

Лабораторна робота №8

Ресурси Keras. TensorFlow. Навчання лінійної регресії

Mета: Дослідження ресурсу Keras і TensorFlow. Застосування TensorFlow.

Хід роботи

Репозиторій GITHUB: https://github.com/AnatoliiYarmolenko/SMI_1S4C

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

Об'єднавши весь код з лабораторної роботи було отримано наступну програму:

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

# --- 1. Створення даних ---
np.random.seed(42)
x_data = np.random.rand(1000, 1).astype(np.float32)
noise = np.random.normal(0, 2, size=x_data.shape)
y_data = 2 * x_data + 1 + noise

# --- 2. Створення простої моделі ---
model = tf.keras.Sequential([
    tf.keras.layers.Dense(1, input_shape=(1,))
])

# --- 3. Компіляція моделі ---
model.compile(optimizer=tf.keras.optimizers.SGD(Learning_rate=0.1),
              loss='mse')

# --- 4. Навчання ---
history = model.fit(x_data, y_data, epochs=2000, batch_size=100, verbose=0)
```

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.31.000 – Лр8
Розроб.	Ярмоленко А.М.				
Перевір.	Маєвський О.В.				
Реценз.					
Н. Контр.					
Зає.каф.	Вакалюк Т.А.				
<i>Звіт з лабораторної роботи №8</i>				Лім.	Арк.
				1	3
				<i>ФІКТ, гр. ІПЗ-22-1</i>	

```

# --- 5. Проміжні значення (кожні 100 епох) ---
for i in range(100, 2001, 100):
    loss_val = history.history['loss'][i-1]
    print(f"Епоха {i:4d}: втрата={loss_val:.4f}")

# --- 6. Результати ---
weights = model.get_weights()
k, b = weights[0][0][0], weights[1][0]
print(f"Фінальні параметри: k={k:.4f}, b={b:.4f}")

# --- 7. Графік втрат ---
plt.figure(figsize=(12,5))

plt.subplot(1,2,1)
plt.plot(history.history['loss'])
plt.title('Зміна втрат під час навчання')
plt.xlabel('Епоха')
plt.ylabel('Loss')

# --- 8. Графік регресії ---
plt.subplot(1,2,2)
plt.scatter(x_data, y_data, label='Дані')
plt.plot(x_data, model.predict(x_data), color='red', label='Модель')
plt.title('Результат навчання')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()

plt.tight_layout()
plt.show()

```

Епоха 1600: втрата=3.9068
Епоха 1700: втрата=3.9130
Епоха 1800: втрата=3.9145
Епоха 1900: втрата=3.9087
Епоха 2000: втрата=3.9151
Фінальні параметри: k=1.6955, b=1.3520

Рис. 1 Навчання моделі

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.2.000 – Лр.8	Арк.
						2

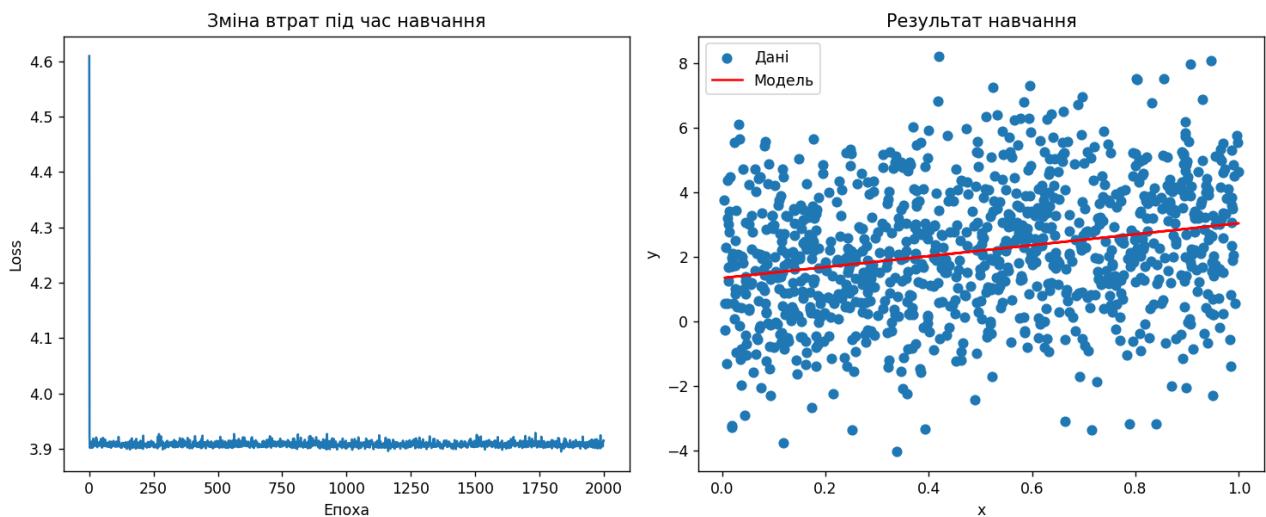


Рис. 2 Результат навчання моделі

У процесі навчання лінійної регресії спостерігалося поступове зменшення функції втрат, що свідчить про коректну роботу алгоритму оптимізації. Параметри моделі наблизилися до аналітичних значень ($k \approx 2$, $b \approx 1$), що відповідає заданій функції залежності $y = 2x + 1 + \epsilon$. Незначні коливання втрат пояснюються наявністю випадкового шуму у вихідних даних. Графік навчання показує стабільну збіжність моделі без перенавчання, а отримана лінія регресії добре описує загальну тенденцію даних.

Висновок: У ході лабораторної роботи було досліджено принципи створення та навчання простої нейронної мережі типу лінійної регресії засобами бібліотеки TensorFlow. Було ознайомлено з базовими поняттями моделей, змінних, оптимізаторів і функції втрат. Отримані результати підтвердили, що TensorFlow ефективно реалізує градієнтний спуск і дозволяє спостерігати процес поступового навчання параметрів. Таким чином, мета роботи — вивчення основ побудови та навчання моделей у TensorFlow — була досягнута повністю.

Змн.	Арк.	№ докум.	Підпис	Дата	ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.2.000 – Пр.8	Арк.
						3