TDnote11

April 14, 2021

1 TD noté du 15/04/2021

L'objectif de ce TD est de construire un classifieur capable de reconnaitre des lettres manuscrites. Le jeu de données est disponible à l'adresse: http://ai.stanford.edu/~btaskar/ocr/. Vous pouvez obtenir des informations sur le format des des données.

Ici, les données sont accessibles via une bibliotheque data_extraction.py. Les données ont globalement le même format que pour *digits*:

- une matrice X, dont chaque ligne représente une image vectorisée.
- un vecteur Y dont les coefficients représente la sortie, selon le code suivant: 0 = 'a', 1 = 'b', ... , 25 = 'z'
- un array numpy Z dont les éléments sont des matrices de 0 et 1 image bitmap de la lettre manuscrite

Pour pouvoir utiliser ces tableaux numpy, il vous faut:

- copier les fichiers *letter.data* et *data_extraction.py* dans un répertoire.
- faire un import data_extraction dans votre programme.

Exemple : L'élément d'indice *1234* est un 'b', ce qui correspond à une sortie de 1 d'après le code décrit ci-dessus.

[1]:
```
import matplotlib.pyplot as plt
import numpy as np

i = 0

X = []
Y = []
Z = []
with open("letter.data", 'r') as infile:
    for line in infile:
        line = line.split()
        V = line[6:]
        V = [int(v) for v in V]
        V = np.array(V)
        X.append(V)
        Y.append(ord(line[1]) - 97)
        V = V.reshape(16,8)
```

```
        Z.append(V)

X = np.array(X)
Y = np.array(Y)
Z = np.array(Z)
```

[2]:
```
X[1234]
```

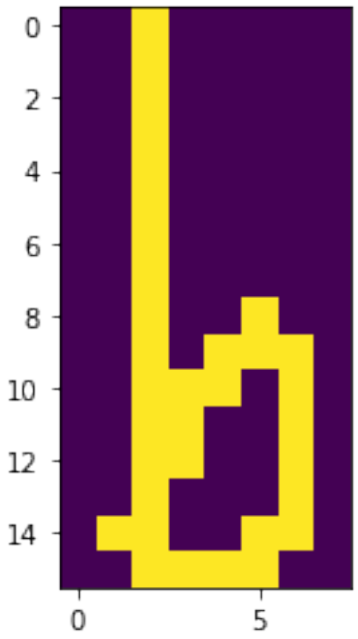
[2]:
```
array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0,
       0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0])
```

[3]:
```
Y[1234]
```

[3]:
```
1
```

[4]:
```
plt.imshow(Z[1234])
```

[4]:
```
<matplotlib.image.AxesImage at 0x1148fa160>
```



Vous devez copier le fichier 'letter.data' dans votre répertoire de travail. Vous pouvez importer alors data_extraction.py pour avoir accès aux tableaux X, Y et Z.

1.1 Exercice 1

Observez les données.

- Les classes sont-elles équilibrées (i.e toutes les classes ont le même effectif)?
- Pouvez-vous expliquer pourquoi, en regardant la manière dont a été construit ce jeu de données? (cf lien au début)
- Quelle est l'effectif de la classe correspondant au 'n'?
- En supposant qu'un classifieur réponde systématiquement 'n', quelle serait son taux d'erreur?

On prendra cette valeur pour baseline. (cela signifie qu'un classifieur faisant plus d'erreur n'a rien appris)

1.2 Exercice 2

En vous servant des fichiers qui ont déjà été produits en TD précédemment:

- Séparez les données en un ensemble d'apprentissage (90%) et un ensemble de test (10%).
- Construisez un classifieur linéaire (on pourra considérer la loss *-log-vraisemblance*) que vous apprendrez sur l'ensemble d'apprentissage.
- Quel taux d'erreur fait ce classifieur sur l'ensemble de test?

1.3 Exercice 3

Même questions que précédement, avec un réseau de neurones à 1 couche cachée. Vous testerez les configurations suivantes:

- couche cachée à 10, 20, et 30 neurones.
- fonction d'activation: ReLU
- Pour ces configurations, quel taux d'erreur sur l'ensemble de test?

[ ]: