DL-2A(OCR)

1. Import Required Libraries

Import pandas as pd

Import numpy as np

Import tensorflow as tf

From sklearn.model\_selection import train\_test\_split

From sklearn.preprocessing import LabelEncoder, StandardScaler

From tensorflow.keras.utils import to\_categorical

Import matplotlib.pyplot as plt

1. Load the Dataset

You can manually download the dataset and load it using:

# Dataset URL

url = ‘https://archive.ics.uci.edu/ml/machine-learning-databases/letter-recognition/letter-recognition.data’

# Define column names from dataset description

Columns = [‘letter’, ‘x-box’, ‘y-box’, ‘width’, ‘high’, ‘onpix’,

‘x-bar’, ‘y-bar’, ‘x2bar’, ‘y2bar’, ‘xybar’, ‘x2ybr’,

‘xy2br’, ‘x-ege’, ‘xegvy’, ‘y-ege’, ‘yegvx’]

# Load data

Df = pd.read\_csv(url, names=columns)

1. Preprocess the Data

# Features and labels

X = df.drop(‘letter’, axis=1).values

Y = df[‘letter’].values

# Encode labels (A-Z -> 0-25)

Label\_encoder = LabelEncoder()

Y\_encoded = label\_encoder.fit\_transform(y)

# One-hot encode labels

Y\_categorical = to\_categorical(y\_encoded)

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y\_categorical, test\_size=0.2, random\_state=42)

# Scale features

Scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

1. Build the DNN Model

Model = tf.keras.Sequential([

Tf.keras.layers.Dense(128, activation=’relu’, input\_shape=(X\_train.shape[1],)),

Tf.keras.layers.Dense(64, activation=’relu’),

Tf.keras.layers.Dense(26, activation=’softmax’) # 26 classes for A-Z

])

Model.compile(optimizer=’adam’,

Loss=’categorical\_crossentropy’,

Metrics=[‘accuracy’])

1. Train the Model

History = model.fit(X\_train, y\_train, epochs=30, batch\_size=32,

Validation\_split=0.2, verbose=1)

1. Evaluate the Model

Loss, accuracy = model.evaluate(X\_test, y\_test)

Print(f”Test Accuracy: {accuracy \* 100:.2f}%”)

1. Plot Accuracy Curve

Plt.plot(history.history[‘accuracy’], label=’Train Accuracy’)

Plt.plot(history.history[‘val\_accuracy’], label=’Validation Accuracy’)

Plt.xlabel(“Epoch”)

Plt.ylabel(“Accuracy”)

Plt.title(“Training and Validation Accuracy”)

Plt.legend()

Plt.grid(True)

Plt.show()