

## Лабораторна робота №8

### Ресурси Keras. TensorFlow. Навчання лінійної регресії

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

```
import numpy as np
import tensorflow.compat.v1 as tf

tf.disable_v2_behavior()

# Parameters
n_samples, batch_size, num_steps = 1000, 100, 20000
learning_rate = 0.0001

# Generate synthetic data
X_data = np.random.uniform(0, 1, (n_samples, 1))
y_data = 2 * X_data + 1 + np.random.normal(0, np.sqrt(2), (n_samples, 1))

# Placeholders for input data
X = tf.placeholder(tf.float32, shape=(batch_size, 1), name='X')
y = tf.placeholder(tf.float32, shape=(batch_size, 1), name='y')

# Define the model
with tf.variable_scope('linear-regression'):
    k = tf.Variable(tf.random_normal([1, 1], mean=0.0, stddev=0.1), name='slope')
    b = tf.Variable(tf.zeros([1,]), name='bias')

# Predictions and loss
y_pred = tf.matmul(X, k) + b
loss = tf.reduce_sum((y - y_pred) ** 2)

# Optimizer
optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(loss)

# Training
display_step = 100
with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    for i in range(num_steps):
        # Sample a batch of data
        indices = np.random.choice(n_samples, batch_size)
        X_batch, y_batch = X_data[indices], y_data[indices]

        # Perform optimization and calculate loss
        _, loss_val, k_val, b_val = sess.run([optimizer, loss, k, b], feed_dict={X: X_batch, y: y_batch})

        # Display progress
        if (i + 1) % display_step == 0:
            print('Epoch %d: Loss=%.8f, k=%.4f, b=%.4f' % (i + 1, loss_val, k_val[0][0], b_val[0]))
```

Рис 1.1 – лістинг програми

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.8						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Прокопчук О.С			Звіт з лабораторної роботи №8			Літ.	Арк.	Аркуші	
Перевір.		Голенко М.Ю.								1	2
Реценз.								ФІКТ, гр. ІПЗ-21-1(2)			
Н. Контр.											
Зав.каф.											

```

Epoch 100: Loss=195.03099060, k=0.8700, b=1.3726
Epoch 200: Loss=185.10311890, k=1.0933, b=1.4151
Epoch 300: Loss=190.32817078, k=1.2257, b=1.3259
Epoch 400: Loss=182.87559509, k=1.3364, b=1.2654
Epoch 500: Loss=259.45812988, k=1.4466, b=1.2115
Epoch 600: Loss=210.76838684, k=1.5476, b=1.1652
Epoch 700: Loss=176.93515015, k=1.6426, b=1.1323
Epoch 800: Loss=176.99557495, k=1.7248, b=1.1012
Epoch 900: Loss=254.67417908, k=1.7852, b=1.0375
Epoch 1000: Loss=199.09140015, k=1.8432, b=1.0219
Epoch 1100: Loss=180.01373291, k=1.8861, b=1.0019
Epoch 1200: Loss=162.13252258, k=1.9323, b=0.9663
Epoch 1300: Loss=229.15866089, k=1.9607, b=0.9558
Epoch 1400: Loss=189.65658569, k=1.9921, b=0.9467
Epoch 1500: Loss=190.78868103, k=2.0139, b=0.9309
Epoch 1600: Loss=211.86308289, k=2.0605, b=0.9520
Epoch 1700: Loss=181.89836121, k=2.0711, b=0.9293
Epoch 1800: Loss=172.75035095, k=2.0852, b=0.9085
Epoch 1900: Loss=158.84463501, k=2.0989, b=0.8899
Epoch 2000: Loss=241.52168274, k=2.1054, b=0.8601
Epoch 2100: Loss=171.45214844, k=2.1370, b=0.8804
Epoch 2200: Loss=301.43460083, k=2.1549, b=0.8794
Epoch 2300: Loss=172.58883667, k=2.1653, b=0.8809
Epoch 2400: Loss=187.92082214, k=2.1556, b=0.8446
Epoch 2500: Loss=198.34300232, k=2.1719, b=0.8512
Epoch 2600: Loss=168.69697571, k=2.1681, b=0.8445
Epoch 2700: Loss=211.45712280, k=2.1920, b=0.8622
Epoch 2800: Loss=200.02955627, k=2.1882, b=0.8475

```

Рис 1.2 – результат виконання

Порівнюючи отримані результати з очікуваними, можна зробити висновок, що процес навчання моделі є нестабільним, оскільки значення помилки (loss) не зменшуються рівномірно, а параметри kkk та bbb коливаються. Це свідчить про потенційні проблеми з рівнем навчання, ініціалізацією ваг або масштабуванням даних, що потребують коригування для стабільного і ефективного навчання моделі.

**Висновок:** На лабораторній роботі мала змогу дослідженні ресурсу Keras і TensorFlow та застосування TensorFlow.

**Посилання на репозиторій:**

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.22.121.15.000 – Лр.7	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2