

Лабораторна робота №8

Ресурси Keras. TensorFlow. Навчання лінійної регресії

Мета: дослідження ресурсу Keras і TensorFlow. Застосування TensorFlow.

Хід роботи:

Завдання 8.1. Використовуючи засоби TensorFlow, реалізувати код наведений нижче та дослідити структуру розрахункового алгоритму. Для виконання розрахунків, можна використовувати онлайн – середовище google – colab.

Лістинг програми:

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

tf.disable_v2_behavior()
tf.disable_eager_execution()
n_samples, batch_size, num_steps = 1000, 100, 20000
learning_rate = 0.001

np.random.seed(9)
tf.set_random_seed(9)

X_data = np.random.uniform(0, 1, (n_samples, 1))
y_data = 2 * X_data + 1 + np.random.normal(0, 2, (n_samples, 1))

X = tf.placeholder(tf.float32, shape=(batch_size, 1), name="X")
y = tf.placeholder(tf.float32, shape=(batch_size, 1), name="y")

with tf.variable_scope("linear-regression"):
    k = tf.Variable(tf.random_normal((1, 1)), name="slope")
    b = tf.Variable(tf.zeros((1,)), name="bias")

y_pred = tf.matmul(X, k) + b
loss = tf.reduce_sum((y - y_pred) ** 2, name="loss")
optimizer = tf.train.GradientDescentOptimizer(learning_rate).minimize(loss)

display_step = 1000
with tf.Session() as session:
    session.run(tf.global_variables_initializer())
```

Змн.	Арк.	№ докум.	Підпис	Дата
Розроб.	Kozlik C.O.			
Перевір.	Маєвський О.В.			
Керівник				
Н. контр.				
Зав. каф.				

ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.13.000 – Пр8

Звіт з лабораторної
роботи №8

Літ.	Арк.	Аркушів
	1	2
ФІКТ, гр. ІПЗ-22-2		

Лістинг програми (продовження):

```
for i in range(1, num_steps + 1):
    indices = np.random.choice(n_samples, batch_size)
    X_batch, y_batch = X_data[indices], y_data[indices]

    _, loss_val, k_val, b_val = session.run([optimizer, loss, k, b],
                                             feed_dict={X: X_batch, y: y_batch})

    if i % display_step == 0:
        print(f"Epoch {i}: Loss={loss_val:.8f}, k={k_val[0][0]:.4f}, b={b_val[0]:.4f}")
```

Результат виконання програми:

```
Epoch 1000: Loss=340.63873291, k=1.9196, b=0.9549
Epoch 2000: Loss=394.48071289, k=1.9989, b=0.9047
Epoch 3000: Loss=358.77084351, k=1.9229, b=0.9977
Epoch 4000: Loss=375.30059814, k=1.8915, b=0.9546
Epoch 5000: Loss=358.75045776, k=1.9867, b=0.9544
Epoch 6000: Loss=537.73065186, k=1.9021, b=1.0305
Epoch 7000: Loss=414.13940430, k=1.9269, b=1.0279
Epoch 8000: Loss=486.94656372, k=1.8462, b=0.8894
Epoch 9000: Loss=411.78726196, k=2.0468, b=1.1080
Epoch 10000: Loss=396.04919434, k=1.8186, b=1.1310
Epoch 11000: Loss=384.62786865, k=1.9511, b=1.0102
Epoch 12000: Loss=536.64892578, k=1.9510, b=1.1747
Epoch 13000: Loss=577.27984619, k=1.8863, b=1.0199
Epoch 14000: Loss=483.66036987, k=1.9647, b=1.0843
Epoch 15000: Loss=454.81878662, k=1.7798, b=1.0553
Epoch 16000: Loss=395.34783936, k=1.8992, b=1.0797
Epoch 17000: Loss=504.53619385, k=1.8556, b=0.9893
Epoch 18000: Loss=368.58813477, k=1.9865, b=0.9980
Epoch 19000: Loss=425.27133179, k=1.9277, b=0.9792
Epoch 20000: Loss=327.81176758, k=1.9287, b=0.9594
```

Рис. 1. Результат виконання програми

У ході роботи була реалізована модель лінійної регресії засобами TensorFlow 1.x. Дані було згенеровано згідно рівняння $y = 2x + 1 + \text{шум}$. Після усіх ітерацій отримано значення, близькі до теоретичних: $k \approx 1.9287$, $b \approx 0.9594$. Також у процесі навчання моделі оптимізатор стохастичного градієнтного спуску поступово зменшував функцію втрат.

Висновок: в ході виконання лабораторної роботи ми дослідили ресурси Keras і TensorFlow; застосували TensorFlow.

Репозиторій: <https://github.com/SofiaKozlyk/AI-systems>

		<i>Козлик С.О.</i>			<i>ЖИТОМИРСЬКА ПОЛІТЕХНІКА. 25.121.13.000 – Пр8</i>	<i>Арк.</i>
		<i>Маєвський О.В.</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>2</i>