

## Лабораторна робота 8

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

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

**Завдання 3.** Використовую данні з пункту 2 визначити відбудеться матч при наступних погодних умовах чи ні: Розрахунки провести з використанням Python.

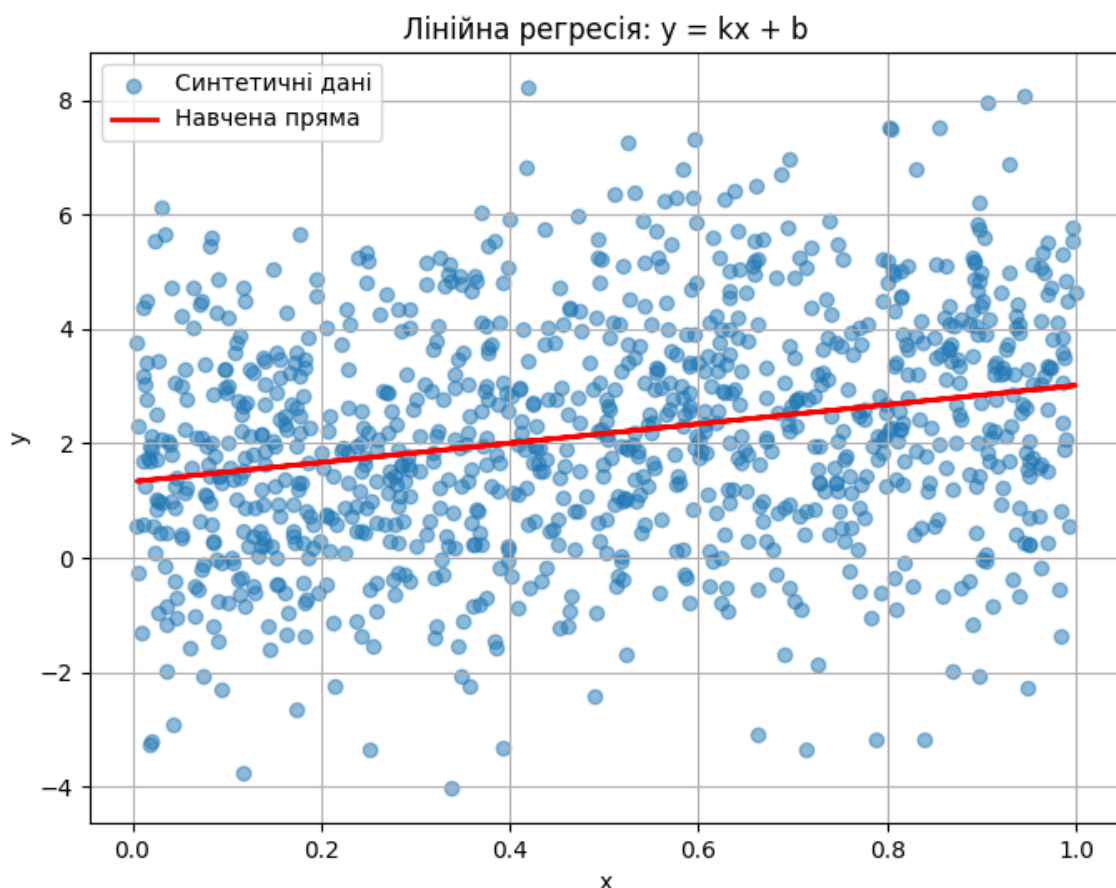


Рис. 1 Графік навчання

					ДУ «Житомирська політехніка».25.121.14.003 – Лр8		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Кольцова Н.О.			Звіт з лабораторної роботи		
Перевір.		Маєвський О.В.					
Керівник							
Н. контр.							
Зав. каф.							
					Літ.	Арк.	Аркушів
						1	9
					ФІКТ Гр. ІПЗ-22-4[1]		

```

Епоха 17400: Loss=4.4054, k=1.6940, b=1.3583
Епоха 17500: Loss=3.2737, k=1.7048, b=1.3639
Епоха 17600: Loss=4.0628, k=1.7145, b=1.3570
Епоха 17700: Loss=3.7047, k=1.7119, b=1.3500
Епоха 17800: Loss=4.3358, k=1.6813, b=1.3251
Епоха 17900: Loss=3.7349, k=1.6999, b=1.3578
Епоха 18000: Loss=3.3307, k=1.6888, b=1.3685
Епоха 18100: Loss=3.7795, k=1.6827, b=1.3604
Епоха 18200: Loss=4.6625, k=1.6810, b=1.3442
Епоха 18300: Loss=3.4166, k=1.6863, b=1.3634
Епоха 18400: Loss=4.2250, k=1.6837, b=1.3597
Епоха 18500: Loss=4.0226, k=1.6734, b=1.3398
Епоха 18600: Loss=4.5975, k=1.6826, b=1.3416
Епоха 18700: Loss=3.8861, k=1.6896, b=1.3899
Епоха 18800: Loss=4.8471, k=1.6929, b=1.3781
Епоха 18900: Loss=3.2578, k=1.6754, b=1.3383
Епоха 19000: Loss=4.3900, k=1.6763, b=1.3308
Епоха 19100: Loss=4.4325, k=1.6816, b=1.3710
Епоха 19200: Loss=3.8351, k=1.6758, b=1.3450
Епоха 19300: Loss=4.1536, k=1.6782, b=1.3574
Епоха 19400: Loss=3.0149, k=1.6820, b=1.3571
Епоха 19500: Loss=4.4957, k=1.7036, b=1.3355
Епоха 19600: Loss=3.6121, k=1.7122, b=1.3543
Епоха 19700: Loss=3.9356, k=1.6985, b=1.3432
Епоха 19800: Loss=3.0075, k=1.7051, b=1.3475
Епоха 19900: Loss=4.8575, k=1.6826, b=1.3130
Епоха 20000: Loss=4.5945, k=1.6876, b=1.3389

```

--- Фінальні метрики моделі ---

k (нахил) = 1.6876

b (зсув) = 1.3389

MSE (на всіх даних) = 3.9003

--- Приклади прогнозів моделі ---

x=0.375, y\_true=2.104, y\_pred=1.963

x=0.951, y\_true=0.231, y\_pred=2.932

x=0.732, y\_true=3.224, y\_pred=2.564

x=0.599, y\_true=3.418, y\_pred=2.340

x=0.156, y\_true=2.432, y\_pred=1.595

x=0.156, y\_true=3.474, y\_pred=1.595

x=0.058, y\_true=2.784, y\_pred=1.431

x=0.866, y\_true=3.651, y\_pred=2.790

x=0.601, y\_true=2.062, y\_pred=2.344

x=0.708, y\_true=-0.906, y\_pred=2.524

Рис. 2 Вивід фінальних метрик та прогнозів моделі

**Висновок:** Під час тренування моделі лінійної регресії спостерігалася поступова стабілізація параметрів (k) та (b) із збільшенням кількості епох. На початкових етапах навчання значення коефіцієнта нахилу та зсуву змінювалися значно, а функція втрат (MSE) коливалася, що свідчить про процес підлаштування моделі під

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Лр8	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

дані. До кінця навчання параметри стабілізувалися на рівнях ( $k = 1.6876$ ) та ( $b = 1.3389$ ), а середня квадратична помилка на всіх даних склала 3.9003, що вказує на достатню точність моделі для подальшого використання. Приклади прогнозів демонструють, що модель здатна відтворювати тенденції даних, хоча на окремих точках спостерігається відхилення від фактичних значень, що може бути наслідком шуму у вихідних даних або лінійного наближення складної залежності. Загалом, модель показує задовільну узгодженість із навчальними даними та готова до практичного застосування або подальшої оптимізації.

**Висновки:** в ході лабораторної роботи було досліджено ресурс Keras і TensorFlow. Застосування TensorFlow.

**Посилання на githab:** <https://github.com/KoltcovaNadiia/Artificial-intelligence-systems-2025>

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Лр8	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

## Лістинг програми:

LR\_8\_task\_1.py

```

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

tf.compat.v1.disable_eager_execution()

# --- Параметри ---
N_samples = 1000
batch_size = 100
num_steps = 20000
display_step = 100
learning_rate = 0.01

# --- Дані ---
np.random.seed(42)
X_data = np.random.uniform(0, 1, (N_samples, 1)).astype(np.float32)
y_data = (2 * X_data + 1 + np.random.normal(0, 2, (N_samples, 1))).astype(np.float32)

# --- Плейсхолдери ---
X = tf.compat.v1.placeholder(tf.float32, shape=(batch_size, 1))
y = tf.compat.v1.placeholder(tf.float32, shape=(batch_size, 1))

# --- Змінні та модель ---
with tf.compat.v1.variable_scope('linear-regression'):
    k = tf.Variable(tf.random.normal([1,1], stddev=0.01), name='slope')
    b = tf.Variable(tf.zeros([1,1]), name='bias')
    y_pred = tf.matmul(X, k) + b
    loss = tf.reduce_mean(tf.square(y - y_pred))
    optimizer =
tf.compat.v1.train.GradientDescentOptimizer(learning_rate).minimize(loss)

# --- Навчання ---
with tf.compat.v1.Session() as sess:
    sess.run(tf.compat.v1.global_variables_initializer())

    for step in range(1, num_steps+1):
        indices = np.random.choice(N_samples, batch_size)
        X_batch, y_batch = X_data[indices], y_data[indices]
        _, loss_val, k_val, b_val = sess.run([optimizer, loss, k, b],
                                             feed_dict={X: X_batch, y: y_batch})

        if step % display_step == 0:
            print(f"Епоха {step}: Loss={loss_val:.4f}, k={k_val[0][0]:.4f},
b={b_val[0][0]:.4f}")

# --- Фінальні метрики ---
y_pred_all = sess.run(tf.matmul(X_data, k) + b)
final_loss = np.mean((y_data - y_pred_all)**2)

```

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Лр8	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

```

print("\n--- Фінальні метрики моделі ---")
print(f"k (нахил) = {k_val[0][0]:.4f}")
print(f"b (зсув) = {b_val[0][0]:.4f}")
print(f"MSE (на всіх даних) = {final_loss:.4f}")

# --- Прогнози ---
print("\n--- Приклади прогнозів моделі ---")
for i in range(10): # показуємо перші 10 прогнозів
    print(f"x={X_data[i,0]:.3f}, y_true={y_data[i,0]:.3f},
y_pred={y_pred_all[i,0]:.3f}")

# --- Графік ---
plt.figure(figsize=(8,6))
plt.scatter(X_data, y_data, alpha=0.5, label='Синтетичні дані')
plt.plot(X_data, y_pred_all, color='red', linewidth=2, label='Навчена пряма')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Лінійна регресія: y = kx + b')
plt.legend()
plt.grid(True)
plt.show()

```

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Лр8	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5