TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN CUỐI KÌ MÔN CÁC HỆ THỐNG PHÂN TÁN

DEEP RESIDUAL LEARNING FOR IMAGE RECOGNITION

Người hướng dẫn: **TS NGUYỄN XUÂN SÂM**
Người thực hiện: **HỒNG QUANG VINH – 186005004**
NGUYỄN ĐẠI THỊNH – 186005035
Lớp**: 18600531**
Khoá**: 2018-2020**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN CUỐI KÌ MÔN CÁC HỆ THỐNG PHÂN TÁN

DEEP RESIDUAL LEARNING FOR IMAGE RECOGNITION

Người hướng dẫn: **TS NGUYỄN XUÂN SÂM**
Người thực hiện: **HỒNG QUANG VINH – 186005004**
NGUYỄN ĐẠI THỊNH – 186005035
Lớp**: 18600531**
Khoá**: 2018-2020**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019

LỜI CẢM ƠN

Nhóm chúng em xin chân thành cảm ơn Thầy Nguyễn Xuân Sâm đã giúp đỡ chúng em hoàn thành đồ án. Những hướng dẫn của Thầy giúp chúng em có một nền tảng lý thuyết đủ để có thể ứng dụng và nghiên cứu phát triển đề tài này. Xin chân thành cảm ơn Thầy.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS Nguyễn Xuân Sâm;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Hồng Quang Vinh

Nguyễn Đại Thịnh

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Các mạng deep nơ-ron là rất khó để huấn luyện, bài viết trình bày một framework học để dễ dàng huấn luyện các mạng sâu (nhiều lớp) hơn đáng kể so với các mạng được sử dụng trước đây. Chúng tôi cải cách lại các tầng là học các hàm dư (residual function) với tham chiếu đến các đầu vào của lớp, thay vì học các hàm không tham chiếu. Chúng tôi cung cấp bằng chứng thực nghiệm toàn diện cho thấy các mạng dư (residual) này dễ tối ưu hóa hơn và có thể đạt được độ chính xác từ độ sâu tăng đáng kể. Trên bộ dữ liệu ImageNet, chúng tôi đánh giá các mạng residual có độ sâu lên tới 152 lớp (layers), sâu hơn 8 lớp (layers) so với lưới VGG [40] nhưng vẫn có độ phức tạp thấp hơn. Một tập hợp các mạng còn lại đạt được lỗi 3.57% trên bộ thử nghiệm ImageNet. Kết quả này đã giành được vị trí số 1 trong nhiệm vụ phân loại ILSVRC 2015. Chúng tôi cũng phân tích về CIFAR-10 với 100 và 1000 lớp.

Độ sâu của các đại diện có tầm quan trọng chính cho nhiều nhiệm vụ nhận dạng hình ảnh. Nhờ đó, chúng tôi có được sự cải thiện tương đối 28% trên bộ dữ liệu phát hiện đối tượng COCO. Mạng dư (residual network) là nền tảng của các bài thi của chúng tôi cho các cuộc thi ILSVRC & COCO 2015, nơi chúng tôi cũng giành được vị trí số 1 về các nhiệm vụ phát hiện ImageNet, cục bộ hóa ImageNet, phát hiện COCO và phân đoạn COCO.

MỤC LỤC

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

DANH MỤC BẢNG

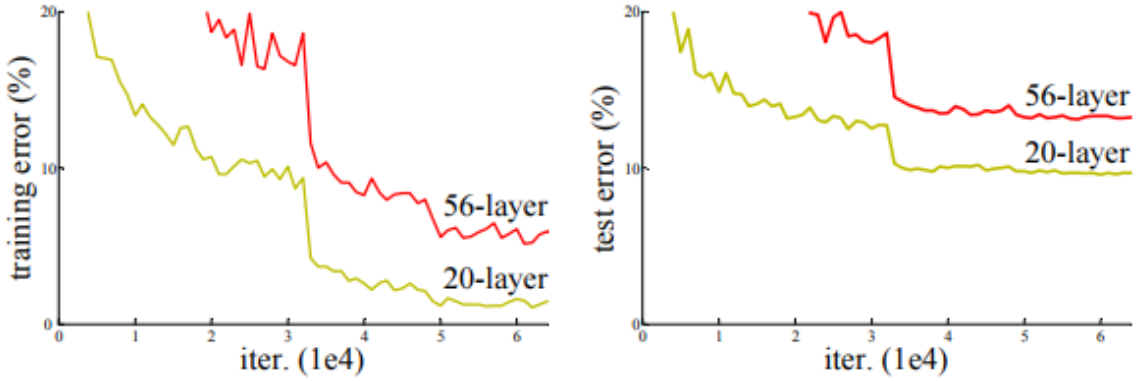
CHƯƠNG 1 – MỞ ĐẦU

1.1 Giới thiệu

Mạng nơ-ron tích chập (Deep Convolutional Neural Network) [22, 21] đã dẫn đến một loạt các bước đột phá để phân loại hình ảnh [21, 49, 39]. Các mạng sâu tích hợp một cách tự nhiên các đặc trưng cấp thấp/trung bình/cao [49] và phân loại theo kiểu đa lớp từ đầu đến cuối, và các cấp độ sâu của các tính năng có thể được làm phong phú bằng số lượng các lớp xếp chồng lên nhau (số lớp). Bằng chứng gần đây [40, 43] cho thấy độ sâu của mạng là rất quan trọng và các kết quả hàng đầu [40, 43, 12, 16] trên bộ dữ liệu ImageNet đầy thách thức [35] đều khai thác các mô hình rất sâu [40], với một độ sâu mười sáu [40] đến ba mươi [16]. Nhiều nhiệm vụ nhận dạng hình ảnh đặc biết khác [7, 11, 6, 32, 27] cũng đã thu được nhiều kết quả từ các mô hình rất sâu.

Được thúc đẩy bởi tầm quan trọng của chiều sâu, một câu hỏi được đặt ra: Việc huấn luyện các mạng tốt hơn có dễ như xếp chồng nhiều lớp không? Một trở ngại để trả lời câu hỏi này là vấn đề rõ ràng về sự biến mất/nổ độ dốc (vanishing/exploding gradient) [14, 1, 8], cản trở sự hội tụ ngay từ đầu. Tuy nhiên, vấn đề này đã được giải quyết chủ yếu bằng cách khởi tạo chuẩn hóa [23, 8, 36, 12] và các lớp chuẩn hóa trung gian [16], cho phép các mạng có hàng chục lớp bắt đầu kết hợp với độ dốc dốc ngẫu nhiên (SGD) với sự lan truyền ngược (Backpropagation) [22].

Khi các mạng sâu hơn có thể bắt đầu hội tụ, một vấn đề xuống cấp đã được chỉ ra: với độ sâu của mạng tăng lên, độ chính xác sẽ bão hòa và sau đó xuống cấp nhanh chóng. Thật bất ngờ, sự xuống cấp như vậy không phải do quá khớp (overfitting) và việc thêm nhiều lớp vào mô hình sâu phù hợp dẫn đến lỗi đào tạo cao hơn, như đã báo cáo trong [10, 41] và được xác minh kỹ lưỡng bởi các thí nghiệm của chúng tôi. Hình 1 cho thấy một ví dụ điển hình.

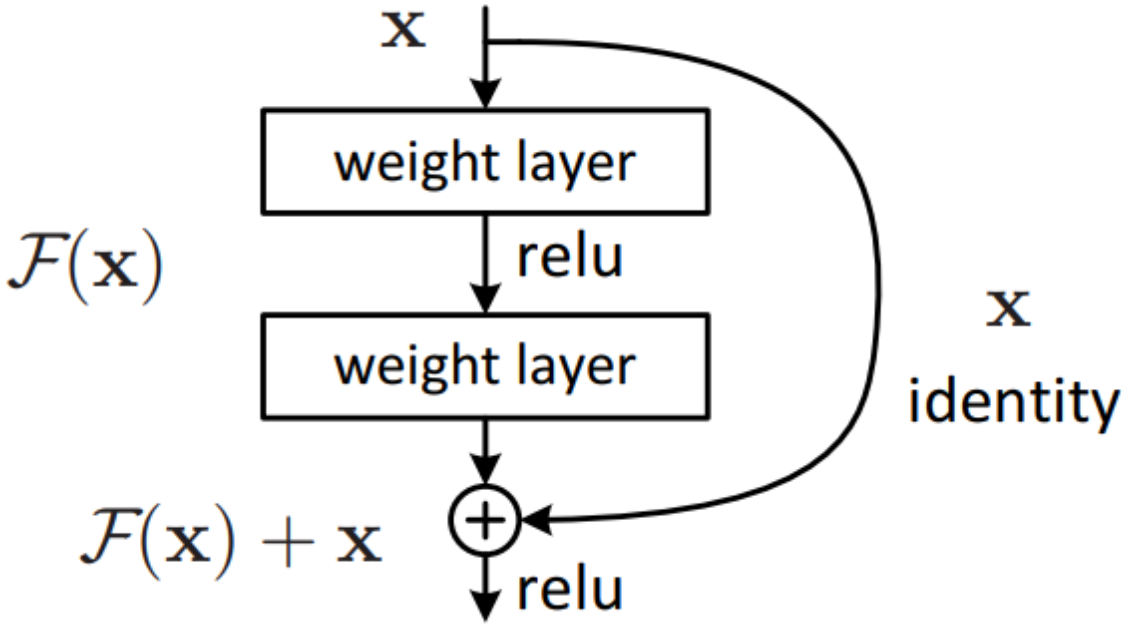


Hình 1: Tỉ lệ lỗi trên CIFAR-10 với mạng 20 và 56 lớp

Sự xuống cấp (về độ chính xác) chỉ ra rằng không phải tất cả các hệ thống đều dễ dàng tối ưu hóa như nhau. Có tồn tại một giải pháp bằng cách xây dựng cho mô hình sâu hơn: các lớp được thêm vào là ánh xạ định danh và các lớp khác được sao chép từ mô hình nông hơn đã học. Sự tồn tại của giải pháp được xây dựng này chỉ ra rằng một mô hình sâu hơn sẽ không tạo ra lỗi đào tạo cao hơn so với đối tác nông hơn. Nhưng các thí nghiệm cho thấy những phương pháp giải quyết hiện tại không thể tìm ra giải pháp tương đối tốt hoặc tốt hơn giải pháp được xây dựng (hoặc không thể thực hiện trong thời gian khả thi).

Trong bài này, chúng tôi giải quyết vấn đề xuống cấp bằng cách giới thiệu một framework học dư (residual learning) sâu. Thay vì hy vọng mỗi một vài lớp xếp chồng fit trực tiếp với mapping cơ bản mong muốn, chúng tôi rõ ràng để các lớp này fit với residual mapping. Chính thức, biểu thị mapping cơ bản mong muốn là H(x), chúng ta để các lớp phi tuyến xếp chồng lên nhau fit với mapping khác của F (x): = H (x) - x. Ánh xạ gốc được đúc lại thành F (x) + x. Chúng tôi đưa ra giả thuyết rằng việc tối ưu hóa residual mapping dễ dàng hơn so với tối ưu hóa ánh xạ gốc, không được ước tính. Nếu ánh xạ định danh là tối ưu, việc đẩy phần dư về 0 sẽ dễ dàng hơn so với điều chỉnh ánh xạ định danh bằng một chồng các lớp phi tuyến.

Công thức của F (x) + x có thể được nhận ra bằng các mạng nơ-ron dẫn nguồn với các kết nối tắt (shortcut connection) trên mạng Cameron (Hình 2). Các kết nối tắt [2, 33, 48] là những kết nối bỏ qua một hoặc nhiều lớp. Trong trường hợp của chúng tôi, các kết nối tắt chỉ đơn giản là thực hiện ánh xạ định danh và các đầu ra của chúng được thêm vào đầu ra của các lớp xếp chồng lên nhau (Hình 2). Các kết nối cắt ngắn danh tính không thêm độ phức tạp cũng như độ phức tạp tính toán. Toàn bộ mạng vẫn có thể được SGD đào tạo từ đầu đến cuối bằng cách truyền ngược và có thể dễ dàng thực hiện bằng các thư viện chung (ví dụ: Caffe [19]) mà không cần sửa đổi phương pháp.



Hình 2: Residual learning: building block

Chúng tôi trình bày các thử nghiệm toàn diện trên ImageNet [35] để hiển thị vấn đề xuống cấp và đánh giá phương pháp của chúng tôi. Chúng tôi chỉ ra rằng: 1) Các mạng dư cực kỳ sâu của chúng tôi rất dễ tối ưu hóa, nhưng các mạng thông thường (đối với các lớp ngăn xếp) biểu hiện lỗi đào tạo cao hơn khi độ sâu tăng; 2) Lưới dư sâu của

chúng tôi có thể dễ dàng đạt độ chính xác từ độ sâu tăng lên đáng kể, tạo ra kết quả tốt hơn đáng kể so với các mạng trước đó.

Hiện tượng tương tự cũng được hiển thị trên bộ CIFAR-10 [20], cho thấy rằng những khó khăn tối ưu hóa và hiệu quả của phương pháp của chúng tôi không giống với một bộ dữ liệu cụ thể. Chúng tôi trình bày các mô hình được đào tạo thành công trên bộ dữ liệu này với hơn 100 lớp và khám phá các mô hình với hơn 1000 lớp.

Trên tập dữ liệu phân loại ImageNet [35], chúng tôi thu được kết quả tuyệt vời bằng các lưới dư cực sâu. Mạng dư 152 lớp của chúng tôi là mạng sâu nhất từng được trình bày trên ImageNet, trong khi vẫn có độ phức tạp thấp hơn so với lưới VGG [40]. Bộ thử nghiệm ImageNet và giành vị trí số 1 trong cuộc thi phân loại ILSVRC 2015. Sự phẫn nộ cực kỳ sâu sắc cũng có hiệu suất khái quát hóa tuyệt vời đối với các nhiệm vụ khác và giúp chúng tôi giành được vị trí số 1 về: Phát hiện ImageNet, cục bộ hóa ImageNet, phát hiện COCO và phân đoạn COCO trong các cuộc thi ILSVRC & COCO 2015. Bằng chứng mạnh mẽ này cho thấy nguyên tắc học tập còn lại là chung và chúng tôi hy vọng rằng nó có thể áp dụng trong các vấn đề về vision và non-vision khác.

1.2 Các công việc liên quan

Residual Representation: Trong nhận dạng hình ảnh, VLAD [18] là một đại diện mã hóa bởi các vectơ dư liên quan đến từ điển và Fisher Vector [30] có thể được coi là một phiên bản xác suất [18] của VLAD. Cả hai đều là đại diện nông mạnh mẽ cho việc tái hiện và phân loại hình ảnh [4, 47]. Đối với lượng tử hóa vectơ, mã hóa vectơ dư [17] được hiển thị là hiệu quả hơn so với mã hóa vectơ gốc.

Trong thị giác ở mức độ thấp và đồ họa máy tính, để giải các phương trình vi phân từng phần (PDEs), phương pháp Multigrid được sử dụng rộng rãi [3] đã định dạng lại hệ thống dưới dạng các bài toán con ở nhiều tỷ lệ, trong đó mỗi bài toán con có thể tương thích với giải pháp còn lại giữa một máy thô và một quy mô tốt hơn. Một thay thế cho Multigrid là tiền điều hòa cơ sở phân cấp [44, 45], dựa trên các biến đại diện cho các

vectơ còn lại giữa hai thang đo. Nó đã được chỉ ra [3, 44, 45] rằng các bộ giải này hội tụ nhanh hơn nhiều so với các bộ giải tiêu chuẩn mà không biết về bản chất còn lại của các giải pháp. Những phương pháp này cho thấy một cải cách tốt hoặc tiền điều kiện có thể đơn giản hóa việc tối ưu hóa.

Shortcut Connection (Kết nối tắt): Thực tiễn và lý thuyết dẫn đến các kết nối tắt [2, 33, 48] đã được nghiên cứu trong một thời gian dài. Một thực hành sớm về đào tạo các tri giác đa lớp (MLP) là thêm một lớp tuyến tính được kết nối từ đầu vào mạng vào đầu ra [33, 48]. Trong [43, 24], một vài lớp trung gian được kết nối trực tiếp với các phân loại phụ trợ để giải quyết các độ dốc biến mất / nổ. Các bài báo của [38, 37, 31, 46] đề xuất các phương pháp để định tâm các phản ứng của lớp, độ dốc và các lỗi lan truyền, được thực hiện bằng các kết nối tắt. Trong [43], một lớp khởi động của người Viking được cấu tạo bởi một nhánh tắt và một vài nhánh sâu hơn.

Đồng thời với công việc của chúng tôi, các mạng [41, 42] hiện các kết nối tắt với các chức năng gating [15]. Các cổng này phụ thuộc vào dữ liệu và có các tham số, trái ngược với các lối tắt nhận dạng của chúng tôi không có tham số. Khi một lối tắt có kiểm soát là khu vực đóng kín (gần bằng 0), các lớp trong mạng cao tốc đại diện cho các chức năng không còn lại. Trái lại, công thức của chúng tôi luôn học các hàm dư; các lối tắt nhận dạng của chúng tôi không bao giờ bị đóng và tất cả thông tin luôn được chuyển qua, với các chức năng còn lại sẽ được học. Ngoài ra, mạng cao tốc đã không chứng minh mức tăng độ chính xác với độ sâu cực kỳ tăng (ví dụ: hơn 100 lớp).

CHƯƠNG II - DEEP RESIDUAL LEARNING

2.1 Residual Learning

Chúng ta hãy xem H (x) như một ánh xạ cơ bản phù hợp với một vài lớp xếp chồng (không nhất thiết là toàn bộ mạng), với x biểu thị các đầu vào cho lớp đầu tiên của các lớp này. Nếu người ta đưa ra giả thuyết rằng nhiều lớp phi tuyến có thể xấp xỉ các hàm phức tạp gần đúng, thì nó tương đương với giả thuyết rằng chúng có thể xấp xỉ các hàm dư, tức là, H (x) - x (giả sử rằng đầu vào và đầu ra có cùng kích thước). Vì vậy, thay vì mong đợi các lớp xếp chồng lên nhau để xấp xỉ H (x), chúng tôi rõ ràng để các lớp này xấp xỉ một hàm dư F (x): = H (x) - x. Do đó, hàm ban đầu trở thành F (x) + x.

Sự cải cách này được thúc đẩy bởi các hiện tượng phản trực giác về vấn đề suy thoái (Hình 1, bên trái). Nếu các lớp được thêm vào có thể được xây dựng dưới dạng ánh xạ định danh, một mô hình sâu hơn sẽ có lỗi đào tạo không lớn hơn đối tác nông hơn. Vấn đề xuống cấp cho thấy rằng người giải có thể gặp khó khăn trong việc xấp xỉ ánh xạ định danh bằng nhiều lớp phi tuyến. Với công thức học dư, nếu ánh xạ định danh là tối ưu, người giải có thể chỉ cần điều chỉnh trọng số của nhiều lớp phi tuyến về 0 để tiếp cận ánh xạ định danh.

Trong các trường hợp thực tế, không chắc là ánh xạ định danh là tối ưu, nhưng cải cách của chúng tôi có thể giúp giải quyết vấn đề. Nếu hàm tối ưu gần với ánh xạ định danh hơn so với ánh xạ bằng 0, thì người giải sẽ dễ dàng tìm thấy các nhiễu loạn có tham chiếu đến ánh xạ định danh hơn là tìm hiểu hàm như một ánh xạ mới. Chúng tôi chỉ ra bằng các thí nghiệm (Hình 7) rằng các hàm dư đã học nói chung có các phản hồi nhỏ, cho thấy rằng ánh xạ định danh cung cấp điều kiện tiên quyết hợp lý.

2.2 Identity Mapping by Shortcuts

Chúng tôi áp dụng học tập còn lại cho mỗi vài lớp xếp chồng lên nhau. Một khối xây dựng được hiển thị trong Hình 2. Chính thức, trong bài báo này, chúng tôi xem xét một khối xây dựng được xác định là:

$y = F(x, \{W_i\}) + x$ (1)

Ở đây x và y là các vectơ đầu vào và đầu ra của các lớp được xem xét. Hàm F(x, {Wi}) biểu thị ánh xạ dư sẽ được học. Đối với ví dụ trong Hình 2 có hai lớp, F = $W_{2\sigma}(W_1 x)$ trong đó σ biểu thị ReLU [29] và các sai lệch được bỏ qua để đơn giản hóa các thông báo. Hoạt động F + x được thực hiện bằng kết nối tắt và bổ sung phần tử. Chúng tôi áp dụng tính phi tuyến thứ hai sau khi bổ sung (xem Hình 2).

Các kết nối tắt trong biểu thức (1) không đưa ra tham số phụ cũng như độ phức tạp tính toán. Điều này không chỉ hấp dẫn trong thực tế mà còn quan trọng trong việc so sánh giữa mạng lưới thông thường và mạng dư. Chúng ta hoàn toàn có thể so sánh các mạng thường / dư đồng thời có cùng số lượng tham số, độ sâu, chiều rộng và chi phí tính toán (ngoại trừ bổ sung phần tử không đáng kể).

Kích thước của x và F phải bằng nhau trong biểu thức (1). Nếu đây không phải là trường hợp (ví dụ: khi thay đổi kênh đầu vào / đầu ra), chúng ta có thể thực hiện phép chiếu tuyến tính Ws bằng các kết nối tắt để khớp với kích thước:

- $y = F(x, \{W_i\}) + W_S x$ (2)

Chúng ta cũng có thể sử dụng ma trận vuông W_S trong biểu thức (1). Nhưng chúng tôi sẽ chỉ ra bằng các thí nghiệm rằng ánh xạ định danh là đủ để giải quyết vấn đề xuống cấp và là kinh tế, và do đó W_S chỉ được sử dụng khi khớp kích thước.

Dạng của hàm dư F là linh hoạt. Các thí nghiệm trong bài viết này liên quan đến một chức năng F có hai hoặc ba lớp (Hình 5), trong khi nhiều lớp hơn là có thể. Nhưng nếu F chỉ có một lớp duy nhất, biểu thức (1) tương tự như một lớp tuyến tính: $y = W_1 x + x$, mà chúng ta chưa quan sát thấy lợi thế.

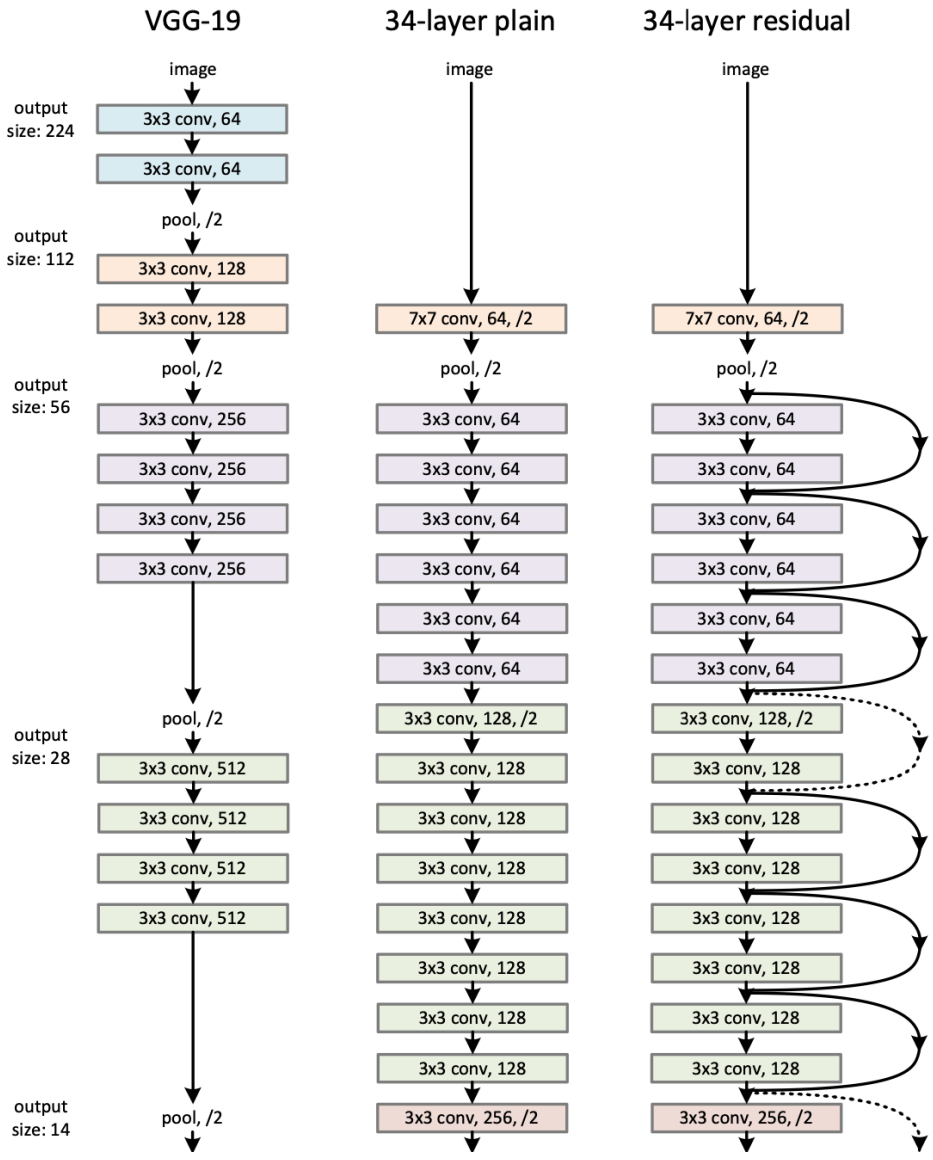
Chúng tôi cũng lưu ý rằng mặc dù các ký hiệu trên là về các lớp được kết nối đầy đủ để đơn giản, nhưng chúng có thể áp dụng cho các lớp chập. Hàm F (x, $\{W_i\}$) có thể gửi lại nhiều lớp chập. Phần bổ sung phần tử được thực hiện trên hai bản đồ tính năng, theo từng kênh.

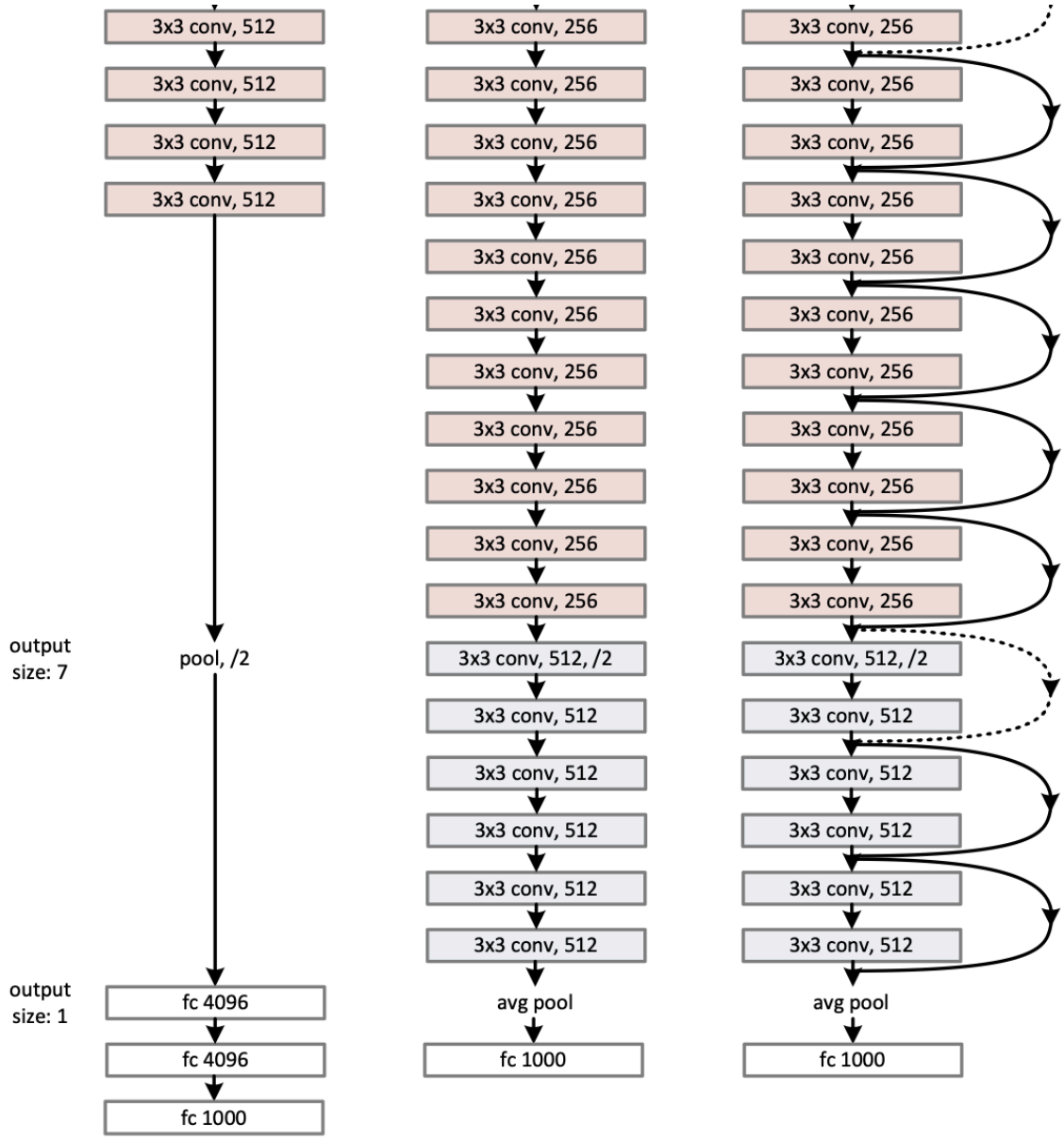
2.3 Kiến trúc mạng (Network Architectures)

Chúng tôi đã thử nghiệm các mạng thường / dư khác nhau và đã quan sát thấy các hiện tượng nhất quán. Để cung cấp các trường hợp để thảo luận, chúng tôi mô tả hai mô hình cho ImageNet như sau.

Plain Network (Mạng thông thường): Đường cơ sở đơn giản của chúng tôi (Hình 3, giữa) chủ yếu được lấy cảm hứng từ triết lý của lưới VGG [40] (Hình 3, bên trái). Các lớp chập hầu hết có các bộ lọc 3 × 3 và tuân theo hai quy tắc thiết kế đơn giản: (i) cho cùng kích thước feature map đầu ra, các lớp có cùng số lượng bộ lọc; và (ii) nếu kích thước bản đồ tính năng giảm một nửa, số lượng bộ lọc được nhân đôi để duy trì độ chính xác thời gian cho mỗi lớp. Chúng tôi thực hiện lấy mẫu trực tiếp bằng các lớp chập có bước tiến là 2. Mạng kết thúc với lớp gộp trung bình toàn cầu và lớp kết nối đầy đủ 1000 chiều với softmax. Tổng số lớp có trọng số là 34 trong Hình 3 (giữa).

Điều đáng chú ý là mô hình của chúng tôi có ít bộ lọc hơn và độ phức tạp thấp hơn so với lưới VGG [40] (Hình 3, bên trái). Đường cơ sở 34 lớp của chúng tôi có 3,6 tỷ FLOP (nhân thêm), chỉ bằng 18% VGG-19 (19,6 tỷ FLOP).


VGG-19
34-layer plain
34-layer residual
image
output size: 224
3x3 conv, 64
3x3 conv, 64
pool, /2
output size: 112
3x3 conv, 128
3x3 conv, 128
pool, /2
output size: 56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
output size: 28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 14
7x7 conv, 64, /2
3x3 conv, 128, /2
3x3 conv, 256, /2



Hình 3: Ví dụ kiến trúc mạng cho ImageNet

Residual network (Mạng dư): Dựa trên mạng đơn giản ở trên, chúng tôi chèn các kết nối tắt (Hình 3, bên phải) để biến mạng thành phiên bản còn lại của đối tác. Các lối tắt nhận dạng (Biểu thức (1)) có thể được sử dụng trực tiếp khi đầu vào và đầu ra có cùng kích thước (các lối tắt đường liền mạch trong Hình. 3). Khi tăng kích thước (các lối tắt

đường chấm trong Hình 3), chúng tôi xem xét hai tùy chọn: (A) Lối tắt vẫn thực hiện ánh xạ định danh, với các mục nhập 0 bổ sung được thêm vào để tăng kích thước. Tùy chọn này giới thiệu không có tham số phụ; (B) Lối tắt chiếu trong Eqn. (2) được sử dụng để khớp với kích thước (được thực hiện bằng các kết cấu 1 × 1). Đối với cả hai tùy chọn, khi các lối tắt đi qua các bản đồ đặc trưng có hai kích cỡ, chúng được thực hiện với sải chân là 2.

CHƯƠNG III – HIỆN THỰC

Việc triển khai ImageNet của chúng tôi tuân theo thực tiễn trong [21, 40]. Hình ảnh được thay đổi kích thước với mặt ngắn hơn được lấy mẫu thường xuyên trong [256, 480] để tăng tỷ lệ [40]. Một hình cắt có kích thước 224 × 224 được lấy mẫu ngẫu nhiên từ một hình ảnh hoặc lật ngang của nó, với trung bình mỗi pixel bị trừ [21]. Việc tăng màu tiêu chuẩn trong [21] được sử dụng. Chúng tôi áp dụng chuẩn hóa hàng loạt (BN) [16] ngay sau mỗi lần tích chập và trước khi kích hoạt, sau [16]. Chúng tôi khởi tạo các trọng số như trong [12] và huấn luyện tất cả các lưới đơn giản / dư từ đầu. Chúng tôi sử dụng SGD với kích thước lô nhỏ là 256. Tốc độ học bắt đầu từ 0.1 và được chia cho 10 khi các cao nguyên lỗi và các mô hình được đào tạo cho số lần lặp lên tới 60 × 104. Chúng tôi sử dụng phân rã trọng lượng 0,0001 và động lượng 0,9. Chúng tôi không sử dụng bỏ học [13], theo thông lệ trong [16].

Trong thử nghiệm, để so sánh các nghiên cứu, chúng tôi áp dụng thử nghiệm 10 vụ tiêu chuẩn [21]. Để có kết quả tốt nhất, chúng tôi áp dụng hình thức tích chập hoàn toàn như trong [40, 12] và tính trung bình điểm số ở nhiều thang đo (hình ảnh được thay đổi kích thước sao cho cạnh ngắn hơn nằm trong {224, 256, 384, 480, 640}).

CHƯƠNG IV – THỬ NGHIỆM

4.1 Phân loại ImageNet

Chúng tôi đánh giá phương pháp của chúng tôi trên bộ dữ liệu phân loại ImageNet 2012 [35] bao gồm 1000 lớp. Các mô hình được đào tạo trên 1.28 triệu hình ảnh đào tạo và được đánh giá dựa trên hình ảnh xác thực 50 nghìn. Chúng tôi cũng có được kết quả cuối cùng trên các hình ảnh thử nghiệm 100 nghìn, được báo cáo bởi máy chủ thử nghiệm. Chúng tôi đánh giá cả tỷ lệ lỗi top-1 và top-5.

Plain Network: Trước tiên chúng tôi đánh giá lưới thông thường 18 lớp và 34 lớp. Lưới thông thường 34 lớp nằm trong Hình 3 (giữa). Lưới thường 18 lớp có dạng tương tự. Xem Bảng dưới để biết các kiến trúc cụ thể.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

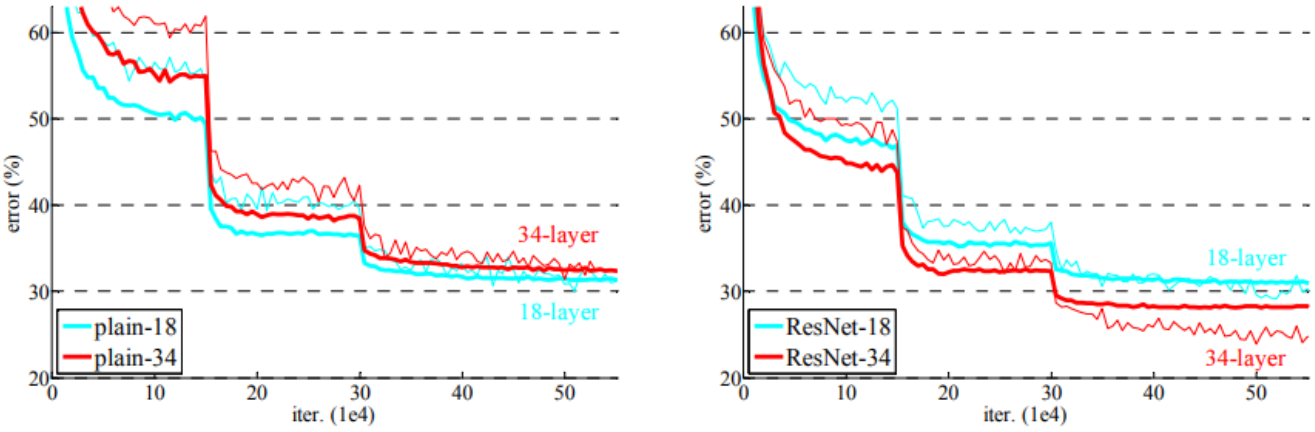
Bảng 1: Kiến trúc cho ImageNet

Kết quả trong Bảng 2 cho thấy, mạng lưới phẳng 34 lớp sâu hơn có lỗi xác nhận cao hơn so với mạng lưới phẳng 18 lớp nông hơn.

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Bảng 2: Tỉ lệ lỗi top-1 với mạng 18 và 34 lớp trên ImageNet

Để tiết lộ lý do, trong Hình 4 (bên trái), chúng tôi tìm ra các lỗi đào tạo / xác nhận của họ trong quá trình đào tạo. Chúng tôi đã quan sát vấn đề xuống cấp - mạng phẳng 34 lớp có lỗi đào tạo cao hơn trong toàn bộ quy trình đào tạo, mặc dù không gian giải pháp của mạng thông thường 18 lớp là không gian con của mạng 34 lớp.



Hình 4: Huấn luyện trên ImageNet

Chúng tôi lập luận rằng khó khăn tối ưu hóa này khó có thể xảy ra do biến mất độ dốc. Các mạng đơn giản này được đào tạo với BN [16], đảm bảo các tín hiệu lan truyền về phía trước có phương sai khác không. Chúng tôi cũng xác minh rằng các gradient lan truyền lạc hậu thể hiện các chỉ tiêu lành mạnh với BN. Vì vậy, không có tín hiệu về phía

trước cũng không biến mất. Trên thực tế, lưới phẳng 34 lớp vẫn có thể đạt được độ chính xác cạnh tranh (Bảng 3), cho thấy rằng bộ giải hoạt động ở một mức độ nào đó. Chúng tôi phỏng đoán rằng lưới thông thường sâu có thể có tốc độ hội tụ thấp theo cấp số nhân, ảnh hưởng đến việc giảm lỗi đào tạo. Lý do cho những khó khăn tối ưu hóa như vậy sẽ được nghiên cứu trong tương lai.

Residual Network (Mạng dư): Tiếp theo, chúng tôi đánh giá lưới dư 18 lớp và 34 lớp (ResNets). Các kiến trúc cơ sở giống như các lưới đơn giản ở trên, hy vọng rằng một kết nối tắt được thêm vào mỗi cặp bộ lọc 3 × 3 như trong Hình 3 (bên phải). Trong so sánh đầu tiên (Bảng 2 và Hình 4 bên phải), chúng tôi sử dụng ánh xạ định danh cho tất cả các lối tắt và không đệm để tăng kích thước (tùy chọn A). Vì vậy, họ không có tham số bổ sung so với các đối tác đơn giản.

Chúng tôi có ba quan sát chính từ Bảng 2 và Hình 4. Đầu tiên, tình huống được đảo ngược với học tập còn lại - ResNet 34 lớp tốt hơn ResNet 18 lớp (bằng 2,8%). Quan trọng hơn, ResNet 34 lớp biểu hiện lỗi đào tạo thấp hơn đáng kể và có thể khái quát hóa với dữ liệu xác nhận. Điều này chỉ ra rằng vấn đề xuống cấp được giải quyết tốt trong cài đặt này và chúng tôi quản lý để đạt được mức tăng độ chính xác từ độ sâu tăng.

Thứ hai, so với đối tác đơn giản của nó, ResNet 34 lớp giảm 3,5% lỗi hàng đầu (Bảng 2), do lỗi đào tạo giảm thành công (Hình 4 bên phải so với bên trái). So sánh này xác minh tính hiệu quả của việc học còn lại trên các hệ thống cực kỳ sâu sắc.

model	top-1 err.	top-5 err.
VGG-16 [40]	28.07	9.33
GoogLeNet [43]	-	9.15
PReLU-net [12]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Bảng 3: Tỉ lệ lỗi xác thực ImageNet

Cuối cùng, chúng tôi cũng lưu ý rằng lưới thường / dư 18 lớp có độ chính xác tương đương (Bảng 2), nhưng ResNet 18 lớp hội tụ nhanh hơn (Hình 4 bên phải so với bên trái). Khi mạng không phải là quá sâu (18 lớp ở đây), bộ giải SGD hiện tại vẫn có thể tìm thấy các giải pháp tốt cho mạng đơn giản. Trong trường hợp này, ResNet giảm bớt tối ưu hóa bằng cách cung cấp tốc độ hội tụ nhanh hơn ở giai đoạn đầu.

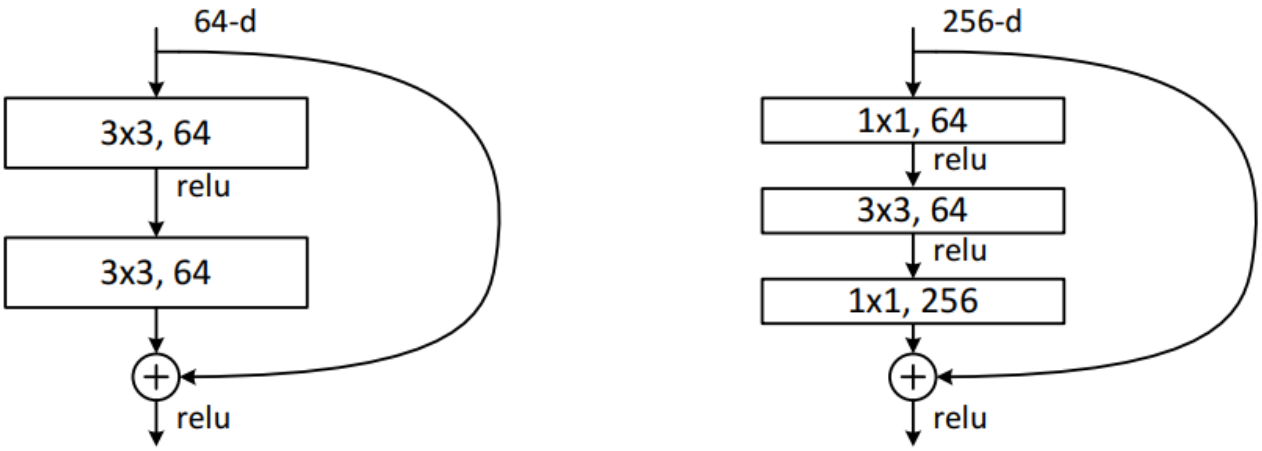
Identity vs. Projection Shortcuts: Chúng tôi đã chỉ ra rằng các lối tắt định danh, không có tham số giúp đào tạo. Tiếp theo, chúng tôi điều tra các lối tắt chiếu (Biểu thức (2)). Trong Bảng 3, chúng tôi so sánh ba tùy chọn: (A) các lối tắt không đệm được sử dụng để tăng kích thước và tất cả các lối tắt đều không có tham số (giống như Bảng 2 và

Hình 4 bên phải); (B) các lối tắt chiếu được sử dụng để tăng kích thước và các lối tắt khác là định danh; và (C) tất cả các lối tắt là hình chiếu.

Bảng 3 cho thấy rằng cả ba tùy chọn đều tốt hơn đáng kể so với đối tác đơn giản. B tốt hơn một chút so với A. Chúng tôi lập luận rằng điều này là do kích thước không đệm trong A thực sự không có học tập dư. C tốt hơn một chút so với B và chúng tôi gán thuộc tính này cho các tham số bổ sung được giới thiệu bởi nhiều (13) lối tắt chiếu. Nhưng sự khác biệt nhỏ giữa A / B / C chỉ ra rằng các lối tắt chiếu không cần thiết để giải quyết vấn đề xuống cấp. Vì vậy, chúng tôi không sử dụng tùy chọn C trong phần còn lại của bài viết này, để giảm độ phức tạp của bộ nhớ / thời gian và kích thước mô hình. Các lối tắt định danh đặc biệt quan trọng để không làm tăng sự phức tạp của các kiến trúc nút cổ chai được giới thiệu ở dưới.

Deeper Bottleneck Architectures (Kiến trúc cổ chai sâu hơn): Tiếp theo, chúng tôi mô tả các mạng sâu hơn của chúng tôi cho ImageNet. Do lo ngại về thời gian đào tạo mà chúng tôi có thể đủ khả năng, chúng tôi sửa đổi khối xây dựng thành một thiết kế tắc nghẽn. Đối với mỗi hàm dư F, chúng tôi sử dụng một ngăn xếp gồm 3 lớp thay vì 2 (Hình 5). Ba lớp là các kết cấu 1 × 1, 3 × 3 và 1 × 1, trong đó các lớp 1 × 1 chịu trách nhiệm giảm và sau đó tăng (khôi phục) kích thước, khiến cho lớp 3 × 3 bị tắc nghẽn với kích thước đầu vào / đầu ra nhỏ hơn . Hình. 5 cho thấy một ví dụ, trong đó cả hai thiết kế có độ phức tạp thời gian tương tự nhau.

Các lối tắt nhận dạng không có tham số đặc biệt quan trọng đối với các kiến trúc nút cổ chai. Nếu lối tắt nhận dạng trong Hình 5 (bên phải) được thay thế bằng phép chiếu, người ta có thể chỉ ra rằng độ phức tạp thời gian và kích thước mô hình được nhân đôi, vì lối tắt được kết nối với hai đầu chiều cao. Vì vậy, các lối tắt nhận dạng dẫn đến các mô hình hiệu quả hơn cho các thiết kế nút cổ chai.



Hình 5: Kiến trúc nút cổ chai

ResNet 50 lớp: Chúng tôi thay thế mỗi khối 2 lớp trong mạng 34 lớp bằng khối nút cổ chai 3 lớp này, dẫn đến ResNet 50 lớp (Bảng 1). Chúng tôi sử dụng tùy chọn B để tăng kích thước. Mô hình này có 3.8 tỷ FLOP.

ResNets 101 lớp và 152 lớp: Chúng tôi xây dựng ResNets 101 lớp và 152 lớp bằng cách sử dụng nhiều khối 3 lớp hơn (Bảng 1). Đáng chú ý, mặc dù độ sâu được tăng lên đáng kể, ResNet 152 lớp (11.3 tỷ FLOP) vẫn có độ phức tạp thấp hơn lưới VGG-16/19 (15,3 / 19,6 tỷ FLOP).

Các ResNets 50/101/152 lớp chính xác hơn các lớp 34 lớp theo tỷ lệ lợi nhuận đáng kể (Bảng 3 và 4). Chúng tôi không quan sát vấn đề xuống cấp và do đó tận hưởng độ chính xác đáng kể từ độ sâu tăng đáng kể. Lợi ích của độ sâu được chứng kiến cho tất cả các số liệu đánh giá (Bảng 3 và 4).

method	top-1 err.	top-5 err.
VGG [40] (ILSVRC'14)	-	8.43†
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Bảng 4: Tỉ lệ lỗi trên mô hình duy nhất của tập xác thực ImageNet

So sánh với các Phương pháp hiện đại: Trong Bảng 4, chúng tôi so sánh với các kết quả mô hình đơn tốt nhất trước đó. ResNets 34 lớp cơ bản của chúng tôi đã đạt được độ chính xác rất cạnh tranh. ResNet gồm 152 lớp của chúng tôi có lỗi xác thực top 5 mô hình duy nhất là 4,49%. Kết quả mô hình duy nhất này vượt trội hơn tất cả các kết quả tập hợp trước đó (Bảng 5). Chúng tôi kết hợp sáu mô hình có độ sâu khác nhau để tạo thành một nhóm (chỉ với hai mô hình 152 lớp tại thời điểm nộp). Điều này dẫn đến 3,57% lỗi hàng đầu trong bộ thử nghiệm (Bảng 5). Mục này đã giành được vị trí số 1 trong ILSVRC 2015.

method	top-5 err. (**test**)
VGG [40] (ILSVRC'14)	7.32
GoogLeNet [43] (ILSVRC'14)	6.66
VGG [40] (v5)	6.8
PReLU-net [12]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Bảng 5: Tỉ lệ lỗi top-5 trên bộ kiểm tra ImageNet

4.2 CIFAR-10 và Phân tích

Chúng tôi đã tiến hành nhiều nghiên cứu hơn về bộ dữ liệu CIFAR-10 [20], bao gồm 50k hình ảnh đào tạo và 10k hình ảnh thử nghiệm trong 10 lớp. Chúng tôi trình bày các thí nghiệm được đào tạo trên tập huấn luyện và đánh giá trên tập kiểm tra. Chúng tôi tập trung vào các hành vi của các mạng cực kỳ sâu sắc, nhưng không thúc đẩy các kết quả hiện đại, vì vậy chúng tôi cố tình sử dụng các kiến trúc đơn giản như sau.

Các kiến trúc đơn giản / dư theo mẫu trong Hình 3 (giữa / phải). Đầu vào mạng là hình ảnh 32 × 32, bị trừ trung bình trên mỗi pixel. Lớp đầu tiên là 3 × 3 kết cấu. Sau đó, chúng tôi sử dụng một chồng gồm 6n lớp với độ phân giải 3 × 3 trên các bản đồ đặc trưng có kích thước {32, 16, 8}, với 2n lớp cho mỗi kích thước bản đồ tính năng. Số lượng bộ lọc tương ứng là {16, 32, 64}. Việc lấy mẫu con được thực hiện bằng các kết quả với bước tiến là 2. Mạng kết thúc với nhóm trung bình toàn cầu, lớp được kết nối đầy đủ 10 chiều và softmax. Có hoàn toàn 6n + 2 lớp có trọng số xếp chồng lên nhau. Bảng sau đây tóm tắt kiến trúc:

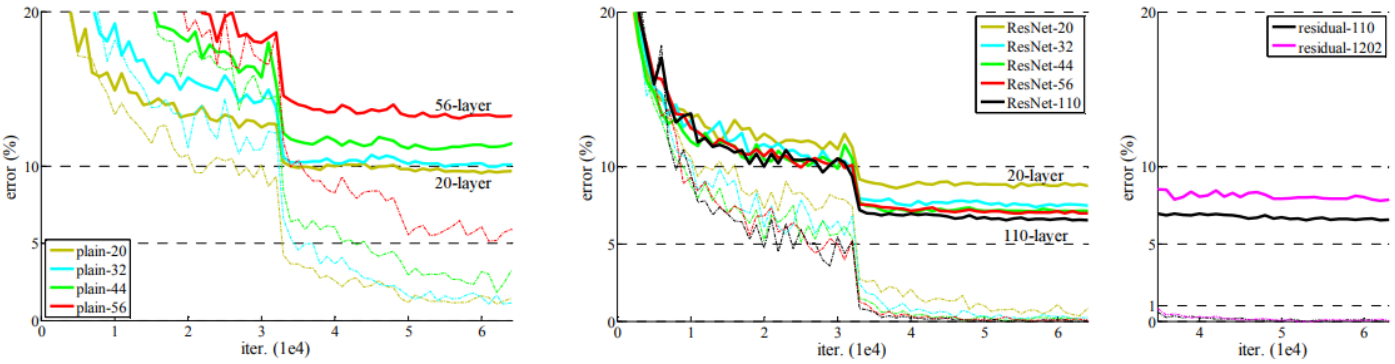
output map size	32×32	16×16	8×8
# layers	1+2n	2n	2n
# filters	16	32	64

Khi các kết nối tắt được sử dụng, chúng được kết nối với các cặp lớp 3 × 3 (hoàn toàn là 3n lối tắt). Trên tập dữ liệu này, chúng tôi sử dụng các lối tắt định danh trong tất cả các trường hợp (nghĩa là tùy chọn A) để các mô hình còn lại của chúng tôi có cùng độ sâu, chiều rộng và số lượng tham số chính xác như các đối tác đơn giản.

Chúng tôi sử dụng phân rã trọng lượng 0,0001 và động lượng 0,9 và áp dụng khởi tạo trọng lượng trong [12] và BN [16] nhưng không giảm. Những mô hình này được đào tạo với kích thước minibatch nhỏ 128 trên hai GPU. Chúng tôi bắt đầu với tỷ lệ học tập là 0,1, chia cho 10 lần lặp 32k và 48 nghìn, và chấm dứt đào tạo ở mức lặp 64k, được xác định trên mức chia 45k / 5k. Chúng tôi thực hiện theo cách tăng dữ liệu đơn giản trong [24] để đào tạo: 4 pixel được đệm ở mỗi bên và một cây trồng 32 × 32 được lấy mẫu ngẫu nhiên từ hình ảnh được đệm hoặc lật ngang của nó. Để thử nghiệm, chúng tôi chỉ đánh giá chế độ xem duy nhất của hình ảnh 32 × 32 ban đầu.

Chúng tôi so sánh n = {3, 5, 7, 9}, dẫn đến các mạng 20, 32, 44 và 56 lớp. Hình 6 (bên trái) cho thấy các hành vi của lưới trơn. Các lưới thông thường sâu chịu đựng độ sâu tăng, và biểu hiện lỗi đào tạo cao hơn khi đi sâu hơn. Hiện tượng này tương tự như trên ImageNet (Hình 4, bên trái) và trên MNIST (xem [41]), cho thấy rằng một khó khăn tối ưu hóa như vậy là một vấn đề cơ bản.

Hình 6 (giữa) cho thấy các hành vi của ResNets. Cũng tương tự như các trường hợp ImageNet (Hình 4, bên phải), ResNets của chúng tôi quản lý để khắc phục khó khăn tối ưu hóa và chứng minh độ chính xác tăng khi độ sâu tăng.



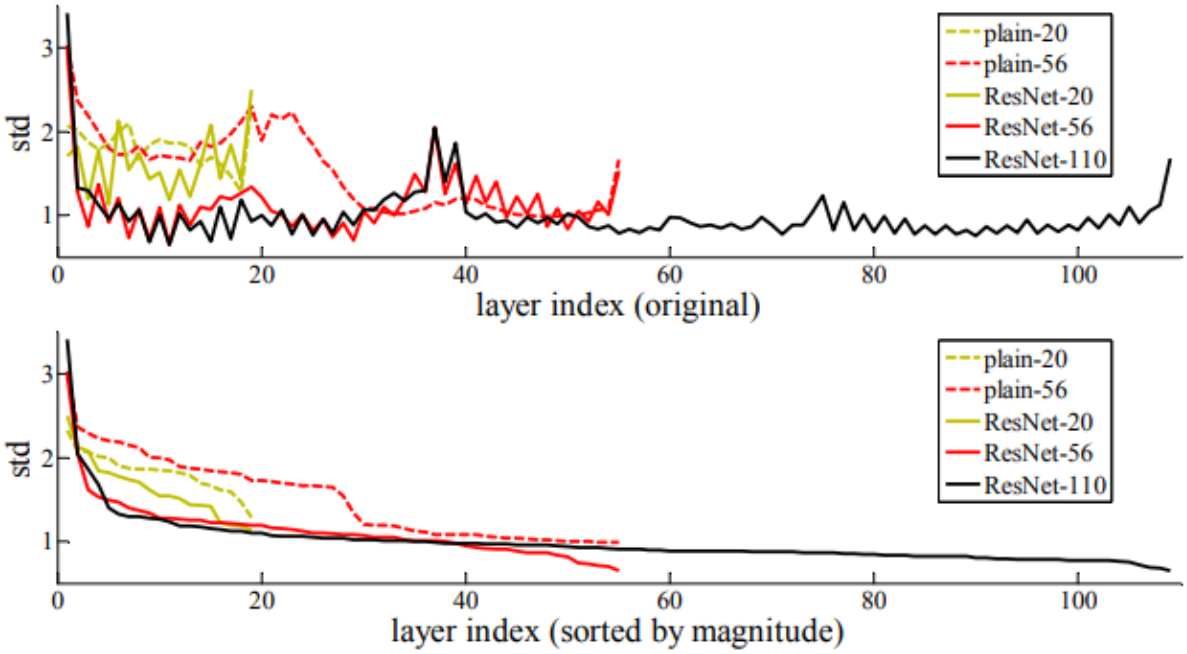
Hình 6: Huấn luyện trên CIFAR-10

Chúng tôi tiếp tục khám phá n = 18 dẫn đến ResNet 110 lớp. Trong trường hợp này, chúng tôi thấy rằng tốc độ học tập ban đầu là 0,1 hơi quá lớn để bắt đầu hội tụ. Vì vậy, chúng tôi sử dụng 0,01 để làm nóng việc đào tạo cho đến khi sai số đào tạo dưới 80% (khoảng 400 lần lặp), sau đó quay lại 0,1 và tiếp tục đào tạo. Phần còn lại của lịch trình học tập là như được thực hiện trước đó. Mạng 110 lớp này hội tụ tốt (Hình 6, giữa). Nó có ít tham số hơn các mạng sâu và mỏng khác như FitNet [34] và Highway [41] (Bảng 6), tuy nhiên là một trong những kết quả tiên tiến nhất (6,43%, Bảng 6).

method			error (%)
Maxout [9]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers	# params	
FitNet [34]	19	2.5M	8.39
Highway [41, 42]	19	2.3M	7.54 (7.72±0.16)
Highway [41, 42]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

Bảng 6: Lỗi phân loại trên tập kiểm tra CIFAR-10

Phân tích các phản hồi của lớp: Hình 7 cho thấy độ lệch chuẩn (std) của các phản ứng lớp. Các câu trả lời là kết quả đầu ra của mỗi lớp 3 × 3, sau BN và trước các phi tuyến khác (ReLU / phép cộng). Đối với ResNets, phân tích này cho thấy cường độ đáp ứng của các hàm dư. Hình 7 cho thấy ResNets thường có phản hồi nhỏ hơn so với các đối tác đơn giản của chúng. Những kết quả hỗ trợ động lực cơ bản của chúng tôi (Sec.3.1) rằng các chức năng còn lại có thể là thường gần gũi hơn với zero hơn các chức năng không còn sót lại. Chúng tôi cũng nhận thấy rằng ResNet sâu hơn có cường độ phản hồi nhỏ hơn, bằng chứng là sự so sánh giữa ResNet-20, 56 và 110 trong Hình 7. Khi có nhiều lớp hơn, một lớp ResNets riêng lẻ có xu hướng thay đổi tín hiệu ít hơn.



Hình 7: Độ lệch chuẩn (std) các lớp phản hồi trên CIFAR-10

Khám phá hơn 1000 lớp. Chúng tôi khám phá một mô hình sâu tích cực của hơn 1000 lớp. Chúng tôi đặt n = 200 dẫn đến mạng 1202 lớp, được đào tạo như mô tả ở trên. Phương pháp của chúng tôi cho thấy không có khó khăn tối ưu hóa và mạng 103 người chơi này có thể đạt được lỗi đào tạo <0,1% (Hình 6, phải). Lỗi kiểm tra của nó vẫn khá tốt (7,93%, Bảng 6).

Nhưng vẫn còn những vấn đề mở trên các mô hình sâu tích cực như vậy. Kết quả thử nghiệm của mạng 1202 lớp này kém hơn so với mạng 110 lớp của chúng tôi, mặc dù cả hai đều có lỗi đào tạo tương tự. Chúng tôi cho rằng điều này là do quá mức. Mạng 1202 lớp có thể lớn không cần thiết (19,4M) cho bộ dữ liệu nhỏ này. Chuẩn hóa mạnh mẽ như maxout [9] hoặc bỏ học [13] được áp dụng để thu được kết quả tốt nhất ([9, 25, 24, 34]) trên tập dữ liệu này. Trong bài báo này, chúng tôi không sử dụng tối đa / bỏ học và chỉ đơn giản là áp dụng chính quy hóa thông qua các kiến trúc sâu và mỏng theo thiết kế, mà không làm sao lãng sự tập trung vào những khó khăn của tối ưu hóa. Nhưng kết

hợp với chính quy hóa mạnh hơn có thể cải thiện kết quả, mà chúng ta sẽ nghiên cứu trong tương lai.

4.3 Nhận diện đối tượng trên PASCAL và MS COCO

Phương pháp của chúng tôi có hiệu suất khái quát tốt trên các nhiệm vụ công nhận khác. Bảng 7 và 8 cho thấy kết quả cơ bản phát hiện đối tượng trên PASCAL VOC 2007 và 2012 [5] và COCO [26]. Chúng tôi áp dụng Faster R-CNN [32] làm phương pháp phát hiện. Ở đây chúng tôi quan tâm đến những cải tiến của việc thay thế VGG-16 [40] bằng ResNet-101. Việc thực hiện phát hiện (xem phụ lục) của việc sử dụng cả hai mô hình là như nhau, vì vậy mức tăng chỉ có thể được quy cho các mạng tốt hơn. Đáng chú ý nhất, trên bộ dữ liệu COCO đầy thách thức, chúng tôi có được mức tăng 6.0% trong số liệu tiêu chuẩn COCO (mAP @ [. 5, .95]), cải thiện tương đối 28%. Việc đạt được này chỉ là do các đại diện đã học.

training data	07+12	07++12
test data	VOC 07 test	VOC 12 test
VGG-16	73.2	70.4
ResNet-101	**76.4**	**73.8**

Bảng 7: Nhận diện đối tượng mAP (%) trên PASCAL VOC

metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	**48.4**	**27.2**

Bảng 8: Nhận diện đối tượng mAP (%) trên COCO

Dựa trên các mạng còn lại sâu, chúng tôi đã giành được vị trí số 1 trong một số bài hát trong các cuộc thi ILSVRC & COCO 2015: Phát hiện ImageNet, bản địa hóa ImageNet, phát hiện COCO và phân đoạn COCO. Các chi tiết có trong phần phụ lục.

TÀI LIỆU THAM KHẢO

[1] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2):157–166, 1994.

[2] C. M. Bishop. Neural networks for pattern recognition. Oxford university press, 1995.

[3] W. L. Briggs, S. F. McCormick, et al. A Multigrid Tutorial. Siam, 2000.

[4] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In BMVC, 2011.

[5] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. IJCV, pages 303–338, 2010.

[6] R. Girshick. Fast R-CNN. In ICCV, 2015.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[8] X. Glorot and Y. Bengio. Understanding the difficulty of trainingdeep feedforward neural networks. In AISTATS, 2010.

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. arXiv:1302.4389, 2013.

[10] K. He and J. Sun. Convolutional neural networks at constrained timecost. In CVPR, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015.

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing coadaptation of feature detectors. arXiv:1207.0580, 2012.

[14] S. Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Diploma thesis, TU Munich, 1991.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.

[17] H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. TPAMI, 33, 2011.

[18] H. Jegou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. TPAMI, 2012.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093, 2014.

[20] A. Krizhevsky. Learning multiple layers of features from tiny images. Tech Report, 2009.

[21] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[22] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1989.

[23] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Muller. Efficient backprop. ¨ In Neural Networks: Tricks of the Trade, pages 9–50. Springer, 1998.

[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeplysupervised nets. arXiv:1409.5185, 2014.

[25] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv:1312.4400, 2013.

[26] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick. Microsoft COCO: Common objects in ´ context. In ECCV. 2014.

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[28] G. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of ´ linear regions of deep neural networks. In NIPS, 2014.

[29] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.

[30] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In CVPR, 2007.

[31] T. Raiko, H. Valpola, and Y. LeCun. Deep learning made easier by linear transformations in perceptrons. In AISTATS, 2012.

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.

[33] B. D. Ripley. Pattern recognition and neural networks. Cambridge university press, 1996.

[34] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In ICLR, 2015.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. arXiv:1409.0575, 2014.

[36] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120, 2013.

[37] N. N. Schraudolph. Accelerated gradient descent by factor-centering decomposition. Technical report, 1998.

[38] N. N. Schraudolph. Centering neural network gradient factors. In Neural Networks: Tricks of the Trade, pages 207–226. Springer, 1998.

[39] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014.

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[41] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. arXiv:1505.00387, 2015.

[42] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. 1507.06228, 2015.

[43] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In CVPR, 2015.

[44] R. Szeliski. Fast surface interpolation using hierarchical basis functions. TPAMI, 1990.

[45] R. Szeliski. Locally adapted hierarchical basis preconditioning. In SIGGRAPH, 2006.

[46] T. Vatanen, T. Raiko, H. Valpola, and Y. LeCun. Pushing stochastic gradient towards second-order methods–backpropagation learning with transformations in nonlinearities. In Neural Information Processing, 2013.

[47] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[48] W. Venables and B. Ripley. Modern applied statistics with s-plus. 1999.

[49] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In ECCV, 2014.