left[-\frac{\|\mathbf{x}^t-\mathbf{m}_i\|^2}{2s_i^2}\right]}{\partial m_{i,j}} = \frac{(x_j^t - m_{i,j})}{s_i^2} \exp\left[-\frac{\|\mathbf{x}^t-\mathbf{m}_i\|^2}{2s_i^2}\right] \\
&= \frac{(x_j^t - m_{i,j})}{s_i^2}\phi_i(\mathbf{x}^t) \\
\frac{\partial E}{\partial m_{i,j}} &= \frac{\partial(-\sum_{\mathbf{x}^t\in\mathcal{Y}} r^t(\sum_{l=1}^R w_l\phi_l(\mathbf{x}^t)+w_0))}{\partial m_{i,j}} \\
&= -\sum_{\mathbf{x}^t\in\mathcal{Y}} r^t\left(w_i\frac{\partial\phi_i(\mathbf{x}^t)}{\partial m_{i,j}}\right) = -\sum_{\mathbf{x}^t\in\mathcal{Y}} r^t w_i \frac{(x_j^t - m_{i,j})}{s_i^2}\phi_i(\mathbf{x}^t) \\
\Delta m_{i,j} &= -\eta\frac{\partial E}{\partial m_{i,j}} = \eta\sum_{\mathbf{x}^t\in\mathcal{Y}} r^t w_i \frac{(x_j^t - m_{i,j})}{s_i^2}\phi_i(\mathbf{x}^t) \\
m_{i,j} &= m_{i,j} + \Delta m_{i,j}, \quad i = 1,\ldots,R, \quad j = 1,\ldots,D
\end{aligned}
$$

Question 3 (25 points sur 100)

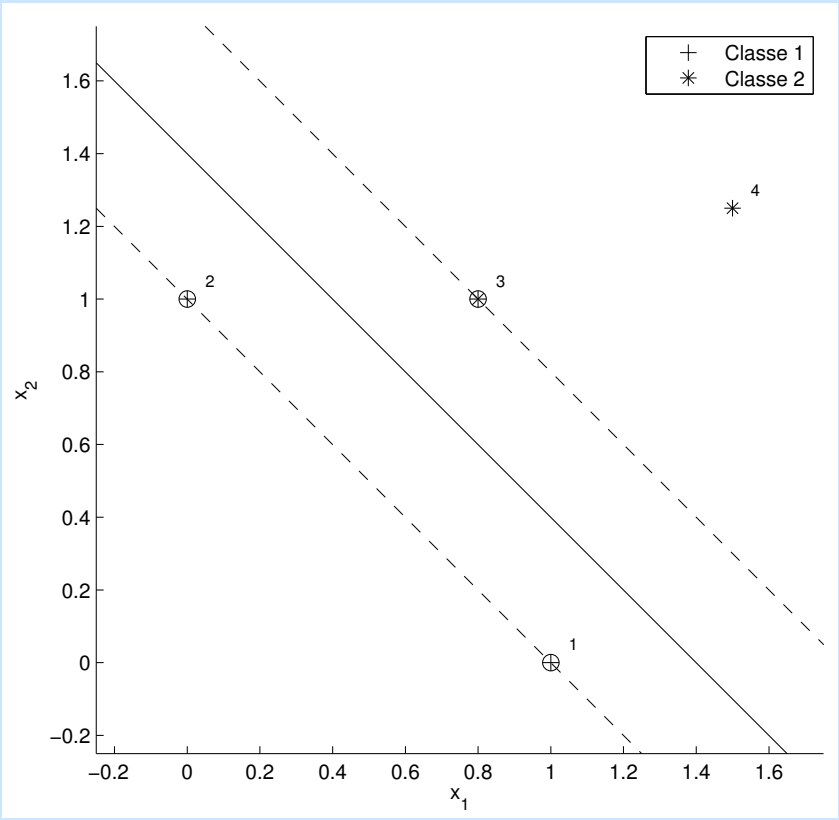
Soit le jeu de données $\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^4$ présenté ici-bas.

$$
\underset{r^1=-1,}{\mathbf{x^1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}}, \quad \underset{r^2=-1,}{\mathbf{x^2} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}}, \quad \underset{r^3=1,}{\mathbf{x^3} = \begin{bmatrix} 0{,}8 \\ 1 \end{bmatrix}}, \quad \underset{r^4=1.}{\mathbf{x^4} = \begin{bmatrix} 1{,}5 \\ 1{,}25 \end{bmatrix}},
$$

Supposons que l'on veut classer ces données avec un classifieur de type Séparateur à vastes marges (SVM) utilisant un noyau linéaire ($K(\mathbf{x},\mathbf{x}') = \langle \mathbf{x},\mathbf{x}'\rangle = (\mathbf{x}')^T\mathbf{x}$), sans marge floue.

(5) (a) Tracez dans votre cahier de réponse les données du jeu $\mathcal{X}$, l'hyper-plan séparateur que l'on obtient avec un SVM sur ces données, les marges géométriques maximales correspondantes et encerclez les données agissant comme vecteurs de support.

Solution:



(10) (b) Donnez les valeurs des poids $\mathbf{w}$ et biais w_0 correspondant au discriminant linéaire maximisant les marges géométriques tracées en (a).

Solution: Trois données sont identifiées comme vecteurs de support : $\mathbf{x}^1$, $\mathbf{x}^2$ et $\mathbf{x}^3$. En travaillant directement dans l'espace d'entrée, ceci nous donne trois équations et trois inconnus.

$$\begin{aligned}
\mathrm{h}(\mathbf{x}^1) &= w_2 x_2^1 + w_1 x_1^1 + w_0 = 1w_2 + 0w_1 + w_0 = -1 \\
\mathrm{h}(\mathbf{x}^2) &= w_2 x_2^2 + w_1 x_1^2 + w_0 = 0w_2 + 1w_1 + w_0 = -1 \\
\mathrm{h}(\mathbf{x}^3) &= w_2 x_2^3 + w_1 x_1^3 + w_0 = 1w_2 + 0{,}8w_1 + w_0 = 1
\end{aligned}$$

La résolution de ce système d'équation peut se faire par la suivante.

$$\begin{aligned}
\mathrm{EQ1} &: w_2 + w_0 = -1 \\
\mathrm{EQ2} &: w_1 + w_0 = -1 \\
\mathrm{EQ3} &: w_2 + 0{,}8w_1 + w_0 = 1 \\
\mathrm{EQ3} - \mathrm{EQ1} - 0.8 \times \mathrm{EQ2} &: (1-1)w_2 + (0{,}8-0{,}8)w_1 + (1-1-0{,}8)w_0 = 1+1+0{,}8 \\
&\rightarrow w_0 = -3{,}5 \\
\mathrm{EQ1} - \mathrm{L4} &: (1-0)w_2 + (1-1)w_0 = -1+3{,}5 \\
&\rightarrow w_2 = 2{,}5 \\
\mathrm{EQ2} - \mathrm{L4} &: (1-0)w_1 + (1-1)w_0 = -1+3{,}5 \\
&\rightarrow w_1 = 2{,}5
\end{aligned}$$

La résolution de ce système d'équations nous donne les valeurs suivantes :

$$\mathbf{w} = [2{,}5 \;\; 2{,}5]^T,\; w_0 = -3{,}5.$$

(10) (c) Calculez la valeur des α^t correspondant au discriminant linéaire maximisant les marges géométriques calculé en (a).
Considérez que la valeur de w_0 calculée en (b) est toujours valide et n'oubliez pas que les valeurs des α^t doivent respecter la contrainte suivante : $\sum_t \alpha^t r^t = 0$.

Solution: Le calcul des α^t se fait selon la formule suivante,

$$\mathrm{h}(\mathbf{x}) = \sum_t \alpha^t r^t (\mathbf{x}^t)^T \mathbf{x} + w_0.$$

Dans la cas présent, nous avons trois vecteurs de support, $\mathbf{x}^1$, $\mathbf{x}^2$ et $\mathbf{x}^3$. Nous calculons toutes les combinaisons de produit scalaire par ce qui suit :

$$\begin{aligned}
(\mathbf{x}^1)^T\mathbf{x}^1 &= 1,\\
(\mathbf{x}^1)^T\mathbf{x}^2 &= (\mathbf{x}^2)^T\mathbf{x}^1 = 0,\\
(\mathbf{x}^1)^T\mathbf{x}^3 &= (\mathbf{x}^3)^T\mathbf{x}^1 = 0{,}8,\\
(\mathbf{x}^2)^T\mathbf{x}^2 &= 1,\\
(\mathbf{x}^2)^T\mathbf{x}^3 &= (\mathbf{x}^3)^T\mathbf{x}^2 = 1,\\
(\mathbf{x}^3)^T\mathbf{x}^3 &= 1{,}64.
\end{aligned}$$

Maintenant, nous allons calculer la valeur de $\mathrm{h}(\mathbf{x}^t)$ pour les trois vecteurs de support :

$$\begin{aligned}
\mathrm{h}(\mathbf{x}^1) &= \alpha^1(-1)1 + \alpha^2(-1)0 + \alpha^3(1)0{,}8 - 3{,}5 = -1\\
&\rightarrow \; -\alpha^1 + 0{,}8\alpha^3 = 2{,}5\\
\mathrm{h}(\mathbf{x}^2) &= \alpha^1(-1)0 + \alpha^2(-1)1 + \alpha^3(1)1 - 3{,}5 = -1\\
&\rightarrow \; -\alpha^2 + \alpha^3 = 2{,}5\\
\mathrm{h}(\mathbf{x}^3) &= \alpha^1(-1)0{,}8 + \alpha^2(-1)1 + \alpha^3(1)1{,}64 - 3{,}5 = 1\\
&\rightarrow \; -0{,}8\alpha^1 - \alpha^2 + 1{,}64\alpha^3 = 4{,}5
\end{aligned}$$

Nous avons donc trois équations et trois inconnues, on peut tenter de résoudre ce système par la suivante,

$$\begin{aligned}
\text{EQ1} \quad &: \quad -\alpha^1 + 0{,}8\alpha^3 = 2{,}5\\
&\rightarrow \quad \alpha^1 = 0{,}8\alpha^3 - 2{,}5\\
\text{EQ2} \quad &: \quad -\alpha^2 + \alpha^3 = 2{,}5\\
&\rightarrow \quad \alpha^2 = \alpha^3 - 2{,}5\\
\text{EQ3} \quad &: \quad -0{,}8\alpha^1 - \alpha^2 + 1{,}64\alpha^3 = 4{,}5\\
\text{EQ3} - 0{,}8 \times \text{EQ1} \quad &: \quad (-0{,}8 + 0{,}8)\alpha^1 - \alpha^2 + (1{,}64 - 0{,}64)\alpha^3 = 4{,}5 - 0{,}8 \times 2{,}5\\
&\rightarrow \quad -\alpha^2 + \alpha^3 = 2{,}5
\end{aligned}$$

Il semble donc qu'il y ait un dépendance linéaire entre les équations (EQ3 $= 0{,}8 \times$ EQ1 $+$ EQ2). Pour résoudre ce problème, nous allons introduire une quatrième équation, découlant de la contrainte sur les valeurs du α^t donnée dans l'énoncé de la question,

$$\sum_t \alpha^t r^t = 0 \rightarrow -\alpha^1 - \alpha^2 + \alpha^3 = 0.$$

Ceci nous ramène donc à :

$$\begin{aligned}
\text{EQ4} \quad &: \quad -\alpha^1 - \alpha^2 + \alpha^3 = 0 \\
\text{EQ3} - \text{EQ4} \quad &: \quad (-0{,}8+1)\alpha^1 + (-1+1)\alpha^2 + (1{,}64-1)\alpha^3 = 4{,}5 \\
&\rightarrow \quad 0{,}2\alpha^1 + 0{,}64\alpha^3 = 4{,}5 \\
&\rightarrow \quad 0{,}2(0{,}8\alpha^3 - 2{,}5) + 0{,}64\alpha^3 = 4{,}5 \\
&\rightarrow \quad (0{,}16 + 0{,}64)\alpha^3 = 4{,}5 + 0{,}5 \\
&\rightarrow \quad \alpha^3 = \frac{5}{0{,}8} = 6{,}25 \\
&\rightarrow \quad \alpha^1 = 0{,}8 \times 6{,}25 - 2{,}5 = 2{,}5 \\
&\rightarrow \quad \alpha^2 = 6{,}25 - 2{,}5 = 3{,}75
\end{aligned}$$

Donc, les valeurs de α^t sont : $\alpha^1 = 2{,}5$, $\alpha^2 = 3{,}75$, $\alpha^3 = 6{,}25$ et $\alpha^4 = 0$.

Question 4 (40 points sur 100)

Répondez aussi brièvement et clairement que possible aux questions suivantes.

(4) (a) Il a été indiqué dans le cours qu'il y a un lien entre le classement logistique et les méthodes paramètriques. Sous cette perspective, de quelle façon peut-on interpréter la signification de la valeur de la fonction $\mathrm{h}(\mathbf{x})$ d'une fonction de classement logistique traitant des données organisées selon deux classes ?

Solution: La valeur de la fonction $\mathrm{h}(\mathbf{x})$ d'une fonction de classement logistique pour des données à deux classes correspond en fait à une estimation de la probabilité *a posteriori* que la donnée $\mathbf{x}$ appartienne de la première classe, $\hat{P}(C_1|\mathbf{x})$, sous hypothèse que les données de la classes C_1 et C_2 suivent un loi normale multivariée avec une matrice de covariance partagée.

(4) (b) Avec un Séparateur à vastes marges (SVM) à marge douce, chaque données d'entraînement $\mathbf{x}^t$ a une variable *slack* ξ^t associée. Que signifie une valeur de $0 < \xi^t < 1$ pour une donnée $\mathbf{x}^t$ particulière relativement à son classement par le SVM ?

Solution: Une donnée $\mathbf{x}^t$ pour laquelle la variable *slack* est $0 < \xi^t < 1$ est bien classée par le SVM (du bon côté de l'hyperplan séparateur), mais se retrouve dans la marge du classifieur.

(4) (c) Avec une analyse en composantes principales (ACP) à noyau, quelle est la taille d'un vecteur propre extrait de la matrice de Gram ?

Solution: Un vecteur propre extrait de la matrice de Gram est un vecteur de taille $N \times 1$, où N est le nombre de données de l'ensemble d'entraînement $\mathcal{X}$ utilisé pour bâtir la matrice de Gram.

(4) (d) Indiquez en quoi un jeu de données de validation peut être intéressant pour faire un ajustement d'hyper-paramètres d'un classifieur.

Solution: Un jeu de données de validation permet d'évaluer la performance d'un classifieurs sur des données distinctes des données utilisées en entraînement, afin donc de mesurer la capacité de généralisation du classifieur. Cependant, comme un grand nombre de configurations de classifieurs peut être testé pour faire de l'ajustement d'hyper-paramètres, il est important que ce jeu de données soit distinct du jeu de données de test utilisé pour évaluer les performances finales du classifieur en généralisation.

(4) (e) Dans un processus d'expérimentation, on identifie quatre éléments format l'expérimentation : les entrées, les sorties, les facteurs contrôlables et les facteurs incontrôlables. Dans une contexte de classement où une série d'expériences consiste en l'ajustement des hyper-paramètres du classifieur, donnez un exemple de facteurs contrôlables.

Solution: Les facteurs contrôlables dans ce type d'expérience seraient simplement les hyper-paramètres que l'on veut optimiser.

(4) (f) On dit souvent que l'utilisation d'une matrice de confusion pour évaluer les performances d'un classifieur permet de s'absoudre des probabilités *a priori* des données (balance entre les classes), comparativement à l'utilisation de l'erreur de classement. Expliquez en vos mots pourquoi ceci est possible.

Solution: La matrice de confusion permet d'obtenir les performances de classement par classe, sans ternir compte du nombre de données pour chaque classe. Ainsi, on peut combiner les performances pour les différentes classes de différentes façons, pour tenir compte de différents compromis de performance selon les classes. En opposition, l'erreur de classement fait une combinaison prédéfinie des performances pour chaque classe, selon les probabilités *a priori* des données. Donc, pour des classes peu fréquentes dans les données, l'impact sur le taux d'erreur de classement des données de ces classes peut être faible relativement à ce qui est souhaité.

(4) (g) Dans un perceptron multi-couche avec fonctions de transfert de type sigmoïde pour chaque neurones, il est nécessaire de normaliser les valeur de sortie désirée à des valeurs telles que $\tilde{r}_i^t \in \{0{,}05, 0{,}95\}$. Expliquez pourquoi cette normalisation est nécessaire.

Solution: Cette normalisation est nécessaire étant donné la saturation des neurones. En effet, pour obtenir des valeurs très proches de valeurs désirées telle que 0 ou 1, il faudrait que les valeur entrée de la fonction sigmoïde soit très grande ou très petite, $\lim_{a \to -\infty} \mathrm{f}_{\mathrm{sig}}(a) = 0$ et $\lim_{a \to \infty} \mathrm{f}_{\mathrm{sig}}(a) = 1$. Ceci fait en sorte que les poids en entrées des neurones de sortie ($a = \mathbf{w}^T\mathbf{x} + w_0$) devront être d'une magnitude très élevés, ce qui risque d'exploiter le neurone dans sa zone de saturation où la correction d'erreur par descente du gradient sera inefficace.

(4) (h) Dans l'algorithme AdaBoost, il est indiqué que l'utilisation de *weak learner* permet d'obtenir de bonnes performances. Expliquez ce qu'est précisément un *weak learner* et spécifiez de quelle façon ce type de classifieur peut être intéressant pour l'algorithme AdaBoost.

Solution: Un *weak learner* est un classifieur pas très performant, mais capable d'avoir un taux d'erreur de classement inférieur à 50% pour des données à deux classes tout en étant relativement instable dans son apprentissage. Un tel classifieur est intéressant pour être utilisé avec AdaBoost car il permet d'avoir une bonne diversité dans la réponse des classifieurs obtenus.

(4) (i) Expliquez de quelle façon pourrait-on utiliser une sélection séquentielle vorace avant pour choisir les classifieurs formant un ensemble à partir d'un bassin de classifieurs, dans une approche de type surproduction et sélection.

Solution: On peut sélectionner les classifieurs par une sélection séquentielle vorace avant comme on fait de la sélection de caractéristiques. Il suffit d'abord d'identifier le classifieur du bassin offrant les meilleures performances. Ensuite, on ajoute le classifieur qui combiné au premier de l'ensemble offre le meilleur gain en performance, parmi les classifieurs restant dans le bassin, et ainsi de suite jusqu'à ce que l'on ait sélectionné autant de classifieurs que nécessaires pour former l'ensemble résultant.

(4) (j) Pourquoi une génération d'un ensemble de classifieurs par la maximisation de la corrélation négative permet souvent d'obtenir de bonnes performances de classement ?

Solution: La génération de classifieurs négativement corrélés permet jusqu'à un certain point de réduire la variance de l'ensemble, d'un point de vue compromis biais-variance. L'effet net est donc de réduire l'erreur quadratique de l'ensemble sur les données. Une explication alternative consiste à dire que la corrélation négative permet d'augmenter la diversité dans les réponses des classifieurs, de sorte que les classifieurs ne feront pas les mêmes erreurs de classement.

Question 5 (15 points bonus)

Supposons que l'on veut appliquer l'algorithme Espérance-Maximisation (EM) à un jeu de données en une dimension, où chaque groupe $\mathcal{G}_i$ est décrit par une loi normale $\mathcal{N}(\mu_i,\sigma_i^2)$, soit :

$$p(x|\mathcal{G}_i,\mu_i,\sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right].$$

Donc, la paramétrisation du clustering par EM est donnée par $\Phi = \{\pi_i,\mu_i,\sigma_i\}_{i=1}^K$. En guise de rappel, la formule de l'espérance de vraisemblance de l'algorithme EM est la suivante :

$$\mathcal{Q}(\Phi|\Phi^l) = \sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p(x^t|\mathcal{G}_i,\Phi^l).$$

(5 (bonus)) (a) Donnez le développement complet permettant de calculer les estimations π_i des probabilités *a priori* des groupes.

Solution: Comme π_i est une probabilité, on a la contrainte que $\sum_i \pi_i = 1$. On résout donc par la méthode de Lagrange :

$$\begin{aligned}\frac{\partial \mathcal{Q}(\Phi|\Phi^l)}{\partial \pi_j} &= \frac{\partial}{\partial \pi_j}\left[\sum_t \sum_i h_i^t \log \pi_i - \lambda\left(\sum_i \pi_i - 1\right)\right] \\ &= \sum_t \frac{h_j^t}{\pi_j} - \lambda = 0.\end{aligned}$$

Comme $\sum_i \pi_i = 1$ et $\sum_i h_i^t = 1$:

$$\begin{aligned}\sum_i \pi_i \sum_t \frac{h_i^t}{\pi_i} &= \sum_i \pi_i \lambda, \\ \sum_t \sum_i h_i^t &= \sum_t 1 = N = \lambda, \\ \frac{1}{\pi_i}\sum_t h_i^t - N &= 0, \\ \pi_i &= \frac{\sum_t h_i^t}{N}.\end{aligned}$$

(5 (bonus)) (b) Donnez le développement complet permettant de calculer les estimations m_i des moyennes μ_i.

Solution: Résolution par $\partial \mathcal{Q}(\Phi|\Phi^l)/\partial m_i = 0$:

$$\begin{aligned}\frac{\partial \mathcal{Q}(\Phi|\Phi^l)}{\partial m_j} &= \frac{\partial}{\partial m_j}\sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l) \\ &= \sum_t \frac{\partial}{\partial m_j}\sum_i h_i^t \log \frac{1}{\sqrt{2\pi}s_i}\exp\left[-\frac{(x^t - m_i)^2}{2s_i^2}\right] \\ &= \sum_t \left(\frac{\partial}{\partial m_j}\sum_i h_i^t \log \frac{1}{\sqrt{2\pi}s_i} + \frac{\partial}{\partial m_j}\sum_i h_i^t \left[-\frac{(x^t - m_i)^2}{2s_i^2}\right]\right) \\ &= \sum_t h_j^t \left(-2\frac{x - m_j}{2s_j^2}\right) = \sum_t h_j^t \frac{m_j - x^t}{s_j^2} \\ &= \sum_t h_j^t m_j - \sum_t h_j^t x^t = 0, \\ m_j &= \frac{\sum_t h_j^t x^t}{\sum_t h_j^t}.\end{aligned}$$

(5 (bonus)) (c) Donnez le développement complet permettant de calculer les estimations s_i^2 des variances σ_i^2.

Solution: Résolution par $\partial \mathcal{Q}(\Phi|\Phi^l)/\partial s_i = 0$:

$$\begin{aligned}
\frac{\partial \mathcal{Q}(\Phi|\Phi^l)}{\partial s_j} &= \frac{\partial}{\partial s_j} \sum_t \sum_i h_i^t \log \pi_i + \sum_t \sum_i h_i^t \log p(\mathbf{x}^t|\mathcal{G}_i,\Phi^l) \\
&= \sum_t \frac{\partial}{\partial s_j} \sum_i h_i^t \log \frac{1}{\sqrt{2\pi} s_i} \exp\left[-\frac{(x^t - m_i)^2}{2s_i^2}\right] \\
&= \sum_t \left(\frac{\partial}{\partial s_j} \left[\sum_i h_i^t \log \frac{1}{\sqrt{2\pi}} - \sum_i h_i^t \log s_i \right] + \frac{\partial}{\partial s_j} \sum_i h_i^t \left[-\frac{(x^t - m_i)^2}{2s_i^2} \right] \right) \\
&= -\sum_t \frac{h_j^t}{s_j} + \sum_t h_j^t (-2) \left[-\frac{(x^t - m_j)^2}{2s_j^3} \right] \\
&= -\sum_t \frac{h_j^t}{s_j} + \sum_t h_j^t \frac{(x^t - m_j)^2}{s_j^3} = 0, \\
\sum_t \frac{h_j^t}{s_j} &= \sum_t h_j^t \frac{(x^t - m_j)^2}{s_j^3}, \\
\sum_t h_j^t &= \frac{1}{s_j^2} \sum_t h_j^t (x^t - m_j)^2, \\
s_j^2 &= \frac{\sum_t h_j^t (x^t - m_j)^2}{\sum_t h_j^t}.
\end{aligned}$$

Question 6 (10 points bonus)

Dans le cadre du projet final du cours, une équipe d'étudiants a suivi une méthode d'apprentissage leur permettant d'obtenir d'excellentes performances en classement sur la jeu de données de test MNIST à l'aide d'un perceptron multi-couche avec momentum. Voici la méthode proposée par l'équipe :

1. Partitionner le jeu d'entraînement $\mathcal{X}_{\text{train}}$ MNIST en trois jeux de données disjoints de taille comparable, soit $\mathcal{X}_{\text{train1}}$, $\mathcal{X}_{\text{train2}}$ et $\mathcal{X}_{\text{train3}}$;
2. Partitionner le jeu de test $\mathcal{X}_{\text{test}}$ MNIST en trois jeux de données disjoints de taille comparable, soit $\mathcal{X}_{\text{test1}}$, $\mathcal{X}_{\text{test2}}$ et $\mathcal{X}_{\text{test3}}$;
3. Tester plusieurs valeurs de taux d'entraînement η du perceptron multi-couche sur le jeu $\mathcal{X}_{\text{train1}}$, avec un topologie fixe (une couche cachée de 5 neurones) et un momentum nul ($\alpha = 0$), et sélectionner la valeur du taux d'apprentissage permettant de maximiser les performances sur le premier jeu de test, $\mathcal{X}_{\text{test1}}$;
4. Tester plusieurs valeurs de taux de momentum α du perceptron multi-couche sur le jeu $\mathcal{X}_{\text{train2}}$, avec un topologie fixe (une couche cachée de 5 neurones) et le taux d'apprentissage

η trouvé à l'étape précédente, et sélectionner la valeur du taux de momentum permettant de maximiser les performances sur le deuxième jeu de test, $\mathcal{X}_{\text{test2}}$;

5. Tester différents nombres de neurones sur la couche cachée du perceptron multi-couche sur le jeu $\mathcal{X}_{\text{train3}}$, avec le taux d'apprentissage η trouvé à l'étape 3 et le taux de momentum α trouvé à l'étape 4, et sélectionner la valeur du nombre de neurones permettant de maximiser les performances sur le troisième jeu de test, $\mathcal{X}_{\text{test3}}$;
6. Faire un nouvel apprentissage avec les valeurs « optimale » du taux d'entraînement η, du taux de momentum α et du nombre de neurones sur la couche cachée trouvés aux trois étapes précédentes avec un nouveau perceptron multi-couche sur le jeu complet d'entraînement $\mathcal{X}_{\text{train}}$ et rapporter les performances de ce classifieur sur le jeu complet de test $\mathcal{X}_{\text{test}}$ comme étant les performances finale du classifieur en généralisation.

D'après vous, est-ce que cette méthodologie est valide afin d'évaluer la capacité de généralisation du classifieur ? Justifiez votre réponse.

Solution: Cette méthode n'est pas recevable pour évaluer la capacité de généralisation du classifieur. En effet, l'ajustement des hyper-paramètres du perceptron multi-couche est effectué en maximisant les performances sur des données provenant du jeu de test. De cette façon, les résultats rapportés sur le jeu de test risquent d'être exagérément bons relativement aux véritables performances du classifieur en généralisation, c'est-à-dire des données inconnues mais provenant du même phénomène.