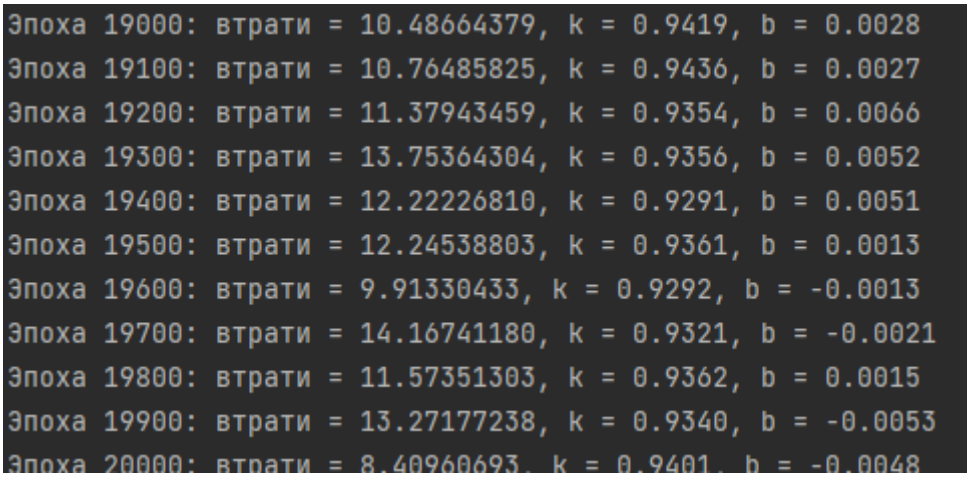
Системи штучного інтелекту.
Лабораторна робота 8.Федорович Дмитро ІПЗ-21-3
https://github.com/Dmitrij3/lab8AI

Завдання №1. Використовуючи засоби TensorFlow, реалізувати код наведений нижче та дослідити структуру розрахункового алгоритму.

```
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'Эпоха {i + 1}: втрати = {loss.numpy():.8f}, k =
{k.numpy()[0][0]:.4f}, b = {b.numpy()[0]:.4f}')
```

```
Эпоха 100: втрати = 80.09137726, k = 0.1110, b = 0.0039
Эпоха 200: втрати = 73.43798065, k = 0.2110, b = -0.0113
Эпоха 300: втрати = 48.29820633, k = 0.3110, b = -0.0078
Эпоха 400: втрати = 34.58003616, k = 0.4110, b = 0.0014
Эпоха 500: втрати = 29.54535675, k = 0.5110, b = -0.0149
Эпоха 600: втрати = 19.23437119, k = 0.6110, b = -0.0058
Эпоха 700: втрати = 20.65388107, k = 0.7110, b = -0.0008
Эпоха 800: втрати = 13.02258492, k = 0.8110, b = -0.0011
Эпоха 900: втрати = 12.46575642, k = 0.9050, b = 0.0003
Эпоха 1000: втрати = 10.77532196, k = 0.9266, b = 0.0007
```



Кожну епоху(ітерацію) модель знижує помилку до мінімального значення, через те що значення k та b стають точніше. У результаті ми отримали, що на початку втрати були ~80, а у кінці ~8.