**Лабораторна робота №8**

**ІПЗ-21-5 Пархомчук Іван**

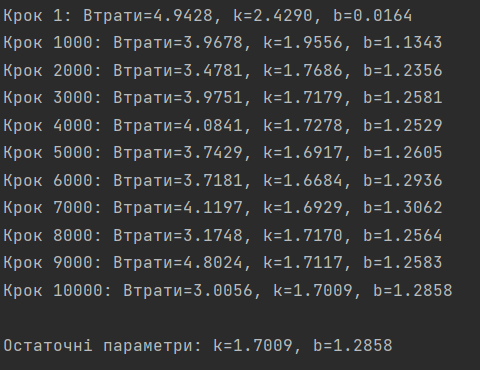
**Мета:** Дослідження ресурсу Keras і TensorFlow. Застосування TensorFlow.

**Завдання 1.** Використовуючи засоби TensorFlow, реалізувати код наведений нижче та дослідити структуру розрахункового алгоритму.

**Лістинг коду**

import tensorflow as tf  
import numpy as np  
  
# Параметри  
n\_samples = 1000 # Кількість зразків  
batch\_size = 100 # Розмір міні-батчу  
num\_steps = 10000 # Кількість кроків  
  
# Генерація синтетичних даних  
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)  
  
# Змінні моделі  
k = tf.Variable(tf.random.normal([1, 1], dtype=tf.float32), name="k")  
b = tf.Variable(tf.zeros([1], dtype=tf.float32), name="b")  
  
  
# Функція для обчислення втрат  
def compute\_loss(X, y):  
 y\_pred = tf.matmul(X, k) + b # Лінійна модель  
 return tf.reduce\_mean((y - y\_pred) \*\* 2)  
  
  
# Оптимізатор  
optimizer = tf.optimizers.SGD(learning\_rate=0.01)  
  
# Тренування моделі  
for step in range(1, num\_steps + 1):  
 # Вибір випадкового міні-батчу  
 indices = np.random.choice(n\_samples, batch\_size)  
 X\_batch, y\_batch = X\_data[indices], y\_data[indices]  
  
 # Виконання одного кроку оптимізації  
 with tf.GradientTape() as tape:  
 loss = compute\_loss(X\_batch, y\_batch)  
 gradients = tape.gradient(loss, [k, b])  
 optimizer.apply\_gradients(zip(gradients, [k, b]))  
  
 # Виведення прогресу кожні 1000 кроків  
 if step % 1000 == 0 or step == 1:  
 print(f"Крок {step}: Втрати={loss.numpy():.4f}, k={k.numpy()[0][0]:.4f}, b={b.numpy()[0]:.4f}")  
  
# Остаточні параметри моделі  
print(f"\nОстаточні параметри: k={k.numpy()[0][0]:.4f}, b={b.numpy()[0]:.4f}")

**Результат виконання**



**Висновок:** в ході виконання лабораторної робота дослідив ресурс Keras і TensorFlow.