**Sử dụng model như là một lớp lớn**

**Sử dụng Mô hình như Lớp**

Trong Keras, chúng ta có khả năng sử dụng mô hình như một lớp, cụ thể, mô hình có thể được coi như một "lớp lớn." Điều này đúng cho cả lớp Sequential và lớp Model. Điều này có nghĩa rằng bạn có thể gọi một mô hình trên một tensor đầu vào và nhận một tensor đầu ra, giống như khi bạn gọi một lớp.

Ví dụ, chúng ta có thể gọi một mô hình trên một đầu vào và thu được một đầu ra:

Nếu mô hình có nhiều tensor đầu vào và nhiều tensor đầu ra, bạn cần gọi nó bằng một danh sách các tensors:



**Tái sử dụng Trọng số và Biểu diễn đã Học**

Một điểm quan trọng là khi bạn gọi một thể hiện của mô hình, bạn đang tái sử dụng các trọng số của mô hình, tương tự như khi bạn gọi một lớp. Điều này có nghĩa rằng trọng số đã học từ mô hình trước đó sẽ được sử dụng lại, và điều này hữu ích khi bạn muốn kết hợp các mô hình đã đào tạo vào mô hình chính của mình.

**Ví dụ Cụ thể**

Một ví dụ cụ thể về cách chúng ta có thể sử dụng mô hình như một lớp là khi xây dựng một mô hình thị giác cho hệ thống xem xuyên độ sâu. Chẳng hạn, chúng ta có hai máy ảnh song song, cách nhau một vài centimet (một inch). Chúng ta muốn tạo một mô hình để phân biệt chiều sâu giữa các vật thể trong hình ảnh, có thể hữu ích trong nhiều ứng dụng.

Thay vì xây dựng hai mô hình độc lập để trích xuất đặc trưng hình ảnh từ máy ảnh trái và máy ảnh phải, chúng ta có thể tái sử dụng một mô hình thị giác chung và sử dụng nó cho cả hai đầu vào. Kết quả là chúng ta chia sẻ cùng một trọng số và biểu diễn, giúp tiết kiệm thời gian và tài nguyên trong việc xử lý hình ảnh và xây dựng mô hình.

Mô hình thị giác Siamese:

A screen shot of a computer program

Description automatically generated

\* Giải thích:

Trong đoạn mã này, một mô hình thị giác Siamese được triển khai bằng cách sử dụng mạng Xception như một phần cơ sở. Hai hình ảnh RGB kích thước 250 × 250 pixel được đưa vào mô hình thông qua các đầu vào left\_input và right\_input. Mô hình Xception được gọi cùng một lần cho cả hai đầu vào để trích xuất đặc trưng từ hình ảnh. Cuối cùng, các đặc trưng trích xuất từ cả hai hình ảnh được hợp nhất vào merged\_features thông qua lớp concatenate, giúp tạo ra một biểu diễn kết hợp của cả hai hình ảnh.

1. Import các modules:

from keras import layers: Import lớp layers từ thư viện Keras để sử dụng trong việc xây dựng mô hình.

from keras import applications: Import lớp applications từ Keras để sử dụng mạng Xception.

from keras import Input: Import lớp Input từ Keras để định nghĩa đầu vào của mô hình.

2. Định nghĩa mạng cơ sở xử lý hình ảnh:

xception\_base = applications.Xception(weights=None, include\_top=False): Tạo một thể hiện của mạng Xception. Tham số weights=None chỉ định rằng không sử dụng trọng số được đã được đào tạo trước. Tham số include\_top=False chỉ định rằng chỉ cần lấy phần tích chập của mạng và bỏ qua phần fully connected (điểm cuối) của mạng.

3. Định nghĩa các đầu vào:

left\_input = Input(shape=(250, 250, 3)) và right\_input = Input(shape=(250, 250, 3)): Định nghĩa hai đầu vào của mô hình, mỗi đầu vào là một hình ảnh có kích thước 250x250 pixel và 3 kênh (RGB). Một đầu vào đại diện cho máy ảnh bên trái và một đầu vào đại diện cho máy ảnh bên phải.

4. Gọi mô hình thị giác hai lần:

left\_features = xception\_base(left\_input): Gọi mô hình Xception (mạng cơ sở) trên đầu vào của máy ảnh bên trái. Điều này sẽ trả về biểu diễn được học từ hình ảnh bên trái.

right\_features = xception\_base(right\_input): Gọi mô hình Xception trên đầu vào của máy ảnh bên phải. Điều này sẽ trả về biểu diễn được học từ hình ảnh bên phải.

5. Kết hợp các biểu diễn:

merged\_features = layers.concatenate([left\_features, right\_features], axis=-1): Sử dụng lớp concatenate để kết hợp thông tin biểu diễn từ hai máy ảnh, cả bên trái và bên phải, thông qua trục cuối cùng (-1). Điều này tạo ra một biểu diễn kết hợp của cả hai hình ảnh, cho phép mô hình phân tích sự sâu và chiều sâu giữa chúng.

Kết quả của mã này là một mô hình thị giác Siamese, trong đó thông tin từ cả hai máy ảnh được kết hợp để giúp mô hình hiểu về chiều sâu và độ sâu giữa các đối tượng trong hình ảnh, có thể hữu ích trong nhiều ứng dụng.

**Kết luận**

Trong tổng hợp, việc sử dụng mô hình như một "lớp lớn" trong Keras là một công cụ mạnh mẽ cho việc xây dựng các mô hình phức tạp và kết hợp các mô hình đã đào tạo trước vào các dự án của bạn. Điều này giúp bạn tiết kiệm thời gian và nỗ lực trong việc phát triển các ứng dụng sâu hơn và thông minh hơn.