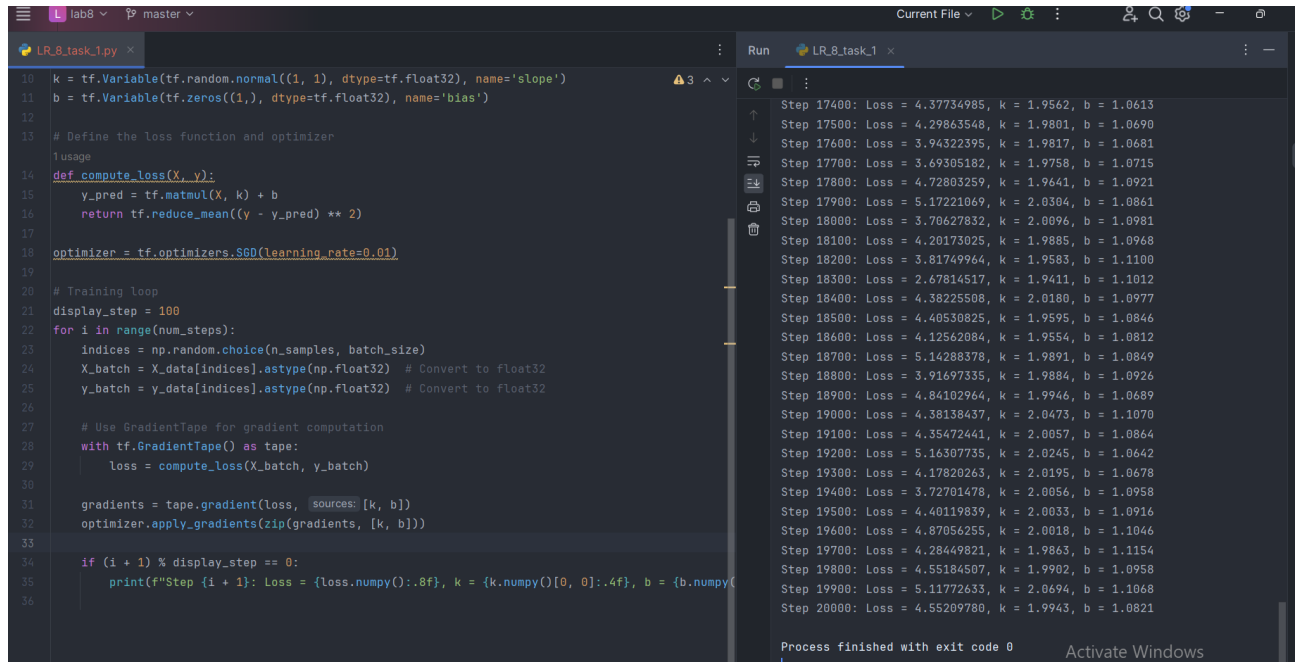


ЛАБОРАТОРНА РОБОТА № 8

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

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



```
10 k = tf.Variable(tf.random.normal((1, 1), dtype=tf.float32), name='slope')
11 b = tf.Variable(tf.zeros((1,)), dtype=tf.float32, name='bias')
12
13 # Define the loss function and optimizer
14 def compute_loss(X, y):
15     y_pred = tf.matmul(X, k) + b
16     return tf.reduce_mean((y - y_pred) ** 2)
17
18 optimizer = tf.optimizers.SGD(learning_rate=0.01)
19
20 # Training loop
21 display_step = 100
22 for i in range(num_steps):
23     indices = np.random.choice(n_samples, batch_size)
24     X_batch = X_data[indices].astype(np.float32) # Convert to float32
25     y_batch = y_data[indices].astype(np.float32) # Convert to float32
26
27     # Use GradientTape for gradient computation
28     with tf.GradientTape() as tape:
29         loss = compute_loss(X_batch, y_batch)
30
31     gradients = tape.gradient(loss, [k, b])
32     optimizer.apply_gradients(zip(gradients, [k, b]))
33
34     if (i + 1) % display_step == 0:
35         print(f"Step {i + 1}: Loss = {loss.numpy():.8f}, k = {k.numpy()[0, 0]:.4f}, b = {b.numpy():.4f}")
36
```

Run LR_8_task_1

Step 17400: Loss = 4.37734985, k = 1.9562, b = 1.0613
Step 17500: Loss = 4.29863548, k = 1.9801, b = 1.0690
Step 17600: Loss = 3.94322395, k = 1.9817, b = 1.0681
Step 17700: Loss = 3.69305182, k = 1.9758, b = 1.0715
Step 17800: Loss = 4.72803259, k = 1.9641, b = 1.0921
Step 17900: Loss = 5.17221069, k = 2.0304, b = 1.0861
Step 18000: Loss = 3.70627832, k = 2.0096, b = 1.0981
Step 18100: Loss = 4.20173025, k = 1.9885, b = 1.0968
Step 18200: Loss = 3.81749964, k = 1.9583, b = 1.1100
Step 18300: Loss = 2.67814517, k = 1.9411, b = 1.1012
Step 18400: Loss = 4.38225508, k = 2.0180, b = 1.0977
Step 18500: Loss = 4.40530825, k = 1.9595, b = 1.0846
Step 18600: Loss = 4.12562084, k = 1.9554, b = 1.0812
Step 18700: Loss = 5.14288378, k = 1.9891, b = 1.0849
Step 18800: Loss = 3.91697335, k = 1.9884, b = 1.0926
Step 18900: Loss = 4.84102964, k = 1.9946, b = 1.0689
Step 19000: Loss = 4.38138437, k = 2.0473, b = 1.1070
Step 19100: Loss = 4.35472441, k = 2.0057, b = 1.0864
Step 19200: Loss = 5.16307735, k = 2.0245, b = 1.0642
Step 19300: Loss = 4.17820263, k = 2.0195, b = 1.0678
Step 19400: Loss = 3.72701478, k = 2.0056, b = 1.0958
Step 19500: Loss = 4.40119839, k = 2.0033, b = 1.0916
Step 19600: Loss = 4.87056255, k = 2.0018, b = 1.1046
Step 19700: Loss = 4.28449821, k = 1.9863, b = 1.1154
Step 19800: Loss = 4.55184507, k = 1.9902, b = 1.0958
Step 19900: Loss = 5.11772633, k = 2.0694, b = 1.1068
Step 20000: Loss = 4.55209780, k = 1.9943, b = 1.0821

Process finished with exit code 0

Рис 1. Результат виконання програми

Висновок: На даній лабораторній роботі, було реалізовано модель лінійної регресії з використанням TensorFlow. Було навчено модель знаходити залежність між синтетично згенерованими даними, які представляють рівняння $y = 2x + 1 + \text{шум}$. У процесі навчання використовуються параметри k (нахил) і b (зсув), які поступово оновлюються методом стохастичного градієнтного спуску (SGD). Для обчислення помилок моделі використовується функція середньоквадратичної похибки (MSE). Код виконує 20,000 ітерацій навчання, на кожній із яких працює з випадковими міні-пакетами даних. Результати тренування відображаються через кожні 100 ітерацій, що дозволяє відстежувати зниження похибки та уточнення параметрів.

Github: [link](#)