

Module - Réseaux de neurones (PRTAL 5)

D.E.A Informatique, 2001-2002

8 janvier 2001, documents autorisés

1 Un problème de robot qui fuit

1.1 Définition du problème (simplement à lire)

On dispose d'un robot cylindrique, doté de deux roues motrices et d'une roue libre (pour l'équilibre). Le robot est doté de 8 capteurs de proximité, disposés régulièrement sur son pourtour (cf. figure 1).

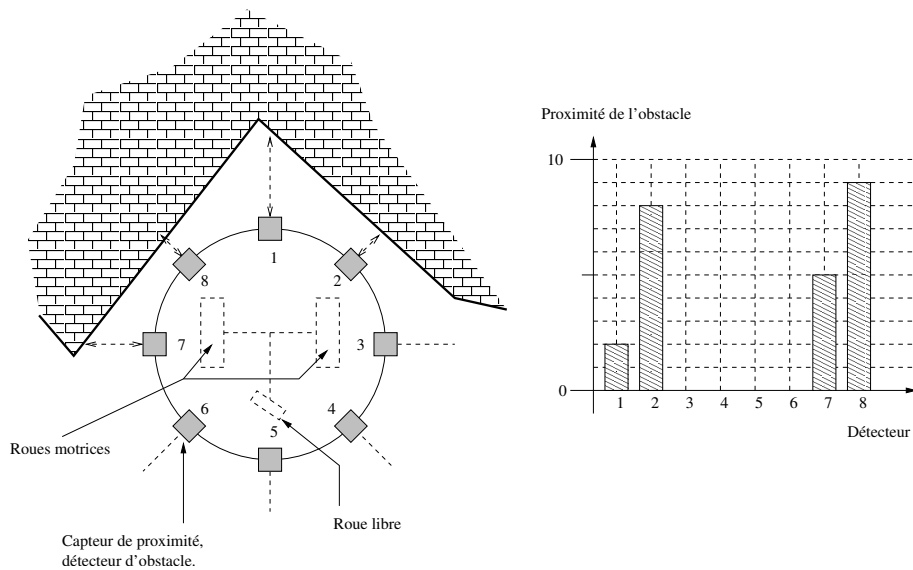


FIG. 1 – Robot vu de dessus (à gauche) et état des 8 capteurs (à droite).

Les capteurs de proximité renvoient une valeur entière entre 0 et 10. Si, dans la direction où "regarde" le capteur, il n'y a pas d'obstacle ou si l'obstacle est éloigné d'une distance supérieure au rayon du robot, le capteur retourne 0. Si il y a un obstacle contre le capteur, il retourne 10. La valeur du capteur prend les valeurs de 0 à 10 dans les cas intermédiaires, proportionnellement à la distance de l'obstacle. Les 8 capteurs fournissent ainsi au robot un jeu de 8

valeurs **entières** entre 0 et 10, indiquant au robot les obstacles qui se trouvent autour de lui.

Les roues motrices sont commandées en vitesse. On dispose donc pour la commande de deux valeurs continues, l'une étant la vitesse à imposer sur la roue gauche, et l'autre la vitesse à imposer sur la roue droite.

L'objectif est, à partir de l'état des 8 capteurs, de calculer les vitesses des deux roues motrices, pour que le robot évite les obstacles. Pour ce faire, on souhaite mettre en oeuvre un pré-traitement des valeurs de capteurs par réseaux de neurones, sans tenir compte des commandes motrices. C'est uniquement ce pré-traitement qui est l'objet des questions qui suivent.

1.2 Constitution d'un ensemble de données

On s'intéresse donc uniquement aux informations fournies par les 8 capteurs de proximité. Ces 8 valeurs sont les données sur lesquelles va travailler le pré-traitement que l'on veut mettre en place. On appellera espace des données l'ensemble de toutes les configurations possibles des 8 capteurs.

Questions :

1. Quelle est la taille de l'espace des données ? Justifiez.
2. **On supposera pour tout le problème** que le robot évolue dans une pièce carrée dont le côté vaut 4 fois le diamètre du robot. Les seuls obstacles sont les murs de la pièce. On construit un ensemble de données en positionnant 100000 fois le robot à un endroit choisi aléatoirement dans la pièce, avec une orientation aléatoire, en mémorisant à chaque fois la configuration de capteurs ainsi obtenue. Selon vous, la répartition des 100000 données ainsi obtenue dans l'espace de données est-elle uniforme ? Justifiez votre réponse en donnant des exemples.
3. En quoi votre réponse à la question précédente incite-t-elle à choisir un apprentissage non supervisé pour rendre compte de ces données.

1.3 Métrique « naïve » de l'espace de données

Questions :

1. Les algorithmes d'apprentissage non supervisé vus en cours mettent généralement en oeuvre deux types de distances. Quelles sont ces deux distances, et quel rôle jouent-elles dans le processus d'auto-organisation.
2. Nous utilisons toute la partie 1.3 la distance euclidienne d pour comparer deux données. Soit $X = (x_1, \dots, x_8)$ et $Y = (y_1, \dots, y_8)$ deux données, on rappelle que

$$d(X, Y) = \sqrt{\sum_{i=1}^{i=8} (x_i - y_i)^2} \quad (1)$$

Calculez $d(A, B)$ et $d(A, C)$, les données A , B et C étant représentées sur la figure 2. Pourquoi peut-on dire que l'usage de la distance euclidienne au niveau de l'espace des données que nous considérons, est inapproprié dans un algorithme d'auto-organisation.

3. Dites en quelques mots¹ les propriétés des données dont devrait rendre compte une distance adaptée.

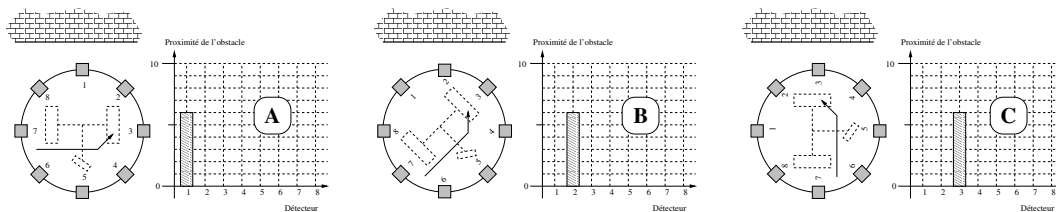


FIG. 2 – Trois positions du robot, et les trois données correspondantes.

1.4 Une métrique adéquate

Pour surmonter l'insuffisance de la distance euclidienne, on introduit une couche de neurones connectés aux capteurs. Soit $X = (x_1, \dots, x_8)$ une configuration des valeurs des capteurs (i.e. une donnée), on « alimente » la couche de neurones selon le principe illustré figure 3. Les poids des neurones sont fixes, et la sortie des neurones est directement la somme pondérée de ses entrées.

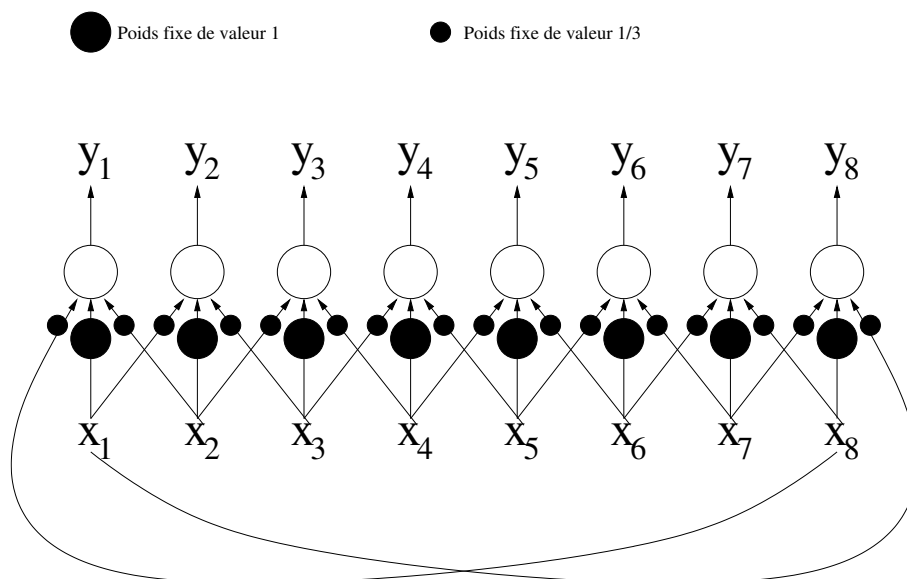


FIG. 3 – Une première couche de neurones traite les valeurs brutes des capteurs.

Questions :

1. Pour estimer la distance séparant deux données, on applique une distance euclidienne, mais cette fois-ci sur les sorties y de la couche de neurones (cf. 3). Calculez à nouveau, à partir des données de la figure 2, les distances entre A et B, puis entre A et C. Que constatez-vous ?
2. Expliquez comment l'ajout de la couche de neurones résout les problèmes évoqués dans la partie 1.3.

¹Pas de formules mathématiques, seulement vos intuitions.

1.5 Apprentissage non supervisé

On utilise la distance définie dans la partie 1.4 pour comparer une donnée aux prototypes d'un des algorithmes d'apprentissage non supervisé vus en cours. Les données présentées sont tirées de la base de 100000 données constituée en partie 1.2.

Questions :

1. Rappelez les grandes étapes communes aux algorithmes d'apprentissage non supervisés vus en cours, c'est-à-dire ceux qui conservent la topologie de l'espace d'entrée.
2. Quel effet désastreux observera-t-on lorsqu'on appliquera un algorithme de ce type sur nos données (indication : cet effet est lié à la répartition des données que nous avons recueillies dans l'espace des configurations possibles de capteurs).
3. Quelles données, parmi les 100000 recueillies, pourra-t-on retirer de la base d'apprentissage pour éviter l'effet désastreux.
4. On réalise l'apprentissage, sur les données restantes après élimination des données qui conduisent à l'effet désastreux. On utilise l'algorithme Growing Neural Gas. Quels sont les deux points qui différencient fondamentalement cet algorithme de l'algorithme des cartes auto-organisatrices de Kohonen ?
5. Selon vous, quelle sera la forme du maillage entre les prototypes ? Justifiez votre intuition.

1.6 Identification de lieux

On désire maintenant se servir de l'état des capteurs pour caractériser des lieux, c'est-à-dire dans le cas présent, différencier les coins des murs droits perçus à l'avant du robot. Pour cela, on peut utiliser deux modèles différents, un perceptron et un modèle de Hopfield. Pour chacun de ces deux cas, rappelez les principes d'apprentissage mis en oeuvre et dites comment vous vous y prendriez pour réaliser la fonction désirée. Des notions de complexité de la fonction à apprendre et de taille de corpus peuvent également être introduites.

2 Question de cours

Quels liens y a-t-il entre la règle de Hebb et la règle du perceptron ? Comment peut-on retrouver cette dernière par des méthodes mathématiques issues des techniques de descente de gradient ?