

# Học truyền

*Transfer Learning*

**TS. Nguyễn Thị Kim Ngân**



# Nội dung

---

- Nguyên nhân mô hình dự báo kém
- Mạng học truyền



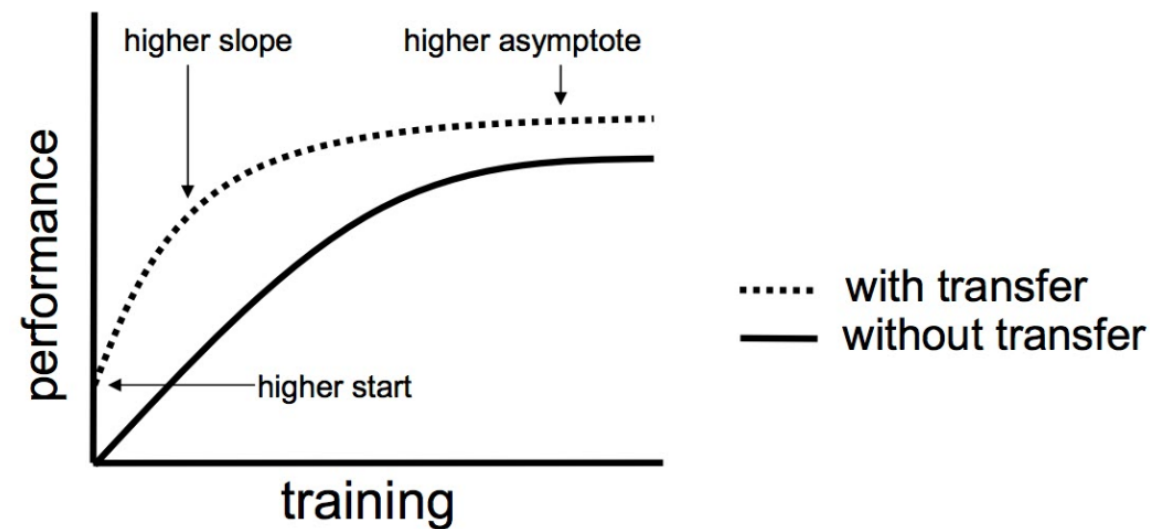
# Nguyên nhân mô hình dự báo kém

---

- Dữ liệu nhỏ không đại diện
- Mô hình mất cân bằng dữ liệu
- Kiến trúc mô hình quá phức tạp
- Quá trình tối ưu hóa gặp khó khăn

# Vai trò của Transfer Learning

- Chuyển giao tri thức: Sử dụng các pretrained-model có chất lượng tốt và độ chính xác cao. => Tận dụng lại các tri thức đã huấn luyện trước đó và cải thiện lại trên nhiệm vụ mới
- Cải thiện accuracy và tiết kiệm chi phí: Nếu sử dụng các pretrained-model thì số epochs huấn luyện mô hình mới có thể ít hơn. Độ chính xác có thể cao hơn so với không dùng Transfer Learning
- Hiệu quả với dữ liệu nhỏ: Tận dụng lại tri thức từ các pretrained-model với cùng tác vụ sẽ giúp các mô hình được huấn luyện dự báo tốt hơn với dữ liệu mới vì mô hình được học trên cả 2 nguồn tri thức đó là dữ liệu huấn luyện và dữ liệu mà nó đã được học trước đó



Hình 1: Sơ đồ so sánh hiệu suất mô hình trước và sau khi áp dụng transfer learning. (Nguồn: Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques).



# Transfer Learning

---

- Quá trình áp dụng tri thức đã được học từ một mô hình trước (pretrained-model) sang bài toán hiện tại được gọi là transfer learning
- Transfer learning sẽ tận dụng lại các đặc trưng được học từ những pretrained-model



# Ví dụ về Transfer Learning

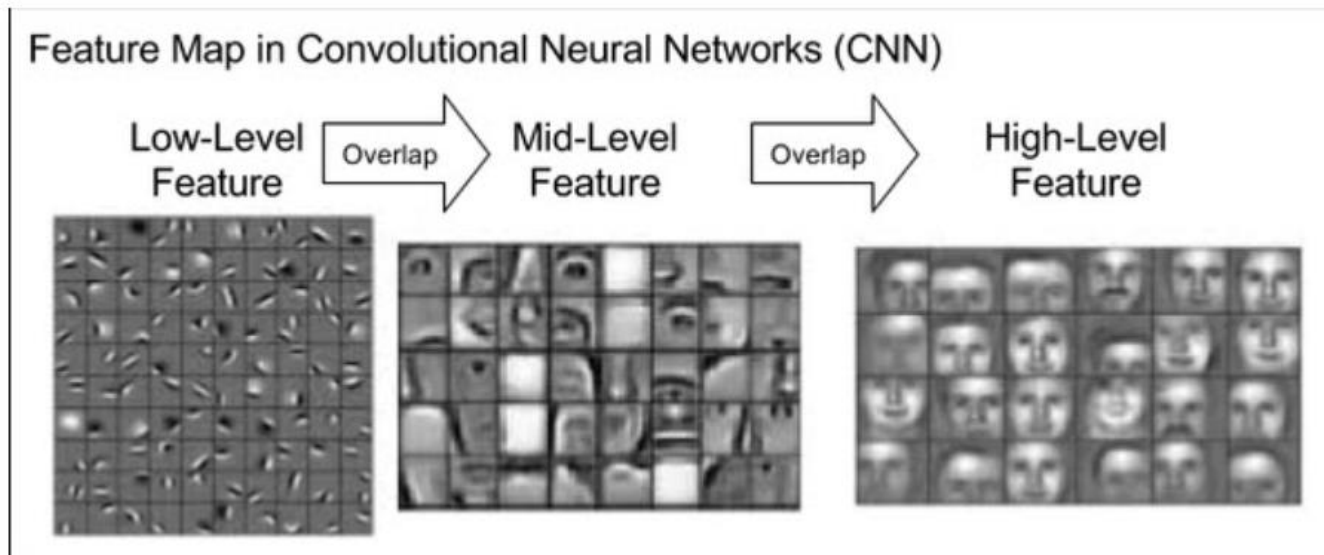
---

Trong bài toán dự báo dog and cat. Chúng ta có 2 nhãn cần phân loại là dog, cat và cả 2 nhãn này đều xuất hiện trong một bộ dữ liệu imagenet

=> Chúng ta kì vọng rằng có thể tận dụng lại các weights từ pretrained-model trên bộ dữ liệu imagenet để huấn luyện lại bài toán nhanh hơn, chuẩn xác hơn

# Kiến trúc mô hình sử dụng Transfer Learning

Các layers CNN về bản chất là một feature extractor mà mỗi một layer CNN sẽ có tác dụng trích lọc đặc trưng theo những level khác nhau

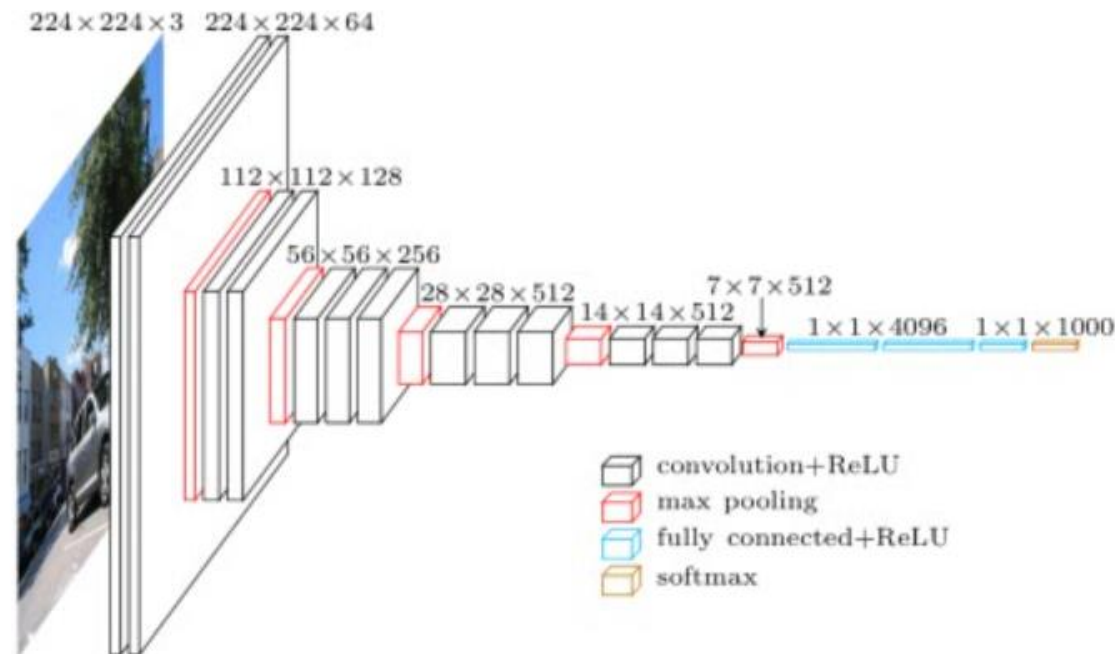


**Hình 2:** Các đặc trưng học được từ mạng CNN. Ở những Convolutional Layers đầu tiên, các bộ lọc phát hiện được các chi tiết chung dưới dạng các nét ngang, dọc và các cạnh của ảnh. Đây là những đặc trưng bậc thấp (low level feature) và khá chung chung. Chúng ta chưa thể nhận biết được vật thể dựa trên những đường nét này. Ở những Convolutional Layers cuối cùng là những đặc trưng bậc cao (high level feature) được tổng hợp từ đặc trưng bậc thấp. Đây là những đặc trưng tốt và có sức mạnh phân loại các classes.

# Kiến trúc mô hình sử dụng Transfer Learning

Transfer learning gồm các phrase:

- **Phrase 1:** Là một mạng Base Network được trích xuất từ một phần của pretrained-model sau khi loại bỏ các top fully connected layers

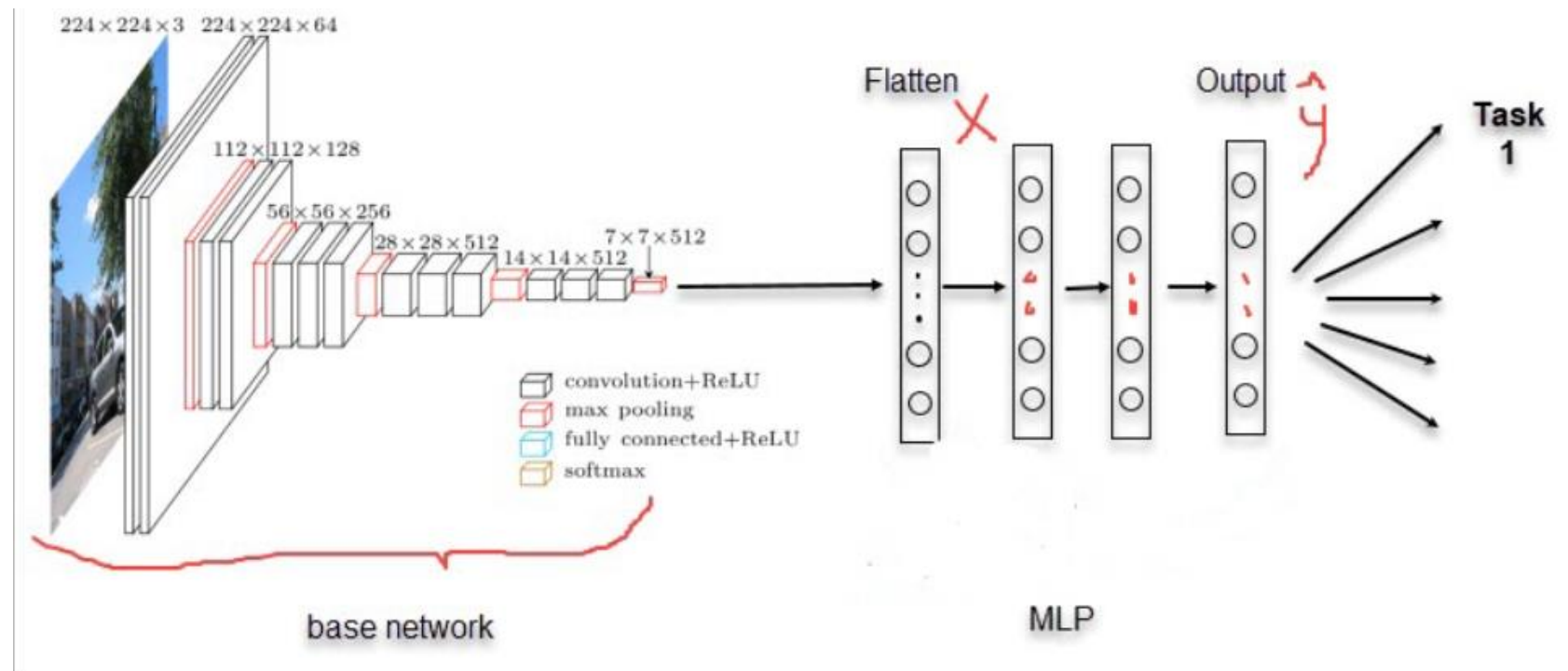


Hình 3: Kiến trúc của mạng VGG16 được sử dụng làm base network trong transfer learning.



# Kiến trúc mô hình sử dụng Transfer Learning

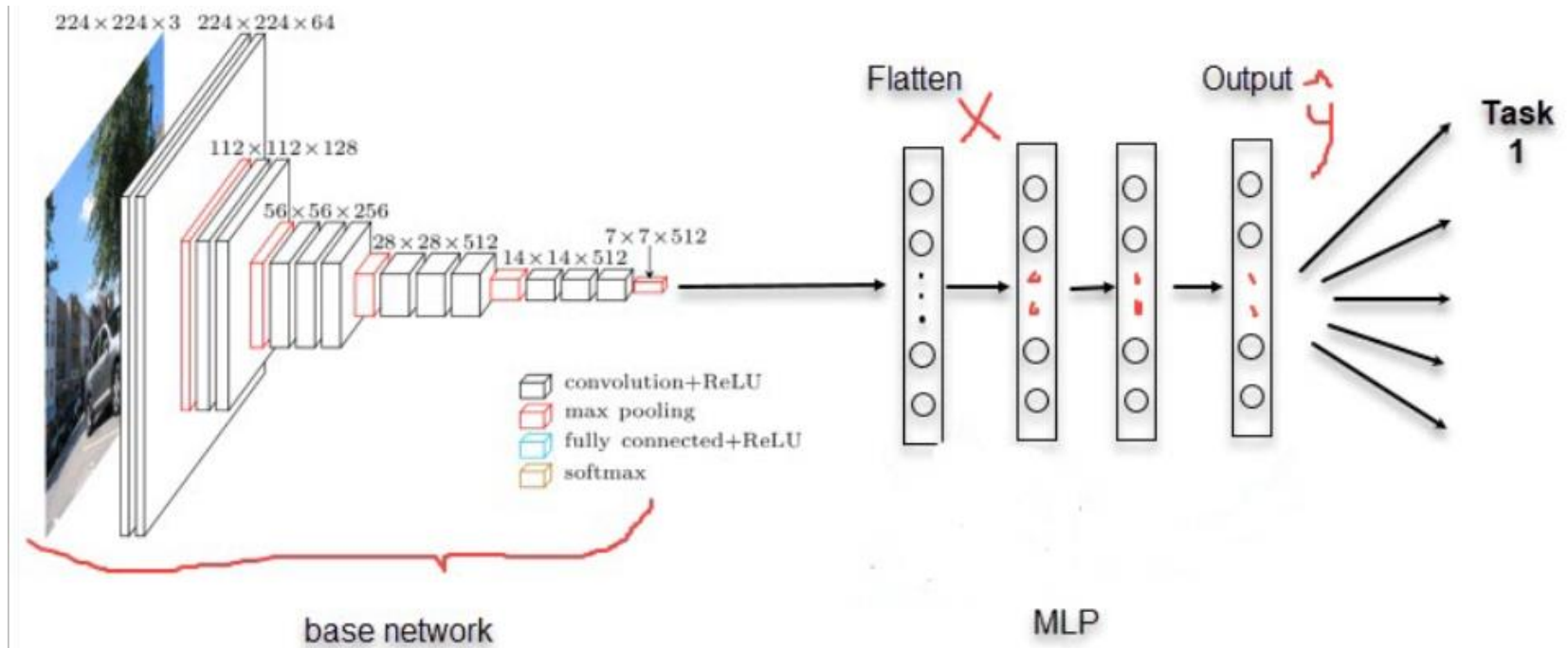
- **Phrase 2:** Là các Fully Connected Layers giúp giảm chiều dữ liệu và tính toán phân phối xác suất ở output. Bản chất Fully Connected Layers chính là một mạng Multiple Layer Perceptron (MLP)



Hình 4: Kiến trúc base network kết hợp với fully connected layers.

# Kiến trúc mô hình sử dụng Transfer Learning

- Khởi tạo tham số:
  - Base network: Mô hình sử dụng lại các weight của base\_network
  - Mạng MLP: Hệ số  $W$  và  $b$  của MLP được khởi tạo ngẫu nhiên
- Dữ liệu sau khi đi qua base\_network sẽ tạo ra những đặc trưng tốt, những đặc trưng này chính là input của mạng MLP



Hình 4: Kiến trúc base network kết hợp với fully connected layers.



## Ví dụ

```
1 from tensorflow.keras.models import load_model, Sequential
2 from tensorflow.keras.layers import Dense, Flatten
3 from tensorflow.keras.applications import MobileNet
4 from tensorflow.keras.optimizers import Adam
5
6
7 base_network = MobileNet(input_shape=(224, 224, 3), include_top = False, weights = 'i
8 flat = Flatten()
9 den = Dense(1, activation='sigmoid')
10
11 model = Sequential([base_network,
12                     flat,
13                     den])
14 model.compile(optimizer=Adam(), loss='binary_crossentropy', metrics = ['accuracy'])
15 model.summary()
```