DL-2B(IMDB)

1. Import Required Libraries

Import tensorflow as tf

From tensorflow.keras.datasets import imdb

From tensorflow.keras.preprocessing.sequence import pad\_sequences

Import matplotlib.pyplot as plt

1. Load and Preprocess IMDB Dataset

# Load dataset with top 10,000 frequent words

Vocab\_size = 10000

(X\_train, y\_train), (X\_test, y\_test) = imdb.load\_data(num\_words=vocab\_size)

# Pad sequences to ensure equal input length

Max\_length = 300

X\_train\_padded = pad\_sequences(X\_train, maxlen=max\_length, padding=’post’)

X\_test\_padded = pad\_sequences(X\_test, maxlen=max\_length, padding=’post’)

1. Build the Deep Neural Network

Model = tf.keras.Sequential([

Tf.keras.layers.Embedding(input\_dim=vocab\_size, output\_dim=32, input\_length=max\_length),

Tf.keras.layers.Flatten(),

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

Tf.keras.layers.Dense(1, activation=’sigmoid’) # Binary output

])

Model.compile(optimizer=’adam’, loss=’binary\_crossentropy’, metrics=[‘accuracy’])

1. Train the Model

History = model.fit(X\_train\_padded, y\_train, epochs=5, batch\_size=128,

Validation\_split=0.2, verbose=1)

1. Evaluate the Model

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

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

1. Plot Accuracy Graph

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()