Bài 38 - Các kiến trúc CNN hiện đại

31 May 2020 - phamdinhkhanh

Menu

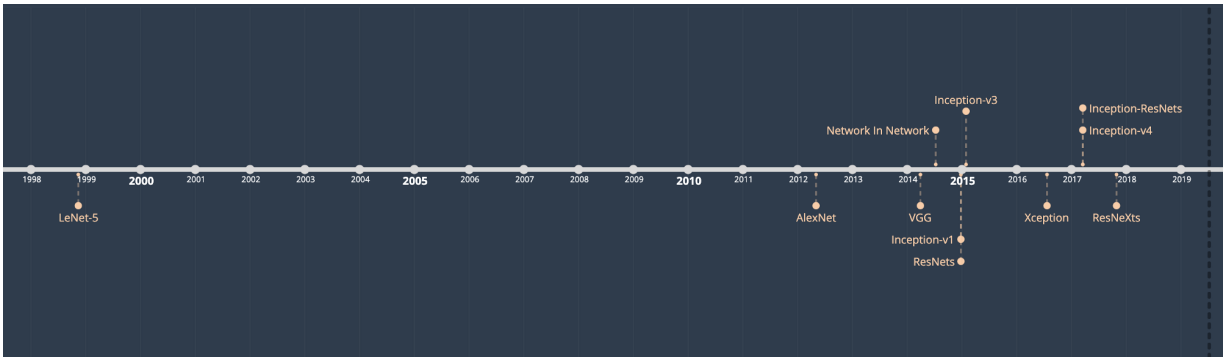
1. Tiến trình phát triển của CNN

1.1. Giới thiệu chung

Mạng CNN ra đời đã thúc đẩy quá trình phát triển của ngành computer vision. Hiện tại có rất nhiều các kiến trúc mạng CNN khác nhau và các kiến trúc mới vẫn đang tiếp tục được khám phá ngày qua ngày. Nhưng ít ai biết rằng đằng sau những khám phá đó là một tiến trình khoa học lâu dài và bền bỉ trong gần 20 năm. Với sự kết hợp đồng bộ của phát triển kiến trúc mạng, khả năng tính toán của máy tính và các phương pháp tối ưu hóa. Bài viết này mình sẽ giới thiệu tới các bạn lược sử hình thành của các kiến trúc CNN tiêu biểu và những đóng góp mang tính cải tiến của những kiến trúc mạng này so với trước đó. Thông qua bài viết bạn đọc sẽ hình dung được lộ trình hình thành và phát triển cho tới ngày nay của những mạng CNN và đồng thời hiểu rõ được đặc trưng trong kiến trúc của từng mạng. Những ưu nhược điểm và cải tiến đã thực hiện so với những kiến trúc mạng trước đó. Trước khi bắt đầu bài này, mình khuyến nghị các bạn hãy đọc qua Bài 8 - Convolutional Neural Network (https://phamdinhkhanh.github.io/2019/08/22/convolutional-neural-network.html) để hiểu rõ hơn về mạng CNN là gì? Sau khi đã nắm được các khái niệm về mạng CNN, chúng ta sẽ dễ dàng hình dung các kiến thức được trình bày tại bài viết này.

1.2. Các dấu mốc quan trọng

Top



Hình 1: Các cột mốc phát triển của mạng CNN. Source: Illustrated: 10 CNN Architectures - Raimi Karim (https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d)

Tiến trình phát triển của các kiến trúc CNN có thể được khái quát qua những dấu mốc quan trọng sau đây:

- 1998: Yan Lecun lần đầu tiên sử dụng mạng tích chập trong tác vụ phân loại chữ số viết tay và đạt hiệu quả cao. Tuy nhiên vào thời điểm đó do chưa có sự phát triển của dữ liệu và khả năng tính toán nên mạng CNN vẫn chưa có cơ hội bùng nổ. Các mô hình machine learning truyền thống như SVM, kNN vẫn được sử dụng phổ biến.
- 2009: Bộ dữ liệu ImageNet được giới thiệu vào năm 2009 là một trong những bộ dữ liệu tạo ra sự thay đổi trong cộng đồng computer vision. Đây là bộ dữ liệu lớn nhất so với các bộ dữ liệu từng có từ trước đến thời điểm đó. Với kích thước lên tới 1 triệu ảnh và phân bố đều trên 1000 nhãn. Các mô hình được huấn luyện trên ImageNet có thể chuyển giao tới rất nhiều những domain dữ liệu khác nhau. Kể từ thời điểm 2010, ImageNet trở thành tiêu chuẩn đo đạc sự phát triển của các thuật toán học có giám sát trong thị giác máy tính.
- 2012: Mạng AlexNet sử dụng tích chập CNN lần đầu tiên vượt qua các phương pháp tạo đặc trưng thủ công truyền thống như HOG, SHIFT và đạt độ chính xác cách biệt trong cuộc thi ImageNet. Dấu mốc đó đã khởi đầu cho xu hướng ứng dụng CNN trong computer vision thay thế cho những thuật toán học máy truyền thống trước kia.
- Liên tiếp vào những năm sau đó, ngày càng xuất hiện nhiều các kiến trúc CNN mới. Chúng được hình thành, phát triển và cải tiến về độ sâu, cách thiết kế block, cách kết nối giữa các block. Lần lượt từ VGG Net, GoogleNet, ResNet, DenseNet,… mà chúng ta sẽ tìm hiểu qua bài viết này được ra đời dựa trên sự kế thừa những ý tưởng cũ và phát triển những ý tưởng mới mẻ. Qúa trình phát triển của các kiến trúc mạng song hành cùng với sự phát triển phần cứng máy tính như các GPU có tốc độ nhanh hơn. Kỹ thuật huấn luyện phân tán và song song trên nhiều GPU cho phép một model huấn luyện chỉ trong vòng một vài tiếng so với việc huấn luyện kéo dài qua nhiều ngày và tốn kém như trước đây. Các framework hỗ trợ deep learning cũng xuất hiện nhiều hơn, được cải tiến và trở thành công cụ đáp ứng mọi nhu cầu cần thiết cho quá trình huấn luyện deep learning. Phổ biến nhất có thể kể tới ba frameworks *pytorch (facebook), tensorflow (google), mxnet (intel)* được phát triển và hậu thuẫn từ những công ty công nghệ hàng đầu thế giới. Kể từ sau ImageNet, các bộ dữ liệu ảnh đã khẳng định vai trò thúc đẩy sự phát triển của ngành AI. Các thuật toán được so sánh với nhau dựa trên kết quả dẫn đầu (*leader board*) từ những bộ dữ liệu chuẩn hoá. Nhờ sự mở rộng của những nền tảng huấn luyện free như google colab, kaggle mà mọi người đều có thể tiếp cận được với AI. Chiến lược phát triển toàn cầu về AI ở các tập đoàn, quốc gia trên thế giới dẫn tới sự hình thành những viện nghiên cứu về AI qui tụ được nhiều nhà khoa học xuất sắc và có những nghiên cứu đột phá.

2. Các pipeline trước CNN

Top

Trước thời điểm 2012, hầu hết các nhà nghiên cứu cho rằng phần quan trọng nhất của một pipeline là sự biểu diễn. SIFT (https://en.wikipedia.org/wiki/Scale-invariant_feature_transform), SURF (https://en.wikipedia.org/wiki/Speeded_up_robust_features), HOG (https://phamdinhkhanh.github.io/2019/11/22/HOG.html) là những phương pháp quan trích chọn đặc trưng thủ công, được áp dụng kết hợp với các thuật toán của machine learning như SVM, MLP, k-NN, Random Forest,….

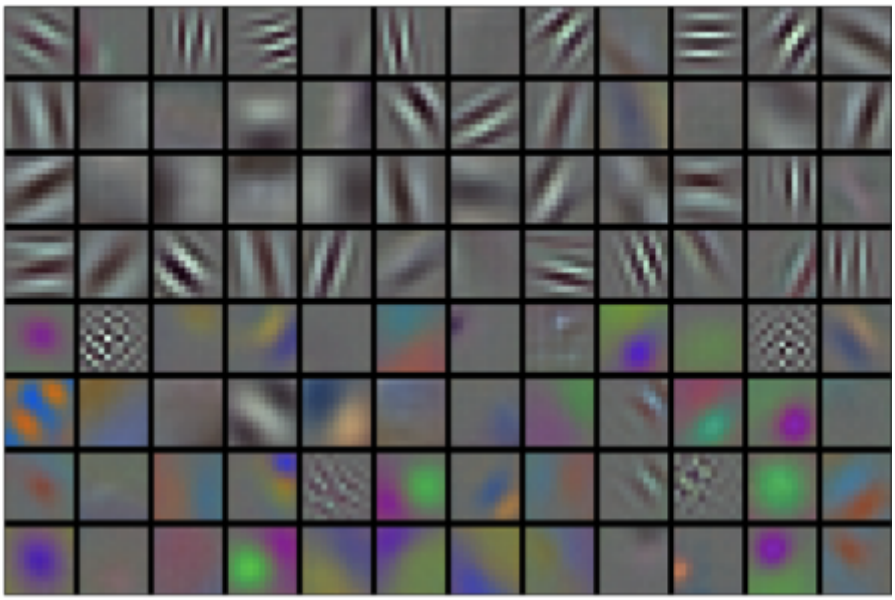
Bạn đọc có thể xem một ví dụ tạo đặc trưng trên HOG (https://phamdinhkhanh.github.io/2019/11/22/HOG.html#42-%E1%BB%A9ng-d%E1%BB%A5ng-trong-feature-engineering) cho các bài toán học có giám sát mà mình đã giới thiệu trước đó.

Đặc điểm của những kiến trúc này đó là:

- Các đặc trưng được tạo ra không có khả năng huấn luyện vì qui luật tạo ra chúng là cố định.
- Pipeline tách rời giữa feature extractors và classifier.

Một nhóm các nhà nghiên cứu đầu ngành cho rằng các đặc trưng là có thể học được thông qua mô hình và để có được sự phức tạp thì các đặc trưng nên được học phân tầng theo nhiều layer. Từ những đặc điểm chung, xuất hiện ở mọi bức ảnh như các đường nét dọc, ngang, chéo tới những đặc trưng riêng giúp nhận biết vật thể.

Một điều khá thú vị là ở các tầng thấp nhất của mạng CNN, mô hình đã học được cách trích xuất đặc trưng giống như các hàm trích lọc đặc trưng truyền thống như HOG, SHIFT.

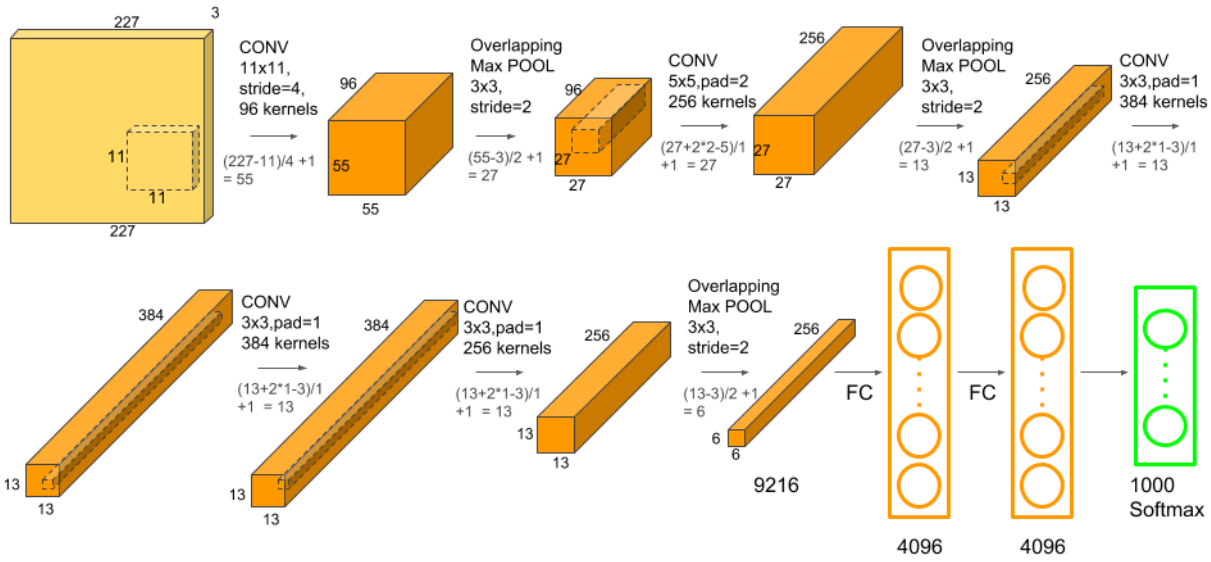


Hình 2: Các đặc trưng được trích xuất từ layer đầu tiên thông qua bộ lọc trong AlexNet.

Hướng nghiên cứu đó vẫn tiếp tục phát triển qua quá trình thử nghiệm các ý tưởng, thuật toán và kiến trúc mới. Đến thời điểm hiện tại đã có ngày càng nhiều các mô hình CNN được khai phá.

3. Đặc trưng chung của các mạng CNN

Top



Hình 3: Mạng Alexnet, một kiến trúc điển hình của CNN.

- Sử dụng tích chập: Các mạng CNN đều trích suất đặc trưng dựa trên nguyên lý tích chập. Bởi vậy tên gọi chung cho chúng là Convolutional Neural Network (Mạng nơ ron tích chập). Để hiểu hơn về nguyên lý tích chập, các bạn có thể tham khảo Bài 8 - Convolutional Neural Network (https://phamdinhkhanh.github.io/2019/08/22/convolutional-neural-network.html).
- Kiến trúc phân tầng: Kiến trúc phân tầng giúp mạng CNN học được đặc trưng ở những cấp độ khác nhau, từ cấp độ low-level (bậc thấp) tới high-level (bậc cao). Theo đó, mức độ chi tiết của hình ảnh cũng tăng tiến dần từ các đặc trưng chung như các đường chéo, ngang, dọc rìa, cạnh tới những các đặc trưng chi tiết hơn giúp phân biệt vật thể như bánh xe, cánh cửa, mui xe (nếu vật thể là xe), tất cả các chi tiết đó được tổng hợp lại và ở layer tích chập cuối cùng ta thu được hình ảnh của một chiếc xe. Để thực nghiệm visualize output của từng block trong mạng CNN bạn đọc có thể tham khảo How to Visualize Filters and Feature Maps in CNN - Machine Learning Mastery (https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/)
- Được huấn luyện trên những bộ dữ liệu lớn: Sẽ không có một sự khác biệt đáng kể giữa mô hình học sâu nhiều tầng và các phương pháp học máy truyền thống nếu chỉ huấn luyện trên một bộ dữ liệu rất nhỏ. Vì dữ liệu nhỏ chỉ cần một không gian biểu diễn nhỏ từ phương pháp truyền thống là đủ để phân biệt các nhãn với nhau. Nhưng trên các bộ dữ liệu lớn, kiến trúc học sâu nhiều tầng đã cho thấy ưu thế vượt trội về độ chính xác và khả năng biểu diễn. Điều này cũng dễ hiểu bởi kích thước mạng nơ ron có thể lên tới hàng chục triệu tham số và lớn hơn rất nhiều so với số lượng tham số của các phương pháp học máy truyền thống dẫn tới khả năng biểu diễn tốt hơn.
- Kích thước layers giảm dần: Hình 3 là kiến trúc của mạng AlexNet, một trong những kiến trúc CNN ta có thể thấy kích thước layers giảm dần theo độ sâu. Thông thường mức độ giảm lý tưởng là cấp số 2. Các nghiên cứu đã chỉ ra rằng việc kích thước layers giảm dần giúp giảm thiểu số lượng tham số của mô hình đáng kể và giúp tạo ra những mạng có kích thước nhẹ hơn và tốc độ dự báo nhanh hơn. Trong khi độ chính xác của mô hình giảm không đáng kể.
- Độ sâu tầng layers tăng dần: Độ sâu của các layers tăng dần nhờ tăng số bộ lọc, thường là theo cấp số nhân. Độ sâu tăng sẽ giúp cho mạng CNN học được đa dạng các đặc trưng hơn. Ở những layer đầu tiên là những đặc trưng chung, chúng khá giống nhau về hình

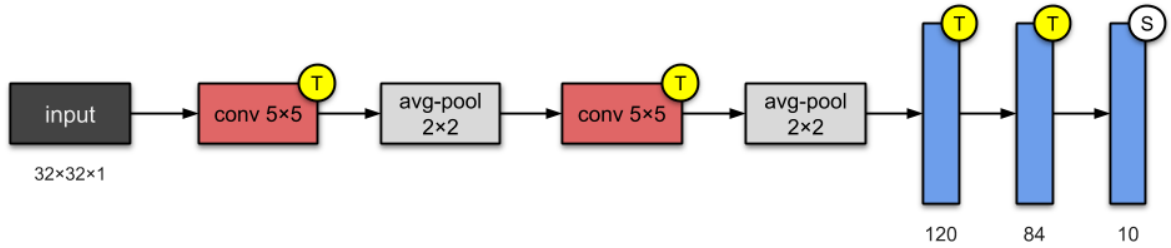
Top

dạng, phương hướng, nên không cần quá nhiều bộ lọc để tạo ra chúng với số lượng lớn. Càng ở những layers sau đòi hỏi độ chi tiết cao hơn thì yêu cầu số lượng bộ lọc nhiều hơn để giúp phân biệt được nhiều chi tiết đặc trưng hơn.

- Sử dụng các fully connected layers để phân loại: Tích chập từ mạng CNN sẽ tạo ra những đặc trưng 2 chiều. Để sử dụng những đặc trưng này vào quá trình phân loại của mạng CNN thì chúng ta phải chuyển chúng thành đặc trưng 1 chiều bằng phương pháp flatten và lan truyền thuận qua các fully connected layers. Đằng sau mỗi một layer là một hàm kích hoạt phi tuyến nhằm gia tăng khả năng biểu diễn giúp cho kết quả phân loại tốt hơn.

4. Các mạng CNN tiêu biểu

4.1. LeNet-5 (1998)



Hình 4: Kiến trúc LeNet. Source: Illustrated: 10 CNN Architectures - Raimi Karim (https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d)

Paper - LeNet-5 - A NN/HMM Hybrid for on-line HandWriting Recognition (http://yann.lecun.com/exdb/publis/pdf/bengio-95.pdf).

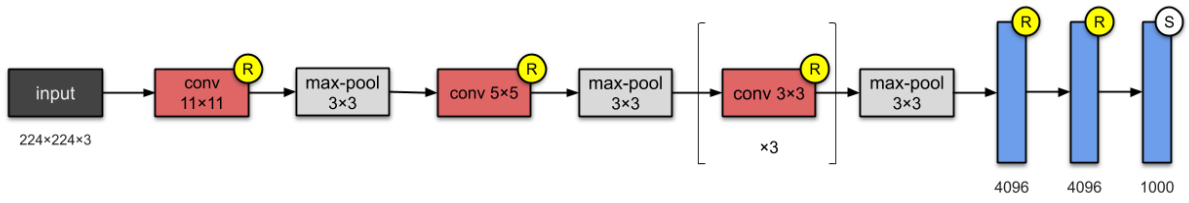
Authors: Yan Lecun, Yoshua Bengio

LeNet-5 là kiến trúc đầu tiên áp dụng mạng tích chập 2 chiều của giáo sư Yan Lecun, cha đẻ của kiến trúc CNN. Model ban đầu khá đơn giản và chỉ bao gồm 2 convolutional layers + 3 fully-connected layers. Mặc dù đơn giản nhưng nó có kết quả tốt hơn so với các thuật toán machine learning truyền thống khác trong phân loại chữ số viết tay như SVM, kNN.

Trong kiến trúc mạng nơ ron đầu tiên, để giảm chiều dữ liệu, Yan Lecun sử dụng Sub-Sampling Layer là một Average-Pooling Layer (các layer nhằm mục đích giảm chiều dữ liệu mà không thay đổi đặc trưng chúng ta còn gọi là Sub-Sampling Layer). Kiến trúc này khó hội tụ nên ngày nay chúng được thay thế bằng Max-Pooling.

Đầu vào của mạng LeNet có kích thước nhỏ (chỉ `32x32`) và ít layers nên số lượng tham số của nó chỉ khoảng 60 nghìn.

4.2. AlexNet (2012)



Top

Paper AlexNet - ImageNet Classification with Deep Convolutional Neural Networks (https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks)

Hình 5: Kiến trúc AlexNet. Source: Illustrated: 10 CNN Architectures - Raimi Karim (https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d)

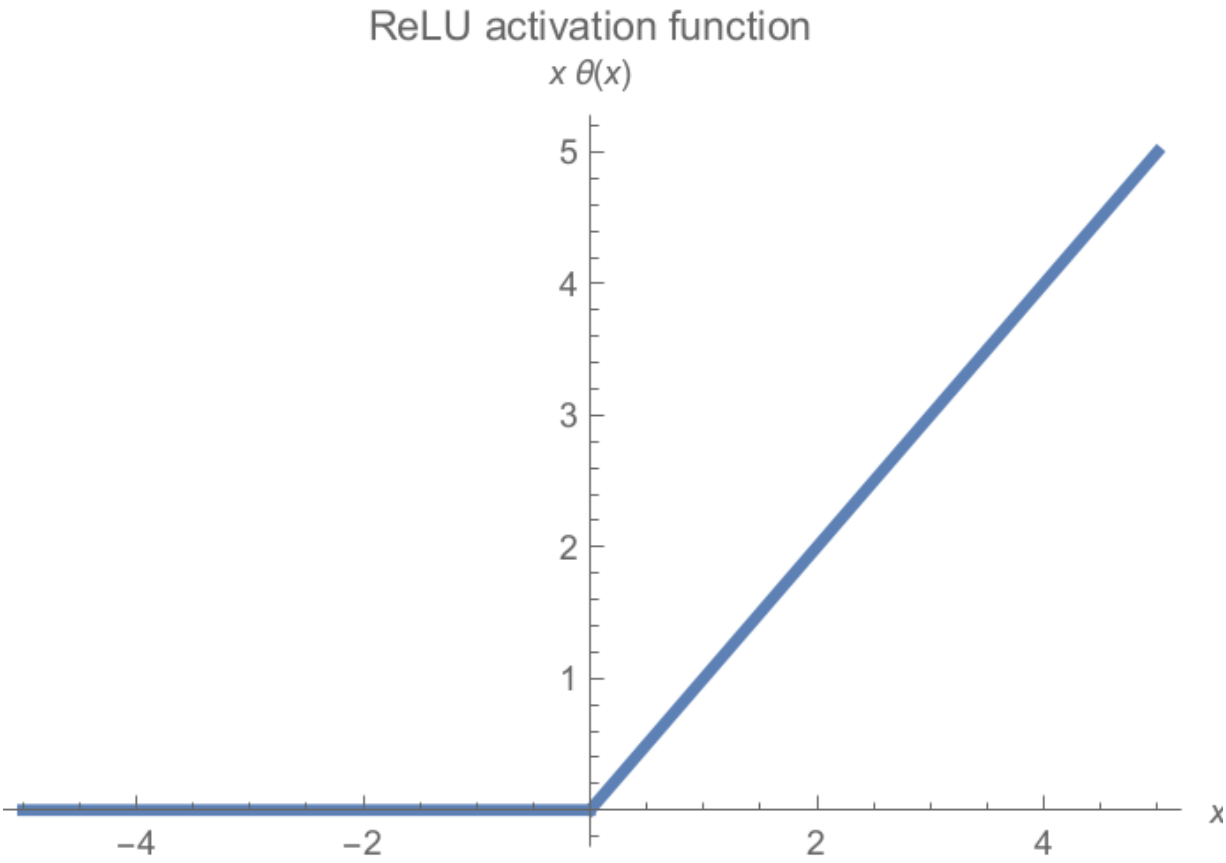
Authors: Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton. University of Toronto, Canada.

AlexNet là mạng CNN được giới thiệu vào năm 2012 bởi Alex Krizhevsky và dành chiến thắng trong cuộc thi ImageNet với cách biệt khá lớn so với vị trí thứ hai. Lần đầu tiên Alex net đã phá vỡ định kiến trước đó cho rằng các đặc trưng được học từ mô hình sẽ không tốt bằng các đặc trưng được tạo thủ công (thông qua các thuật toàn SUFT, HOG, SHIFT). Ý tưởng của AlexNet dựa trên LeNet của Yan Lecun và cải tiến ở các điểm:

- Tăng kích thước đầu vào và độ sâu của mạng.
- Sử dụng các bộ lọc (*kernel* hoặc *filter*) với kích thước giảm dần qua các layers để phù hợp với kích thước của đặc trưng chung và đặc trưng riêng.
- Sử dụng local normalization để chuẩn hóa các layer giúp cho quá trình hội tụ nhanh hơn.

Ngoài ra mạng còn cải tiến trong quá trình optimizer như:

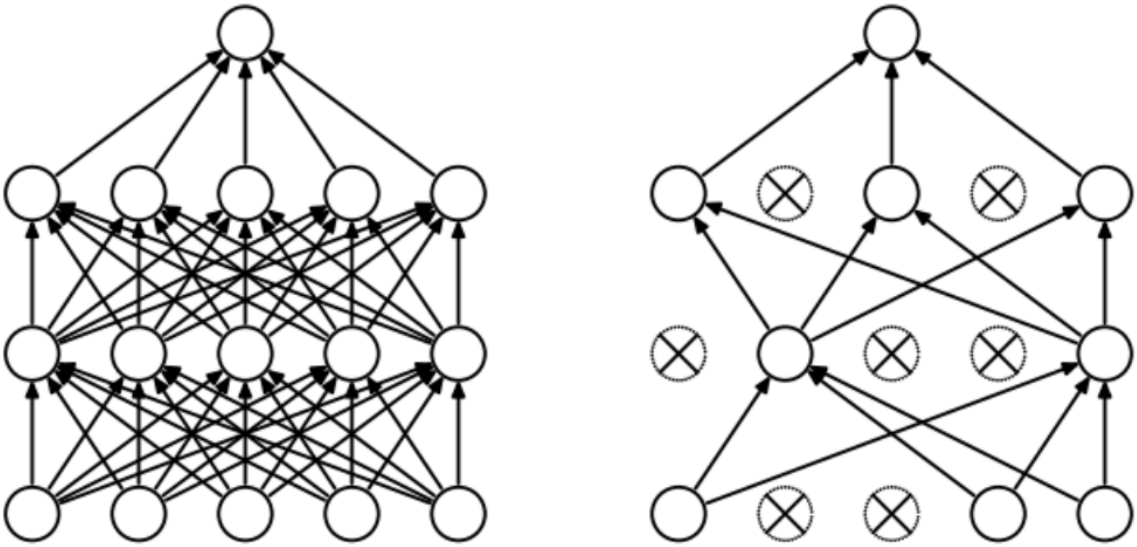
- Lần đầu tiên sử dụng activation là ReLU (Rectified Linear Unit) thay cho Sigmoid. ReLU là hàm có tốc độ tính toán nhanh nhờ đạo hàm chỉ có 2 giá trị {0, 1} và không có lũy thừa cơ số e như hàm sigmoid nhưng vẫn tạo ra được tính phi tuyến (`non-linear`).



Hình 6: Hàm ReLU công thức $\theta(x) = \max(0, x)$.

- Sử dụng dropout layer giúp giảm số lượng liên kết neural và kiểm soát overfitting.

Top

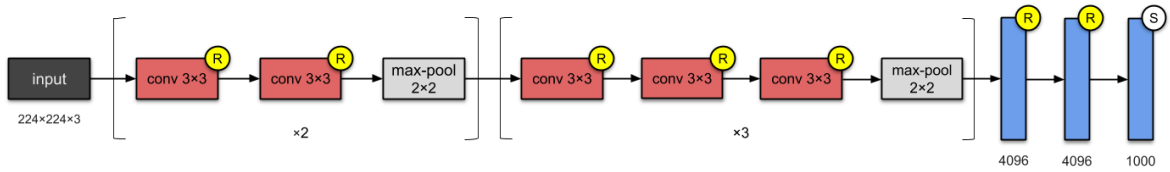


Hình 7: Phương pháp dropout có số lượng các liên kết mạng bị giảm so với trước đó làm mô hình ít phức tạp hơn. Đồng thời đây cũng là một dạng ensemble model giúp giảm thiểu được overfitting.

- Qua các layers, kích thước output giảm dần nhưng độ sâu tăng dần qua từng kernel.

Mạng AlexNet có resolution của input và số lượng layer lớn hơn nên số lượng tham số của nó lên tới 60 triệu, lớn hơn so với LeNet rất nhiều.

4.3. VGG-16 (2014)



Hình 8: Kiến trúc VGG-16. Source: Illustrated: 10 CNN Architectures - Raimi Karim (https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d)

Paper VGG-16 - Very Deep Convolutional Networks for Large-Scale Image Recognition (https://arxiv.org/abs/1409.1556)

Author: Karen Simonyan, Andrew Zisserman. University of Oxford, UK

Với VGG-16, quan điểm về một mạng nơ ron sâu hơn sẽ giúp ích cho cải thiện độ chính xác của mô hình tốt hơn. Về kiến trúc thì VGG-16 vẫn dữ các đặc điểm của AlexNet nhưng có những cải tiến:

- Kiến trúc VGG-16 sâu hơn, bao gồm 13 layers tích chập 2 chiều (thay vì 5 so với AlexNet) và 3 layers fully connected.
- Lần đầu tiên trong VGG-16 chúng ta xuất hiện khái niệm về khối tích chập (block). Đây là những kiến trúc gồm một tập hợp các layers CNN được lặp lại giống nhau. Kiến trúc khối đã khởi nguồn cho một dạng kiến trúc hình mẫu rất thường gặp ở các mạng CNN kể từ đó.
- VGG-16 cũng kế thừa lại hàm activation ReLU ở AlexNet.

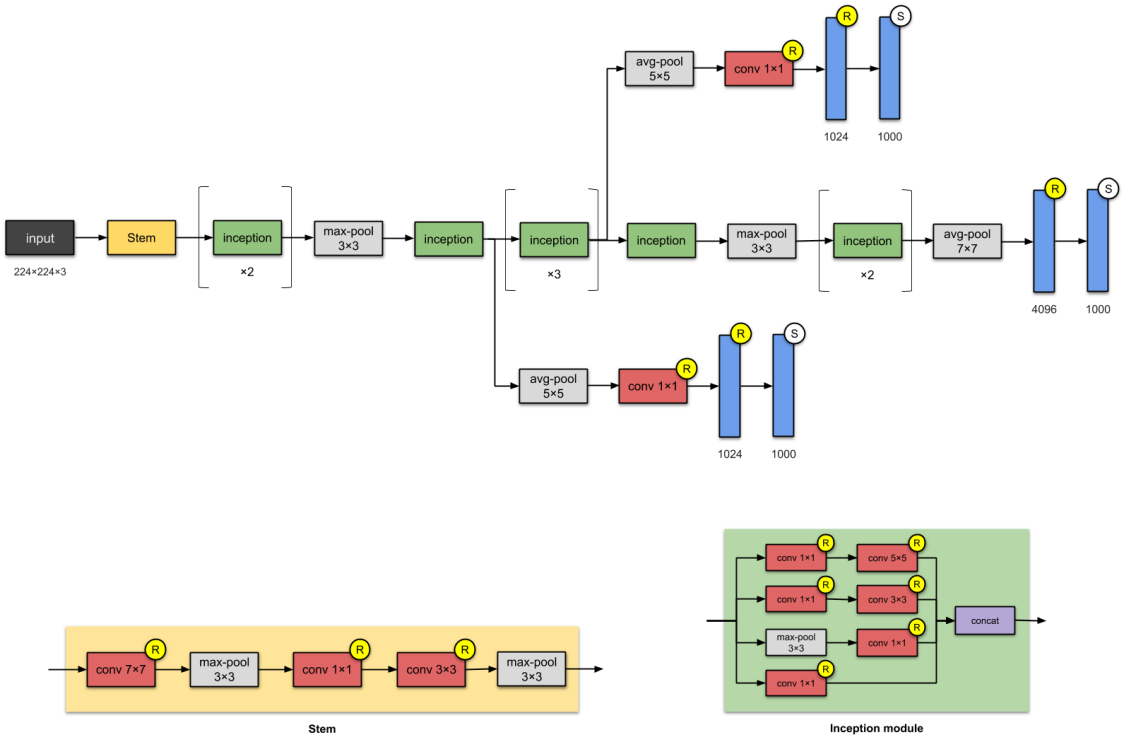
Top

- VGG-16 cũng là kiến trúc đầu tiên thay đổi thứ tự của các block khi xếp nhiều layers CNN + max pooling thay vì xen kẽ chỉ một layer CNN + max pooling. Một bạn có câu hỏi tại Forum Machine Learning Cơ Bản (https://www.facebook.com/groups/machinelearningcoban/?multi_permalinks=968538746936866) về sự thay đổi này sẽ giúp cho VGG net cải thiện hơn như thế nào? Các layers CNN sâu hơn có thể trích lọc đặc trưng tốt hơn so với chỉ 1 layers CNN.
- VGG-16 chỉ sử dụng các bộ lọc kích thước nhỏ `3x3` thay vì nhiều kích thước bộ lọc như AlexNet. Kích thước bộ lọc nhỏ sẽ giúp giảm số lượng tham số cho mô hình và mang lại hiệu quả tính toán hơn. VD: Nếu sử dụng 2 bộ lọc kích thước `3 x 3` trên một features map (là output của một layer CNN) có độ sâu là 3 thì ta sẽ cần `n_filters x kernel_size x kernel_size x n_channels = 2 x 3 x 3 x 3 = 54` tham số. Nhưng nếu sử dụng 1 bộ lọc kích thước `5 x 5` sẽ cần `5 x 5 x 3 = 75` tham số. 2 bộ lọc `3 x 3` vẫn mang lại hiệu quả hơn so với 1 bộ lọc `5 x 5`.

Mạng VGG-16 sâu hơn so với AlexNet và số lượng tham số của nó lên tới 138 triệu tham số. Đây là một trong những mạng mà có số lượng tham số lớn nhất. Kết quả của nó hiện đang xếp thứ 2 trên bộ dữ liệu ImageNet validation ở thời điểm public. Ngoài ra còn một phiên bản nữa của VGG-16 là VGG-19 tăng cường thêm 3 layers về độ sâu.

Bắt đầu từ VGG-16, một hình mẫu chung cho các mạng CNN trong các tác vụ học có giám sát trong xử lý ảnh đã bắt đầu hình thành đó là các mạng trở nên sâu hơn và sử dụng các block dạng `[Conv2D*n + Max Pooling]`.

4.4. GoogleNet - Inception-V1 (2014)



Hình 9: Kiến trúc GoogleNet - Inception version 1.

Paper Inception-V1 - Going Deeper with Convolutions (https://arxiv.org/abs/1409.4842)

Top

Authors: Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Google, University of Michigan, University of North Carolina

Mạng Inception-V1 đã dành chiến thắng ở cuộc thi ImageNet vào năm 2015. Kiến trúc này đã giải quyết một câu hỏi lớn trong mạng CNN đó là sử dụng `kernel_size` với kích thước bao nhiêu thì hợp lý. Các kiến trúc mạng nơ ron trước đó đều sử dụng các bộ lọc với đa dạng các kích thước `11x11`, `5x5`, `3x3` cho tới nhỏ nhất là `1x1`. Một khám phá được đưa ra bởi bài báo đó là việc cùng kết hợp đồng thời các bộ lọc này vào cùng một block có thể mang lại hiệu quả đó chính là kiến trúc khối Inception.

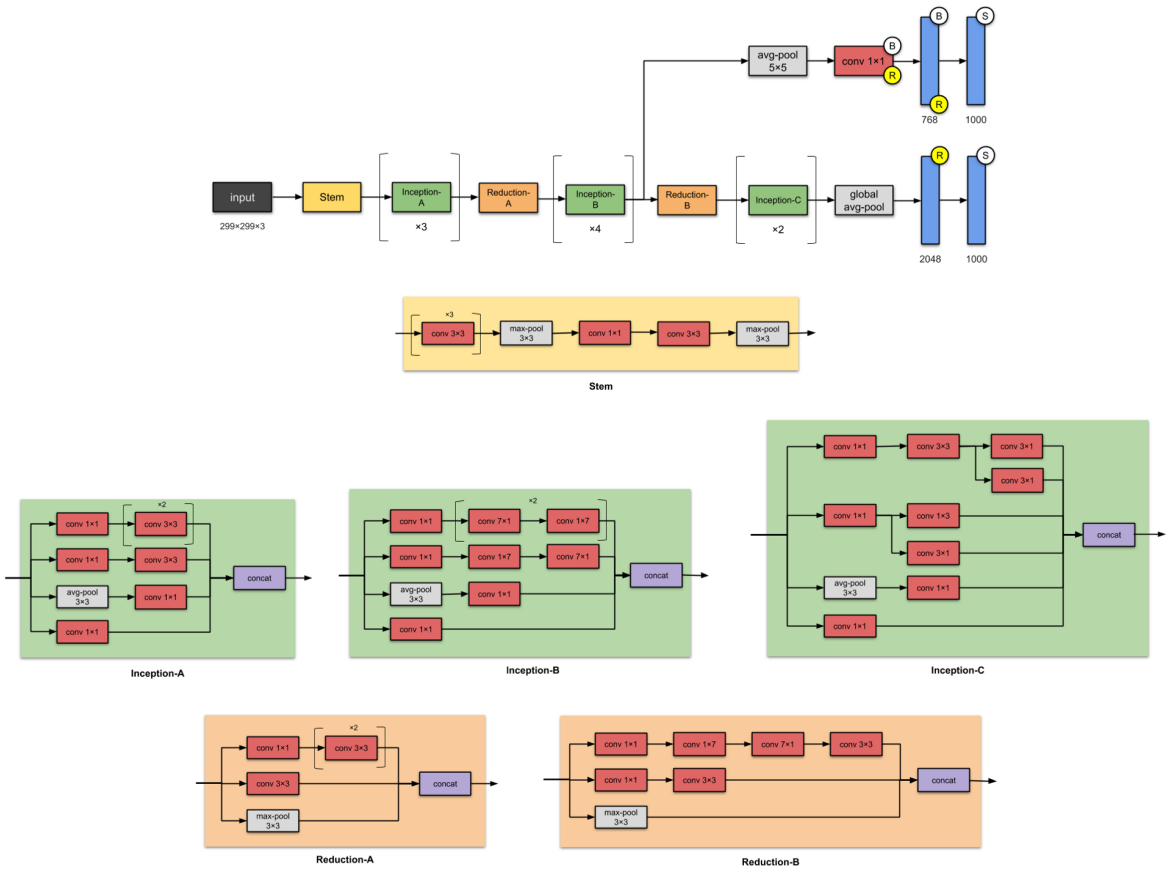
Khối Inception:

- Khối Inception sẽ bao gồm 4 nhánh song song. Các bộ lọc kích thước lần lượt là `1x1`, `3x3`, `5x5` được áp dụng trong Inception Module giúp trích lọc được đa dạng đặc trưng trên những vùng nhận thức có kích thước khác nhau.
- Ở đầu các nhánh 1, 2, 4 từ trên xuống, phép tích chập `1x1` được sử dụng trên từng điểm ảnh như một kết nối fully connected nhằm mục đích giảm độ sâu kênh và số lượng tham số của mô hình. Ví dụ: Ở block trước chúng ta có kích thước `width x height x channels = 12 x 12 x 256`. Sau khi áp dụng 32 bộ lọc kích thước `1x1` sẽ không làm thay đổi `width, height` và độ sâu giảm xuống `32`, output shape lúc này có kích thước là `12 x 12 x 32`. Ở layer liền sau, khi thực hiện tích chập trên toàn bộ độ sâu, chúng ta chỉ khởi tạo các bộ lọc có độ sâu 32 thay vì 256. Do đó đó số lượng tham số giảm đi một cách đáng kể.
- Nhánh thứ 3 từ trên xuống chúng ta giảm chiều dữ liệu bằng một layer max-pooling kích thước 3x3 và sau đó áp dụng bộ lọc kích thước 1x1 để thay đổi số kênh.
- Các nhánh áp dụng padding và stride sao cho đầu ra có cùng kích cỡ chiều dài và chiều rộng. Cuối cùng ta concatenate toàn bộ kết quả đầu ra của các khối theo kênh để thu được output có kích thước bằng với input.

Khối Inception được lặp lại 7 lần trong kiến trúc Inception-V1. Toàn bộ mạng bao gồm 22 Layers, lớn hơn gần gấp đôi so với VGG-16. Nhờ áp dụng tích chập `1x1` giúp tiết kiệm số lượng tham số xuống chỉ còn 5 triệu, ít hơn gần 27 lần so với VGG-16.

4.5. GoogleNet - Inception-V3 (2015)

Top



Hình 10: Kiến trúc GoogleNet - Inception version 3.

Paper Inception-V3 Rethinking the Inception Architecture for Computer Vision (https://arxiv.org/abs/1512.00567)

Authors: Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Google, University College London

Inception-V3 là kế thừa của Inception-V1 bao gồm 24 triệu tham số. Toàn bộ các layer tích chập của Inception-V3 được theo sau bởi một layer batch normalization và một ReLU activation. Batch normalization là kỹ thuật chuẩn hóa đầu vào theo từng minibatch tại mỗi layer theo phân phối chuẩn hóa $\mathbf{N}(0,1)$, giúp cho quá trình huấn luyện thuật toán nhanh hơn.

Inception-V3 giải quyết được vấn đề thắt cổ chai (representational bottlenecks). Tức là kích thước của các layers không bị giảm một cách đột ngột. Đồng thời Inception-V3 có một cách tính toán hiệu quả hơn nhờ sử dụng phương pháp nhân tố (factorisation methods).

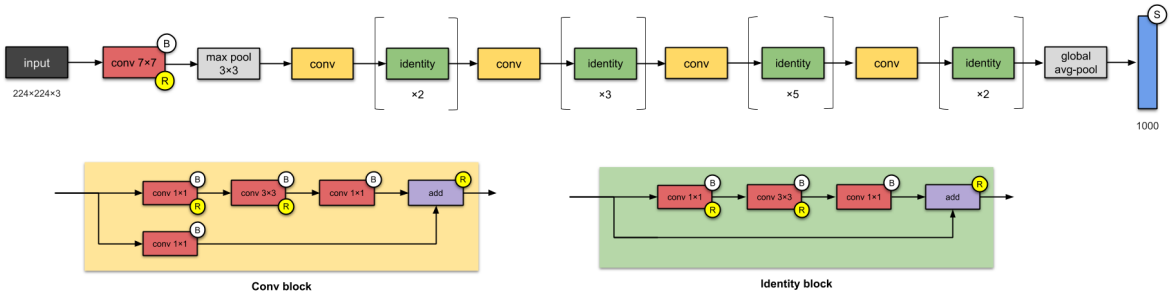
Hiện tại Inception module bao gồm 4 version. Chúng ta hãy cùng xem qua các điểm đặc biệt ở từng version.

- Inception-A: Cải tiến so với Inception module V1. Tại nhãnh thứ nhất thay 1 layer tích chập `5 x 5` bằng 2 layer tích chập `3 x 3` liên tiếp giúp giảm số lượng tham số từ 25 về 18 và tăng độ sâu cho mô hình.
- Inception-B: Cải tiến so với Inception-A. Thay tích chập `3 x 3` bằng tích chập `7 x 7` ở nhánh thứ nhất và nhánh thứ 2. Đồng thời chúng ta phân tích nhân tố tích chập `7 x 7` thành 2 tích chập liên tiếp `7 x 1` và `1 x 7` số lượng tham số sẽ ít hơn so với tích chập 2 tích chập `3 x 3` liên tiếp. Nhờ đó số lượng tham số giảm từ 18 xuống còn 14.
- Inception-C: Cải tiến so với Inception-B. Thay tích chập `7 x 1` bằng tích chập `3 x 1` và `1 x 7` bằng `1 x 3` và đồng thời thay vì đặt layer `3 x 1` và `1 x 3` liên tiếp thì đặt chúng song song. Kiến trúc này giúp giảm số lượng tham số từ 14 về còn 6.

Top

Ngoài ra ở Inception-V3 chúng ta còn sử dụng 2 kiến trúc giảm chiều dữ liệu là Reduction-A và Reduction-B.

4.6. ResNet-50 (2015)



Hình 11: Kiến trúc ResNet bao gồm 2 khối đặc trưng là khối tích chập (Conv Block) và khối xác định (Identity Block).

Paper - ResNet - Deep Residual Learning for Image Recognition (https://arxiv.org/abs/1512.03385)

Authors: Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Microsoft

ResNet là kiến trúc được sử dụng phổ biến nhất ở thời điểm hiện tại. ResNet cũng là kiến trúc sớm nhất áp dụng batch normalization. Mặc dù là một mạng rất sâu khi có số lượng layer lên tới 152 nhưng nhờ áp dụng những kỹ thuật đặc biệt mà ta sẽ tìm hiểu bên dưới nên kích thước của ResNet50 chỉ khoảng 26 triệu tham số. Kiến trúc với ít tham số nhưng hiệu quả của ResNet đã mang lại chiến thắng trong cuộc thi ImageNet năm 2015.

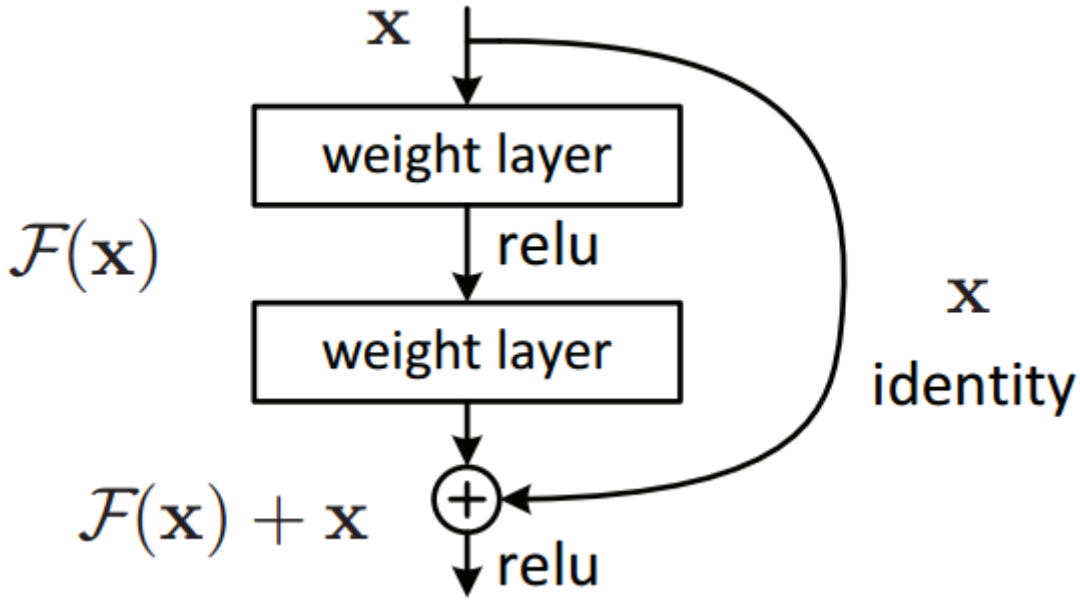
Những kiến trúc trước đây thường cải tiến độ chính xác nhờ gia tăng chiều sâu của mạng CNN. Nhưng thực nghiệm cho thấy đến một ngưỡng độ sâu nào đó thì độ chính xác của mô hình sẽ bão hòa và thậm chí phản tác dụng và làm cho mô hình kém chính xác hơn. Khi đi qua quá nhiều tầng độ sâu có thể làm thông tin gốc bị mất đi thì các nhà nghiên cứu của Microsoft đã giải quyết vấn đề này trên ResNet bằng cách sử dụng kết nối tắt.

Các kết nối tắt (skip connection) giúp giữ thông tin không bị mất bằng cách kết nối từ layer sớm trước đó tới layer phía sau và bỏ qua một vài layers trung gian. Trong các kiến trúc base network CNN của các mạng YOLOv2, YOLOv3 và gần đây là YOLOv4 bạn sẽ thường xuyên thấy các kết nối tắt được áp dụng.

ResNet có khối tích chập (Convolutional Bock, chính là Conv block trong hình) sử dụng bộ lọc kích thước `3 x 3` giống với của InceptionNet. Khối tích chập bao gồm 2 nhánh tích chập trong đó một nhánh áp dụng tích chập `1 x 1` trước khi cộng trực tiếp vào nhánh còn lại.

Khối xác định (Identity block) thì không áp dụng tích chập `1 x 1` mà cộng trực tiếp giá trị của nhánh đó vào nhánh còn lại.

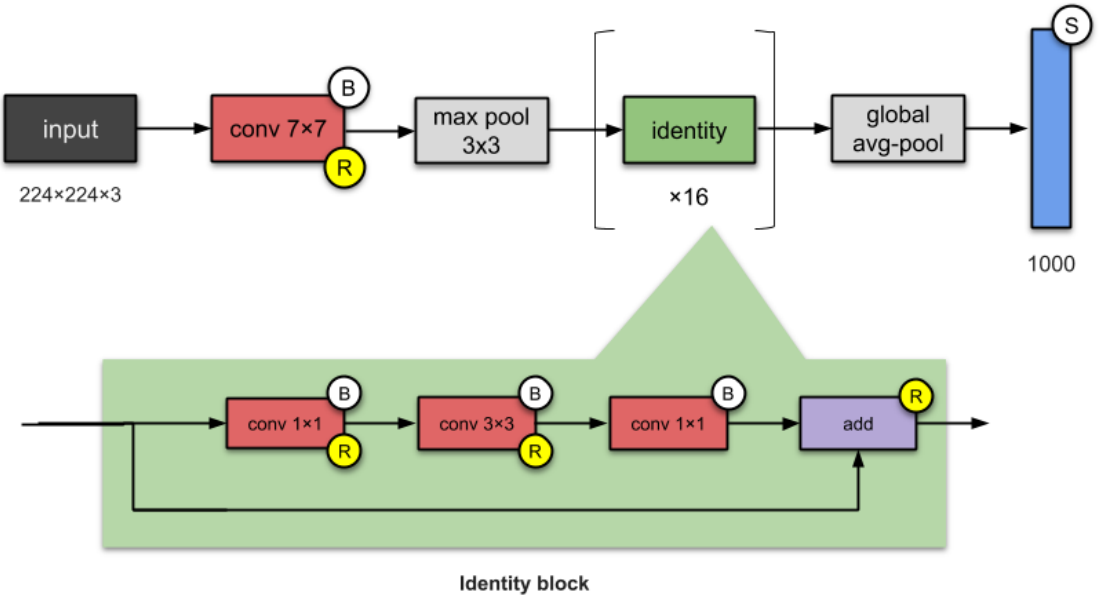
Top



Hình 12: Cộng trực tiếp đầu vào của khối với nhánh còn lại trong khối Identity block.

Gỉa sử chúng ta có $\mathbf{x}$ là đầu vào của khối xác định. Chúng ta cần ánh xạ đầu vào $\mathbf{x}$ thành hàm $f(\mathbf{x})$. Để tìm ra ánh xạ chuẩn xác tương đương với hàm $f(\mathbf{x})$ là một việc khá khó. Nhưng nếu cộng thêm ở đầu ra thành $\mathbf{x} + f(\mathbf{x})$ thì chúng ta sẽ qui về tham số hóa độ lệch, tức cần tham số hóa phần dư $f(\mathbf{x})$. Tìm ánh xạ theo phần dư sẽ dễ hơn nhiều vì chỉ cần tìm giá trị $f(\mathbf{x})$ sao cho nó gần bằng 0 là có thể thu được một ánh xạ chuẩn xác. Tại một khối xác định, chúng ta sẽ áp dụng một layer activation ReLU sau mỗi xen kẽ giữa những tầng trọng số.

Mặc dù có kiến trúc khối kế thừa lại từ GoogleNet nhưng ResNet lại dễ tóm tắt và triển khai hơn rất nhiều vì kiến trúc cơ sở của nó chỉ gồm các khối tích chập và khối xác định. Ta có thể đơn giản hóa kiến trúc của ResNet-50 như hình bên dưới:



Hình 13: Kiến trúc tóm tắt của mạng ResNet-50.

4.7. DenseNet (2016)

Ở ResNet chúng ta phân tách hàm số thành một hàm xác định và một hàm phi tuyến:

Top

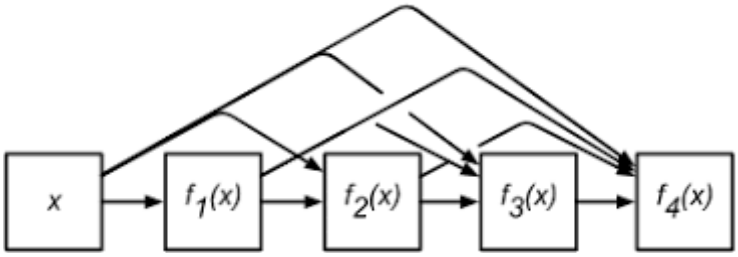
$$f(\mathbf{x}) = \mathbf{x} + g(\mathbf{x})$$

Cùng nhắc lại công thức khai triển Taylor tại $x = 0$:

$$f(x) = f(0) + f'(x)x + \frac{f''(x)}{2!}x^2 + \cdots + \frac{f^{(n)}(x)}{n!}x^n + o(x^n)$$

Ta có thể thấy công thức của ResNet cũng gần tương tự như khai triển taylor tại đạo hàm bậc nhất, $g(\mathbf{x})$ tương ứng với thành phần số dư. Khai triển Taylor sẽ càng chuẩn xác nếu chúng ta phân rã được số dư thành nhiều đạo hàm bậc cao hơn.

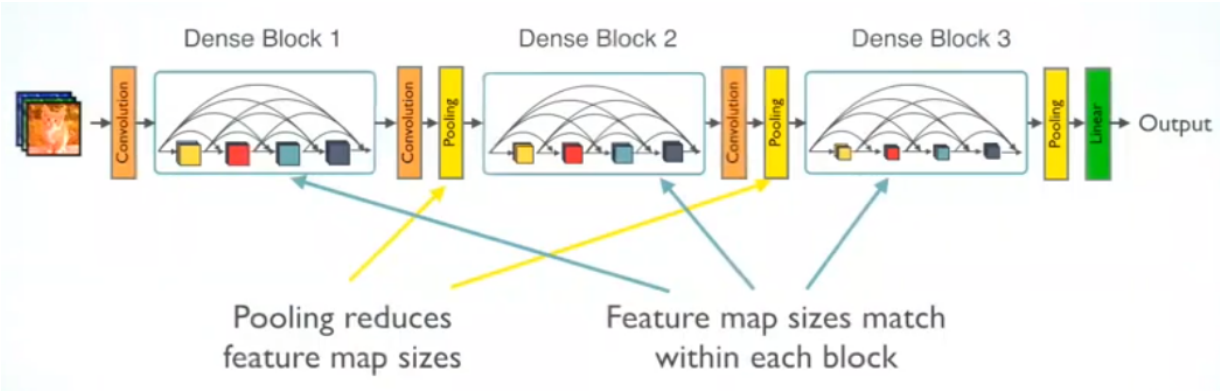
Ý tưởng của DenseNet cũng như vậy, chúng ta sẽ sử dụng một mạng lưới các kết nối tắt dày đặc để liên kết các khối với nhau.



Từ đầu vào $\mathbf{x}$ ta sẽ áp dụng liên tiếp một chuỗi các ánh xạ liên tiếp với cấp độ phức tạp tăng dần:

$$\mathbf{x} \to f_1(\mathbf{x})\ \mathbf{x} \to f_2(\mathbf{x}, f_1(\mathbf{x}))\ \ldots\ \mathbf{x} \to f_4(\mathbf{x}, f_3(\mathbf{x}, f_2(\mathbf{x}, f_1(\mathbf{x}))))$$

DenseNet sẽ khác so với ResNet đó là chúng ta không cộng trực tiếp $\mathbf{x}$ vào $f(\mathbf{x})$ mà thay vào đó, các đầu ra của từng phép ánh xạ có cùng kích thước dài và rộng sẽ được concatenate với nhau thành một khối theo chiều sâu. Sau đó để giảm chiều dữ liệu chúng ta áp dụng tầng chuyển tiếp (translation layer). Tầng này là kết hợp của một layer tích chập giúp giảm độ sâu và một max pooling giúp giảm kích thước dài và rộng. Các bạn sẽ dễ dàng hình dung hơn qua hình vẽ bên dưới:



Hình 14: Kiến trúc DenseNet.

Và bên dưới là chi tiết của từng layers trong DenseNet.

Top

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Kết quả là DenseNet121 chỉ với 8 triệu tham số nhưng có độ chính xác cao hơn so với ResNet50 với gần 26 triệu tham số trên bộ dữ liệu ImageNet.

DenseNet đồng thời cũng áp dụng BatchNormalization trước khi thực hiện tích chập ở các tầng chuyển tiếp nên giảm được triệt tiêu đạo hàm (*vanishing gradient descent*).

5. Tổng kết

Như vậy thông qua bài giới thiệu này, các bạn đã nắm rõ được gần hết tiến trình phát triển của các kiến trúc mạng CNN từ giai đoạn bắt đầu (kiến trúc LeNet) đến những cuối những năm 2016 (*ResNet, DenseNet*). Mình xin tổng kết các dấu mốc của từng mạng như sau:

- LeNet (1998): Là mạng đầu tiên áp dụng tích chập 2 chiều.
- AlexNet (2012): Làm mạng áp dụng CNN đầu tiên chiến thắng trong cuộc thi ImageNet. Phá vỡ lối mòn sử dụng các đặc trưng thủ công từ các thuật toán truyền thống như HOG, SHIFT, SURF thay cho các đặc trưng được huấn luyện trong các tác vụ học có giám sát của thị giác máy tính.
- VGG-16 (2014): Hình thành một xu hướng cải thiện độ chính xác của các mạng học sâu thông qua gia tăng độ sâu của chúng.
- GoogleNet - InceptionV1 (2014): Kết hợp nhiều bộ lọc có kích thước khác biệt vào cùng một khối. Định hình kiến trúc khối cho các kiến trúc mạng CNN chuẩn sau này.
- ResNet-50 (2015): Sử dụng kết nối tắt để ánh xạ các đầu vào từ những layer trước đó tới những layer sau. Là kiến trúc mạng rất sâu nhưng có số tham số nhỏ hơn nhờ kế thừa những kỹ thuật từ GoogleNet.
- DenseNet (2016): Là bước phát triển tiếp theo cua ResNet khi kế thừa kiến trúc khối và phát triển kết nối tắt theo một mạng dày đặc.

Ngoài những kiến trúc tiêu biểu mang tính dấu mốc đã được mình liệt kê trên, vẫn còn những kiến trúc khác không nằm trong top đầu của cuộc thi ImageNet nhưng cũng được sử dụng rộng rãi như MobileNet, SqueezeNet, NasNet. Gần đây thì kiến trúc EfficientNet dựa trên việc tìm kiếm tối ưu trên không gian các tham số Depth, Width và Channel đã được google phát triển và tạo ra kết quả SOTA trên bộ dữ liệu ImageNet. Nhưng có lẽ mình sẽ viết tiếp ở một bài khác.

6.Tài liệu tham khảo

Top

1. Đắm mình vào học sâu - Chapter 7 - Mạng nơ ron tích chập sâu hiện đại (https://d2l.aivivn.com/chapter_convolutional-modern/alexnet_vn.html)

2. Các kiến trúc mạng CNN - dlapplication.github.io (https://dlapplications.github.io/2018-07-06-CNN/)
3. Illustrated: 10 CNN Architectures - Raimi Karim (https://towardsdatascience.com/illustrated-10-cnn-architectures-95d78ace614d)
4. Bài 8 - Convolutional Neural Network - Khanh blog (https://phamdinhkhanh.github.io/2019/08/22/convolutional-neural-network.html)
5. Overview of CNN research: 25 years history and the current trends (https://ieeexplore.ieee.org/iel7/7152138/7168553/07168655.pdf)
6. CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more (https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5)

Top