Introduction à l'apprentissage - M1 Informatique - 2014-2015
Mini-projet

Instructions

Vous réaliserez ce mini-projet par groupes de 2 étudiants.

Vous devrez rendre sur arche d'ici le vendredi 27 mars un rapport concis (illustré par des diagrammes et des captures d'écrans), ainsi que le programme demandé. Vous pourrez utiliser dans plusieurs questions le logiciel GINNet disponible sur `http://ginnet.gforge.inria.fr`

1 Classification de données bidimensionnelles

On dispose de quatre jeux de données (à récupérer sur arche.univ-lorraine.fr) sur lesquels on va tester différentes méthodes de classification. Les fichiers `data1.txt`, `data2.txt` et `data3.txt` contiennent chacun 1000 échantillons dans $\mathbb{R} \times \mathbb{R}$. A l'aide d'un logiciel adapté, visualisez les données de ces fichiers. Commentez qualitativement la répartition des points.

1.1 K-means

En choisissant le nombre de classes en fonction de la visualisation initiale des données, appliquez la méthode `k-means` aux trois fichiers de données bidimensionnelles. Visualisez et commentez la composition des classes.

1.2 Kohonen

Réalisez une classification par carte auto-organisatrice de Kohonen de taille minimale 8×8 pour chacun des trois fichiers de données bidimensionnelles.

Comment pourriez-vous utiliser les résultats pour améliorer la classification obtenue par `k-means` ?

2 Sonar

2.1 Classification

Récupérez les données `sonar.data` sur Arche. Transformez le fichier de façon à pouvoir l'importer avec GINNet. A l'aide des outils d'apprentissage non supervisé, analysez les données, leur classification, leur répartition, ... etc. en vous inspirant de ce qui a été fait précédemment sur les données bidimensionnelles.

2.2 Discrimination

Réalisez un apprentissage supervisé à l'aide d'un perceptron multicouche. Vous ferez vos propres choix en ce qui concerne la répartition des bases de données et l'architecture du réseau neuronal.

2.3 STDP

On considère un réseau totalement connecté de 60 neurones LIF ayant chacun en entrée une des 60 coordonnées des données de `sonar.data`. On applique un apprentissage STDP à ce modèle, avec une variation de poids :

$$\Delta w_{ij} = \begin{cases} w_1 & \text{si} \quad 0 < t_i^{(f)} - t_j^{(f)} < t_0 \\ -w_2 & \text{sinon} \end{cases}$$

A l'issue de cet apprentissage, on munit chaque neurone d'une étiquette `Rock` si sa fréquence moyenne de décharge est plus élevée pour une entrée de type `Rock` que pour une entrée de type `Mine`, ou d'une étiquette `Mine` dans le cas contraire.

Ce modèle est ensuite utilisé pour discriminer les signaux de sonar : lorsqu'une entrée est fournie, on observe les neurones émettant le plus d'impulsions, et la classe attribuée à l'entrée est l'étiquette majoritaire parmi ces neurones.

Programmez ce modèle et testez-le. Vous préciserez clairement tous les paramètres utilisés.

3 Classification de signaux de parole

Le fichier `parole.dt` contient 6772 échantillons correspondant chacun à un vecteur en dimension 241. Les 240 premières dimensions sont obtenues à partir de signaux audio courts (prononciation de différents phonèmes) par une analyse en énergie temps-fréquence multicanale, tandis que la dernière dimension indique le phonème prononcé (1 à 39) : on compte en général 36 phonèmes dans la langue française (en gros les différentes prononciations des consonnes et des voyelles), mais ces données expérimentales prennent en compte des cas particuliers supplémentaires.

Réalisez un apprentissage supervisé à l'aide d'un perceptron multicouche (vous pourrez tester plusieurs architectures). Observez les performances obtenues. En quoi ce problème est-il particulièrement difficile ?