Introduction à l'apprentissage machine (GIF-4101 / GIF-7005)
Département de génie électrique et de génie informatique
Automne 2019

UNIVERSITÉ LAVAL

EXAMEN

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

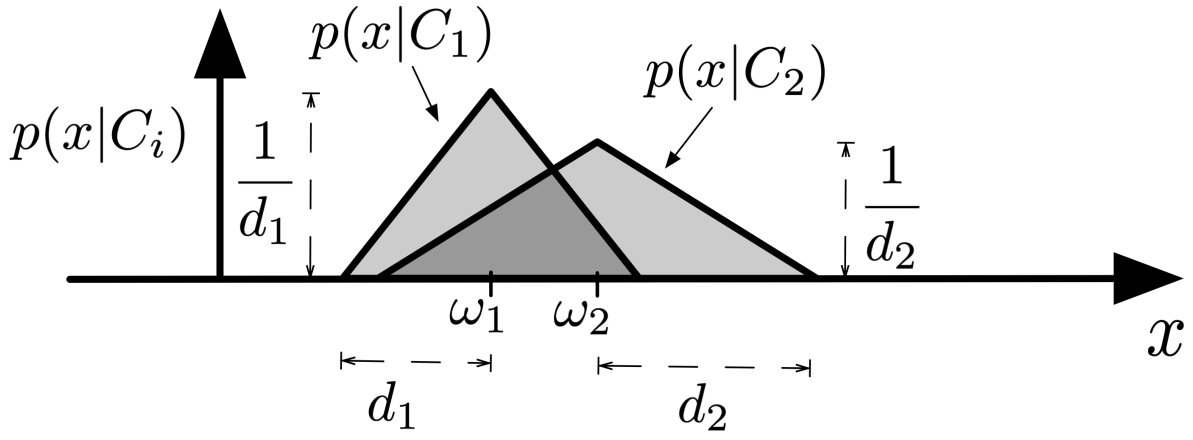
Pondération : Cet examen compte pour 40% de la note finale.

Question 1 (20 points sur 100)

Soit un problème de classement à deux classes et en une dimension, dont les vraisemblances de classe sont décrites par les densités de probabilité suivantes, paramétrée chacune par un centre ω_1 et une demi-largeur d_i :

$$p(x|C_1) = \begin{cases} \frac{x+d_1-\omega_1}{d_1^2} & x \in [\omega_1 - d_1, \omega_1[\\ \frac{-x+d_1+\omega_1}{d_1^2} & x \in [\omega_1, \omega_1 + d_1] \\ 0 & \text{autrement} \end{cases}, \quad p(x|C_2) = \begin{cases} \frac{x+d_2-\omega_2}{d_2^2} & x \in [\omega_2 - d_2, \omega_2[\\ \frac{-x+d_2+\omega_2}{d_2^2} & x \in [\omega_2, \omega_2 + d_2] \\ 0 & \text{autrement} \end{cases},$$

qui sont représentées dans ce qui suit.

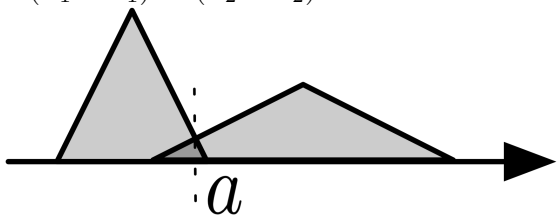


Sans perte de généralité, on suppose que $\omega_2 > \omega_1$.

(9) (a) Supposons que les probabilités a priori sont égales ($P(C_1) = P(C_2) = 0{,}5$). La règle de décision à utiliser pour assigner une donnée x à la classe C_1 ou C_2 selon cette modélisation change selon les configurations des densités, c'est-à-dire les valeurs de ω_1, ω_2, d_1 et d_2. Voici une première configuration possible.

Configuration 1 : $(\omega_1 - d_1) < (\omega_2 - d_2) < (\omega_1 + d_1) < (\omega_2 + d_2)$

$$\mathrm{h}(x) = \begin{cases} C_1 & (\omega_1 - d_1) < x < a, \\ C_2 & a < x < (\omega_2 + d_2), \\ \text{indéfini} & \text{autrement.} \end{cases}$$

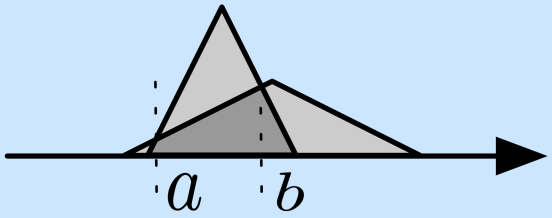


De façon similaire, donnez la règle de décision et le tracé des densités pour les trois autres configurations possibles.

Solution:

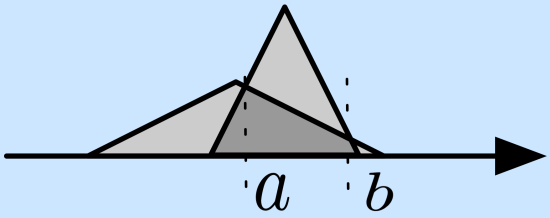
Configuration 2 : $(\omega_2 - d_2) < (\omega_1 - d_1) < (\omega_1 + d_1) < (\omega_2 + d_2)$

$$\mathrm{h}(x) = \begin{cases} C_2 & (\omega_2 - d_2) < x < a, \\ C_1 & a < x < b, \\ C_2 & b < x < (\omega_2 + d_2), \\ \text{indéfini} & \text{autrement.} \end{cases}$$



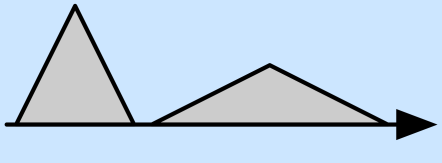
Configuration 3 : $(\omega_1 - d_1) < (\omega_2 - d_2) < (\omega_2 + d_2) < (\omega_1 + d_1)$

$$\mathrm{h}(x) = \begin{cases} C_1 & (\omega_1 - d_1) < x < a, \\ C_2 & a < x < b, \\ C_1 & b < x < (\omega_1 + d_1), \\ \text{indéfini} & \text{autrement.} \end{cases}$$



Configuration 4 : $(\omega_1 + d_1) < (\omega_2 - d_2)$

$$\mathrm{h}(x) = \begin{cases} C_1 & (\omega_1 - d_1) < x < (\omega_1 + d_1), \\ C_2 & (\omega_2 - d_2) < x < (\omega_2 + d_2), \\ \text{indéfini} & \text{autrement.} \end{cases}$$



(5) (b) Pour la configuration 1 présentée précédemment avec probabilités a priori égales, mais où on suppose que $\omega_1 < (\omega_2 - d_2) < (\omega_1 + d_1) < \omega_2$. Déterminez l'équation pour calculer la valeur de a qui devrait être utilisée comme seuil dans la règle de décision.

Solution: Le seuil a correspond au point où les probabilités *a posteri* sont égales pour les deux classes. Comme les probabilités *a priori* sont égales, ceci revient à trouver le point où les vraisemblances de classes ont la même valeur dans l'interval $[(\omega_2 - d_2), (\omega_1 + d_1)]$ et à isoler le x.

$$\begin{aligned} p(x|C_1) = \frac{-x + d_1 + \omega_1}{d_1^2} &= \frac{x + d_2 - \omega_2}{d_2^2} = p(x|C_2), \\ -xd_2^2 + d_1 d_2^2 + \omega_1 d_2^2 &= xd_1^2 + d_2 d_1^2 - \omega_2 d_1^2, \\ x(d_1^2 + d_2^2) &= (\omega_1 + d_1)d_2^2 + (\omega_2 - d_2)d_1^2, \\ x &= \frac{(\omega_2 - d_2)d_1^2 + (\omega_1 + d_1)d_2^2}{d_1^2 + d_2^2} = a. \end{aligned}$$

(6) (c) Toujours selon la configuration 1 et $\omega_1 < (\omega_2 - d_2) < (\omega_1 + d_1) < \omega_2$, déterminez l'équation pour calculer le taux d'erreur bayésien optimal avec cette modélisation, c'est-à-dire le taux d'erreur que l'on obtiendrait pour classer des données suivant ces densités.

Solution: On va estimer le taux d'erreur bayésien optimal en calculant l'aire des densités pour les régions où la décision consiste à choisir une autre classe. Dans le cas présent, pour la densité de la classe C_1, la région se situe dans l'interval $[a, \omega_1 + d_1]$, et similairement pour C_2, la région se situe dans l'interval $[\omega_2 - d_2, a]$. On suppose toujours des probabilités a priori égales, soit $P(C_1) = P(C_2) = \frac{1}{2}$.

$$\begin{aligned} l = p(a|C_1) = \frac{-a + d_1 + \omega_1}{d_1^2} &= \frac{a + d_2 - \omega_2}{d_2^2} = p(a|C_2), \\ E_{\text{Bayes}} = P(C_1)\,\frac{(a - (\omega_2 - d_2))l}{2} &+ P(C_2)\,\frac{(\omega_1 + d_1 - a)l}{2} \\ &= (\omega_1 + d_1 + d_2 - \omega_2)\frac{l}{4}. \end{aligned}$$

Question 2 (20 points sur 100)

Soit le jeu de données suivant $\mathcal{X} = \{(\mathbf{x}^t, r^t)\}_{t=1}^{8}$, en deux dimensions :

$$\mathbf{x}^1 = [0{,}65\ 0{,}6]^\top, \quad \mathbf{x}^2 = [0{,}25\ 0{,}55]^\top, \quad \mathbf{x}^3 = [0{,}2\ 0{,}7]^\top, \quad \mathbf{x}^4 = [0{,}4\ 0{,}5]^\top,$$
$$\mathbf{x}^5 = [0{,}7\ 0{,}35]^\top, \quad \mathbf{x}^6 = [0{,}4\ 0{,}3]^\top, \quad \mathbf{x}^7 = [0{,}15\ 0{,}35]^\top, \quad \mathbf{x}^8 = [0{,}1\ 0{,}45]^\top.$$

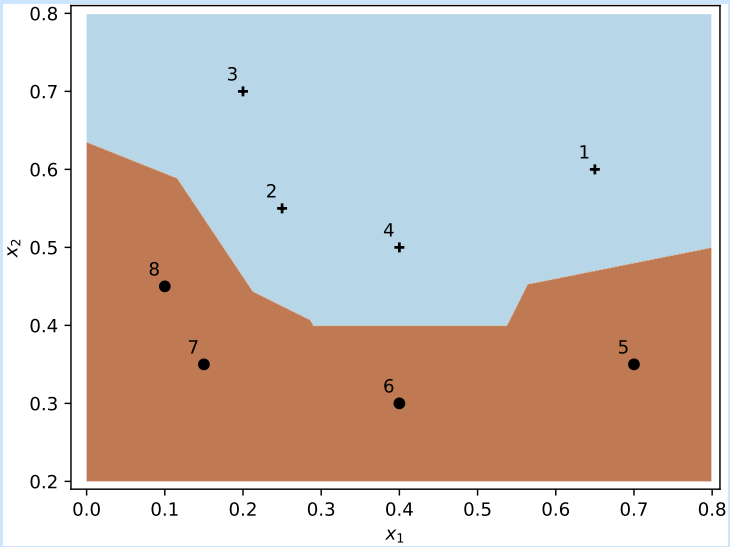
Les étiquettes de ces données sont $r^1 = r^2 = r^3 = r^4 = -1$ et $r^5 = r^6 = r^7 = r^8 = 1$.
La matrice des distances $D_\mathcal{X}$ donne la distance euclidienne entre toutes les paires possibles de données :

$$D_\mathcal{X} = \begin{bmatrix} \mathrm{d}(\mathbf{x}^1,\mathbf{x}^1) & \mathrm{d}(\mathbf{x}^1,\mathbf{x}^2) & \cdots & \mathrm{d}(\mathbf{x}^1,\mathbf{x}^N) \\ \mathrm{d}(\mathbf{x}^2,\mathbf{x}^1) & \mathrm{d}(\mathbf{x}^2,\mathbf{x}^2) & \cdots & \mathrm{d}(\mathbf{x}^2,\mathbf{x}^N) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{d}(\mathbf{x}^N,\mathbf{x}^1) & \mathrm{d}(\mathbf{x}^N,\mathbf{x}^2) & \cdots & \mathrm{d}(\mathbf{x}^N,\mathbf{x}^N) \end{bmatrix}$$

$$= \begin{bmatrix} 0{,}0000 & 0{,}4031 & 0{,}4610 & 0{,}2550 & 0{,}2550 & 0{,}3905 & 0{,}5590 & 0{,}5701 \\ 0{,}4031 & 0{,}0000 & 0{,}1581 & 0{,}1500 & 0{,}4924 & 0{,}2915 & 0{,}2236 & 0{,}1803 \\ 0{,}4610 & 0{,}1581 & 0{,}0000 & 0{,}2500 & 0{,}6103 & 0{,}4472 & 0{,}3536 & 0{,}2693 \\ 0{,}2550 & 0{,}1500 & 0{,}2500 & 0{,}0000 & 0{,}3606 & 0{,}2500 & 0{,}3202 & 0{,}3162 \\ 0{,}2550 & 0{,}4924 & 0{,}6103 & 0{,}3606 & 0{,}0000 & 0{,}3041 & 0{,}5500 & 0{,}6083 \\ 0{,}3905 & 0{,}2915 & 0{,}4472 & 0{,}2500 & 0{,}3041 & 0{,}0000 & 0{,}2550 & 0{,}3354 \\ 0{,}5590 & 0{,}2236 & 0{,}3536 & 0{,}3202 & 0{,}5500 & 0{,}2550 & 0{,}0000 & 0{,}1118 \\ 0{,}5701 & 0{,}1803 & 0{,}2693 & 0{,}3162 & 0{,}6083 & 0{,}3354 & 0{,}1118 & 0{,}0000 \end{bmatrix}.$$

(8) (a) Tracez les régions de décision selon les données d'entraînement présentée en préambule de la question pour un classifieur de type plus proche voisin (avec un seul voisin, $k = 1$) utilisant une distance euclidienne. Tracez le tout dans la **feuille de réponse fournie** à la fin du questionnaire. Donnez également le taux de classement selon une méthodologie *leave-one-out* avec cette configuration, sur ces données. Utilisez l'indice de donnée de valeur la plus faible pour briser les cas d'égalité de distances.

Solution:



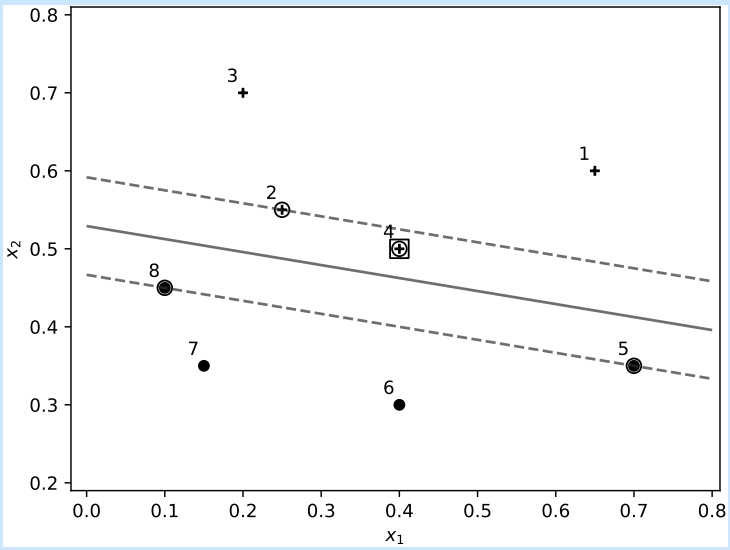
Données mal classées par *leave-one-out* : $\mathbf{x}^5$, et $\mathbf{x}^6$. Ceci correspond à un taux de classement de 75% $(6/8)$.

(12) (b) Nous obtenons le résultat suivant en effectuant l'entraînement d'un SVM linéaire à **marge douce** avec ces données, avec comme valeur de paramètre de régularisation $C = 100$:

$$\begin{array}{llll} \alpha^1 = 0{,}00, & \alpha^2 = 51{,}56, & \alpha^3 = 0{,}00, & \alpha^4 = 100{,}00, \\ \alpha^5 = 58{,}44, & \alpha^6 = 0{,}00, & \alpha^7 = 0{,}00, & \alpha^8 = 93{,}11, \\ w_0 = 8{,}47 & & & \end{array}$$

En utilisant la **feuille de réponse fournie** à la fin du questionnaire, tracez les éléments suivants :
— Frontière de décision du SVM (droite continue, —-) ;
— Frontières de la marge (droites en traits pointillés, - - - -) ;
— Vecteurs de support (encerclez les points, ◯) ;
— Données dans la marge (encadrez les points, □).

Solution:



Question 3 (20 points sur 100)

Supposons que l'on veut entraîner un autoencodeur à deux couches, la première couche comme encodeur et l'autre comme décodeur, toutes deux déterminée selon un apprentissage en ligne par rétropropagation des erreurs. Les poids de ces deux couches sont distincts pour les besoins de l'entraînement.

Dans ce réseau, une fonction de transfert linéaire est utilisée ($\mathrm{f}_{\mathrm{lin}}(x) = x$). La sortie d'un neurone de la couche d'encodage se modélise comme suit :

$$z_i^t = (\mathbf{w}_i^{\mathrm{enc}})^\top \mathbf{x}^t + w_{i,0}^{\mathrm{enc}}, \quad i = 1, \ldots, K.$$

La sortie de la couche de décodage se modélise comme suit, ce qui correspond à la donnée d'entrée reconstruite :

$$\hat{x}_j^t = (\mathbf{w}_j^{\text{dec}})^\top \mathbf{z}^t + w_{j,0}^{\text{dec}}, \quad j = 1, \ldots, D.$$

Le critère de performance utilisé est l'erreur quadratique de reconstruction E_{rec}^t, calculée sur une donnée $\mathbf{x}^t$:

$$e_j^t = (x_j^t - \hat{x}_j^t), \qquad E_{\text{rec}}^t = \frac{1}{2}\|\mathbf{x}^t - \hat{\mathbf{x}}^t\|^2 = \frac{1}{2}\sum_{j=1}^{D}(e_j^t)^2.$$

Répondez aux questions suivantes sur l'entraînement d'un tel réseau de neurones.

(10) (a) Détaillez les équations pour mettre à jour les poids des neurones de la couche du décodeur, par descente du gradient et en utilisant l'erreur de reconstruction E_{rec}^t d'une donnée.

Solution: La descente du gradient requiert de calculer les dérivées partielles de la fonction d'erreur, ici l'erreur de reconstruction, selon les paramètres optimisés, ici $\mathbf{w}_j^{\text{dec}}$ et $w_{j,0}^{\text{dec}}$, $j = 1, \ldots, D$. Développons d'abord les dérivées partielles par chaînage des dérivées :

$$\frac{\partial E_{\text{rec}}^t}{\partial w_{j,i}^{\text{dec}}} = \frac{\partial E_{rec}^t}{\partial e_j^t}\frac{\partial e_j^t}{\partial \hat{x}_j^t}\frac{\partial \hat{x}_j^t}{\partial w_{j,i}^{\text{dec}}},$$
$$\frac{\partial E_{\text{rec}}^t}{\partial w_{j,0}^{\text{dec}}} = \frac{\partial E_{rec}^t}{\partial e_j^t}\frac{\partial e_j^t}{\partial \hat{x}_j^t}\frac{\partial \hat{x}_j^t}{\partial w_{j,0}^{\text{dec}}}.$$

Pour l'erreur, les dérivées sont :

$$\frac{\partial E_{\text{rec}}^t}{\partial e_j^t} = \frac{\partial}{\partial e_j^t}\frac{1}{2}\sum_{l=1}^{D}(e_l^t)^2 = e_j^t,$$
$$\frac{\partial e_j^t}{\partial \hat{x}_j^t} = \frac{\partial}{\partial \hat{x}_j^t}(x_j^t - \hat{x}_j^t) = -1.$$

Pour les poids de la couche de décodage, les dérivées sont :

$$\frac{\partial \hat{x}_j^t}{\partial w_{j,i}^{\text{dec}}} = \frac{\partial}{\partial w_{j,i}^{\text{dec}}}\sum_{l=1}^{K} w_{j,l}^{\text{dec}} z_l^t + w_{j,0}^{\text{dec}} = z_i^t,$$
$$\frac{\partial \hat{x}_j^t}{\partial w_{j,0}^{\text{dec}}} = \frac{\partial}{\partial w_{j,0}^{\text{dec}}}\sum_{l=1}^{K} w_{j,l}^{\text{dec}} z_l^t + w_{j,0}^{\text{dec}} = 1.$$

En intégrant tout cela, on obtient :

$$\frac{\partial E_{\text{rec}}^t}{\partial w_{j,i}^{\text{dec}}} = -e_j^t\, z_i^t, \qquad \frac{\partial E_{\text{rec}}^t}{\partial w_{j,0}^{\text{dec}}} = -e_j^t.$$

La règle d'apprentissage selon la descente du gradient consiste donc à appliquer itérativement les mises à jour suivantes :

$$\begin{aligned}\Delta w_{j,i}^{\text{dec}} &= -\eta \frac{\partial E_{\text{rec}}^t}{\partial w_{j,i}^{\text{dec}}} = \eta e_j^t \, z_i^t, \quad i = 1, \ldots, K, \quad j = 1, \ldots, D, \\ \Delta w_{j,0}^{\text{dec}} &= -\eta \frac{\partial E_{\text{rec}}^t}{\partial w_{j,0}^{\text{dec}}} = \eta e_j^t, \quad j = 1, \ldots, D.\end{aligned}$$

(10) (b) Détaillez maintenant les équations pour mettre à jour les poids des neurones de la couche de l'encodeur, toujours par descente du gradient et en utilisant l'erreur de reconstruction E_{rec}^t d'une donnée.

Solution: Développons maintenant les dérivées partielles par chaînage des dérivées pour $\mathbf{w}_j^{\text{enc}}$ et $w_{j,0}^{\text{enc}}$, $j = 1, \ldots, K$:

$$\begin{aligned}\frac{\partial E_{\text{rec}}^t}{\partial w_{j,i}^{\text{enc}}} &= \frac{\partial E_{rec}^t}{\partial z_j^t} \frac{\partial z_j^t}{\partial w_{j,i}^{\text{enc}}}, \\ \frac{\partial E_{\text{rec}}^t}{\partial w_{j,0}^{\text{enc}}} &= \frac{\partial E_{rec}^t}{\partial z_j^t} \frac{\partial z_j^t}{\partial w_{j,0}^{\text{enc}}}.\end{aligned}$$

Pour l'erreur, les dérivées sont :

$$\begin{aligned}\frac{\partial E_{\text{rec}}^t}{\partial z_j^t} &= \frac{\partial}{\partial z_j^t} \frac{1}{2} \sum_{k=1}^{D} (e_k^t)^2 = \sum_{k=1}^{D} e_k^t \frac{\partial e_k^t}{\partial z_j^t} = \sum_{k=1}^{D} e_k^t \frac{\partial e_k^t}{\partial \hat{x}_k^t} \frac{\partial \hat{x}_k^t}{\partial z_j^t}, \\ \frac{\partial \hat{x}_k^t}{\partial z_j^t} &= \frac{\partial}{\partial z_j^t} \sum_{l=1}^{K} w_{k,l}^{\text{dec}} z_l^t + w_{k,0}^{\text{dec}} = w_{k,j}^{\text{dec}}.\end{aligned}$$

Pour les poids de la couche d'encodage :

$$\frac{\partial z_j^t}{\partial w_{j,i}^{\text{enc}}} = \frac{\partial}{\partial w_{j,i}^{\text{enc}}} \sum_{l=1}^{K} w_{j,l}^{\text{enc}} x_l^t + w_{j,0}^{\text{enc}} = x_i^t, \qquad \frac{\partial z_j^t}{\partial w_{j,0}^{\text{enc}}} = \frac{\partial}{\partial w_{j,0}^{\text{enc}}} \sum_{l=1}^{K} w_{j,l}^{\text{enc}} x_l^t + w_{j,0}^{\text{enc}} = 1.$$

En intégrant tout cela, on obtient :

$$\frac{\partial E_{\text{rec}}^t}{\partial w_{j,i}^{\text{enc}}} = -x_i^t \sum_{k=1}^{D} w_{k,j}^{\text{dec}} e_k^t, \qquad \frac{\partial E_{\text{rec}}^t}{\partial w_{j,0}^{\text{enc}}} = -\sum_{k=1}^{D} w_{k,j}^{\text{dec}} e_k^t.$$

La règle d'apprentissage selon la descente du gradient consiste donc à appliquer itérativement les mises à jour suivantes :

$$\begin{aligned}\Delta w_{j,i}^{\text{enc}} &= -\eta \frac{\partial E_{\text{rec}}^t}{\partial w_{j,i}^{\text{enc}}} = \eta x_i^t \sum_{k=1}^{D} w_{k,j}^{\text{dec}} e_k^t, \quad i = 1, \ldots, D, \quad j = 1, \ldots, K, \\ \Delta w_{j,0}^{\text{enc}} &= -\eta \frac{\partial E_{\text{rec}}^t}{\partial w_{j,0}^{\text{enc}}} = \eta \sum_{k=1}^{D} w_{k,j}^{\text{dec}} e_k^t, \quad j = 1, \ldots, K.\end{aligned}$$

Question 4 (40 points sur 100)

Répondez aussi brièvement et clairement que possible aux questions suivantes.

(4) (a) Voici la formalisation mathématique du théorème de Bayes pour le classement, tel que présenté en classe.

$$P(C_i|\mathbf{x}) = \frac{P(C_i)\, p(\mathbf{x}|C_i)}{p(\mathbf{x})}$$

Définissez textuellement ce que signifie chacun des termes mathématiques suivants : $P(C_i)$, $p(\mathbf{x}|C_i)$, $p(\mathbf{x})$ et $P(C_i|\mathbf{x})$.

Solution:
- Probabilité a priori ($P(C_i)$) : probabilité d'observer une instance de la classe C_i ;
- Vraisemblance de classe ($p(\mathbf{x}|C_i)$) : vraisemblance qu'une observation de la classe C_i soit $\mathbf{x}$;
- Évidence ($p(\mathbf{x})$) : vraisemblance d'observer la donnée $\mathbf{x}$;
- Probabilité a posteriori ($P(C_i|\mathbf{x})$) : probabilité qu'une observation $\mathbf{x}$ appartienne à la classe C_i.

(4) (b) Expliquez quel sera le niveau de performance et la complexité attendue d'un modèle d'apprentissage supervisé lorsque celui-ci est dit à biais faible et variance élevée.

Solution: Un modèle d'apprentissage à biais faible et variance élevée est un modèle qui est en mesure de modéliser avec une précision élevée les données d'entraînement, effectuant ainsi un **surapprentissage** important. En effet, les modèles générés peuvent varier grandement d'un entraînement à l'autre, soit suite à des échantillonnages différents de données d'entraînement provenant d'un même phénomène, soit des variations aléatoires de l'algorithme d'apprentissage. Ce type de modèle tend à offrir des performances en généralisation très variables sur des données non vues durant l'entraînement.

(4) (c) Dans un contexte de classement paramétrique avec modèles de lois normales multivariées, indiquez quelle(s) simplification(s) aux matrices de covariance permet d'obtenir un modèle de classement linéaire.

Solution: Le partage de la matrice de covariance entre les classes du problème permet d'obtenir des modèles de classement linéaires.

(4) (d) Dans la méthode de classement par les k-plus proches voisins, expliquez l'effet de l'augmentation de la valeur de k (nombre de voisins) sur les frontières de décision et les performances, supposant que l'on a un N (taille du jeu d'entraînement) faible. Précisez également ce qui arrive dans le cas limite où la valeur de k se rapproche de celle de N.

Solution: En augmentant le nombre de voisins pour un classement par les k-plus proches voisins, on va adoucir les frontières de décision et séparer l'espace selon les emplacements où les données de chaque classe sont les plus présentes (régions denses en données). Dans le cas limite où k se rapproche de N, l'algorithme aura tendance à assigner les données à traiter aux classes les plus fréquentes (ayant les probabilités *a priori* les plus élevées).

(4) (e) Expliquez pourquoi le critère des moindres carrés est préférable au critère du perceptron pour des modèles de discriminants linéaires devant être entraînés sur des données non linéairement séparables.

Solution: L'entraînement de discriminants linéaires avec le critère du perceptron sur des données non linéairement séparables diverge généralement, avec comme résultats des modèles peu performants. L'entraînement d'un discriminant linéaire par une régression selon le critère des moindres carrés est beaucoup plus stable, générant des solutions souvent acceptables, sans nécessairement être optimal avec ces modèles.

(4) (f) Expliquez comment un SVM à noyau est en mesure de traiter avec succès des données de classement non linéairement séparables en comparaison avec un SVM linéaire.

Solution: Un SVM à noyau est un discriminant linéaire appliqué dans un espace mathématique de plus haute dimensionnalité que l'espace d'entrée, qui est engendré par la fonction noyau. Les données sont souvent pratiquement linéairement séparables dans cet espace mathématique, et même si elles ne le sont pas, les marges douces permettent de tolérer des données mal classées ou dans la marge. Un SVM linéaire traite les données directement dans l'espace d'entrée et risque donc d'avoir beaucoup de difficultée à produire un modèle traitant bien les données non linéaires.

(4) (g) Expliquez pourquoi l'utilisation d'un jeu de données de validation, indépendant du jeu d'entraînement et de test, est nécessaire pour arrêter l'entraînement et sélectionner le modèle, avec des réseaux de neurones tels que les perceptrons multicouches.

Solution: Les réseaux de neurones sont des modèles pouvant avoir une capacité et flexibilité importante pour modéliser les données. En ce sens, un entraînement ces modèles sur un grand nombre d'itérations peut faire un surapprentissage important des données d'entraînement. Le jeu de validation est ainsi utilisé comme jeu de données de contrôle pour interrompre l'apprentissage lorsque le surapprentissage devient important, et sélectionner le modèle vu durant l'apprentissage qui semblait généraliser le mieux.

(4) (h) Il a été dit dans le cours que les réseaux de neurones profonds modernes sont des modèles qui possèdent des mécanismes leur permettant de bien gérer les données à haute dimensionnalité. Expliquez comment ceci est effectué concrètement dans les réseaux de neurones à convolution, par exemple pour traiter des images.

Solution: Les réseaux de neurones à convolution comportent des filtres de taille relativement réduite (ex. 3×3 ou 5×5 valeurs), qui sont généralement convolués sur l'ensemble des données d'un canal organisé temporellement ou spatialement. Ces filtres comportent donc assez peu de valeurs numériques à apprendre, tout en permettant des opérations assez complexes exploitant la structure des données traitées.

(4) (i) Expliquez la façon la plus courante d'effectuer un transfert de représentation avec des réseaux de neurones profonds, dans un contexte de classement.

Solution: Une approche courante consiste à retirer une ou quelques couches de neurones en sortie d'un réseau entraîné sur un jeu de données permettant de bien modéliser les données du domaine dans lequel on travail, tout en conservant le reste du réseau qui correspond à la représentation apprise. Ensuite, on crée de nouvelles couches de sortie selon la nouvelle tâche que l'on veut effectuer, les entraînant avec des données de la nouvelle tâche. On peut aussi raffiner (*fine-tuning*) les poids de la représentation transférée avec les données de la nouvelle tâche.

(4) (j) Lorsqu'on implémente un modèle de classement supervisé avec scikit-learn, indiquez le nom des trois méthodes principales de l'interface que l'on doit définir, en décrivant la fonction de chacune d'elle en une phrase.

Solution:
- `fit` : pour apprendre le modèle sur un jeu de données ;
- `predict` : pour effectuer les prédictions de valeurs (inférence) d'un modèle entraîné sur un certain jeu de données ;
- `score` : pour obtenir la performance du modèle sur un jeu de données.