**ЛАБОРАТОРНА РОБОТА № 8**

**Ресурси Keras. TensorFlow. Навчання лінійної регресії**

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

**Хід роботи:**

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

Лістинг коду:

import numpy as np  
import tensorflow.compat.v1 as tf  
  
tf.disable\_v2\_behavior()  
  
n\_samples = 1000  
batch\_size = 100  
num\_steps = 20000  
display\_step = 100  
learning\_rate = 0.001  
  
X\_data = np.random.uniform(0, 1, (n\_samples, 1))  
y\_data = 2 \* X\_data + 1 + np.random.normal(0, 2, (n\_samples, 1))  
X = tf.placeholder(tf.float32, shape=(batch\_size, 1), name="X")  
y = tf.placeholder(tf.float32, shape=(batch\_size, 1), name="y")  
  
def linear\_regression\_model():  
 with tf.variable\_scope("linear-regression"):  
 k = tf.Variable(tf.random\_normal((1, 1)), name="slope")  
 b = tf.Variable(tf.zeros((1,)), name="bias")  
 y\_pred = tf.matmul(X, k) + b  
 return y\_pred, k, b  
  
y\_pred, k, b = linear\_regression\_model()  
loss = tf.reduce\_sum((y - y\_pred) \*\* 2, name="loss")  
optimizer = tf.train.GradientDescentOptimizer(learning\_rate).minimize(loss)  
  
with tf.Session() as session:  
 session.run(tf.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 = session.run(  
 [optimizer, loss, k, b], feed\_dict={X: X\_batch, y: y\_batch}

)  
  
 if step % display\_step == 0:  
 print(  
 f"Epoch {step}: Loss={loss\_val:.8f}, k={k\_val[0][0]:.4f}, b={b\_val[0]:.4f}"  
 )

A screen shot of a black screen

Description automatically generated

Рис. 1. Результат виконання програми

Посилання на репозиторій на GitHub: <https://github.com/vladyslavgeyna/artificial-intelligence-systems/tree/main/lab8>.

***Висновки:*** в ході виконання лабораторної роботи мидослідили ресурси Keras і TensorFlow. Застосували TensorFlow.