

## Лабораторна №8

### «Ресурси Keras. TensorFlow. Навчання лінійної регресії»

Мета: Дослідження ресурсу Keras і TensorFlow. Застосування TensorFlow.

#### Хід роботи

Необхідно навчити нейронну мережу (один нейрон) відновлювати лінійну залежність виду:

Завдання полягає у знаходженні коефіцієнтів  $k$  та  $b$  на основі зашумлених даних. Істинні значення:  $k = 2$ ,  $b = 1$ .

#### Програмна реалізація

Було написано скрипт мовою Python з використанням бібліотеки tensorflow. Оскільки методичні вказівки базуються на першій версії TensorFlow, у коді використано модуль сумісності tensorflow.compat.v1.

Алгоритм роботи програми:

1. Генерація 1000 випадкових точок  $X$  та обчислення  $Y = 2X + 1 + \text{шум}$ .
2. Оголошення змінних  $k$ ,  $b$  з випадковою ініціалізацією.
3. Визначення функції втрат Loss Function як середньоквадратичної помилки MSE.
4. Використання оптимізатора градієнтного спуску GradientDescentOptimizer для мінімізації помилки.
5. Запуск сесії навчання на 2000 епох.

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow.compat.v1 as tf

tf.disable_eager_execution()

print("--- ЛАБОРАТОРНА РОБОТА №8: TensorFlow Linear Regression ---")

num_samples = 1000
batch_size = 100
epochs = 2000

X_data = np.random.rand(num_samples).astype(np.float32)
noise = np.random.normal(0, 1.41, num_samples).astype(np.float32)
y_data = X_data * 2 + 1 + noise
```

Перевір.	Маєвський О		
Реценз.			
Н. Контр.			
Зав.каф.			

```

x_ph = tf.placeholder(tf.float32, shape=[None])
y_ph = tf.placeholder(tf.float32, shape=[None])

k = tf.Variable(tf.random_normal([1]), name='slope')
b = tf.Variable(tf.zeros([1]), name='bias')

y_pred = tf.multiply(x_ph, k) + b

cost = tf.reduce_sum(tf.pow(y_pred - y_ph, 2)) / (2 * batch_size)

learning_rate = 0.05
optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(cost)

init = tf.global_variables_initializer()

loss_history = []

with tf.Session() as sess:
    sess.run(init)

    print("Починаємо навчання...")

    for epoch in range(epochs):
        indices = np.random.choice(num_samples, batch_size)
        x_batch = X_data[indices]
        y_batch = y_data[indices]

        _, current_cost, current_k, current_b = sess.run(
            [optimizer, cost, k, b],
            feed_dict={x_ph: x_batch, y_ph: y_batch}
        )

        loss_history.append(current_cost)

        if (epoch + 1) % 100 == 0:
            print(f"Епоха {epoch+1}: Cost={current_cost:.4f}, k={current_k[0]:.4f}, b={current_b[0]:.4f}")

    print("\nНавчання завершено!")
    final_k = sess.run(k)[0]
    final_b = sess.run(b)[0]
    print(f"Фінальні результати: k = {final_k:.4f}, b = {final_b:.4f}")
    print(f"Очикувані (істинні): k = 2.0000, b = 1.0000")

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

plt.scatter(X_data, y_data, color='blue', alpha=0.2, label='Вхідні дані (з шумом)')

y_line = final_k * X_data + final_b

```

		Чайковський А.В				Арк.
		Масєвський О				
Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Лр.1	2

```

plt.plot(X_data, y_line, color='red', linewidth=3, label=f'Лінія регресії:
y={final_k:.2f}x+{final_b:.2f}')

plt.title("TensorFlow: Лінійна регресія")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()
plt.grid(True)
plt.show()

```

Результати роботи У процесі навчання програма виводила поточні значення коефіцієнтів. Після 2000 епох модель зійшлася до значень, близьких до істинних

```

To enable the following instructions: SSE3 SSE4.1 SSE4.2 AVX AVX2 FMA, in other operations, rebuild Tensor
Flow with the appropriate compiler flags.
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
I0000 00:00:1766011802.693656    3948 mlir_graph_optimization_pass.cc:437] MLIR V1 optimization pass is no
t enabled
Починаємо навчання...
Епоха 100: Cost=1.1694, k=2.1012, b=0.9127
Епоха 200: Cost=1.0286, k=2.0285, b=0.8705
Епоха 300: Cost=0.7623, k=2.0159, b=0.9306
Епоха 400: Cost=0.9805, k=2.0358, b=0.9388
Епоха 500: Cost=1.1635, k=2.0181, b=0.9388
Епоха 600: Cost=1.0947, k=1.9977, b=0.9426
Епоха 700: Cost=1.0925, k=1.9833, b=0.9855
Епоха 800: Cost=0.8607, k=1.9524, b=0.9525
Епоха 900: Cost=0.9701, k=1.9589, b=0.9495
Епоха 1000: Cost=1.0165, k=1.9442, b=0.9665
Епоха 1100: Cost=0.7370, k=1.9478, b=0.9707
Епоха 1200: Cost=1.1037, k=1.9493, b=0.9905
Епоха 1300: Cost=0.8064, k=1.9421, b=0.9390
Епоха 1400: Cost=0.8691, k=1.9792, b=0.9621
Епоха 1500: Cost=1.1612, k=1.9824, b=0.9562
Епоха 1600: Cost=1.0116, k=1.9774, b=0.9653
Епоха 1700: Cost=0.8474, k=1.9964, b=0.9759
Епоха 1800: Cost=1.0056, k=1.9905, b=0.9827
Епоха 1900: Cost=0.6727, k=1.9823, b=0.9505
Епоха 2000: Cost=0.8695, k=1.9602, b=0.9406

Навчання завершено!
Фінальні результати: k = 1.9614, b = 0.9443

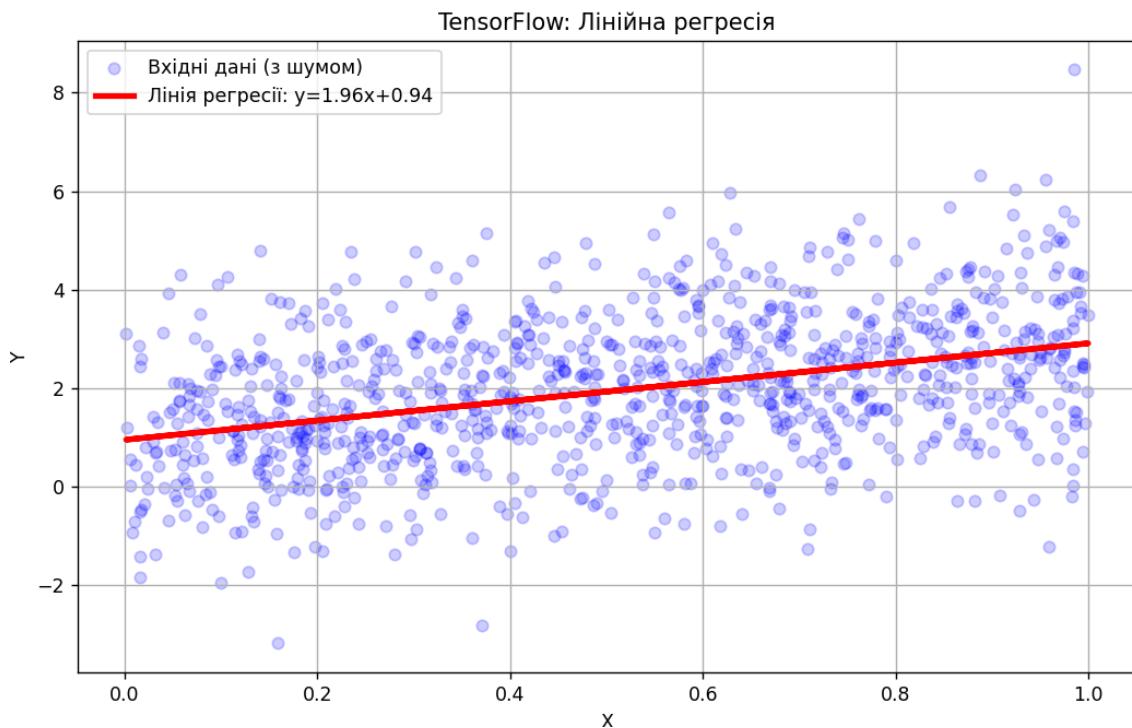
```

In 72 Cell 15 (2100 selected) Spaces: 4 HTF: 0 CPU: 0.1s Path:

		Чайковський А.В					Арк.
		Масевський О					
Змн.	Арк.	№ докум.	Підпис	Дата		ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Пр.1	3

Графічна інтерпретація:

На отриманому графіку синіми точками позначено вхідні дані, а червона лінія — це побудована моделлю лінійна регресія.



Як видно з графіка, червона лінія проходить через центр розподілу даних, що свідчить про коректну роботу алгоритму. Отримані значення  $k$  та  $b$  близькі до 2.0 та 1.0 відповідно.

Висновок: на даній лабораторній роботі я дослідив ресурс Keras і TensorFlow. Також застосував TensorFlow

GIT- [AntonChaikovskyi/AI-systems](#)

Чайковський А.В	Маєвський О	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.23.121.15.001 – Пр.1			Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	4