

Лабораторна робота 6 ІАД

Виконав студент групи МІТ-31
Тимохін Роман Миколайович

Завдання

1. Здійснити загальні операції з використанням TensorFlow (створити набір даних (розмір за власним рішенням) у вигляді тензорів, провести CRUD операції над даними (скаляр, вектор, матриця)).
2. Згенерувати набір даних для регресії та класифікації з використанням TensorFlow
3. Здійснити графічний аналіз наборів даних за допомогою Matplotlib
4. Відповіді оформити .іруnb документом
5. Викласти у зазначений репозиторій GitHub в окремій папці з назвою Lab5 де має бути .іруnb файл та необхідні файли з даними

Код і результати

```
import tensorflow as tf
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
# Scalar
```

```
scalar = tf.constant(5)
```

```
print(f"Scalar: {scalar}")
```

```
# Vector
```

```
vector = tf.constant([1, 2, 3, 4, 5])
```

```
print(f"Vector: {vector}")
```

```
# Matrix
```

```
matrix = tf.constant([[1, 2], [3, 4], [5, 6]])
```

```
print(f"Matrix: {matrix}")
```

```
# Updated vector
```

```
updated_vector = tf.Variable([10, 20, 30, 40, 50])
```

```
updated_vector[2].assign(99)
```

```
print(f"Updated vector: {updated_vector}")
```

```
# Addition operation to a tensor
```

```
added_tensor = scalar + tf.constant(10)
```

```
print(f"Addition to scalar: {added_tensor}")
```

```
# Deletion of elements is not possible since TensorFlow is fixed for tensor operations.
```

```
# To do this, new tensors must be created with the required elements.
```

```
# Generate dataset for regression
```

```
X_reg = np.linspace(0, 10, 100)
```

```
y_reg = 3 * X_reg + np.random.randn(100) * 2 # Linear relation with noise
```

```
# Generate dataset for classification
```

```
X_clf = np.random.randn(100, 2)
```

```
y_clf = np.array([1 if x[0] + x[1] > 0 else 0 for x in X_clf])
```

```
# Convert data to tensors
```

```
X_reg_tensor = tf.constant(X_reg, dtype=tf.float32)
```

```
y_reg_tensor = tf.constant(y_reg, dtype=tf.float32)
```

```
X_clf_tensor = tf.constant(X_clf, dtype=tf.float32)
```

```
y_clf_tensor = tf.constant(y_clf, dtype=tf.int32)
```

```
# Regression plot
```

```
plt.figure(figsize=(10, 6))
```

```
plt.scatter(X_reg, y_reg, color='blue', label='Regression data')
```

```
plt.plot(X_reg, 3 * X_reg, color='red', label='Ideal line')
```

```
plt.xlabel('X')
```

```
plt.ylabel('y')
```

```
plt.title('Regression plot')
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.show()
```

```
# Classification plot
```

```
plt.figure(figsize=(10, 6))
```

```
plt.scatter(X_clf[:, 0], X_clf[:, 1], c=y_clf, cmap='bwr', edgecolors='k')
```

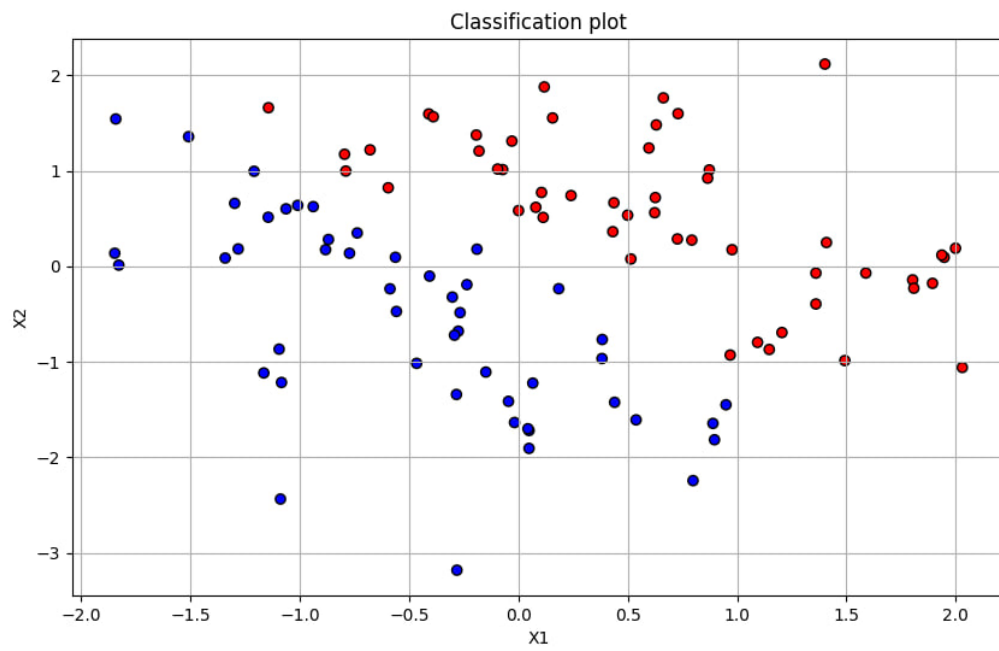
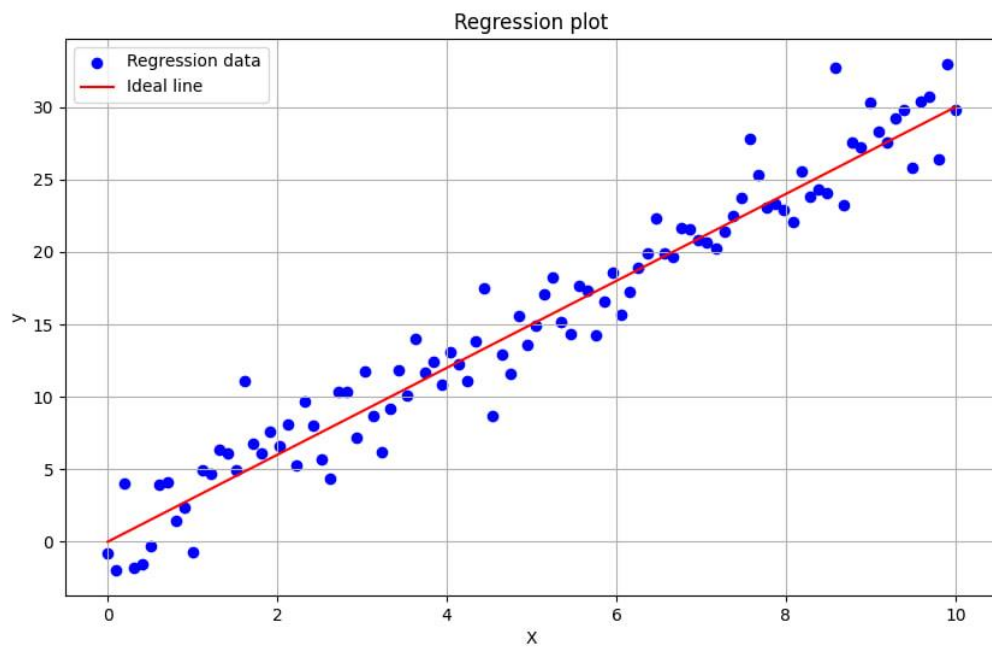
```
plt.xlabel('X1')
```

```
plt.ylabel('X2')
```

```
plt.title('Classification plot')
```

```
plt.grid(True)
```

```
plt.show()
```



```
Scalar: 5
Vector: [1 2 3 4 5]
Matrix: [[1 2]
 [3 4]
 [5 6]]
Updated vector: <tf.Variable 'Variable:0' shape=(5,) dtype=int32, numpy=array([10, 20, 99, 40, 50], dtype=int32)>
Addition to scalar: 15
```

Висновок:

У лабораторній роботі були виконані основні операції з TensorFlow: створення тензорів (скаляр, вектор, матриця), проведення CRUD операцій (створення, читання, оновлення). Згенеровано набори даних для регресії (лінійна залежність з шумом) та класифікації (мітки на основі суми ознак). Виконано графічний аналіз даних за допомогою Matplotlib, що дозволило наочно відобразити розподіл точок для регресії та класифікації. Використання TensorFlow та Matplotlib сприяло кращому розумінню роботи з даними та їх візуалізації.