

Tensorflow

TensorFlow là một trong những framework hàng đầu để triển khai các thuật toán học sâu, được sử dụng thường xuyên trong các dự án thực tế.

Biểu diễn dữ liệu trong NumPy và TensorFlow

Khi xây dựng mạng nơ-ron, việc hiểu cách **biểu diễn dữ liệu** trong NumPy và TensorFlow rất quan trọng để viết code đúng và tránh lỗi không cần thiết.

1. Ma trận (Matrix) và mảng (Array)

- Ma trận là một **mảng 2 chiều (2D array)** gồm các số.
- Kích thước của ma trận được viết theo **số hàng × số cột** (rows × columns).

Ví dụ – Ma trận 2×3:

```
x = np.array([[1, 2, 3],  
              [4, 5, 6]])
```

- Có 2 hàng, 3 cột → ma trận 2×3.

2. Vector hàng và vector cột

- **Vector hàng (row vector):** Ma trận có 1 hàng và nhiều cột → kích thước 1×n.
- **Vector cột (column vector):** Ma trận có nhiều hàng và 1 cột → kích thước n×1.

Ví dụ – Row vector (1×2):

```
x = np.array([[200, 17]])
```

3. Sự khác biệt về dấu ngoặc

- `np.array([200, 17])` : Mảng 1 chiều (1D array), không phải là ma trận.
- `np.array([[200, 17]])` : Ma trận 2D có kích thước 1×2.
- `np.array([[200], [17]])` : Ma trận 2D có kích thước 2×1.

⇒ **TensorFlow thường yêu cầu input là ma trận 2D**, vì vậy việc sử dụng **hai cặp dấu ngoặc vuông** `[...]` là quan trọng để đảm bảo đúng kích thước đầu vào.

4. Tensor là gì?

- `tf.Tensor` = cách TensorFlow biểu diễn dữ liệu (như ma trận).
- Có thể chuyển đổi giữa `Tensor` và `NumPy array` bằng `.numpy()`

Xây dựng Neural Network với Tensorflow

Trong Tensorflow, một neural network có thể được cài đặt dễ dàng sử dụng Sequential.

```
model = tf.keras.Sequential([
    tf.keras.layers.Dense(units=3, activation='sigmoid'),
    tf.keras.layers.Dense(units=1, activation='sigmoid')
])
```

Sau đó mô hình được huấn luyện rất đơn giản

```
model.compile(optimizer='adam', loss='binary_crossentropy')
model.fit(X, Y)
```

- `compile()` thiết lập cách huấn luyện mô hình.
- `fit(X, Y)` huấn luyện mô hình với dữ liệu đầu vào và nhãn tương ứng.

Cuối cùng mô hình có thể được sử dụng đơn giản như sau:

```
prediction = model.predict(X_new)
```