SORBONNE UNIVERSITÉS

Examen d'apprentissage artificiel

Année universitaire 2018-2019

| Nom | Prénom |
|---|---|
| | |

## Exercice SVM

On donne les points suivants : $x = \begin{bmatrix} 4 & 3 \\ 0 & 2 \\ 0 & 0 \\ 2 & 0 \end{bmatrix}$ d'étiquettes $y = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}$

1. Représentez ces points, sont-ils linéairement séparables ?

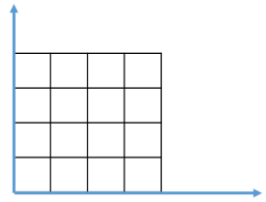


*Figure 1*

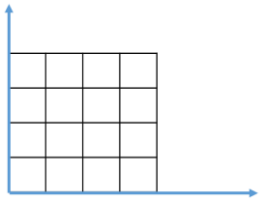


*Figure 2*

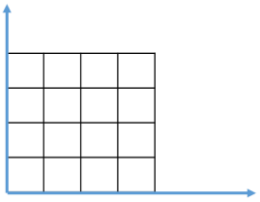


*Figure 3*

2. Exprimer les contraintes de séparabilité pour chaque exemple

(1) $1(4w_0+3w_1+b)>1$
(2) $1(2w_1+b)>1$
(3) $-1(b)>1$
(4) $-1(2w_0+b)>1$

3. Calculer les paramètres **(*w*,*b*)** de l'hyperplan optimal et représenter le sur la Figure 1. Que vaut la marge ? Quels sont les vecteurs supports ?

(2) et (3) ➔$w_1>1$
(3) et (4) ➔$w_0<0$
On veut minimiser $w_0^2+w_1^2$➔$w_0=0$ et $w_1=1$
On veut que $\min_{i=1,N} y_i(\boldsymbol{w^T x_i} + b) = 1$➔b=-1
La marge vaut $\frac{2}{\|\mathbf{w}\|} = \frac{2}{1} = 2$
Pour les vecteurs support, $y_i(\boldsymbol{w^T x_i} + b) = 1$ ➔les points 2 et 3 sont les vecteurs support

4. Dans le cas où on souhaiterait résoudre le problème des SVM par un solver de problème quadratique, déterminer les matrices d, A, B, e à passer au solver et la matrice z qu'il renverrait. Le solver considéré résout $\min_z \frac{1}{2}\boldsymbol{z^T A z} - \boldsymbol{d^T z} \quad \mathbf{sc}\ \boldsymbol{Bz} \leq \mathbf{e}$

$\boldsymbol{z} = \begin{bmatrix} b \\ w_1 \\ w_0 \end{bmatrix}$, $\mathbf{d} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$, $\mathbf{B} = -\begin{bmatrix} 1 & 4 & 3 \\ 1 & 0 & 2 \\ -1 & 0 & 0 \\ -1 & 2 & 0 \end{bmatrix}$ et $\mathbf{e} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}$

5. Classer le point $\begin{bmatrix} 3 \\ 0 \end{bmatrix}$ en déterminant la valeur de f(***x***).

$f(x) = w_0 x + w_1 y + b = y - 1 = -1$ ➔ classe -1

6. On rajoute maintenant un point d'étiquette *-1* en x=[0 1.9]$^T$ et on utilise le SVM à marge souple. Rappeler le principe du SVM à marge souple. Représenter Figure 2 l'hyperplan optimal obtenu avec un C très grand et Figure 3, celui obtenu avec un C très petit. Justifier votre réponse.

## Question réseaux de neurones

1. Quel sont les 2 principaux intérêts du pooling spatial ?

Invariance en translation
Sous-echantillonnage

2. Qu'est ce que le momentum, pourquoi améliore-t-il les résultats ?

Il sert à lisser les gradients au cours du temps ➔ Moins d'effet 'yoyo' ➔ apprentissage plus rapide, moins sensible au batch

3. Donner le principal intérêt d'utiliser une convolution 1D (justifier votre réponse)

Diminue la profondeur des images, sans changer la résolution

4. Une couche d'entrée possède 10 neurones et la couche cachée 5 neurones. Quel est le nombre maximum de connexion entre ces deux couches ?

5*10=50

5. En considérant l'image d'entrée suivante et une convolution par $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ avec un stride de 2 et un padding de 0, donner l'image résultant de la convolution.

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 |

| | | |
|---|---|---|
| 4 | 3 | 3 |
| 3 | 3 | 3 |
| 4 | 3 | 4 |

6. Quelle dernière couche doit-on ajouter à un réseau de neurones si on souhaite prédire des probabilités d'appartenance aux classes ? Comment procède-t-elle ?

Une couche de soft-max

$$y_k^t = \frac{\exp(y_k^t)}{\sum_{k'} \exp(y_{k'}^t)}$$

7. Quels sont les avantages et inconvénients d'une fonction d'activation RELU par rapport à la sigmoïde ?

**Avantages**
Très rapide à calculer (pas d'exponentielle)
Permet au réseau de converger beaucoup plus vite (six fois)
Forts gradients (0 ou 1)
**Inconvénient**
Certains neurones « meurent » en ne produisant que des zéros (les poids sont tels que la sortie est nulle pour toutes les entrées). Comme le gradient devient nul, ils ne se réactivent jamais

8. Dans un CNN, ajouter une couche de max-pooling décroit forcément le nombre de paramètres du modèle ? Justifier votre réponse

Non, si le stride est de 1

9. Rappeler le principe du dropout en apprentissage et en test. Est-il vrai que cette méthode diminue le nombre de paramètres à estimer lors de l'apprentissage du réseau ?

pour chaque epoch, enlever aléatoirement des neurones lors de l'apprentissage
En test, tous les neurones sont utilisés
Les sous-réseaux ont moins de paramètres à estimer mais le réseau complet en a autant

10. En considérant l'image d'entrée ci-dessous et une couche de max-pooling de taille 3x3 avec un stride de 2 et un padding de 0, donner l'image en sortie du max-pooling.

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 1 | 4 | 0 | 1 |
| 0 | 0 | 1 | 6 | 1 | 5 | 5 |
| 1 | 4 | 4 | 5 | 1 | 4 | 1 |
| 4 | 1 | 5 | 1 | 6 | 5 | 0 |
| 1 | 0 | 6 | 5 | 1 | 1 | 8 |
| 2 | 3 | 1 | 8 | 5 | 8 | 1 |
| 0 | 9 | 1 | 2 | 3 | 1 | 4 |

| | | |
|---|---|---|
| 4 | 6 | 5 |
| 6 | 6 | 8 |
| 9 | 8 | 8 |

11. Quelles solutions proposez-vous éviter l'overfitting lors de l'apprentissage de réseaux de neurones (4 propositions)

Data Augmentation
Weight Sharing
Early Stopping
Dropout

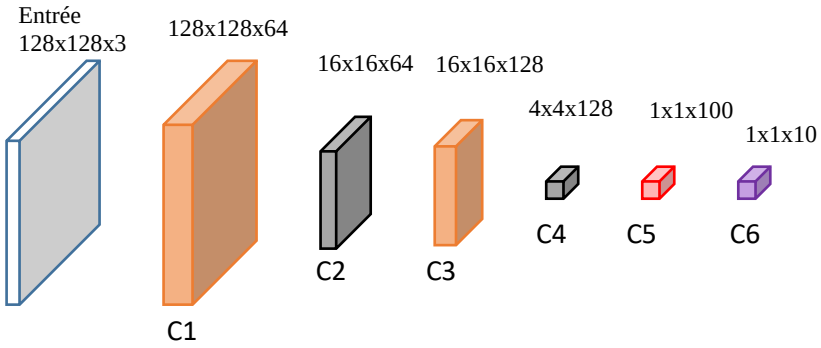
**12.** Supposons que dans le réseau VGG16 on remplace toutes les fonctions d'activation RELU par des activations linéaires. Est-ce une bonne idée ? Pourquoi ?

Non, le réseau perd ses capacités de gérer des données non linéairement séparables

13. Pour un problème de classification binaire, la sortie peut être composée de 1 ou deux neurones. Déterminer dans chacun des cas comment est codée la sortie, et comment réaliser l'optimisation (fonction perte, fonction d'activation de la dernière couche, …)

Avec un seul neurone :
La sortie est codée 0/1. On met une sigmoide et la fonction perte : erreur quadratique
Avec 2 neurones :
La sortie est codee (10) ou (01). On utilise une fonction d'activation softmax et en fonction perte l'entropie croisee

14. On considère l'architecture suivante.




Sachant que :

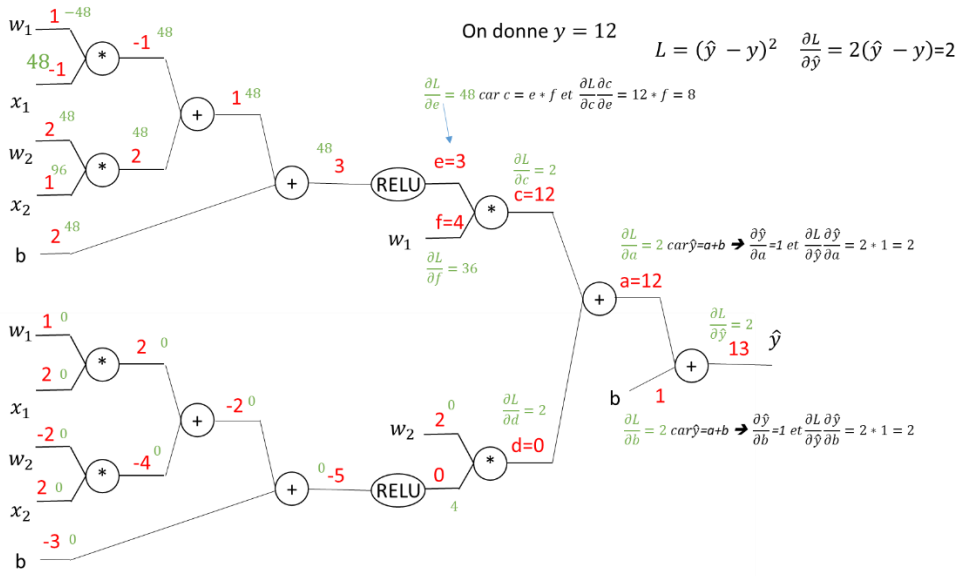
- Chaque couleur représente un type de couche parmi « convolution + RELU », de « max-pooling », « fully-connected+RELU » et « fully-connected+softmax » ;
- Le padding est réalisé en complétant les bords de zéros autant que nécessaire pour les couches de convolution et de pooling ;
- Les couches convolutionnelles sont réalisées avec des filtres 3x3 ; les couches de pooling sont réalisées avec des filtres 11x11 ;
  a. Identifier les différentes couches. Donner le stride et le padding des couches convolutionnelles et de pooling.
  b. Que réalise ce réseau ? Classification en combien de classes, régression, … ?
  c. Combien y a-t-il de paramètres à estimer dans le réseau (poser le calul)?

1. C1 : convolution3x3, stride 1, padding 1
   C2 : max-pooling 11x11, stride=4, padding=5
   C3 : convolution 3x3, stride 1, padding 1
   C4 : max-pooling 11x11, stride 0, padding 5
   C5 : fully connected+RELU
   C6 : fully connected+
2. Classification en 10 classes
3. 64x(3*3*3+1)+128*(3x3x64+1)+100x4x4*128+128+100*10+10

## Apprentissage d'un réseau

1. Tracer l'évolution de la fonction d'activation RELU et de sa dérivée

2. Estimer la sortie $\hat{y}$ du réseau de neurone suivant (écrire les valeurs en rouge, étape par étape).




3. Sachant que l'on utilise une MSE comme fonction de perte, donner l'expression de la fonction de perte *L* en fonction de la sortie estimée $\hat{y}$ et de la sortie désirée *y*. En déduire l'expression de $\partial L/(\partial \hat{y})$

4. Réaliser la rétro-propagation du gradient pour estimer la dérivée de *L* par rapport aux paramètres du réseau (écrire les valeurs en vert ou bleu sur le schéma). On donne *y*=12