le cnam

Sujet UE RCP209
Apprentissage, réseaux de neurones et modèles graphiques

Année universitaire 2017–2018

Examen 1ère session : 9 février 2018

Responsable : Michel CRUCIANU

Durée : 3h00

Seuls documents autorisés : 2 pages A4 recto-verso, manuscrites.

Les téléphones mobiles et autres équipements communicants (PC, tablette, etc.) doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve. Calculatrice autorisée.

Sujet de 5 pages, celle-ci comprise.

Vérifiez que vous disposez bien de la totalité des pages du sujet en début d'épreuve et signalez tout problème de reprographie le cas échéant.

---

1. Entre les procédures de validation croisée *leave one out* et *k-fold*, laquelle est la plus coûteuse ? Laquelle a le plus fort biais et pourquoi ? (**2 points**)

   Correction : *K-fold* est moins coûteuse car seulement $k \ll N$ modèles sont appris et non N. *K-fold* a aussile plus fort biais car l'apprentissage se fait à chaque fois sur $\frac{N(k-1)}{k}$ données $\left(\frac{N(k-1)}{k} < N-1\right)$.

2. Arbres de décision (**2 points**). Définition et interprétation de l'entropie d'un nœud par rapport à la variable cible. Comment l'algorithme de construction de l'arbre utilise cette grandeur ?

   Correction : L'entropie d'un nœud S est définie comme $H(S) = -\sum_{1=1}^{m} p_i \log(p_i)$ ou p_i est la fréquence d'apparition de la valeur n_i de la variable cible dans le nœud. $H(S)$ mesure l'écart de la distribution de la variable cible par rapport à la distribution uniforme ($H(S) = 0$ si S est homogène : tous les éléments de S ont la même valeur). L'algorithme utilise cette grandeur pour décider sur quelle variable (attribut) créer un nouveau test.

3. Boosting (**2 points**). Dans l'algorithme AdaBoost, quelle condition doivent remplir les classifieurs faibles pour garantir le bon fonctionnement de la procédure ? Expliquez pourquoi.

   Correction : Le classifieur faible doit avoir un comportement de base un peu meilleur que l'aléatoire : taux d'erreurs inférieur à 0.5 pour une classification binaire (c'est-à-dire qu'il ne se trompe pas plus d'une fois sur deux en moyenne, si la répartition des classes est équilibrée). Le taux d'erreurs intervient dans le réajustement des poids w_m : pour un taux $e_m < 0.5$ le coefficient α_m qui guide la pondération du classifieur m devient négatif, ce qui inverse le fonctionnement de l'algorithme (encourage les classifications mauvaises dans le mélange des classifieurs).

4. SVM Lineaires (**2 points**). Dans le cas des données non séparables linéairement, expliquez le rôle de la constante C dans le problème d'optimisation dual.

   Correction : C est une variable de pénalisation des points mal classés faisant un compromis entre la largeur de la marge et les points mal classés. Dans le problème dual elle définit les conditions d'admissibilité duale $C \geq \alpha_i \geq 0, i = 1, \ldots, n$. Elle joue donc le rôle d'une constante de régularisation (la régularisation est d'autant plus forte que C est proche de 0).

5. Algorithmes à noyaux. (**2 points**) Expliquez brièvement l'astuce à noyaux. Quels algorithmes permet-elle de rendre non-linéaires ?

   Correction : L'astuce à noyaux permet de rendre non-linéaire tout algorithme linéaire qui utilise des produits scalaires. L'idée est de transposer les données dans

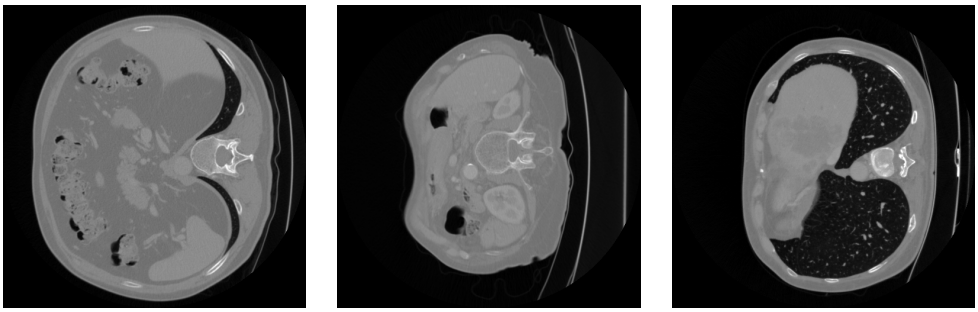
un autre espace de dimension plus grande et ensuite appliquer l'algorithme linéaire sur les données projetées. Par la théorème de Mercer, si K est un noyau défini positif $K : R^d \times R^d \to R$ alors $K(x,y) = \langle \phi(x), \phi(y) \rangle$ ou $\phi : R^d \to \mathcal{H}, x \to \phi(x)$ est une transformation de R^d vers un espace de Hilbert $\mathcal{H}$ qu'on n'a pas besoin d'expliciter : tout algorithme qui utilise seulement des produits scalaires entres les échantillons de données peut tout de suite s'appliquer dans l'espace $\mathcal{H}$ via le produit scalaire $\langle \phi(x), \phi(y) \rangle = K(x,y)$.

6. Deep Learning et réseaux convolutifs (**4 points**). On s'intéresse aux différences entre les réseaux convolutifs modernes (post 2012) et les réseaux des années 1980 (LeNet).
   (a) En quoi le réseau AlexNet (2012) diffère-t-il du réseau LeNet ?
   (b) Pourquoi le réseau VGG (2014) ne contient que des convolutions 3×3 ?
   (c) Quelles sont les spécificité du réseau GoogLeNet (2015) ?
   (d) Quelle est la nouveauté architecturale principale du réseau ResNet (2015) ?

   Correction :
   (a) Pas de différence fondamentale dans l'architecture macroscopique / CNN des années 80 : alternance de couches convolution / pooling puis couches complètement connectées. Simplement des réseaux avec plus de couches et plus de neurones / filtres par couches (60000 vs 60 M paramètres). Des différences micro : ReLU, dropout.
   (b) En empilant plusieurs convolutions 3x3, on peut modéliser des convolutions plus larges que 3x3, tout en limitant le nombre de paramètres
   (c) Module Inception, on a le loss qui apparaît à différents niveaux de profondeur du réseau, réseau complètement convolutif.
   (d) Connexion résiduelle qui permet de plus facilement modéliser l'identité $\Rightarrow$ on peut apprendre des réseaux plus profonds.

7. Transfert (**3 points**)
   On souhaite mettre en place un système de diagnostic médical basé sur des techniques d'apprentissage profond, afin de reconnaître les trois classes suivantes dans les images : pancréas, estomac, tumeur. On dispose d'une base avec 500 images scanner étiquetées par classes, dont certaines sont montrées dans la figure 1.

   — Est-il possible d'entraîner des réseaux convolutifs modernes (*e.g.* VGG, ResNet) « *from scratch* » sur ces données ? Si oui, décrire la méthodologie. Sinon, justifier.
   — Est-il possible d'utiliser des réseaux convolutifs modernes (*e.g.* VGG, ResNet) pré-entraînés sur ImageNet sur ces données ? Si oui, décrire la méthodologie. Sinon, justifier.



a) Pancréas b) Estomac c) Tumeur

FIGURE 1 – Base d'images médicales annotée en trois classes : pancréas, estomac, tumeur.

Correction :

— 1pt Non, il n'y a pas de assez de données annotées, on est dans un régime de sur-apprentissage extrême avec des réseaux ayant $\sim 10^7 - 10^8$ paramètres.

— 2pts Oui, on peut faire du transfert, ce qui consiste à prendre un réseau pré-entraîné sur ImageNet, supprimer la couche des classes d'ImageNet et ajouter une couche correspondant aux 4 classes de la base cible. Dans ce contexte de transfert, on apprendra uniquement les paramètres de la dernière couche, les autres seront gelés. On peut également essayer de fine-tuner, mais le risque de sur-apprentissage est important étant donné le faible nombre d'exemples.

8. Agrégation (pooling) (**2 points**)
On considère la carte 2D C suivante (taille 5×5), résultat de l'application d'une convolution sur une image I d'entrée.

$$C = \begin{bmatrix} 11 & -5 & 1 & -2 & 0 \\ 1 & 3 & 0 & 0 & 5 \\ 8 & 4 & 15 & -10 & 4 \\ 8 & 6 & 5 & 3 & 7 \\ 3 & 0 & -2 & 9 & 3 \end{bmatrix}$$

— Si on fait subir une translation à l'image d'entrée, quel va être l'impact sur la carte C ? Comment s'appelle cette propriété ?

— On considère maintenant la région 3×3 encadrée de la carte C centrée au neurone de valeur 15. On effectue un pooling de type max dans cette région. Quel va être la sortie du pooling pour une version translatée de un pixel de l'image d'entrée I ?

Correction :

— La translation va être répercutée directement sur la carte $C \Rightarrow$ équivariance (on peut commuter translation et convolution).

— Le max va être inchangé $\forall$ translation $(t_x, t_y) \in \pm 1$ px $\Rightarrow$ invariance.

9. Prédiction structurée (**1 point**). Quelle est la différence entre les méthodes d'apprentissage structuré et les méthodes de classification ? (3-4 lignes max).

   Correction : La prédiction structurée permet d'entraîner des modèles à prédire des sorties discrètes quelconques, beaucoup plus générales que les méthodes de classification (limitées à prédire une classe en sortie). En particulier, l'intérêt est de pouvoir modéliser la corrélation entre les variables de sortie.