

ЛАБОРАТОРНА РОБОТА № 8

Ресурси Keras. TensorFlow. Навчання лінійної регресії

Мета: дослідження ресурсу Keras і TensorFlow. Застосування TensorFlow.

Хід роботи

Завдання 8.1.

Завдання полягало у реалізації коду з навчального посібника для наближення лінійної функції $f = kx + b$ (де цільові значення $k = 2$ та $b = 1$). Початковий код був написаний для застарілої версії TensorFlow 1.x, що призвело до помилок сумісності.

Лістинг програми LR_8_task_1.py:

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)
m = 1000
X_data = np.random.rand(m, 1).astype(np.float32)
epsilon = np.random.normal(0, 0.5, (m, 1)).astype(np.float32)
y_data = 2 * X_data + 1 + epsilon

k = tf.Variable(tf.random.normal([1, 1]))
b = tf.Variable(tf.zeros([1]))

learning_rate = 0.0005
epochs = 20000
batch_size = 200

optimizer = tf.optimizers.SGD(learning_rate)

for epoch in range(1, epochs + 1):
    indices = np.random.choice(m, batch_size)
    X_batch = X_data[indices]
    y_batch = y_data[indices]

    with tf.GradientTape() as tape:
        y_pred = tf.matmul(X_batch, k) + b
        loss = tf.reduce_sum(tf.square(y_pred - y_batch))

    grads = tape.gradient(loss, [k, b])
    optimizer.apply_gradients(zip(grads, [k, b]))

    if epoch % 1000 == 0:
        print(f"Епоха {epoch}: {loss.numpy():.6f}, k={k.numpy()[0][0]:.4f}, b={b.numpy()[0]:.4f}")
```

					ДУ «Житомирська політехніка».25.121.12.000 – Лр8						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Затилук Д.О.			Звіт з лабораторної роботи			Лім.	Арк.	Аркушів	
Перевір.		Масевський О. В.								1	3
Керівник								ФІКТ Гр. ІПЗ-22-3			
Н. контр.											
Зав. каф.											

```
plt.scatter(X_data, y_data, alpha=0.3, label="Дані")
plt.plot(X_data, X_data * k.numpy()[0][0] + b.numpy()[0], color='red', label="Передбачена лінія")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()
plt.show()

output_model_file = 'model.pkl'
with open(output_model_file, 'wb') as f:
    pickle.dump(regressor, f)

with open(output_model_file, 'rb') as f:
    regressor_model = pickle.load(f)

y_test_pred_new = regressor_model.predict(X_test)
print("\nNew mean absolute error =", round(sm.mean_absolute_error(y_test,
y_test_pred_new), 2))
```

```
Епоха 1000: 47.520920, k=1.9256, b=1.0863
Епоха 2000: 41.904526, k=1.9130, b=1.0953
Епоха 3000: 43.033813, k=1.9100, b=1.0889
Епоха 4000: 41.044807, k=1.9273, b=1.0869
Епоха 5000: 43.953590, k=1.9327, b=1.0689
Епоха 6000: 51.363861, k=1.9254, b=1.0760
Епоха 7000: 47.622303, k=1.9406, b=1.0759
Епоха 8000: 50.781769, k=1.9396, b=1.0827
Епоха 9000: 40.120747, k=1.9149, b=1.1001
Епоха 10000: 56.158546, k=1.9061, b=1.0715
Епоха 11000: 50.498825, k=1.9280, b=1.0944
Епоха 12000: 48.172012, k=1.9281, b=1.0834
Епоха 13000: 54.997482, k=1.9066, b=1.0879
Епоха 14000: 50.518562, k=1.9165, b=1.0834
Епоха 15000: 51.272049, k=1.9253, b=1.0917
Епоха 16000: 46.007286, k=1.9107, b=1.0929
Епоха 17000: 53.709320, k=1.9254, b=1.0794
Епоха 18000: 53.353722, k=1.9406, b=1.0756
Епоха 19000: 53.459763, k=1.9141, b=1.0735
Епоха 20000: 54.492744, k=1.9120, b=1.0844
```

Рис.8.1. Результат виконання програми

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр8	Арк.
		Масвський О. В..				2
Змн.	Арк.	№ докум.	Підпис	Дата		

Figure 1

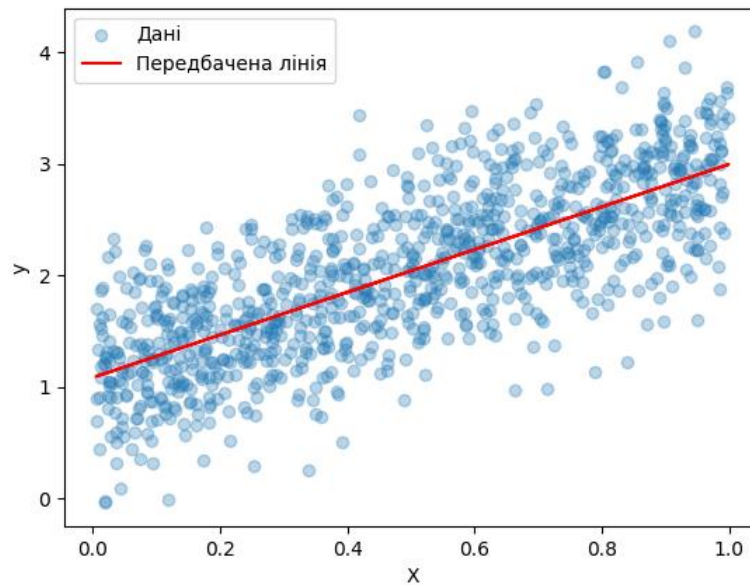


Рис.8.2. Графік функції

Значення коефіцієнтів k (нахил) та b (зсув), отримані моделлю, дуже близькі до істинних значень (2.0 та 1.0), на яких генерувалися дані. Це підтверджує, що алгоритм градієнтного спуску успішно мінімізував функцію втрат, і модель знайшла правильну лінійну залежність.

Висновок: Я дослідив ресурс Keras і TensorFlow. Застосував TensorFlow.

Посилання на Git: https://github.com/ipz223-zdo/AIS_Labs

		Затилук Д.О.			ДУ «Житомирська політехніка».25.121.12.000 – Лр8	Арк.
		Масвський О. В..				3
Змн.	Арк.	№ докум.	Підпис	Дата		