

## ЛАБОРАТОРНА РОБОТА № 8

**Тема:** «Ресурси Keras. TensorFlow. Навчання лінійної регресії».

**Мета:** Дослідження ресурсу Keras і TensorFlow. Застосування TensorFlow.

Без налаштування та встановлення так як використовується *Google Collab* – в якому влаштоване все для опрацювання завдання.

**Код:**

```
import tensorflow as tf
import numpy as np

# 1. Data generation
np.random.seed(42) # For reproducibility
x_data = np.random.uniform(0, 1, 1000).reshape(-1, 1)
y_data = 2 * x_data + 1 + np.random.normal(0, 2, size=(1000, 1))

# 2. Creating data as tensors
X = tf.convert_to_tensor(x_data, dtype=tf.float32)
y = tf.convert_to_tensor(y_data, dtype=tf.float32)

# 3. Initializing model variables
k = tf.Variable(tf.random.normal([1], stddev=0.1), name='k')
b = tf.Variable(tf.zeros([1]), name='b')

# 4. Building the linear regression model
def model(X):
    return k * X + b

# 5. Defining the loss function and optimizer
def compute_loss(y_true, y_pred):
    return tf.reduce_mean(tf.square(y_true - y_pred))

optimizer = tf.optimizers.SGD(learning_rate=0.01)

# 6. Training the model
num_epochs = 20000
batch_size = 100
num_batches = len(x_data) // batch_size

for epoch in range(num_epochs):
    for i in range(num_batches):
        # Selecting a random mini-batch
```

					ДУ «Житомирська політехніка».24.121.12.000 – Лр8			
Змн.	Арк.	№ докум.	Підпис	Дата				
Виконав		Левкович О.О.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Іванов Д.А.						1
Керівник								4
Н. контр.							ФІКТ Гр. ІПЗк-23-1	
Зав. каф.								

```

        batch_indices = np.random.choice(len(x_data), batch_size)
        x_batch = tf.convert_to_tensor(x_data[batch_indices],
dtype=tf.float32)
        y_batch = tf.convert_to_tensor(y_data[batch_indices],
dtype=tf.float32)

        # Optimization within GradientTape
        with tf.GradientTape() as tape:
            y_pred = model(x_batch)
            loss = compute_loss(y_batch, y_pred)

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

        # Printing results every 100 epochs
        if (epoch + 1) % 100 == 0:
            y_pred_full = model(X)
            full_loss = compute_loss(y, y_pred_full).numpy()
            print(f"Epoch {epoch + 1}: Loss={full_loss:.4f}, k={k.numpy()[0]:.4f},
b={b.numpy()[0]:.4f}")

print("\nTraining completed!")
print(f"Final parameters: k={k.numpy()[0]:.4f}, b={b.numpy()[0]:.4f}")

```

### Результат:

		Левкович О.О.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр8	Арк.
		Іванов Д.А.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

Epoch 14600: Loss=3.9000, k=1.7027, b=1.3344
Epoch 14700: Loss=3.9001, k=1.7081, b=1.3551
Epoch 14800: Loss=3.9016, k=1.6782, b=1.3138
Epoch 14900: Loss=3.9011, k=1.7188, b=1.3696
Epoch 15000: Loss=3.8999, k=1.6971, b=1.3524
Epoch 15100: Loss=3.9004, k=1.6999, b=1.3225
Epoch 15200: Loss=3.8999, k=1.7049, b=1.3388
Epoch 15300: Loss=3.9008, k=1.6960, b=1.3156
Epoch 15400: Loss=3.9003, k=1.6901, b=1.3291
Epoch 15500: Loss=3.9015, k=1.6825, b=1.3137
Epoch 15600: Loss=3.9005, k=1.6881, b=1.3253
Epoch 15700: Loss=3.9000, k=1.7301, b=1.3350
Epoch 15800: Loss=3.9029, k=1.7076, b=1.3957
Epoch 15900: Loss=3.9005, k=1.6952, b=1.3227
Epoch 16000: Loss=3.9002, k=1.6915, b=1.3309
Epoch 16100: Loss=3.9000, k=1.7010, b=1.3531
Epoch 16200: Loss=3.9000, k=1.6865, b=1.3404
Epoch 16300: Loss=3.8999, k=1.6999, b=1.3456
Epoch 16400: Loss=3.9000, k=1.6773, b=1.3455
Epoch 16500: Loss=3.8999, k=1.7001, b=1.3307
Epoch 16600: Loss=3.9003, k=1.6963, b=1.3251
Epoch 16700: Loss=3.9000, k=1.7196, b=1.3311
Epoch 16800: Loss=3.9005, k=1.6484, b=1.3489
Epoch 16900: Loss=3.9028, k=1.6814, b=1.4076
Epoch 17000: Loss=3.8999, k=1.6787, b=1.3473
Epoch 17100: Loss=3.9016, k=1.7289, b=1.3712
Epoch 17200: Loss=3.9005, k=1.6990, b=1.3196
Epoch 17300: Loss=3.8999, k=1.6858, b=1.3508
Epoch 17400: Loss=3.8999, k=1.7012, b=1.3479
Epoch 17500: Loss=3.9012, k=1.6933, b=1.3124
Epoch 17600: Loss=3.9000, k=1.7077, b=1.3534
Epoch 17700: Loss=3.8999, k=1.7068, b=1.3465
Epoch 17800: Loss=3.9002, k=1.7000, b=1.3264
Epoch 17900: Loss=3.9018, k=1.6723, b=1.3147
Epoch 18000: Loss=3.9006, k=1.6923, b=1.3763
Epoch 18100: Loss=3.9000, k=1.6875, b=1.3416
Epoch 18200: Loss=3.9003, k=1.6858, b=1.3313
Epoch 18300: Loss=3.9000, k=1.6816, b=1.3423
Epoch 18400: Loss=3.9000, k=1.6991, b=1.3368
Epoch 18500: Loss=3.9001, k=1.6822, b=1.3391
Epoch 18600: Loss=3.8999, k=1.6898, b=1.3467
Epoch 18700: Loss=3.9001, k=1.7155, b=1.3259
Epoch 18800: Loss=3.9001, k=1.7067, b=1.3562
Epoch 18900: Loss=3.9006, k=1.7137, b=1.3637
Epoch 19000: Loss=3.9004, k=1.7120, b=1.3177
Epoch 19100: Loss=3.8999, k=1.6882, b=1.3587
Epoch 19200: Loss=3.8999, k=1.6944, b=1.3512
Epoch 19300: Loss=3.9004, k=1.6645, b=1.3851
Epoch 19400: Loss=3.8999, k=1.6816, b=1.3564
Epoch 19500: Loss=3.9000, k=1.6951, b=1.3576
Epoch 19600: Loss=3.9001, k=1.6775, b=1.3423
Epoch 19700: Loss=3.8999, k=1.6875, b=1.3490
Epoch 19800: Loss=3.9001, k=1.7070, b=1.3549
Epoch 19900: Loss=3.8999, k=1.7017, b=1.3502
Epoch 20000: Loss=3.8999, k=1.6931, b=1.3522

Training completed!
Final parameters: k=1.6931, b=1.3522

```

Рис. 1. Результат ітерацій.

Висновок результату роботи коду:

Модель лінійної регресії навчилася на даних, і підсумкові параметри  $k$  та  $b$  після 20000 епох складають:

- $k = 1.6931$  ( $k = 2$ ),

		Левкович О.О.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр8	Арк.
		Іванов Д.А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

- $b = 1.3522b = 1.3522$  (очікувалося  $b = 1b = 1$ ).

Відхилення можуть значити таке -

1. Випадковий шум у даних: При генерації даних додавався шум, який ускладнив точне відновлення параметрів.

2. Темп навчання: Хоча процес мінімізації втрат стабілізувався,  $k_k$  і  $b_b$  не досягли ідеальних значень. Можливо, варто переглянути швидкість навчання чи кількість епох.

3. Початкові параметри: Випадкові початкові значення  $k_k$  і  $b_b$  могли вплинути на підсумкові результати.

Підсумок:

Модель успішно наблизилася до справжніх значень коефіцієнтів, мінімізуювши помилку, але через шум і гіперпараметри результат має невеликі відхилення. Для покращення точності можна:

- Збільшити кількість епох,
- Зменшити рівень шуму даних,
- Підібрати оптимальні гіперпараметри (швидкість навчання, початкові значення).

**Висновок:** Під час лабораторної роботи було досліджено ресурси Keras і TensorFlow, зокрема використано TensorFlow для реалізації та навчання моделі лінійної регресії.

**Github** - <https://github.com/TAMOTO24/-Intelligen-Systems>

		Левкович О.О.			ДУ «Житомирська політехніка». 24.121.12.000 – Лр8	Арк.
		Іванов Д.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4