**Міністерство освіти та науки України**

**Державний вищий навчальний заклад “Київський національний економічний університет ім. Вадима Гетьмана”**

**Кафедра інформаційних систем в економіці**

**Дисципліна “** **Нейромережі та нейроконтроллери”**

**ЗВІТ**

**з лабораторної роботи №8(10)**

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

**Підготував:**

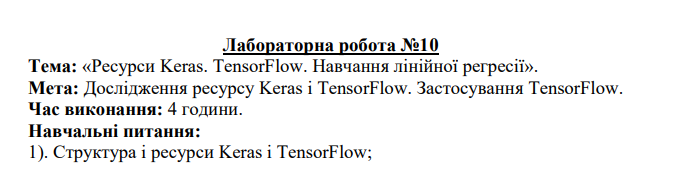
студент 5 курсу гр. ІШІ-501

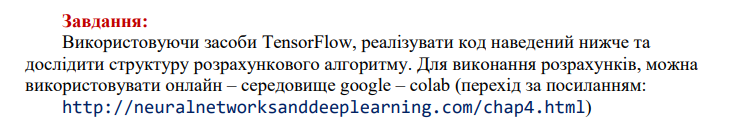
спеціальності «8і09»

Летич А.А.

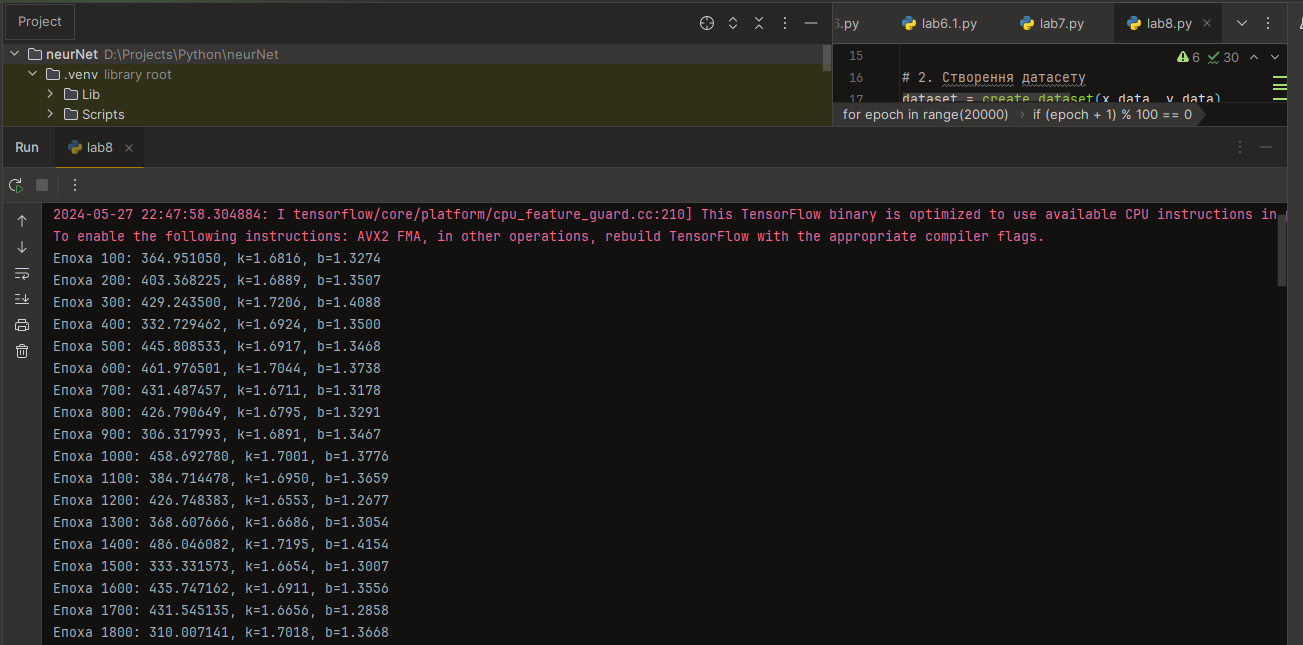
**Прийняв:**

Маєвський О.В.





Посилання на github: <https://github.com/SubAtom131/Neural-networks>

Результат роботи програми:   


Текст програми:

import tensorflow as tf  
import numpy as np  
  
# 1. Створення випадкових даних  
np.random.seed(42)  
x\_data = np.random.rand(1000, 1).astype(np.float32)  
y\_data = (2 \* x\_data + 1 + np.random.normal(0, 2, (1000, 1))).astype(np.float32)  
  
  
# Функція для створення датасету  
def create\_dataset(x\_data, y\_data):  
 dataset = tf.data.Dataset.from\_tensor\_slices((x\_data, y\_data))  
 return dataset.shuffle(buffer\_size=1000).batch(100).repeat()  
  
  
# 2. Створення датасету  
dataset = create\_dataset(x\_data, y\_data)  
iterator = iter(dataset)  
  
# 3. Ініціалізація змінних для моделі  
k = tf.Variable(tf.random.normal([1]), name='k', dtype=tf.float32)  
b = tf.Variable(tf.zeros([1]), name='b', dtype=tf.float32)  
  
  
# 4. Визначення функції моделі  
def model(X):  
 return k \* X + b  
  
  
# 5. Визначення функції втрат  
def loss\_fn(y\_true, y\_pred):  
 return tf.reduce\_sum(tf.square(y\_true - y\_pred))  
  
  
# 6. Визначення оптимізатора  
optimizer = tf.optimizers.SGD(learning\_rate=0.001)  
  
# 7. Виконання тренування моделі  
for epoch in range(20000):  
 for \_ in range(10): # Виконання 10 кроків навчання на кожній епосі  
 x\_batch, y\_batch = next(iterator)  
 with tf.GradientTape() as tape:  
 y\_pred = model(x\_batch)  
 loss = loss\_fn(y\_batch, y\_pred)  
 gradients = tape.gradient(loss, [k, b])  
 optimizer.apply\_gradients(zip(gradients, [k, b]))  
  
 if (epoch + 1) % 100 == 0:  
 print(f"Епоха {epoch + 1}: {loss.numpy():.6f}, k={k.numpy()[0]:.4f}, b={b.numpy()[0]:.4f}")