Numéro d'anonymat :

Contrôle de Connaissance
L2 Informatique - Introduction à l'apprentissage automatique- Université Paris-Sud

Mardi 11 Décembre 2018
Durée : 2h00
Documents autorisés : supports et notes de cours

Partie I : Apprentissage supervisé

1 Perceptron [10+1 pts]

On cherche à écrire un algorithme permettant de distinguer entre les courriels importants des autres. On considérera donc une tâche de classification binaire. Pour entraîner notre algorithme, nous disposons de N données d'apprentissage : $(\boldsymbol{x}_i, \tilde{y}_i)_{i=1,..,N}$ où $\boldsymbol{x}_i$ représente la i-ème données et $\tilde{y}_i$ son étiquette ($+1$ si le courriel est important, -1 sinon).

1.1 Questions de cours [2 pts]

1. En quoi consiste le modèle du perceptron (vous définierez les variables d'entrées, de sorties et les paramètres).
2. Donner deux jeux de données, un pour lequel le Percetron peut les classifier et un autre pour lequel il ne peut pas.
3. Illustrer par un schéma comment détecter le phénomère de surajustement (overfitting) sur les courbes du taux d'erreur de l'ensemble d'entraînement et de l'ensemble de test.
4. Comment s'exprime les sorties du perceptron multicouches en fonctions des entrées et des poids ?

1.2 Le perceptron binaire "classique" [4 pts]

On s'attaque ici au problème sous l'angle du perceptron binaire non probabiliste. Nous supposerons que nous disposons des données suivantes :

$$\tilde{y} = +1 : \left\{ \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}, \begin{pmatrix} -1 \\ -0.5 \end{pmatrix} \right\} \text{ et } \tilde{y} = -1 : \left\{ \begin{pmatrix} 0 \\ -3 \end{pmatrix}, \begin{pmatrix} 1 \\ -2 \end{pmatrix} \right\}$$

et que le vecteur de poids initial est donné par $\mathbf{w'}^t = (0, -1, 1)$

1. Indiquer quelles sont les données bien classées et les données mal classées <u>et</u> les représenter sur un graphe.
2. Tracer la fontière de décision correspondant au $\mathbf{w'}^t$ initial.
3. Effectuer une époque sur ces données (en les prenant dans l'ordre que vous voulez) et indiquer la nouvelle valeur des paramètres du percetron.
4. Sur le graphe précédent, représenter la nouvelle frontière de décision.
5. L'algorithme peut-il converger sur ces données ?

1.3 Perceptron multicouche avec ReLU [4+1 pts]

Sur le précédent problème, on a rencontré les données suivantes

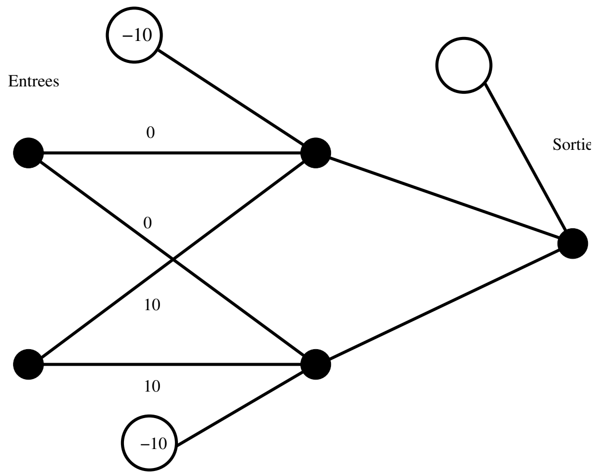
$$\tilde{y} = -1 : \left\{ \begin{pmatrix} -1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \end{pmatrix} \right\}$$

On se propose d'étudier le fonctionnement d'un perceptron multicouche, avec des fonctions d'activation de type ReLU pour la couche intermédiaire. La fonction d'activation ReLU se comporte de la façon suivante :

$$\mathrm{ReLU}(x) = \max(0, x) \tag{1}$$

1. Expliquer pourquoi il est impossible de classifier ces données à l'aide d'un perceptron classique.
2. Sur un graphique, représenter l'allure de cette fonction d'activation.

On souhaite tester la machine suivante :



où $w_{11}^{(1)} = 0$, $w_{12}^{(1)} = 10$, $w_{10}^{(1)} = -10$ d'une part et $w_{21}^{(1)} = 0$, $w_{22}^{(1)} = -10$, $w_{20}^{(1)} = 0$ d'autre part. Les valeurs de $w_0^{(2)}$, $w_1^{(2)}$ et $w_2^{(2)}$ reste indéfinis pour le moment.

On peut donc exprimer la valeur des neurones de la couche intermédiaire et le neurone de sortie par

$$z_1 = \mathrm{ReLU}(w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{10}^{(1)}) \tag{2}$$

$$z_2 = \mathrm{ReLU}(w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + w_{20}^{(1)}) \tag{3}$$

$$y = \mathrm{sign}(w_1^{(2)} z_1 + w_2^{(2)} z_2 + w_0^{(2)}) \tag{4}$$

Dans la suite, on va chercher à étudier comment est organiser le partitionnement de l'espace.

1. Tracer dans le plan (x_1,x_2) les séparatrice correspondant aux neurones z_1 et z_2.
2. Indiquer sur la même figure, les régions de (x_1,x_2) pour lesquelles z_1 est positif ou nul, et pareil pour z_2.
3. En déduire où, dans le plan (z_1,z_2) se situent les projections des données (en utilisant un schéma).
4. Toujours dans le plan (z_1,z_2), tracer une séparatrice sur le schéma permettant d'obtenir une bonne classification et, en déduire les paramètre de la dernière couche.
5. **Bonus** : Calculer le gradient de y par rapport à $w_{11}^{(1)}$.

Partie II : Apprentissage par renforcement

2 Les bandits [3 pts]

Rappeler comment fonctionne le modèle des bandits (on considérera un ensemble de n bras/bandits) :

1. Comment fonctionne chaque bras ?
2. Comment s'effectue l'apprentissage des paramètrse $Q^*(a)$.
3. Comment choisit-on une action dans le cas de la stratégie ϵ-glouton.
4. Que se passe-t-il lorsqu'on choisit $\epsilon = 0.5$ dans cette stratégie ?

3 Equation de Bellman [3 pts]

1. Rappeler comment s'exprime la fonction de valeur $Q_\pi(s, a)$ en fonction de : $p(s'|s, a)$, $r(s, a, s')$ et $V_\pi(s)$.
2. Expliquer pourquoi on a $V_\pi(s) = \sum_{a \in \mathcal{A}(s)} \pi(a|s) Q_\pi(s, a)$
3. En déduire une équation de la fonction de valeur $Q_\pi(s, a)$ ne dépendant plus de $V_\pi(s)$.

4 Application [4 pts]

On considère ici un environnement de type GridWorld. Sur la figure (**??**), l'image de gauche montre les récompenses obtenues en fonction de la case sur laquelle l'agent **arrive**, au milieu la valeur de la fonction de valeur $V_\pi(s)$ pour une politique π quelconque avec une valeur $\gamma = 0.8$ et à droite une numérotation des états.

0	10	0
0	0	5
-1	0	-1

6.2	9.4	8.9
1.8	6.0	5.6
-0.5	0.8	2.2

0	1	2
3	4	5
6	7	8

On considérera que cet univers fonctionne de la façon suivante :

— l'agent peut depuis chaque case de déplacer dans les 4 directions ou rester sur place.

— Quelque-soit le mouvement choisi, on a une probabilité 0.2 de rester dans l'état actuel et 0.8 d'aller vers l'état voulu par l'action.

— Si l'agent veut se diriger contre un mur, il sera obligé à rester sur place.

Répondez aux questions suivantes

1. Pour chaque état, calculer la fonction $Q_\pi(s, a)$
2. En déduire une politique π' meilleur que la politique π.
3. Appliquer une itération de l'algorithme "Value iteration" décrit ci-dessous, en partant de $V_{\pi'}^{(0)}(s) = V_\pi(s)$ afin d'estimer la fonction de valeur π'.
4. En déduire une nouvelle politique π''. Semble-t-elle meilleur que π' ?

Rappel : *Value Iteration*

Data: une politique π
$V^{(0)}(s) = 0 \; \forall s$;
$t = 0$;
while $t < t_{\max}$ **do**
  for *all* s' **do**
    $V_\pi^{(t+1)}(s) = \sum_{s'} p(s'|s,a) \left[r(s') + \gamma V_\pi^{(t)} \right]$
  end
  $t = t + 1$
end

Algorithm 1: How to write algorithms