Report paper: Deep Residual Learning for Image Recognition

Giáp Ngọc Hiệu

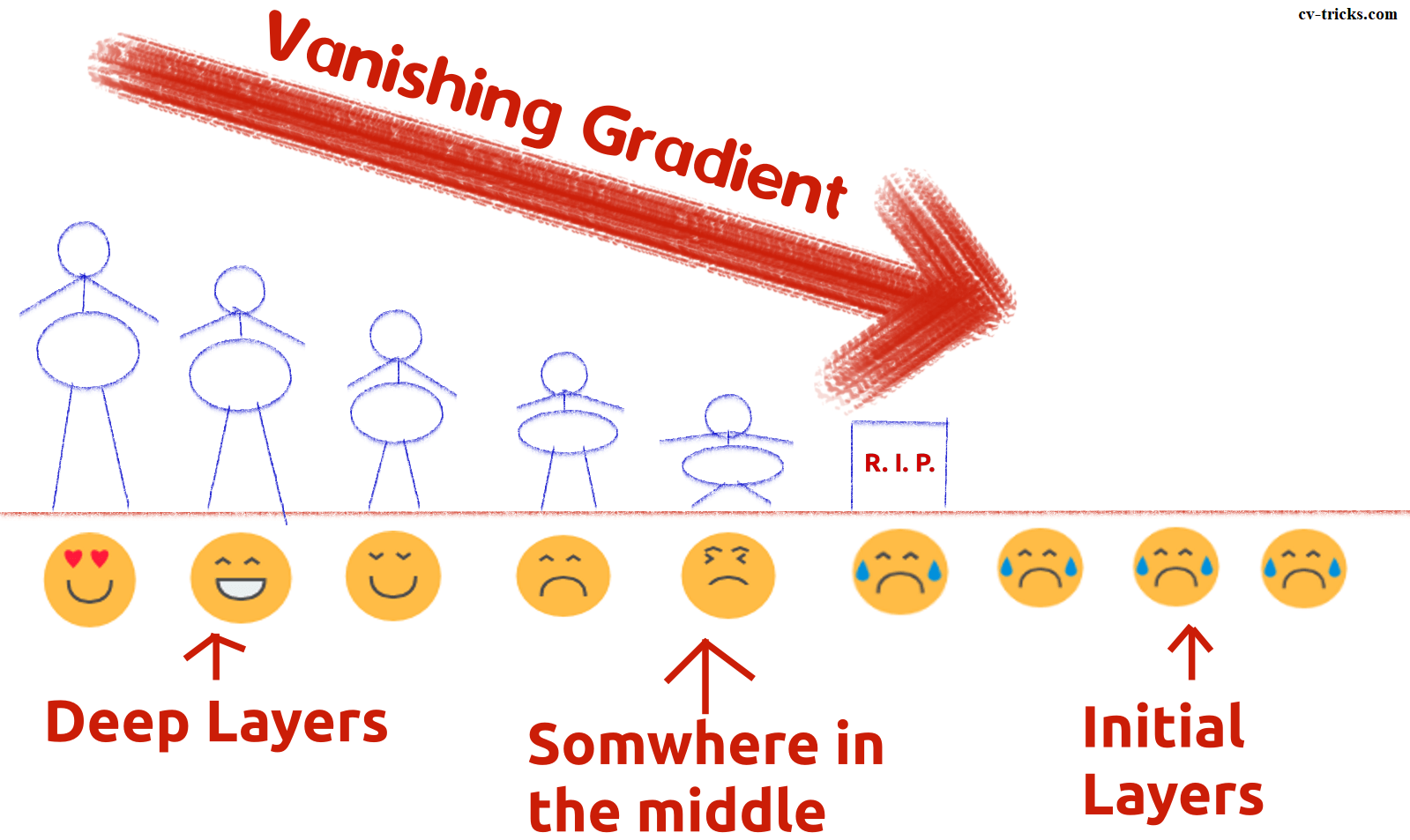
**Tóm tắt nội dung**

Bài report này là phần tóm tắt lại ý tưởng, nội dung chính, những điều làm được của paper

Deep Residual Learning for Image Recognition

1. **Ý tưởng**

Mạng nơ-ron tích chập sâu đã tạo ra nhiều đột phá trong thị giác máy tính, từ các mạng sâu đến rất sâu. Các mạng sâu có thể học và nắm bắt các đặc trưng cấp thấp cho đến các đặc trưng cấp cao của dữ liệu, làm cho việc phân loại trở nên dễ dàng hơn. Nhưng mạng càng sâu thì càng khó để đào tạo. Theo như thực nghiệm của tác giả, khi độ sâu mạng tăng lên, độ chính xác sẽ bão hòa và sau đó giảm xuống. Đây không phải do overfitting mà là do vanishing gradient (gradient cực kì nhỏ) gây ra, làm cho mạng không thể cập nhật và học được.



Giả sử ta có 1 mạng có N layer

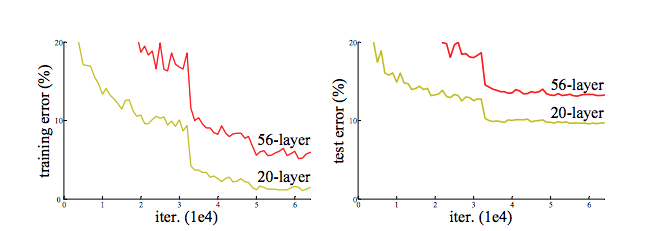


Nếu là identity function:





Từ đây ta có thể thấy nếu gradient của các layer càng nhỏ, khi lan truyền ngược nhân lại với nhau sẽ càng tiến về 0. Điều này làm cho parameter ở các layer gần input không thể cập nhật hoặc cập nhật rất ít, không đóng góp vào việc học của model. Dẫn đến model dù sâu nhưng thực ra chỉ có các layer ở cuối học và có thể không đủ khả năng học được các feature chính của dữ liệu.



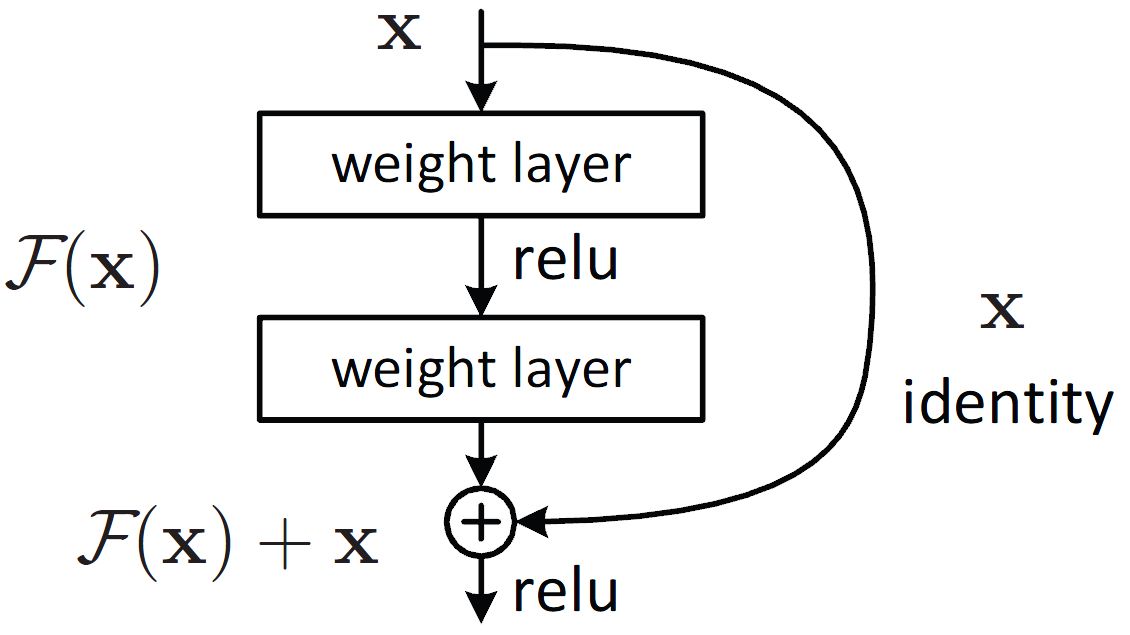
Hình. Thực nghiệm của tác giả

Từ đó, tác giả đã đưa ra ý tưởng thay vì để các layer học các underlying mapping (ánh xạ cơ bản), ta cho mạng học các residual mapping (ánh xạ dư thừa) bằng cách thêm “shortcut” hoặc “skip connection”.

Gọi  là ánh xạ ban đầu, ta để mạng học , từ đó ta có 

Ở đây, F(x) đại diện cho "residual " mà mạng cố gắng học, và x là đầu vào của mạng. Kết quả H(x) thu được bằng cách cộng F(x) với đầu vào x. Như vậy, mạng không cần học phép biến đổi hoàn chỉnh từ đầu, nó chỉ tập trung vào việc học thông tin bổ sung (residual) cần được thêm vào đầu vào, làm cho gradient có thể lưu thông dễ dàng hơn trong quá trình lan truyền ngược, thậm chí qua những mạng cực kỳ sâu. Nhờ đó cho phép đào tạo các mạng sâu hơn, dễ dàng hơn mà vẫn đảm bảo được độ chính xác và khắc phục vấn đề gradient vanishing.

1. **Phương pháp**
   1. **Residual Block**



Residual block là một thành phần quan trọng trong Residual Learning do tác giả đề xuất, đóng vai trò chính giúp huấn luyện các mạng sâu hơn và đạt hiệu suất tốt hơn. Theo tác giả,  có thể bao gồm nhiều lớp hơn.



Trong đó:

 là dữ liệu đầu vào

 là residual mapping

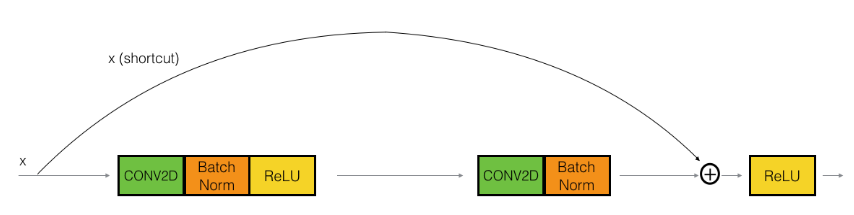
 là đầu ra của Residual Block



Khi không cần học ở nhóm layer này, mạng sẽ điều hướng và học như identity function.

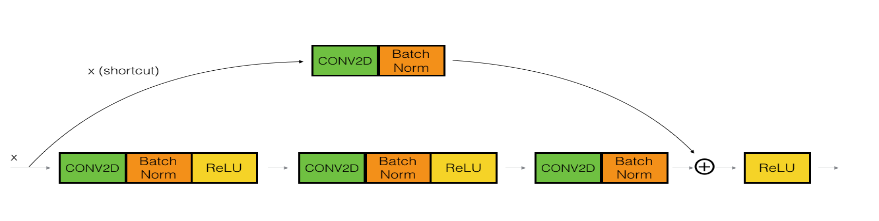
Gradient qua các layer có thể nhỏ dần và thậm chí bằng 0, do đó skip connection giúp thông tin truyền ngược lại dễ dàng hơn.

* 1. **Identity Shortcut**



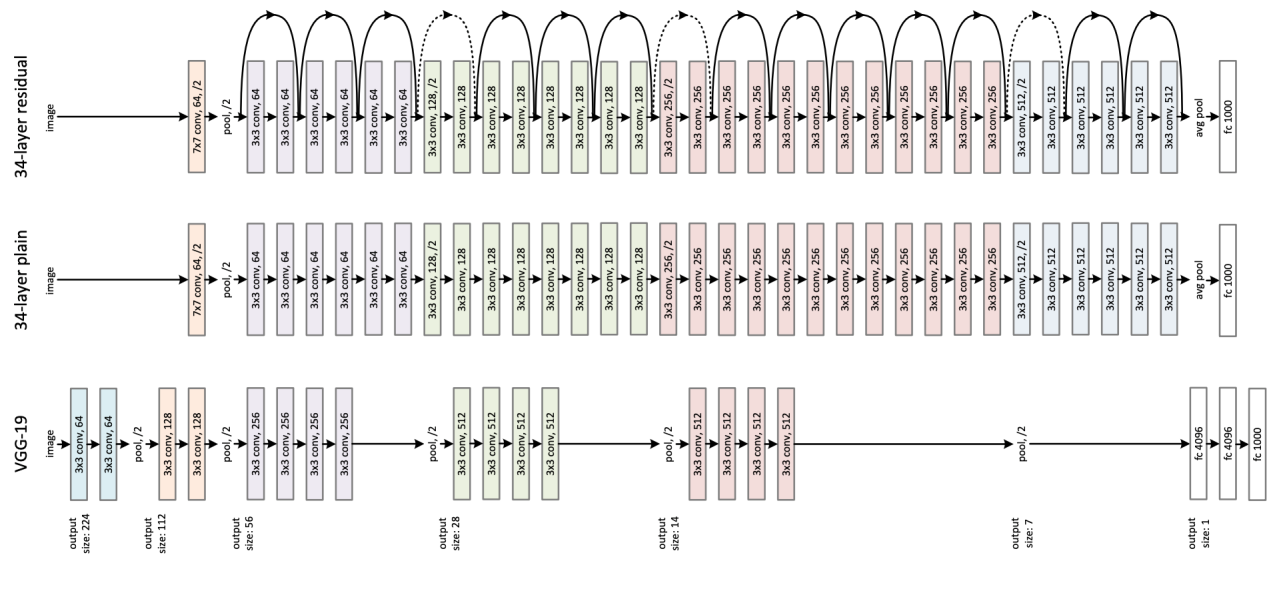
Identity Shortcut là kiểu tiêu chuẩn được sử dụng trong ResNets, đối với trường hợp kích thước đầu vào có cùng kích thước với đầu ra.

* 1. **Projection Shortcut**



Projection Shortcut được sử dụng trong trường hợp kích thước đầu vào khác với kích thước đầu ra. Việc biến đổi kích thước đầu vào có cùng kích thước với đầu ra có thể thực hiện bằng 1x1 convolution, trong bài báo, tác giả sử dụng 1x1 convolution với stride bằng 2 để giảm kích thước đầu vào xuống 2 lần. Ở đây shortcut không dùng activation vì mục đích chỉ để biến đổi kích thước đầu vào.

* 1. **Network Architectures**



**Plain Network:** đượctác giả lấy cảm hứng từ mạng VGG bao gồm 34 layer. Hầu hết các lớp convolution đều có 3×3 filters và tuân theo hai quy tắc:

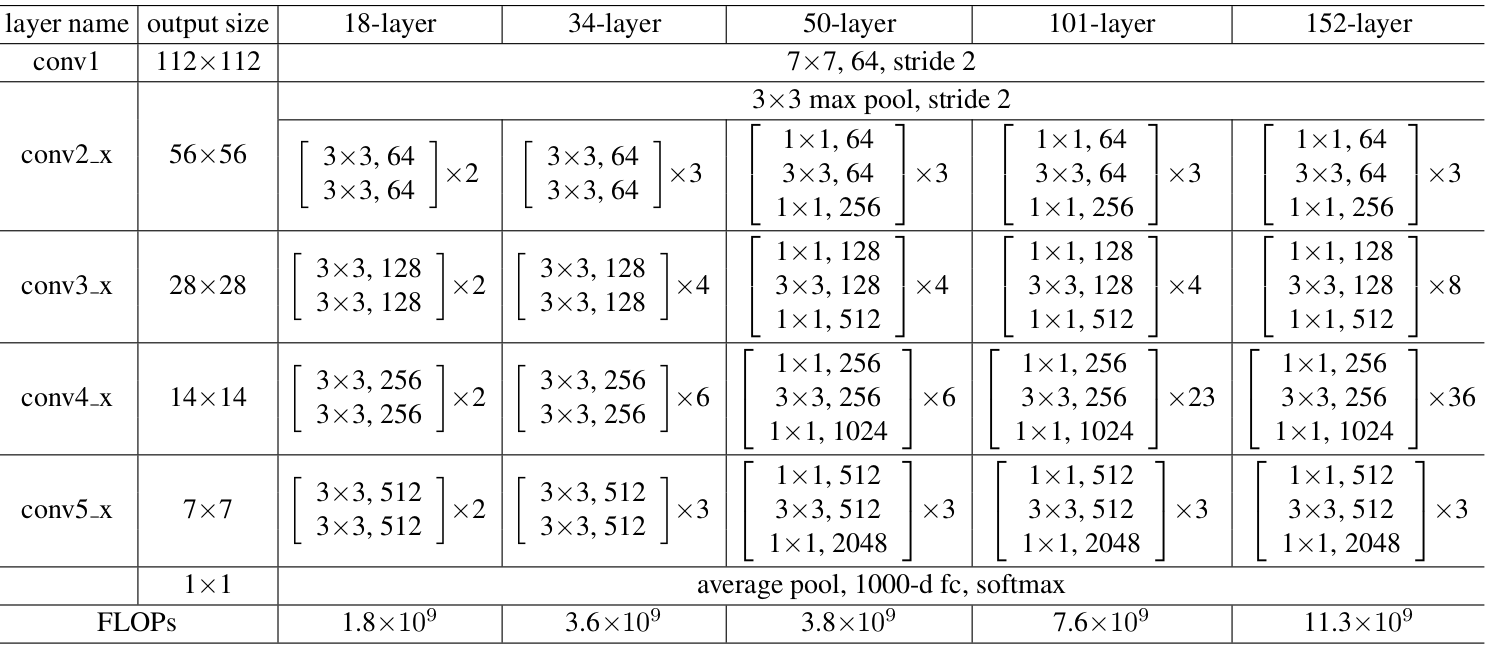
**+** Các lớp có cùng kích thước đầu ra sẽ có cùng số filters

**+** Nếu kích thước đầu ra giảm đi một nửa, thì số filters sẽ tăng gấp đôi. Thực hiện downsampling bằng các lớp convolution có stride bằng 2.

Theo cách này, model của tác giả có độ phức tạp thấp hơn so với mạng VGG khi có số layer nhiều hơn.

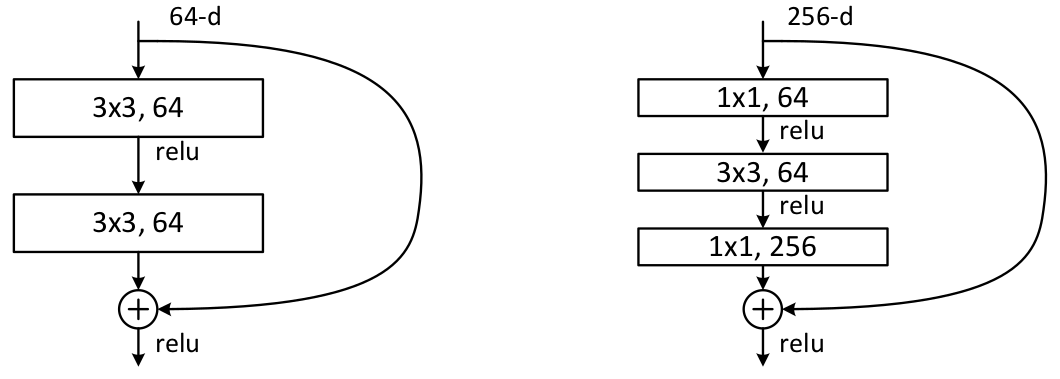
**Residual Network**: dựa vào plain network, thêm các skip connection (shortcut)

Tác giả đã phát triển Resnet với nhiều biến thể như Resnet50, Resnet101, …



Hình. Architectures for ImageNet (paper)

* 1. **Deeper Bottleneck Architectures**



Do lo ngại về việc model càng sâu thì thời gian đào tạo càng lớn, tác giả đã giới thiệu việc sửa đổi các residual block thành bottleneck để giảm thời gian training (giảm số parameters).

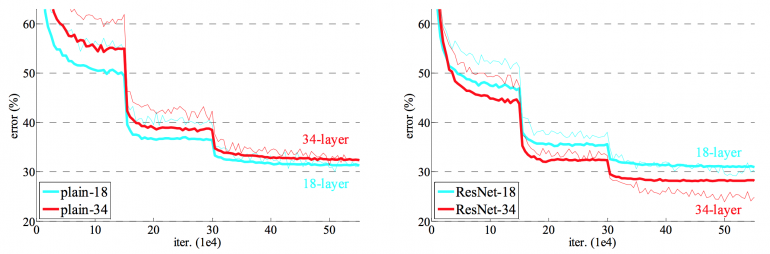
Một bottleneck bao gồm 3 layer:

+ 1x1 Convolution làm giảm dimension

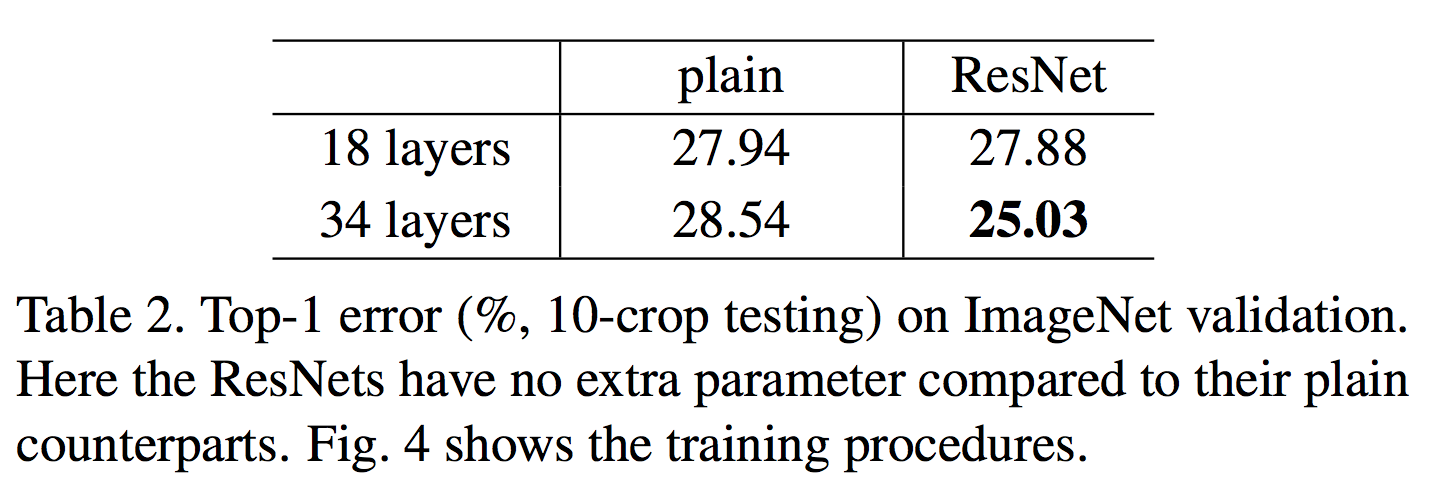
+ 3x3 Convolution thực hiện việc học

+ 1x1 Convolution khôi phục lại dimension

1. **Kết quả**

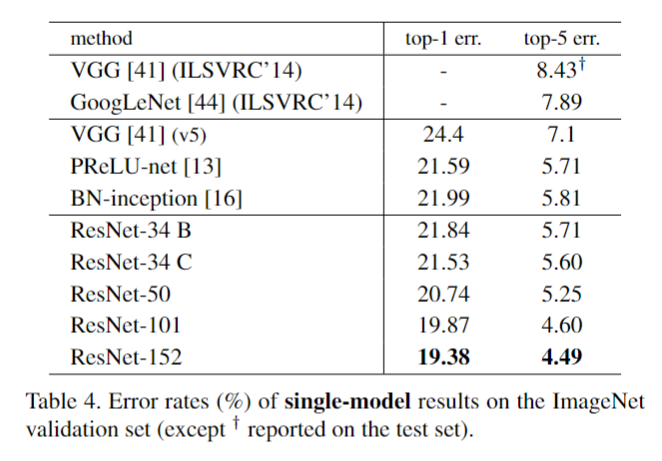
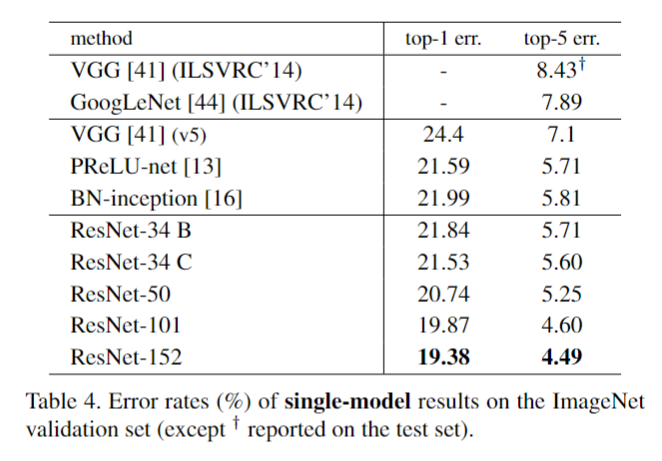


Hình. Training on ImageNet. Thin curves denote training error, and bold curves denote validation error of the center crops.



Resnet34 có hiệu suất cao hơn so với Plain và Resnet18, cho thấy vấn đề degradation đã được giải quyết.

Ngoài ra, Resnet cũng đã vượt qua các model tại thời điểm đó trên nhiều nhiệm vụ khác nhau.

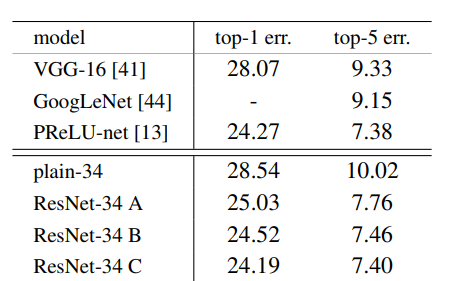
1. **Note**

- Đối với vấn đề kích thước đầu vào khác với kích thước đầu ra, tác giả có thử với 3 lựa chọn:

+ A: Sử dụng identity shortcut cho toàn bộ mạng, phần dimension bị thiếu ở input sẽ được zero-padding để thực hiện add với output

+ B: Sử dụng projection shortcut cho các block có kích thước input và output khác nhau, còn lại là identity shortcut

+ C: Sử dụng toàn bộ là projection shortcut



Kết quả thu được cho thấy:

+ B tốt hơn A: vì phần skip connection của A là zero-padding nên sẽ không học được như B

+ C tốt hơn B: vì skip connection ở C được học hoàn toàn so với B

Theo tác giả, để cân bằng độ chính xác và số parameter, lựa chọn B sẽ được chọn làm model chính.

- Đối với các mạng sâu hơn (50 layer trở đi), tác giả khuyên nên dùng bottleneck architectures để giảm thời gian đào tạo mà vẫn đạt được độ chính xác cao.

- Lợi ích của Resnet:

+ Cho phép xây dựng và dào tạo các mạng sâu hơn, thậm chí hàng ngàn layer, tránh được vanishing gradient bằng các skip connection

+ Resnet cải thiện hiệu suất vượt trội so với các model khác

+ Giúp việc đào tạo hội tụ nhanh hơn do gradient truyền lại dễ dàng trong quá trình lan truyền ngược

+ Có thể áp dụng cho các nhiệm vụ khác nhau trong thị giác máy tính.

- Một số thách thức với Resnet:

+ Đôi khi vẫn tiềm ẩn hiện tượng vanishing gradient đối với các model rất sâu

+ Các kiến trúc mạng sâu của Resnet có thể yêu cầu nhiều bộ nhớ và tài nguyên tính toán

+ Resnet hoạt động rất tốt trên các nhiệm vụ của thị giác máy tính, nhưng có thể không hiệu quả đối với các loại dữ liệu khác như text, audio,..