



VietAI

# Convolutional Neural Networks (cont.)

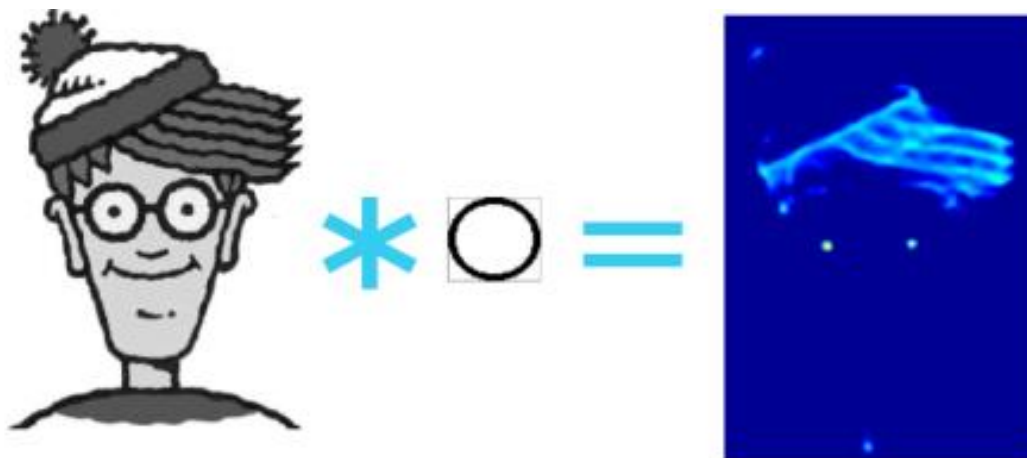


VietAI Teaching Team

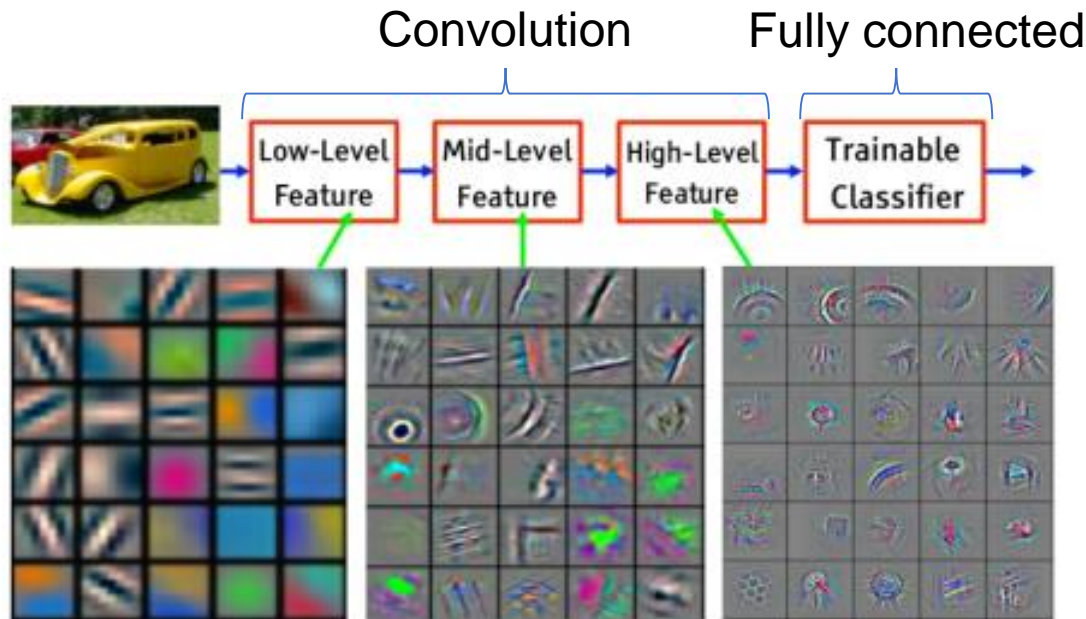
# Nội dung

1. Nhắc lại CNN
2. Một số kiến trúc CNN và đặc điểm
  - a) Sự phát triển của CNN
  - b) LeNet-5
  - c) AlexNet
  - d) ZFNet
  - e) VGGNet
  - f) GoogLeNet
  - g) ResNet
  - h) DenseNet
3. Transfer Learning

# 1 Nhắc lại CNN

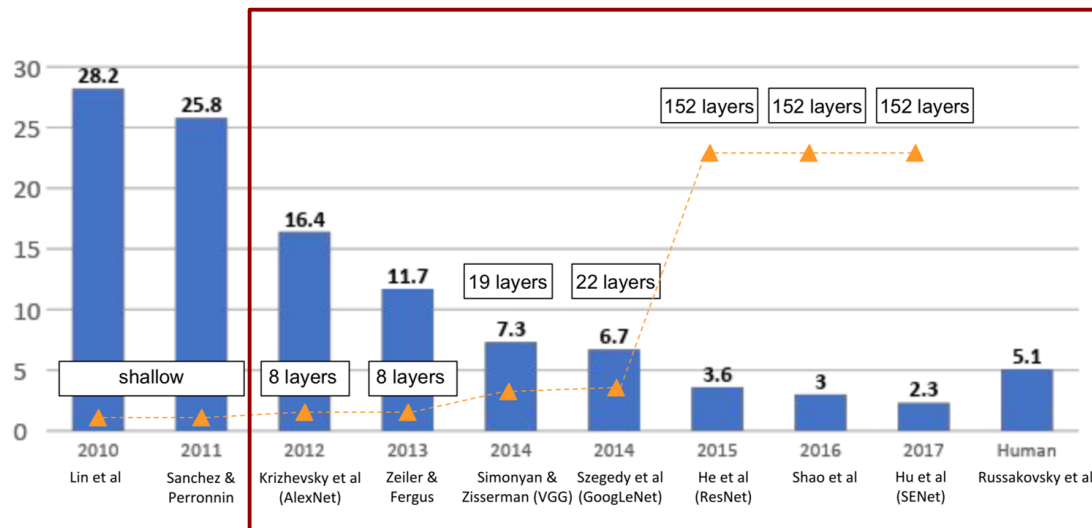


# 1 Nhắc lại CNN



## Learning Hierarchical Representations

## 2 Sự phát triển của CNN



CNN milestones

LeNet (1993)

Deep CNN

IMAGENET

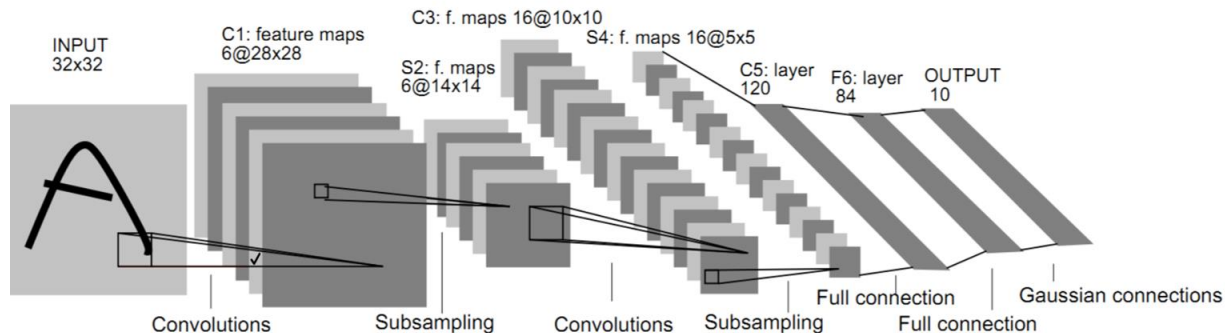
ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- > 14M ảnh
- > 20K class

Tổ chức cuộc thi thường niên từ 2010  
+ Classification  
+ Detection

Đặc trưng nổi bật so với các mạng neural network tại thời điểm bấy giờ

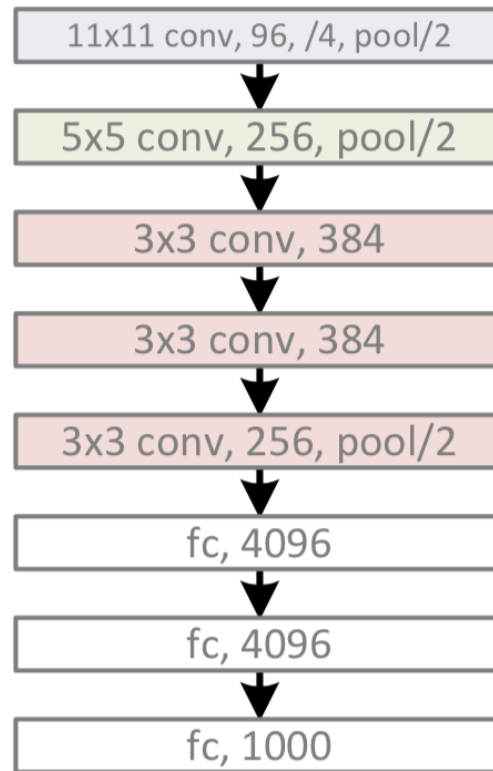
- Convolutions
- Subsampling (pooling)
- Fully-connected outputs



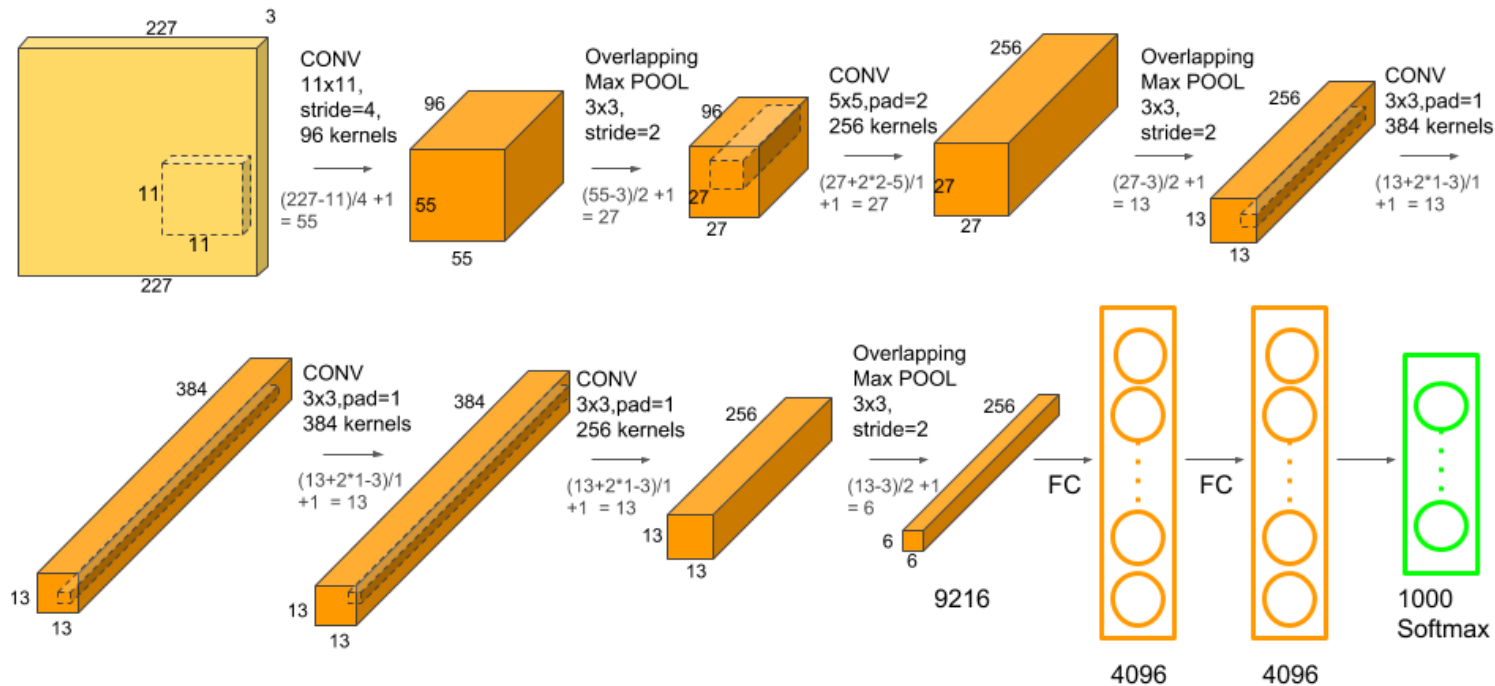
## 2 AlexNet – ILSVRC 2012 Winner

Đặc trưng:

- ReLU activation
- Dropout



## 2 AlexNet – ILSVRC 2012 Winner

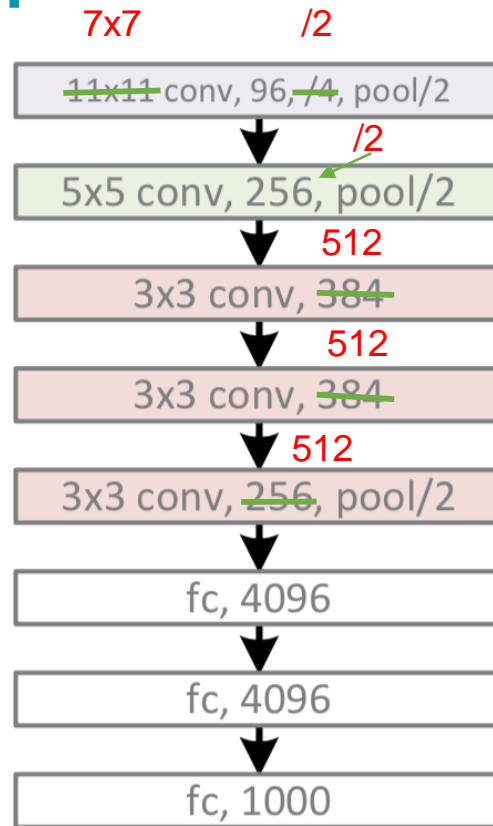




## 2 ZFNet – ILSVRC 2013 Winner

Đặc trưng:

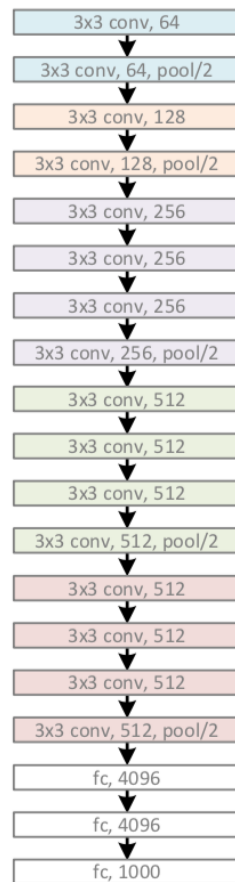
- Dựa trên AlexNet
- Điều chỉnh hyperparameter hiệu quả thông qua visualizing features
- Sử dụng filters cỡ nhỏ



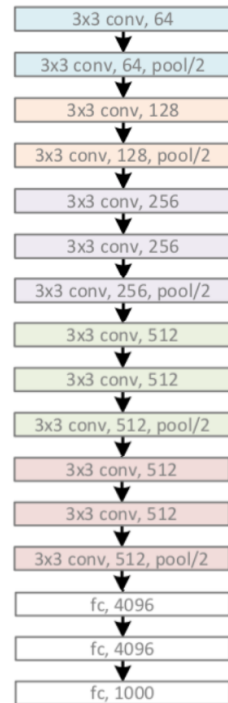
## 2 VGGNet – ILSVRC 2014 1st Runner Up

Đặc trưng:

- Sử dụng conv 3x3
  - receptive field hiệu quả
  - số lượng trọng số ít hơn (3 filter 3x3 với 1 filter 7x7)
- Thiết kế đơn giản - Modularized design
- Dễ dàng “lắp” thêm module:  
VGG11 -> VGG13 -> VGG16 -> VGG19
- Số lượng parameters lớn: VGG16 - 138M



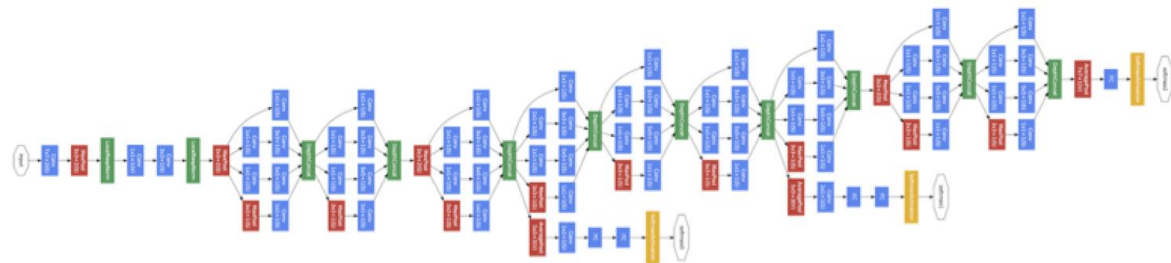
**VGG19**



**VGG16**

## 2 GoogLeNet - ILSVRC 2014 Winner

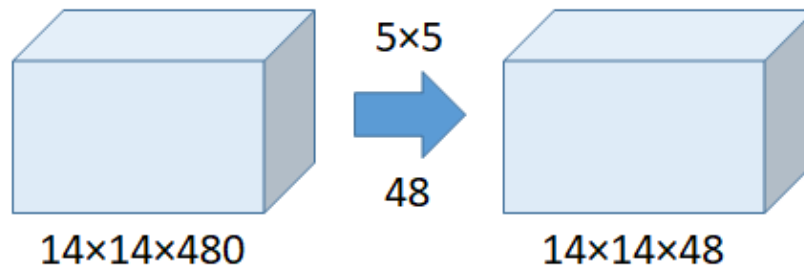
- Bottleneck layer
  - Giúp giảm số chiều của feature maps
- Inception Module - Multiple branches
  - Lấy cảm hứng từ Network in Network
- Shortcut 1x1
- Batch Normalization
- Global Pooling



Convolution  
Pooling  
Softmax  
Other

## 2 GoogLeNet - ILSVRC 2014 Winner

- **Bottleneck layer**
  - Giúp giảm số chiều của feature maps
- Inception Module - Multiple branches
  - Lấy cảm hứng từ Network in Network
- Shortcut 1x1
- Batch Normalization
- Global Pooling



**Không sử dụng bottleneck**

$$[5 \times 5 \text{ conv}, 48] \ 14 \times 14 \times 480 \times 5 \times 5 \times 48 = 112,896,000p$$

## 2 GoogLeNet - ILSVRC 2014 Winner

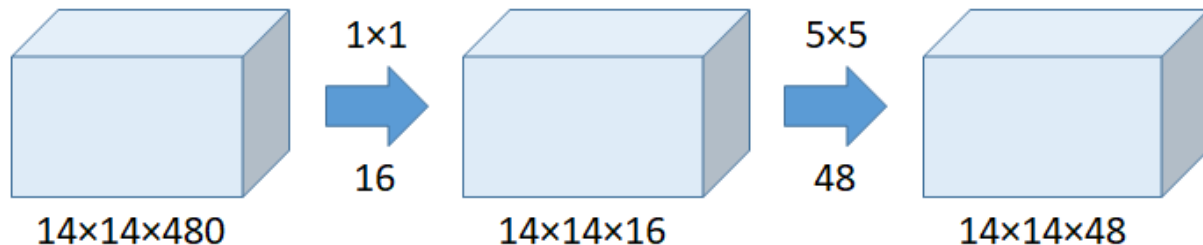
- **Bottleneck layer**
  - Giúp giảm số chiều của feature maps
- Inception Module - Multiple branches
  - Lấy cảm hứng từ Network in Network
- Shortcut 1x1
- Batch Normalization
- Global Pooling

### Sử dụng bottleneck

[1x1 conv, 16]  $14 \times 14 \times 480 \times 1 \times 1 \times 16 = 1,505,280p$

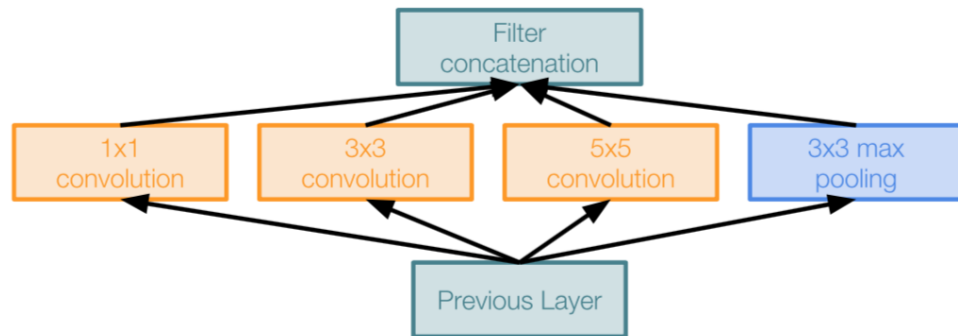
[5x5 conv, 48]  $14 \times 15 \times 16 \times 5 \times 5 \times 48 = 4,032,000p$

**Tổng:**  $5,537,280p \ll 112,896,000p$



## 2 GoogLeNet - ILSVRC 2014 Winner

- Bottleneck layer
  - Giúp giảm số chiều của feature maps
- **Inception Module - Multiple branches**
  - Lấy cảm hứng từ Network in Network
- Shortcut 1x1
- Batch Normalization
- Global Pooling

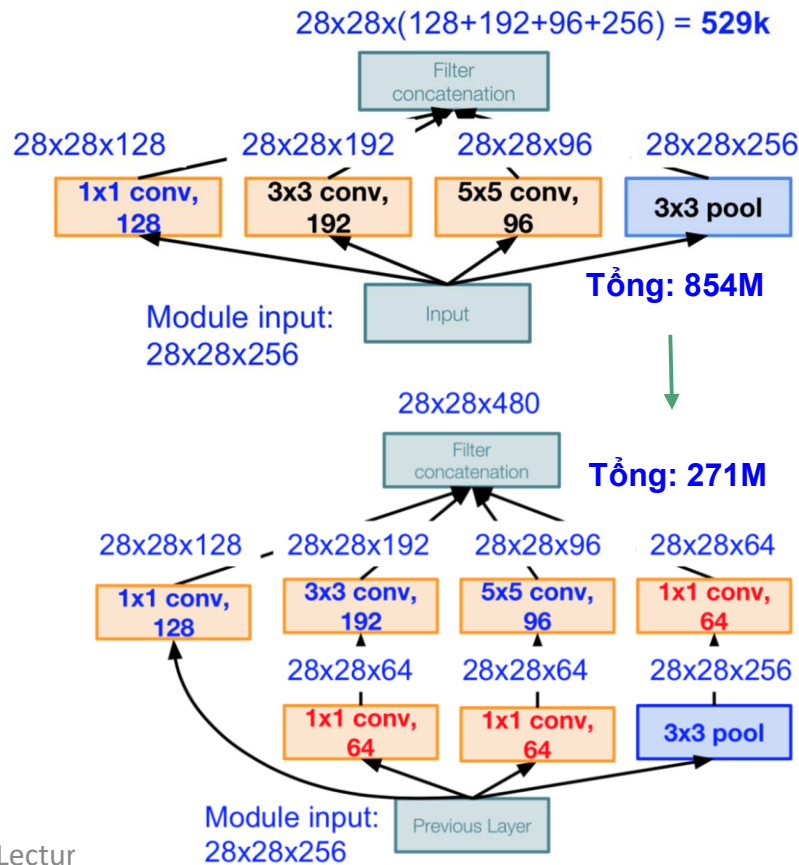


*Kích cỡ của filter nào là tốt nhất ?*

→ Sử dụng nhiều filters/operators cùng 1 lúc

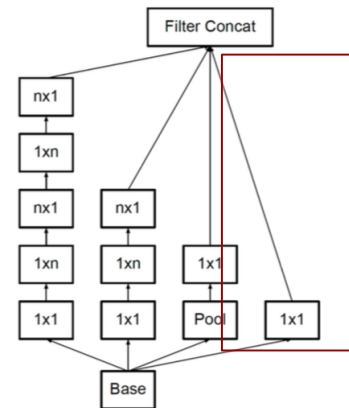
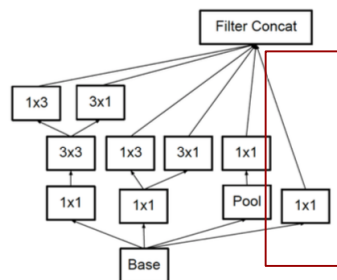
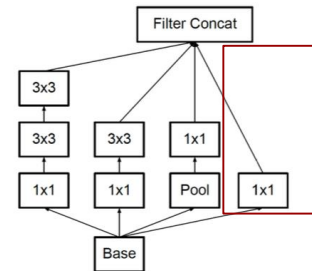
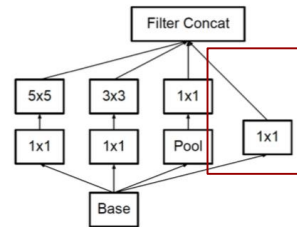
## 2 GoogLeNet - ILSVRC 2014 Winner

- **Bottleneck layer**
  - Giúp giảm số chiều của feature maps
- **Inception Module - Multiple branches**
  - Lấy cảm hứng từ Network in Network
- Shortcut 1x1
- Batch Normalization
- Global Pooling



## 2 GoogLeNet - ILSVRC 201

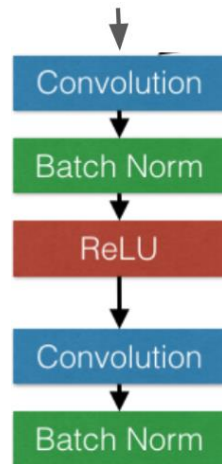
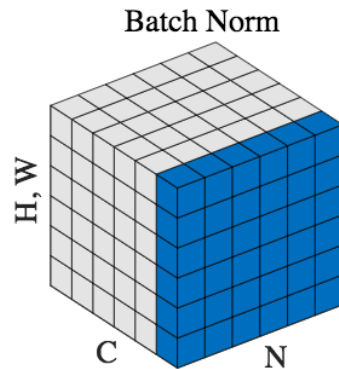
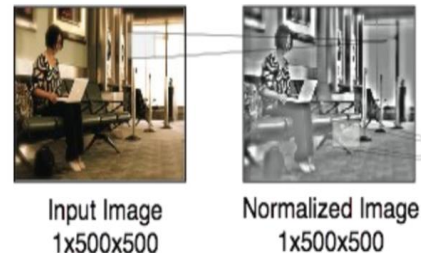
- Bottleneck layer
  - Giúp giảm số chiều của feature maps
- Inception Module - Multiple branches
  - Lấy cảm hứng từ Network in Network
- **Shortcut 1x1**
- Batch Normalization
- Global Pooling





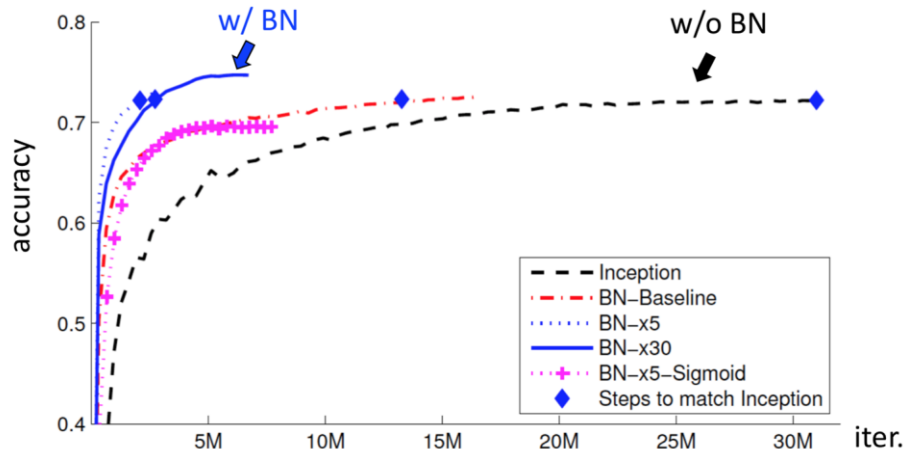
## 2 GoogLeNet - ILSVRC 2014 Winner

- Bottleneck layer
  - Giúp giảm số chiều của feature maps
- Inception Module - Multiple branches
  - Lấy cảm hứng từ Network in Network
- Shortcut 1x1
- **Batch Normalization**
  - Áp dụng Normalizing input cho mỗi layer và normalize theo trên cả mini-batch
- Global Pooling



## 2 GoogLeNet - ILSVRC 2014 Winner

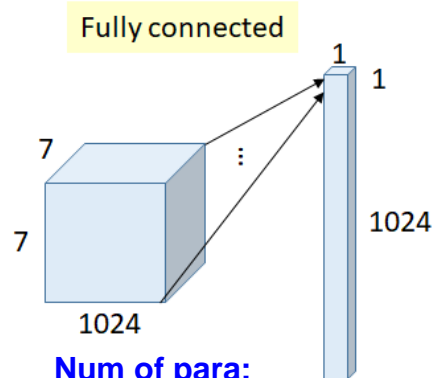
- Bottleneck layer
  - Giúp giảm số chiều của feature maps
- Inception Module - Multiple branches
  - Lấy cảm hứng từ Network in Network
- Shortcut 1x1
- **Batch Normalization**
- Global Pooling



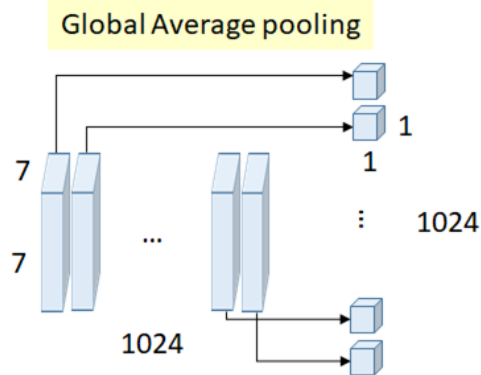
BN giúp tăng tốc quá trình training, giúp mạng ít nhạy cảm hơn với tham số khởi đầu và góp phần làm regularization

## 2 GoogLeNet - ILSVRC 2014 Winner

- Bottleneck layer
  - Giúp giảm số chiều của feature maps
- Inception Module - Multiple branches
  - Lấy cảm hứng từ Network in Network
- Shortcut 1x1
- Batch Normalization
- **Global Pooling**



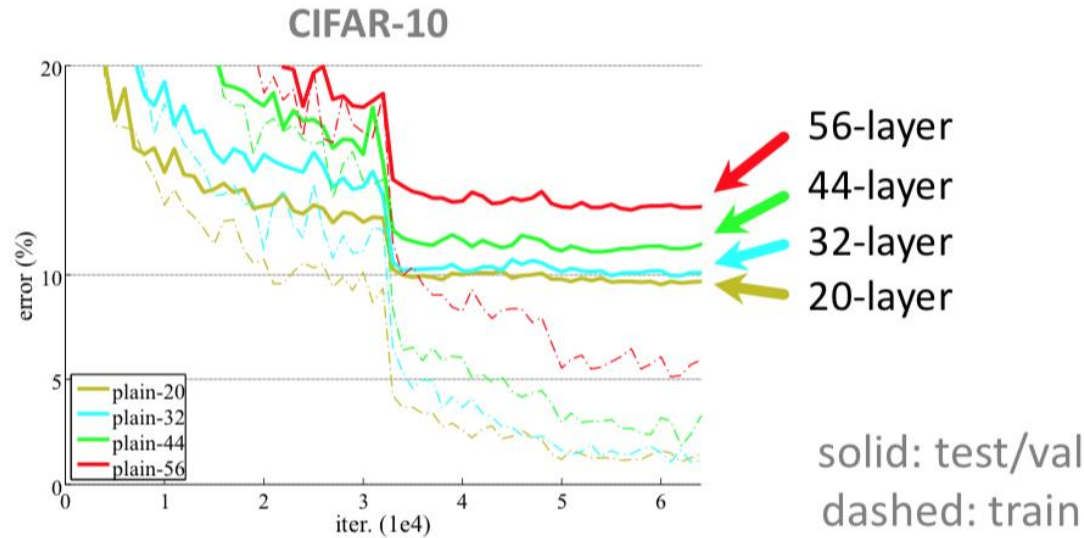
Num of para:  
 $7 \times 7 \times 1024 \times 1024 = 51.3M$



Num of para: 0  
 + improved 0.6% top-1 acc

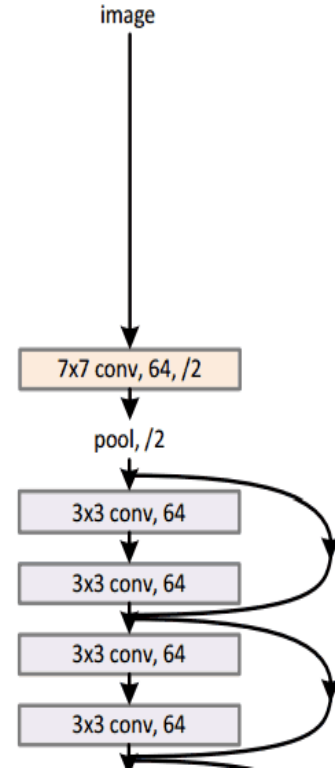
2

# ResNet - ILSVRC 2015 Winner

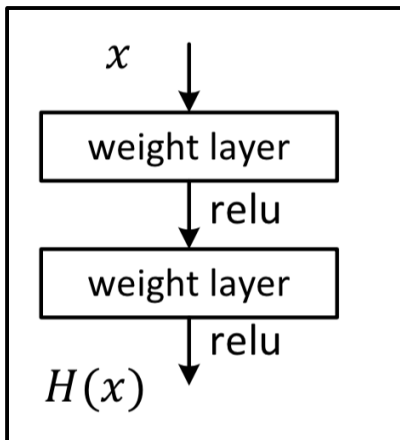


Đơn thuần xếp chồng thêm các layers

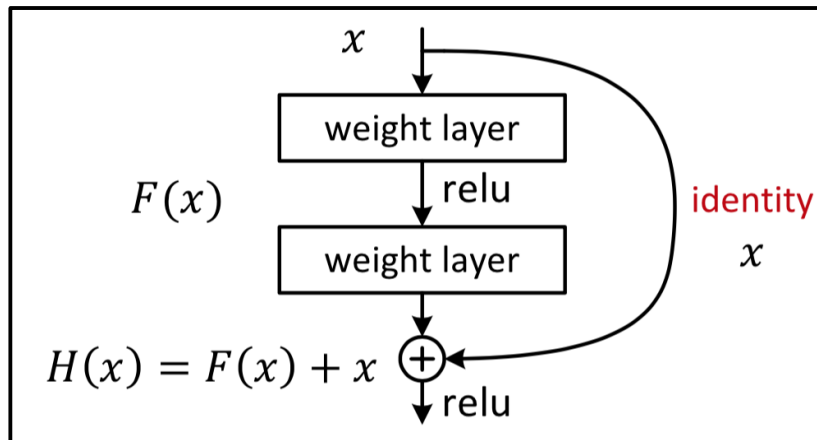
34-layer residual



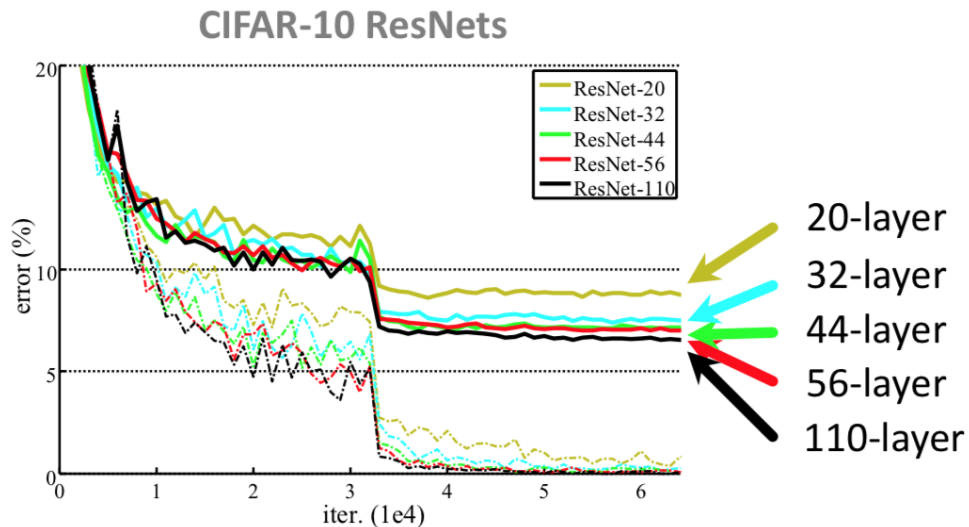
Plain net

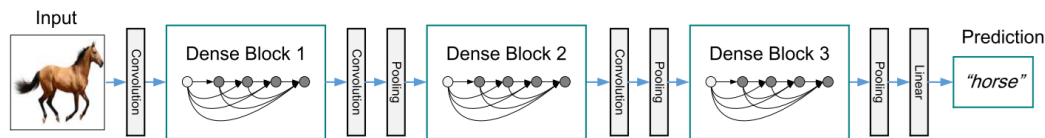


Residual net

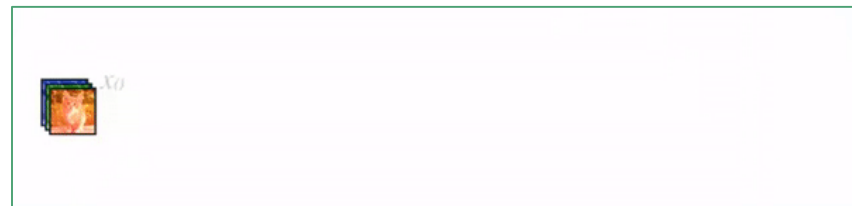


## 2 ResNet - ILSVRC 2015 Winner

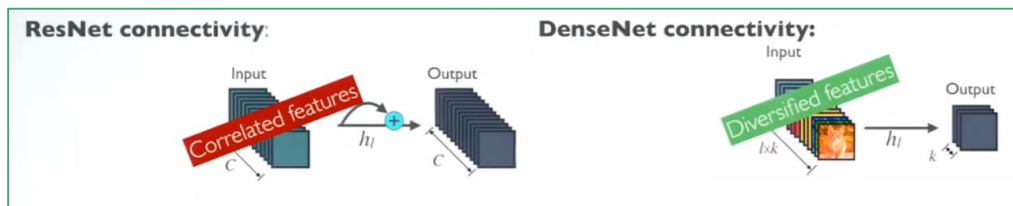




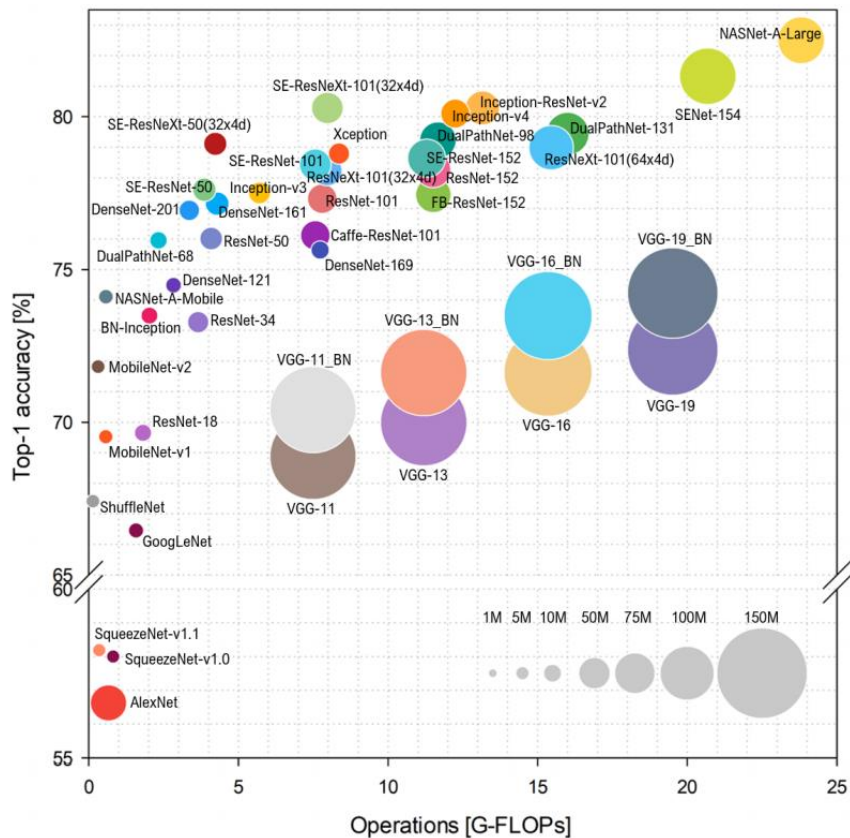
- Dùng Shortcut connection nối mọi layers trước đó đến layer sau (Densely connected)
  - strong gradient flow - deep supervision
  - efficient parameters used
- Concat các lớp thay vì cộng các đường dẫn ở ResNet
  - re-used features (coarse to fine prediction)



Feed-forward flow



## 2 Các kiến trúc CNN

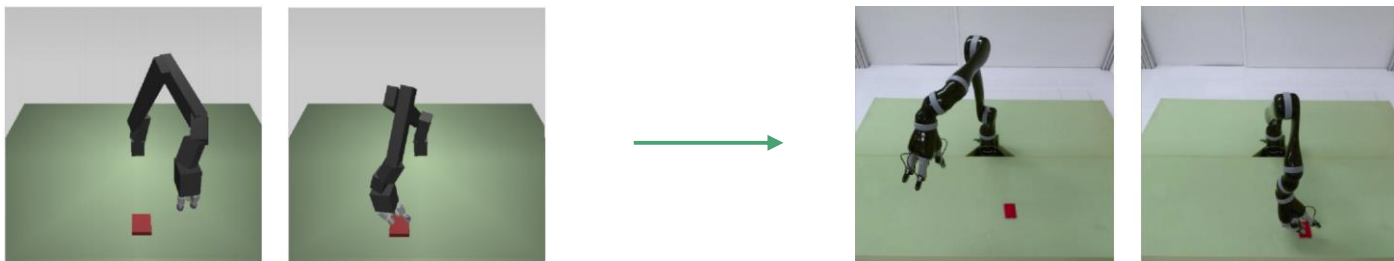


Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	98 MB	0.749	0.921	25,636,712	-
ResNet101	171 MB	0.764	0.928	44,707,176	-
ResNet152	232 MB	0.766	0.931	60,419,944	-
ResNet50V2	98 MB	0.760	0.930	25,613,800	-
ResNet101V2	171 MB	0.772	0.938	44,675,560	-
ResNet152V2	232 MB	0.780	0.942	60,380,648	-
ResNeXt50	96 MB	0.777	0.938	25,097,128	-
ResNeXt101	170 MB	0.787	0.943	44,315,560	-
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-



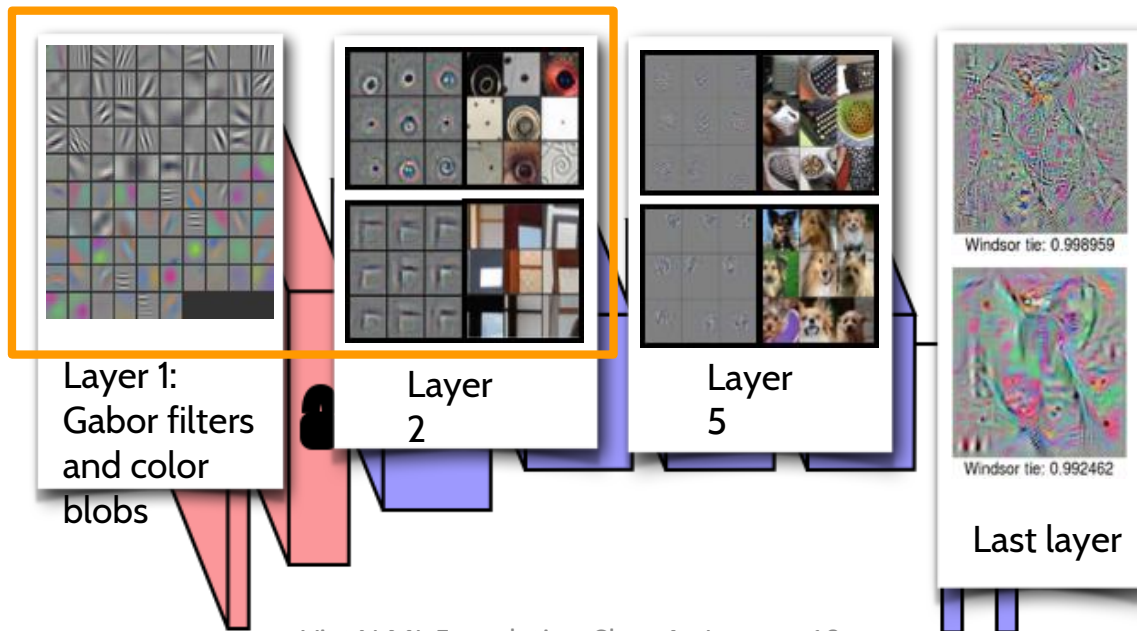
## 3 Transfer Learning

Trên thực tế, rất ít người tự thiết kế kiến trúc mạng riêng và train lại hoàn toàn trên 1 tập dataset. Phương pháp hay được sử dụng là dùng lại một “pretrained” model có kiến trúc thông dụng (Ex. Resnet,...) đã được train trước đó trên một tập dataset lớn, sau đó “train lại một phần network” (retrain) trên tập dataset của bài toán. Phương pháp này được gọi là **Transfer learning**.



### 3 Transfer Learning

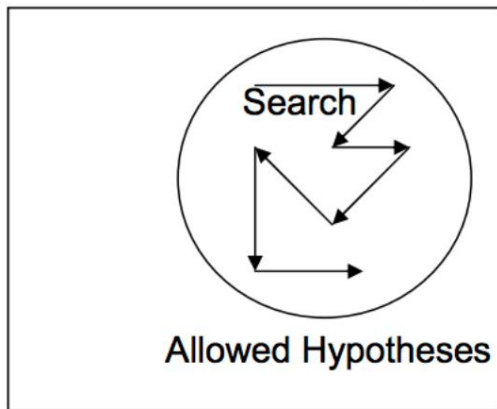
Representation mà một mô hình học được trên một tập dataset lớn thường có tính “universal”, đặc biệt là ở những lớp đầu tiên.



# 3 Transfer Learning

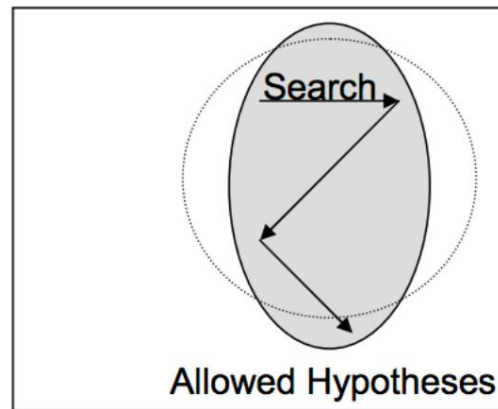
Việc lựa chọn pretrained model đã train trên bộ dataset có tính chất tương tự (hoặc task tương tự) giúp phần learned representation được transfer hiệu quả hơn ở task mới.

**Inductive Learning**



All Hypotheses

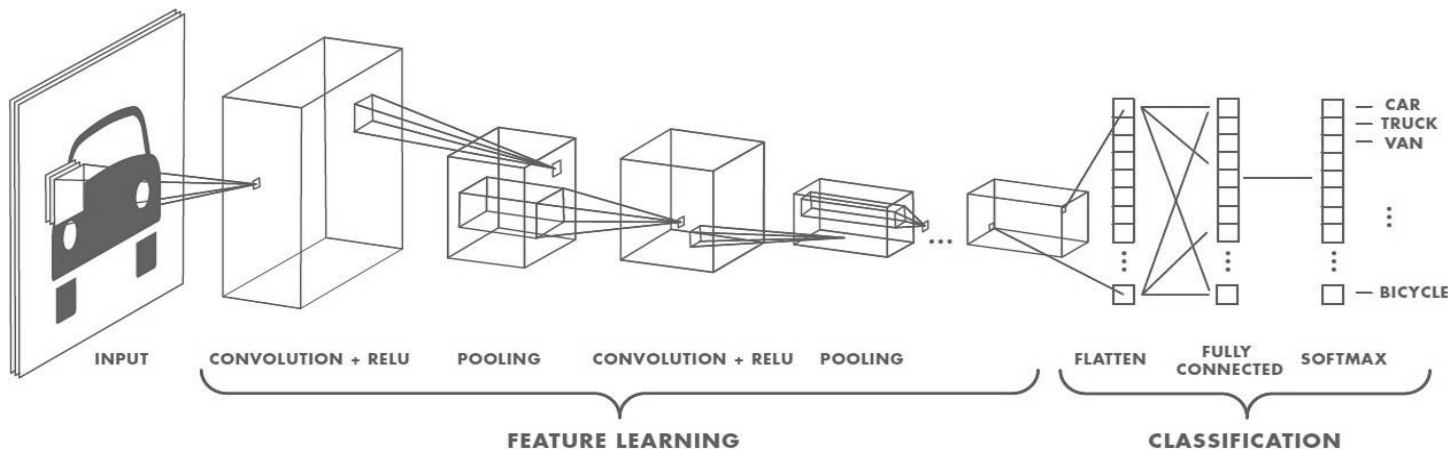
**Inductive Transfer**



All Hypotheses

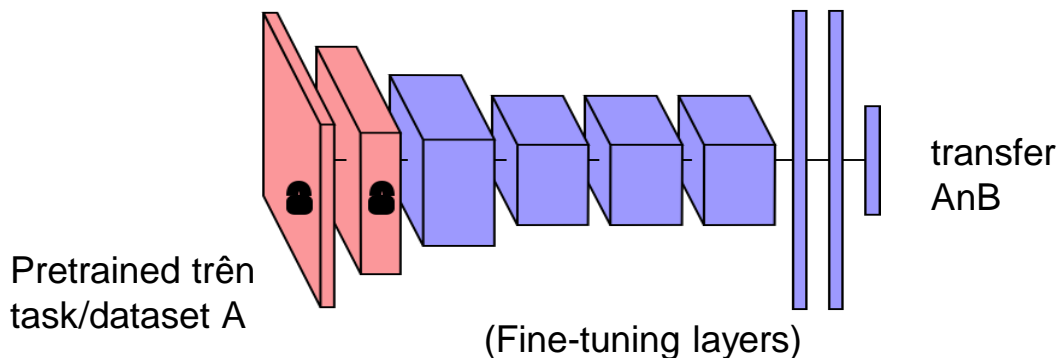
### 3 Một số kỹ thuật Transfer Learning

- **Giữ nguyên phần feature learning:** Sử dụng pretrained model, ta “đóng băng” tất cả trọng số phần feature learning, chỉ train lại một hoặc nhiều lớp Fully-connected được sử dụng như bộ Classification, hoặc thay thế lớp FC bằng 1 mô hình Machine learning khác và train lại (SVM, KNN,...)



### 3 Một số kỹ thuật Transfer Learning

- **Train lại một phần feature learning:** Với khởi tạo trọng số là trọng số của pretrained model, ta train (fine-tuning) lại 1 phần của mạng đóng vai trò feature extractors ở phía gần cuối và train lại phần Classification. Trên một tập dataset đủ lớn, việc này giúp model học được những feature (representation) phù hợp hơn với bộ dataset (task) của bài toán.



3

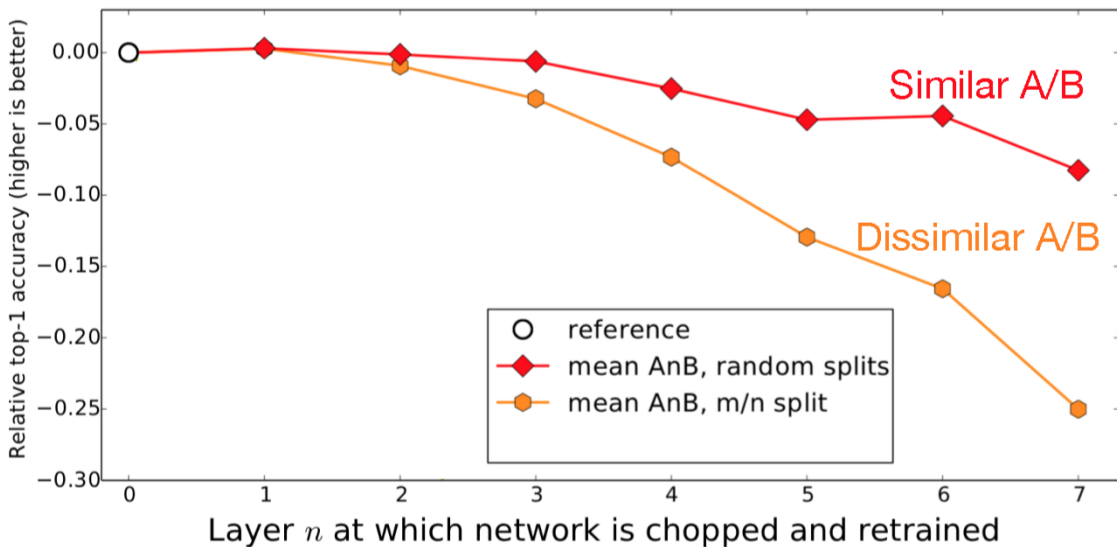
## Một số kỹ thuật Transfer Learning

- Train lại toàn bộ model, với khởi tạo là trọng số của pretrained model.

Lựa chọn retrain ở lớp nào thì hiệu quả?

### 3 Một số kỹ thuật Transfer Learning

Việc lựa chọn sử dụng lớp nào để thực hiện retrain phụ thuộc vào nhiều yếu tố, đặc biệt là: **Sự tương đồng của 2 task** (dataset domain,...), **kích cỡ data của bài toán thực tế**, ...



# Tài liệu tham khảo

1. Asifullah Khan et al. (2019). A Survey of the Recent Architectures of Deep Convolutional Neural Networks
2. [Simone Bianco et al. \(2018\). Benchmark Analysis of Representative Deep Neural Network Architectures](#)
3. <https://keras.io/applications/>
4. Stanford CS231n (2018)
5. UoT CSC2523 Deep Learning in Computer Vision course (2018)