**ЛАБОРАТОРНА РОБОТА № 8**

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

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

**Хід роботи**

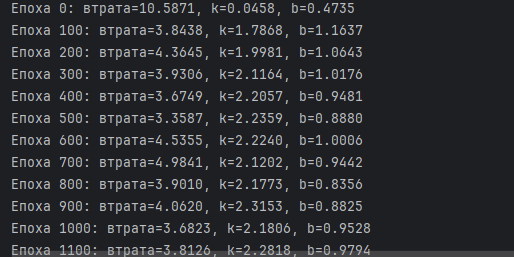
Завдання: Використовуючи засоби TensorFlow, реалізувати код наведений нижче та дослідити структуру розрахункового алгоритму. Для виконання розрахунків, можна використовувати онлайн – середовище google – colab (перехід за посиланням: <http://neuralnetworksanddeeplearning.com/chap4.html>)

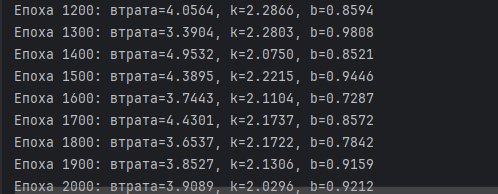
Лістинг програми:

import numpy as np  
import tensorflow.compat.v1 as tf  
  
tf.disable\_v2\_behavior() # Використовуємо TensorFlow 1.x у середовищі 2.x  
  
# Генеруємо дані для навчання  
x\_data = np.random.rand(1000).astype(np.float32) # 1000 випадкових точок  
y\_data = 2.0 \* x\_data + 1.0 + np.random.normal(0, 2, x\_data.shape) # y = 2x + 1 + шум  
  
# Оголошення placeholder для X та y  
X = tf.placeholder(tf.float32, shape=[None, 1], name="X") # Матриця розмірності (міні-батч × 1)  
y = tf.placeholder(tf.float32, shape=[None], name="y") # Вектор довжини розмір міні-батча  
  
# Ініціалізація параметрів моделі  
k = tf.Variable(tf.random.normal([1]), name="k") # k ініціалізується нормальним розподілом  
b = tf.Variable(tf.zeros([1]), name="b") # b початково нульове  
  
# Модель лінійної регресії  
y\_pred = tf.squeeze(tf.matmul(X, tf.reshape(k, [-1, 1])) + b) # Передбачення моделі  
  
# Функція втрат  
loss = tf.reduce\_mean(tf.square(y\_pred - y))  
  
# Оптимізатор  
optimizer = tf.train.GradientDescentOptimizer(learning\_rate=0.1)  
train\_op = optimizer.minimize(loss)  
  
# Ініціалізація змінних  
init = tf.global\_variables\_initializer()  
  
# Розмір міні-батчу  
batch\_size = 100

# Функція для вибору міні-батчу  
def get\_batch(x, y, batch\_size):  
 indices = np.random.choice(len(x), batch\_size)  
 return x[indices].reshape(-1, 1), y[indices]  
  
  
# Тренування моделі  
with tf.Session() as sess:  
 sess.run(init)  
  
 for epoch in range(2001):  
 x\_batch, y\_batch = get\_batch(x\_data, y\_data, batch\_size)  
 \_, loss\_value, k\_value, b\_value = sess.run(  
 [train\_op, loss, k, b],  
 feed\_dict={X: x\_batch, y: y\_batch}  
 )  
  
 if epoch % 100 == 0:  
 print(f"Епоха {epoch}: втрата={loss\_value:.4f}, k={k\_value[0]:.4f}, b={b\_value[0]:.4f}")

Виконання програми:





Посилання на ГітХаб: https://github.com/Kn211mna/AI-YT

**Висновок**: в ході виконання лабораторної роботи опрацював спеціалізовані бібліотеки та мову програмування Python дослідити методи неконтрольованої класифікації даних у машинному навчанні.