

Лабораторна робота 7 ІАД

Підготував студент 3 курсу МІТ-31

Крижановський Олександр

Завантаження та аналіз даних

- Завантажте набори даних із зазначених посилань, використовуючи **pandas** для аналізу

```
import pandas as pd

# Завантаження даних
data_3 = pd.read_csv("path_to_file_3.csv") # Замініть на відповідний шлях до файлу
data_5 = pd.read_csv("path_to_file_5.csv")

# Аналіз даних
print(data_3.info())
print(data_5.info())
print(data_3.describe())
print(data_5.describe())
```

Очистка даних:

- Видалення пропущених значень (NaN), нормалізація та кодування категорій.

```
data_3 = data_3.dropna()
data_5 = data_5.dropna()
```

Розділення на вхідні змінні та цільові значення:

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Для регресії (дані з лабораторної 3)
X_3 = data_3.iloc[:, :-1].values # Усі колонки, крім останньої
y_3 = data_3.iloc[:, -1].values  # Остання колонка (ціль)

# Для класифікації (дані з лабораторної 5)
X_5 = data_5.iloc[:, :-1].values
y_5 = data_5.iloc[:, -1].values

# Масштабування
scaler = StandardScaler()
X_3 = scaler.fit_transform(X_3)
X_5 = scaler.fit_transform(X_5)

# Розділення на тренувальний і тестовий набори
X_3_train, X_3_test, y_3_train, y_3_test = train_test_split(X_3, y_3, test_size=0.2, random_state=42)
X_5_train, X_5_test, y_5_train, y_5_test = train_test_split(X_5, y_5, test_size=0.2, random_state=42)
```

Побудова нейронних мереж

1. Регресія (дані з лабораторної 3):

```
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense

# Модель нейронної мережі
model_regression = Sequential([
    Dense(64, activation='relu', input_shape=(X_3_train.shape[1],)),
    Dense(32, activation='relu'),
    Dense(1) # Вихідний шар для регресії
])

model_regression.compile(optimizer='adam', loss='mse', metrics=['mae'])

# Тренування моделі
model_regression.fit(X_3_train, y_3_train, epochs=50, batch_size=32, validation_split=0.2)

# Оцінка моделі
loss, mae = model_regression.evaluate(X_3_test, y_3_test)
print(f"Регресія - MAE: {mae}")
```

Класифікація (дані з лабораторної 5):

```

# Перекодування цільової змінної (якщо потрібно)
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
y_5_train = encoder.fit_transform(y_5_train)
y_5_test = encoder.transform(y_5_test)

# Модель нейронної мережі
model_classification = Sequential([
    Dense(64, activation='relu', input_shape=(X_5_train.shape[1],)),
    Dense(32, activation='relu'),
    Dense(len(set(y_5_train)), activation='softmax') # Вихідний шар для класифікації
])

model_classification.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Тренування моделі
model_classification.fit(X_5_train, y_5_train, epochs=50, batch_size=32, validation_split=0.2)

# Оцінка моделі
loss, accuracy = model_classification.evaluate(X_5_test, y_5_test)
print(f"Класифікація - Accuracy: {accuracy}")

```

Результати

1. Вивести графіки втрат (loss) для кожної моделі під час тренування:

```

import matplotlib.pyplot as plt

# Для регресії
history_reg = model_regression.history.history
plt.plot(history_reg['loss'], label='Train Loss')
plt.plot(history_reg['val_loss'], label='Validation Loss')
plt.legend()
plt.title("Регресія: Зміна Loss під час навчання")
plt.show()

# Для класифікації
history_class = model_classification.history.history
plt.plot(history_class['loss'], label='Train Loss')
plt.plot(history_class['val_loss'], label='Validation Loss')
plt.legend()
plt.title("Класифікація: Зміна Loss під час навчання")
plt.show()

```

Висновки

На основі отриманих результатів можна зробити висновки про:

- Точність і якість побудованих моделей.
- Можливі причини помилок (недостатня кількість даних, невідповідна архітектура NN тощо).

- Досвід застосування TensorFlow для задач машинного навчання.