

### REPORTE DE PRÁCTICA

#### IDENTIFICACIÓN DE LA PRÁCTICA

Práctica	6	Nombre de la práctica	Clasificación múltiple
Fecha	14/06/2025	Nombre del profesor	Alma Nayeli Rodríguez Vázquez
Nombre del estudiante		Jesús Alberto Aréchiga Carrillo	

#### OBJETIVO

El objetivo de esta práctica consiste en implementar una red neuronal para clasificación múltiple.

#### PROCEDIMIENTO

Realiza la implementación siguiendo estas instrucciones.

Implementa una red neuronal para clasificación múltiple en Python. Para ello, considera el set de datos del archivo "dataset\_multiclassOK.csv".

#### IMPLEMENTACIÓN

Agrega el código de tu implementación aquí.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('dataset_multiclassOK.csv')
data.head()

m,n = data.shape
array = data.values
X = array[:,0:n-1]
Y = array[:,n-1]

#Normalizar
mean = X.mean(axis=0)
std = X.std(axis=0, ddof=1)
X = (X-mean)/std

#Parametros
alpha = 0.1
numClases = np.size(np.unique(Y))
w1 = np.ones((10,5))
w2 = np.ones((4,10))
epochMax = 200
```

```
#One hot encoding
D = np.zeros((m,numClases))
for i in range(m):
    D[i,Y[i]-1] = 1

def sigmoid(v):
    y = 1 / (1 + np.exp(-v))
    return y

def softmax(v):
    y = np.exp(v) / np.sum(np.exp(v))
    return y

#Entrenamiento
convEntropia = np.zeros(epochMax)
error = np.zeros(m)
for epoch in range(epochMax):
    for i in range(m):
        #Calcular la salida de la red
        xi = X[i,:]
        v1 = np.dot(w1, xi)
        y1 = sigmoid(v1)

        v = np.dot(w2, y1)
        y = softmax(v)

        #Retropropagacion
        di = D[i,:]
        e = di - y
        delta = e

        e1 = np.matmul(w2.transpose(), delta)
        delta1 = y1*(1-y1)*e1

        #Regla Delta generalizada
        dw1 = alpha * delta1.reshape(1,10).transpose() * xi
        dw2 = alpha * np.reshape(delta, (4,1)) * np.reshape(y1, (1,10))

        w1 = w1 + dw1
        w2 = w2 + dw2
        error[i] = np.sum(np.abs(e))
    convEntropia[epoch] = np.sum(error)/m

plt.plot(convEntropia)
plt.xlabel('Epocas')
plt.ylabel('Error')
plt.show()

y = np.zeros((m,numClases))
for i in range(m):
    xi = X[i,:]
    v1 = np.dot(w1, xi)
    y1 = sigmoid(v1)
```

```
v = np.dot(w2, y1)
y[i] = np.round(softmax(v))

acc = np.sum(y*D)/m
acc

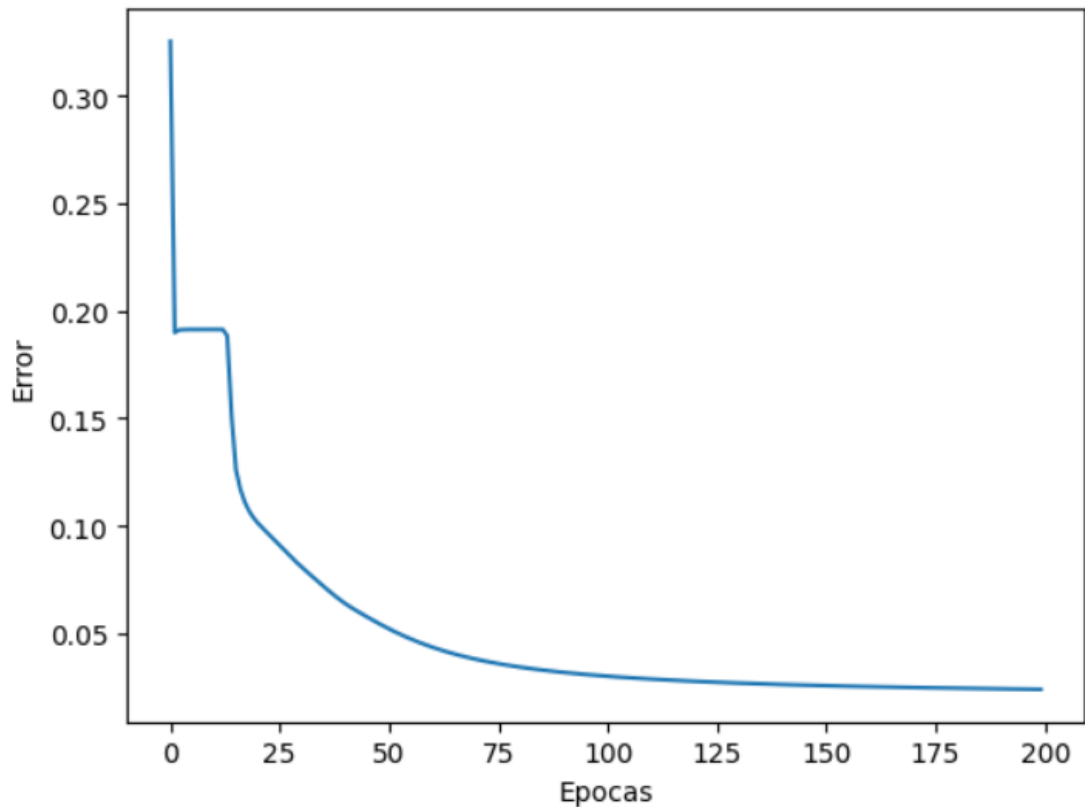
#Prediccion
cliente = [65, 1,1,32000,45000]
cliente = (cliente-mean)/std
v1_ = np.dot(w1, cliente)
y1_ = sigmoid(v1_)

v_ = np.dot(w2, y1_)
y_ = np.round(softmax(v_))
print('Cliente =>', 'clase', np.argmax(y_)+1)
```

### RESULTADOS

Agrega la(s) imagen(es) con los resultados obtenidos en los espacios indicados.

#### Gráfica de convergencia.



Captura de pantalla en la que se observe la salida de la red neural “y”.

```
y
array([[1., 0., 0., 0.],
       [1., 0., 0., 0.],
       [1., 0., 0., 0.],
       ...,
       [0., 1., 0., 0.],
       [0., 1., 0., 0.],
       [0., 1., 0., 0.]])
```

Captura de pantalla en la que se observe la salida de la red neural “y” para un nuevo dato de entrada, es decir, la predicción para un nuevo dato de entrada.

```
y_
array([0., 0., 0., 1.])
```

## CONCLUSIONES

Escribe tus observaciones y conclusiones.

La clasificación múltiple es una técnica fundamental en el aprendizaje automático que permite asignar instancias a una de varias categorías posibles. A diferencia de la clasificación binaria, este enfoque implica una mayor complejidad debido al manejo simultáneo de múltiples clases y la necesidad de representaciones internas que capten mejor las diferencias entre ellas.

A lo largo de este proyecto se ha demostrado la importancia del preprocesamiento de datos, la codificación adecuada de las clases (como el uso de one-hot encoding) y la elección de funciones de activación y evaluación que se ajusten al problema multiclase. La clasificación múltiple es ampliamente aplicable en contextos reales como el diagnóstico médico, el reconocimiento de imágenes, la segmentación de clientes y la predicción de comportamientos, entre otros. En resumen, la clasificación múltiple representa una herramienta poderosa para la toma de decisiones automatizada, siempre que se aborde con modelos bien estructurados, datos representativos y métricas de evaluación adecuadas.