

Taller 3: Modelos No-Lineales para Clasificación

Aprendizaje de Máquina
Universidad Tecnológica de Pereira
Pereira, Colombia

Neural Networks

Utilizando la base de datos *Digits* de *sklearn*, construir una red neuronal para clasificación, que permita identificar diferentes tipos de dígitos.

1. La base de datos a utilizar se encuentra en el módulo ***from sklearn.datasets import load_digits***
2. Investigue y utilice la función *train_test_split* para realizar la partición de la base de datos entre los conjuntos de **Train** y **Test**
3. Realice diferentes pruebas de configuración de la red neuronal, con el fin de encontrar el mejor conjunto de parámetros que realiza el reconocimiento (#-Capas, #-Neuronas por capa).
4. reporte la matriz de confusión y el *accuracy* del modelo para las diferentes configuraciones de las redes encontradas.

El siguiente ejemplo realiza la carga de la base de datos:

```
from sklearn.datasets import load_digits
digits = load_digits()
print(digits.data.shape)
import matplotlib.pyplot as plt
plt.gray()
plt.matshow(digits.images[0])
plt.show()
```

Support Vector Machines

Utilizando la base de datos *Wine* de *sklearn*, construir una SVM para clasificación, que permita identificar las respectivas etiquetas.

1. La base de datos *Wine* se encuentra en el módulo ***from sklearn.datasets import load_wine***
2. Investigue y utilice la función *train_test_split* para realizar la partición de la base de datos entre los conjuntos de **Train** y **Test**

3. Realice diferentes pruebas de configuración de la SVM, con el fin de encontrar el mejor conjunto de parámetros que realiza el reconocimiento (#-Capas, #-Neuronas por capa). Pruebe diferentes tipos de Kernel para cada base de datos y discuta a partir de los resultados obtenidos el que obtiene un mejor desempeño
4. reporte la matriz de confusión y el *accuracy* del modelo para las diferentes configuraciones de las SVMs encontradas.

Comparación de Métodos: Face Recognition

En este experimento trabajaremos con la base de datos *Face in the Wild*. Esta base de datos es una colección de imágenes JPEG de famosos. Todos los detalles de la base de datos se encuentran disponibles en la web oficial: <http://vis-www.cs.umass.edu/lfw/>

Para cargar la base de datos, utilizaremos la función *fetch* que descarga la base de datos para su procesamiento:

```
from sklearn.datasets import fetch_lfw_people  
  
lfw_people = fetch_lfw_people(min_faces_per_person=70, resize=0.4)
```

Un ejemplo para verificar el contenido de la base de datos es:

```
print lfw_people.target_names  
  
data = lfw_people.data  
  
target = lfw_people.target  
  
print data.shape, target.shape  
  
# Ejemplo para graficar  
  
import matplotlib.pyplot as plt  
  
plt.matshow(lfw_people.images[0])  
  
plt.show()
```

En este punto se deben implementar dos estrategias de clasificación una basada en Redes Neuronales y la otra en SVM, la idea es que prueben diferentes configuraciones y comparen los desempeños arrojados para las diferentes estrategias de clasificación

Acerca del trabajo

Fecha de entrega: 20 de abril 11:59pm. Correo hernan.garcia@utp.edu.co

Importante: Reporte la media y desviación estándar. La medida de desempeño es el *accuracy* (porcentaje de acierto). Importante. Se debe presentar un informe, junto con los scripts que se usaron para generar los experimentos y resultados que aparezcan en el informe. El informe debe incluir los resultados solicitados, con una discusión sobre los mismos. El informe no tiene que incluir marco teórico.

Resolver en parejas. Se cuantificará las diferentes configuraciones para clasificación que realicen por grupo.