**Практична робота 6**

**Дослідження методів оптимізації інформаційно-управляючих систем**

Мета: Використати навички роботи з TensorFlow для Python для імплементації нейронних мереж в інформаційно-управляючі системи.

Отримати практичні навички побудови мереж глибокого навчання для роботи з текстовою інформацією.

import matplotlib.pyplot as plt

from tensorflow.keras.datasets import reuters

from tensorflow.keras.preprocessing import sequence

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Embedding, LSTM, Dropout, Dense, Bidirectional

from tensorflow.keras.optimizers import Adam

num\_words = 10000

max\_review\_length = 500

(X\_train, y\_train), (X\_test, y\_test) = reuters.load\_data(num\_words=num\_words)

X\_train = sequence.pad\_sequences(X\_train, maxlen=max\_review\_length)

X\_test = sequence.pad\_sequences(X\_test, maxlen=max\_review\_length)

embedding\_vector\_length = 50

model = Sequential([

    Embedding(num\_words, embedding\_vector\_length, input\_length=max\_review\_length),

    Dropout(0.3),

    Bidirectional(LSTM(128, return\_sequences=True)),

    Dropout(0.3),

    LSTM(64),

    Dropout(0.3),

    Dense(128, activation='relu'),

    Dropout(0.3),

    Dense(46, activation='softmax')

])

optimizer = Adam(learning\_rate=0.0005)

model.compile(loss='sparse\_categorical\_crossentropy', optimizer=optimizer, metrics=['accuracy'])

print(model.summary())

train\_history = model.fit(

    X\_train, y\_train,

    epochs=20,

    batch\_size=16,

    validation\_split=0.2,

    verbose=2

)

scores = model.evaluate(X\_test, y\_test, verbose=1)

print(f"Точність: {scores[1] \* 100:.2f}%")

def show\_train\_history(train\_history, train\_metric, val\_metric):

    plt.plot(train\_history.history[train\_metric])

    plt.plot(train\_history.history[val\_metric])

    plt.title('Тренування')

    plt.ylabel(train\_metric)

    plt.xlabel('Епоха')

    plt.legend(['Навчання', 'Валідація'], loc='upper left')

    plt.show()

show\_train\_history(train\_history, 'accuracy', 'val\_accuracy')

show\_train\_history(train\_history, 'loss', 'val\_loss')





