Nom de l'étudiant: ____________________

---

FACULTE DES ARTS ET DES SCIENCES
DEPARTEMENT D'INFORMATIQUE ET DE RECHERCHE OPERATIONNELLE

TITRE DU COURS: **Algorithmes d'apprentissage**
SIGLE DU COURS : **IFT6265 A97**

NOM DU PROFESSEUR: Yoshua Bengio

DATE DE L'EXAMEN INTRA A97: 21 octobre 1997 | HEURE: 15h30 - 17h30
SALLE: Z-325

DIRECTIVES PEDAGOGIQUES:
- Aucune documentation permise (sauf, si vous voulez, des références sur les règles de différentiation).
- Répondre directement sur le questionnaire.
- Soyez brefs et précis dans vos réponses.

---

1. Pour chacun des énoncés qui suivent, est-il VRAI ou FAUX? Donnez aussi une justification à votre réponse (très important!). Plusieurs de ces questions concernent l'ensemble $\mathcal{B}_n$ des fonctions booléennes sur des vecteurs booléens, $\{-1, 1\}^n \to \{-1, 1\}$.

   (a) Les exemples sont situés sur des coins de l'hypercube $[-1, 1]^n$.

   (b) Les fonctions pour lesquelles un seul vecteur d'entrée donne 1 (ou, -1) sont linéairement séparables.

   (c) La notion d'interpolation s'étend naturellement à $\mathcal{B}_n$.

   (d) Pour les fonctions de $\mathcal{R}^n \to \mathcal{R}$, plus on a de variables, plus la généralisation est facile.

   (e) La généralisation dans $\mathcal{B}_n$ est facile.

2. Nous considérons le cas classique de l'apprentissage comme choix d'une fonction parmi un ensemble, pour minimiser l'erreur empirique. Donner graphiquement la relation qualitative les variables suivantes:

   (a) nombre d'exemples $\to$ erreur d'apprentissage, erreur de généralisation

(b) capacité $\rightarrow$ erreur d'apprentissage, erreur de généralisation

(c) nombre d'exemples $\rightarrow$ capacité optimale

(d) capacité $\rightarrow$ erreur asymptotique (quand nombre d'exemples $\rightarrow \infty$).

3. Donner trois manières de réduire la capacité effective d'un algorithme d'apprentissage supervisé basé sur la descente de gradient appliquée aux paramètres d'un réseaux de neurones multi-couche

avec minimisation de l'erreur quadratique.

4. Il s'agit de montrer ici que l'algorithme *K-Means* jouit d'une convergence superlinéaire vers un minimum (qui peut être local) de l'erreur de reconstruction,

$$C = \sum_{t=1}^{l} 0.5(\mu_{s(z_t)} - z_t)^2$$

où $z_1 \dots z_l$ est l'ensemble d'apprentissage (des vecteurs dans $\mathcal{R}^n$),

$$s(z) = \text{argmin}_{i=1}^{K}(\mu_i - z)^2$$

et $\theta = (\mu_1, \dots, \mu_K)$ est l'ensemble des vecteurs de positions des $K$ prototypes, qui sont les paramètres libres de l'apprentissage. Soit $m = nK$ (donc $\theta \in \mathcal{R}^m$). Pour obtenir une convergence superlinéaire, il suffit de montrer que l'algorithme d'apprentissage effectue des modifications des paramètres selon la formule de Newton,

$$\theta \leftarrow \theta - \left(\frac{\partial^2 C}{\partial \theta^2}\right)^{-1} \frac{\partial C}{\partial \theta}$$

où $\frac{\partial C}{\partial \theta}$ est le vecteur de gradients ($\in \mathcal{R}^m$), et $\frac{\partial^2 C}{\partial \theta^2}$ est la matrice Hessienne ($\in \mathcal{R}^{m \times m}$), dont l'élément $(i, j)$ est la dérivée seconde $\frac{\partial^2 C}{\partial \theta_i \partial \theta_j} = \frac{\partial}{\partial \theta_i}(\frac{\partial C}{\partial \theta_j})$.

5. L'hypothèse *i.i.d.* sur l'ensemble de données d'apprentissage est généralement violée, en particulier à cause des dépendences séquentielles. Il existe plusieurs algorithmes d'apprentissage qui permettent en principe d'apprendre ces dépendences séquentielles entre les exemples successifs. Décrivez le principe de base de l'un deux, en expliquant comment sa formulation mathématique lui permet de représenter (et donc potentiellement, d'apprendre), ces dépendences.

6. Expliquer de manière intuitive pourquoi est probablement vraie la conjecture selon laquelle le *bagging* réduit l'erreur de généralisation. Appuyez-vous sur la décomposition bruit/biais/variance de l'erreur quadratique.

7. Considérons la situation suivante. On utilise une classe de fonctions paramétrisée pour prendre une décision $w_t$ à chaque temps discret $t$ concernant l'allocation de notre capital entre plusieurs actifs: Cette décision est donc représentée par un vecteur de poids $w_t = (w_{t,1}, w_{t,2}, \ldots, w_{t,n})$ positifs et sommant à un, qui représentent la fraction de notre avoir placée dans chacun des actifs $1, 2, \ldots, n$. La fonction de décision est
$$w_t = f(x_t, w_{t-1}, \theta).$$
Notez que la décision au temps $t$ dépend de la décision au temps $t-1$. Ceci est important car les changements d'allocation entre $t$ et $t-1$ impliquent des pertes (frais de transaction). Le critère à maximiser est le rendement total (entre le début $t=1$ et la fin de la séquence $t=T$):
$$C = \sum_{t=1}^{T} C_t$$
avec
$$C_t = \sum_{i=1}^{n} \log(w_{t,i} r_{t,i}) + \log(1 - \lambda |w_{t,i} - w_{t-1}|)$$
où $r_{t,i}$ est donné pendant l'apprentissage et est le ratio de rendement de chaque actif, $\lambda$ est une constante représentant les frais de transactions (proportionnels au montant de la transaction), et on prend $w_0$ comme une constante donnée (le portefeuille initial).

Faites un diagramme de flots correspondant à ces équations et montrez comment le gradient $\frac{\partial C}{\partial \theta}$ pourrait être calculé par rétropropagation à travers le temps (dans le sens inverse des flèches du diagramme de flots).

Expliquez comment ce modèle pourrait en principe permettre d'apprendre l'impact des décisions $w_t$ sur les performances ultérieures $C_\tau$ en $\tau > t$.