TP1 : Réegression scalaire

Master IPS

28 Octobre 2022

(Pour le jeu de données de Reuters, essayez d'améliorer la précision du modèle utilisé, en changeant les paramètres du modèle, tels que le nombre de couches et leurs unités, et l'utilisation d'autres fonctions d'activation pour les couches cachées...)

On essaiera de prédire le prix moyen des maisons dans une certaine région de Boston au milieu des années 1970, à partir de quelques données sur la région pendant cette période, comme le taux de criminalité, le taux d'impôt local sur la propriété, etc. L'ensemble de données que nous allons utiliser comporte très peu de données, seulement 506 au total, réparties en 404 échantillons d'apprentissage et 102 échantillons de test, et chaque caractéristique des données d'entrée a une échelle différente.

Préparation des données :

1. Importer le jeu de données Boston housing de la librairie keras.datasets :

```
from keras.datasets import boston housing

(train data, train targets), (test data, test targets) = boston housing.load data()
```

Il serait problématique d'introduire dans un réseau de neurones des valeurs qui sont toutes dans des intervalles très différents. Le réseau pourrait être capable de s'adapter automatiquement à de telles données hétérogènes, mais cela rendrait certainement l'apprentissage plus difficile.

2. Normaliser les données. Pour chaque caractéristique des données d'entrée (une colonne dans la matrice des données d'entrée), il faut soustraire la moyenne de la caractéristique et la diviser par son écart type, de sorte que les valeurs de la caractéristique soient centrées autour de 0.

Construction du réseau :

3. Le faible nombre d'échantillons disponibles pour l'apprentissage entrainera une contrainte sur la construction du modèle, laquelle ?

4. La configuration typique pour la régression scalaire consiste à utiliser une couche de sortie avec une seule unité, sans fonction d'activation. Quel sera le résultat du réseau si une fonction d'activation sigmoid est appliquée à la sortie ?

5. Choisir une fonction de perte adéquate à ce problème, et en précisant l'erreur absolue moyenne (mae : Elle s'agit simplement de la valeur absolue de la différence entre les prédictions et les cibles) comme mesure à suivre pendant l'apprentissage compiler votre réseau. Vous pouvez utiliser l'optimisateur que vous préférez.

Validation de l'approche :

Puisque on a si peu de données, l'ensemble de validation finirait par être très petit. En conséquence, les scores de validation peuvent changer considérablement en fonction des données choisies pour la validation et ceux choisis pour l'apprentissage. La meilleure pratique dans de telles situations est d'utiliser la validation par K-fold

6. Faire l'apprentissage de votre modèle en 100 époques et 4-fold de validation.

7. le score de validation moyen. Sachant que les prix varient de 10 000$ à 50 000$ et qu'un score de 0.5 signifie que les prédictions sont décalées de 500$ en moyenne, votre modèle performe-t-il bien sur ce problème ?

8. Si votre modèle n'obtient pas des résultats satisfaisants, que pouvez-vous faire pour l'améliorer ? Essayer au moins une procédure pour améliorer votre modèle et effectuer une évaluation finale sur les données de test.