**Лабораторна робота №8**

Ресурси Keras. TensorFlow. Навчання лінійної регресії

**Виконав:** ІПЗ-21-3 Осипчук Антон Олексійович

**Github:** <https://github.com/AntonOsypchuk1/ai_lab/tree/main/lab8>

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

Лістинг програми:

import numpy as np

import tensorflow as tf

n\_samples, batch\_size, num\_steps = 1000, 100, 20000

X\_data = np.random.uniform(1, 10, (n\_samples, 1)).astype(np.float32)

y\_data = 2 \* X\_data + 1 + np.random.normal(0, 2, (n\_samples, 1)).astype(np.float32)

X\_data = (X\_data - np.mean(X\_data)) / np.std(X\_data)

y\_data = (y\_data - np.mean(y\_data)) / np.std(y\_data)

k = tf.Variable(tf.random.normal((1, 1), stddev=0.1), name='slope')

b = tf.Variable(tf.zeros((1,)), name='bias')

optimizer = tf.keras.optimizers.SGD(learning\_rate=0.001)

display\_step = 100

for i in range(num\_steps):

indices = np.random.choice(n\_samples, batch\_size)

X\_batch, y\_batch = X\_data[indices], y\_data[indices]

with tf.GradientTape() as tape:

y\_pred = tf.matmul(X\_batch, k) + b

loss = tf.reduce\_sum((y\_batch - y\_pred) \*\* 2)

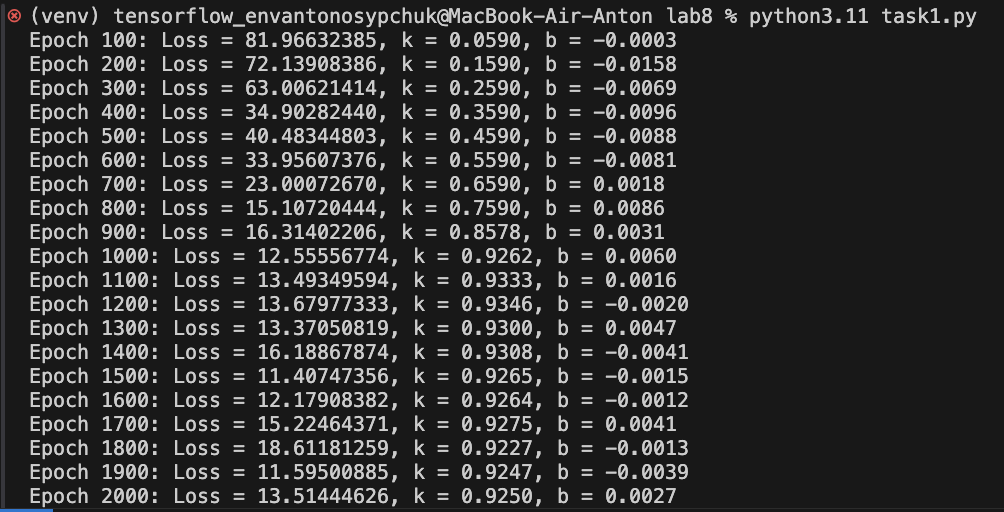
gradients = tape.gradient(loss, [k, b])

clipped\_gradients = [tf.clip\_by\_value(g, -1.0, 1.0) for g in gradients]

optimizer.apply\_gradients(zip(clipped\_gradients, [k, b]))

if (i + 1) % display\_step == 0:

print(f'Epoch {i + 1}: Loss = {loss.numpy():.8f}, k = {k.numpy()[0][0]:.4f}, b = {b.numpy()[0]:.4f}')

Результат виконання:  


Для кожної ітерації (epoch) обчислюється поточна помилка. На основі цієї помилки оновлюються параметри k та b для зменшення Loss, використовуючи градієнтний спуск.

На початку навчання значення Loss велике (81.96). З кожною ітерацією значення поступово зменшується, досягаючи близько 10-15, що свідчить про покращення моделі.

З кожною ітерацією модель наближається до мінімального значення помилки, а параметри k і b стають точнішими. Параметр k наближається до значення ~0.93-0.94. Параметр b коливається близько 0, що свідчить про те, що його вплив на результат мінімальний або модель знаходить рівновагу в даних.

В результаті виконання програми вдалося побудувати модель, яка поступово зменшує помилку (Loss) та уточнює значення параметрів k і b. Остаточна модель має значення k ~0.93 і b ~0, що відповідає оптимальному наближенню до навчальних даних. Хоча модель не досягла ідеального результату (Loss не став нульовим), вона демонструє стабільне навчання з поступовим покращенням.