Report paper: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

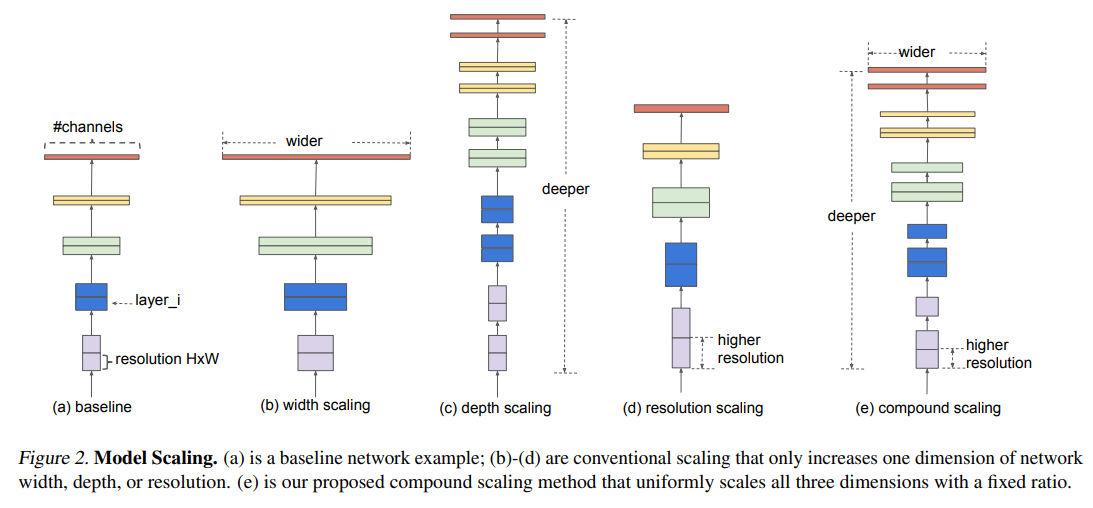
Giáp Ngọc Hiệu

**Tóm tắt nội dung**

Bài report này là phần tóm tắt lại ý tưởng, nội dung chính, những điều làm được, giải của paper EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

1. **Ý tưởng**

Convolutional Neural Networks (ConvNets) thông thường xây dựng với một kiến trúc có só lượng tham số nhất định, sau đó có thể được mở rộng bằng cách thay đổi chiều sâu, chiều rộng và độ phân giải để có độ chính xác cao hơn nếu có nhiều tài nguyên hơn. Tác giả đã đề xuất một phương pháp chia tỷ lệ mới giúp chia tỷ lệ đồng đều tất cả các kích thước này một cách hiệu quả bằng cách sử dụng compound coefficient. Bên cạnh đó, tác giả còn sử dụng neural architecture search tìm kiếm kiến ​​trúc nơ-ron để thiết kế một mạng cơ sở mới và mở rộng quy mô để có được một nhóm các mô hình, được gọi là EfficientNets, đạt được độ chính xác và hiệu quả cao hơn nhiều so với các ConvNet trước đó.



\* Tìm kiếm kiến trúc mạng neural (Neural Architect Search) là phương pháp tự động hóa việc thiết kế kiến trúc mạng neural. Phương pháp này đã được sử dụng để thiết kế các kiến trúc mạng neural vượt trội so với các kiến trúc mạng neural được thiết kế thủ công mà điển hình nhất nằm ở nhiệm vụ phân loại ảnh.

\* Depth: số lớp trong mạng, mạng càng sâu thì càng có nhiều khả năng hoạt động hiệu quả hơn.

\* Width: số lượng kênh trong một lớp tích chập, mạng càng rộng thì càng có xu hướng nắm bắt các tính năng chi tiết và càng dễ đào tạo.

\* Resolution: là độ phân giải của hình ảnh đầu vào. Hình ảnh càng lớn thường có độ chính xác càng cao với FLOPS cao hơn.

\* FLOPS là viết tắt của "Floating Point Operations Per Second". Trong mạng nơ-ron, FLOPS thường được sử dụng để đo lường khả năng tính toán của mô hình.

1. **Phương pháp**
   1. **Compound Model Scaling**

**Problem Formulation**

Lớp ConvNet i có thể được định nghĩa là một hàm: 

Trong đó  là toán tử,  là đầu ra,  là đầu vào với kích thước . Một ConvNet  có thể được biểu diễn bằng một danh sách các lớp :

N = 

Trên thực tế, các lớp ConvNet thường được phân chia thành nhiều giai đoạn và tất cả các lớp trong mỗi giai đoạn đều có chung kiến ​​trúc. Do đó, có thể định nghĩa một ConvNet là:



Trong đó  biểu thị lớp  được lặp lại  lần trong giai đoạn ,  là hình dạng đầu vào X của lớp 

Bằng cách giữ nguyên , model scaling đơn giản hóa vấn đề thiết kế đối với các ràng buộc tài nguyên mới, nhưng nó vẫn là một không gian thiết kế rộng lớn để lựa chọn các  khác nhau cho mỗi lớp. Để tiếp tục giảm không gian thiết kế, tác giả ràng buộc tất cả các lớp phải được chia tỷ lệ đồng nhất với tỷ lệ không đổi. Mục tiêu là:



trong đó w, d, r là các hệ số để chia tỷ lệ chiều rộng, độ sâu và độ phân giải của mạng; là các tham số được xác định trước trong mạng cơ sở.

**Scaling Dimensions**

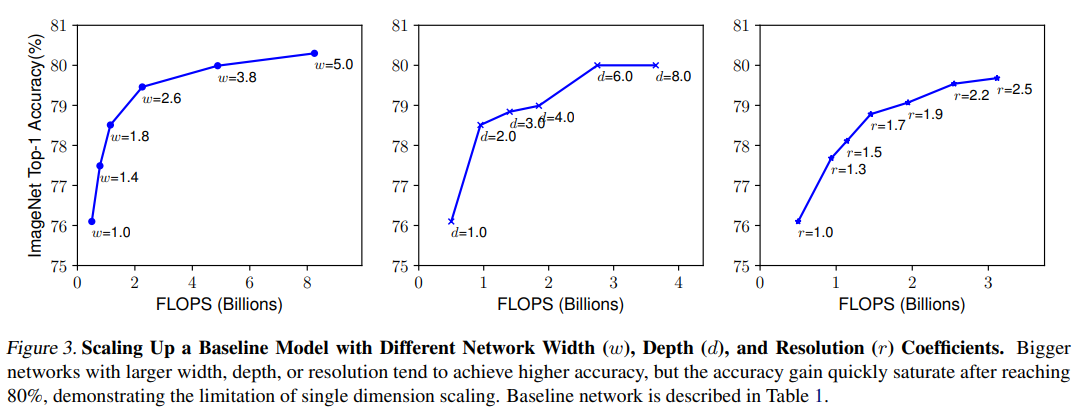
Do d, w, r tối ưu phụ thuộc lẫn nhau và các giá trị thay đổi dưới các ràng buộc tài nguyên khác nhau, các phương pháp thông thường chủ yếu mở rộng ConvNet theo một trong các kích thước này:

**- Depth (d)**: ý tưởng là ConvNet sâu hơn có thể nắm bắt các tính năng phong phú và phức tạp hơn, đồng thời khái quát hóa tốt các tác vụ mới. Tuy nhiên, các mạng sâu hơn cũng khó đào tạo hơn do vấn đề độ dốc biến mất. Mặc dù một số kỹ thuật, chẳng hạn như skip connection và batch normalization có thể khắc phục, nhưng độ chính xác của mạng rất sâu thường giảm đi.

**- Width (w)**: các mạng rộng hơn có xu hướng nắm bắt được nhiều đặc trưng chi tiết hơn và dễ đào tạo hơn, thường được sử dụng cho các mô hình có kích thước nhỏ. Tuy nhiên, các mạng cực rộng nhưng nông có xu hướng gặp khó khăn trong việc nắm bắt các đặc trưng cấp cao hơn.

**- Resolution (r)**: với hình ảnh đầu vào có độ phân giải cao hơn, ConvNets có khả năng nắm bắt được các mẫu chi tiết hơn và đạt được độ chính xác cao hơn.

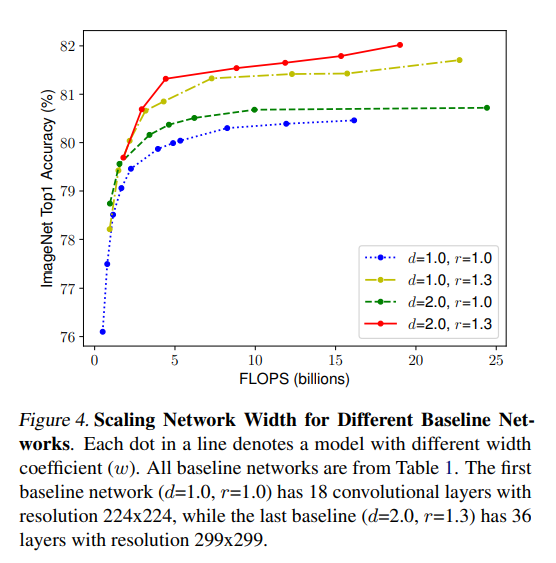
Từ các phân tích trên, tác giả quan sát được rằng việc điều chỉnh bất kỳ kích thước nào về chiều rộng, độ sâu hoặc độ phân giải của mạng sẽ cải thiện độ chính xác, nhưng mức tăng sẽ giảm dần đối với các mô hình lớn hơn.



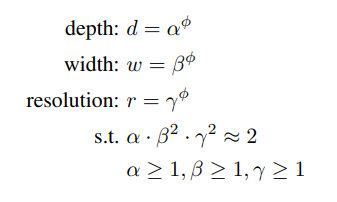
**Compound Scaling**

Theo kinh nghiệm, tác giả cho rằng các kích thước tỷ lệ khác nhau không độc lập. Để kiểm tra, tác giả tiến hành thử nghiệm và so sánh tỷ lệ chiều rộng theo các độ sâu và độ phân giải mạng khác nhau. Từ đó, tác giả đưa ra quan sát thứ hai:

Để đạt được độ chính xác và hiệu quả cao hơn, điều quan trọng là phải cân bằng tất cả các kích thước về chiều rộng, chiều sâu và độ phân giải của mạng trong quá trình thay đổi quy mô ConvNet.



Trong paper này, tác giả đề xuất một phương pháp chia tỷ lệ hỗn hợp mới (compound scaling method), sử dụng hệ số hỗn hợp (compound coefficient) φ để chia tỷ lệ đồng đều cho chiều rộng, độ sâu và độ phân giải của mạng theo nguyên tắc:



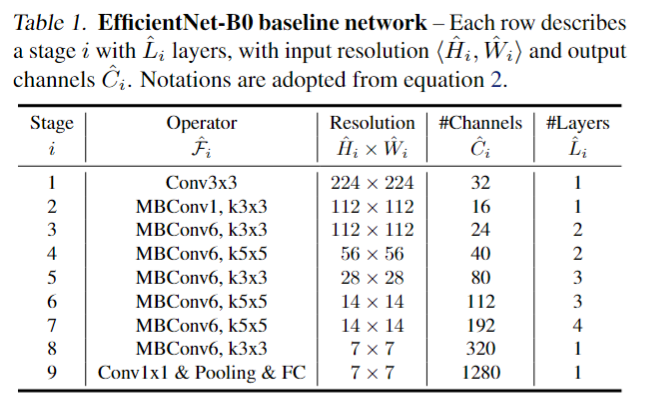
Trong đó φ được gọi là compound coefficient, α, β và γ là các hằng số có thể tìm thấy bằng một tìm kiếm dạng lưới nhỏ.

φ là hệ số do người dùng chỉ định để kiểm soát lượng tài nguyên khả dụng. Trong khi α, β và γ gán các tài nguyên này cho độ sâu, độ rộng và độ phân giải của mạng tương ứng.

Tác giả đã nhận thấy rằng FLOPS của phép toán tích chập thông thường tỷ lệ với . Do hoạt động tích chập chi phối chi phí tính toán trong ConvNets, nên việc scaling trên Convnet sẽ tăng số lượng FLOPS lên , do đó hạn chế để tăng tổng số FLOPS lên .

* 1. **EfficientNet Architecture**

Vì việc mở rộng mô hình không làm thay đổi các toán tử lớp  trong mạng cơ sở, nên việc có một mạng cơ sở tốt cũng rất quan trọng. Do đó, tác giả đã phát triển một mạng cơ sở mới được gọi là EfficientNet bằng cách tận dụng tìm kiếm kiến ​​trúc thần kinh đa mục tiêu để tối ưu hóa cả độ chính xác và FLOPS. Tìm kiếm của tác giả tạo ra một mạng được đặt tên là EfficientNet-B0. Tác giả thực hiện đánh giá phương pháp chia tỷ lệ của mình bằng cách sử dụng các ConvNets hiện có và EfficientNet.



Bắt đầu từ EfficientNet-B0, tác giả áp dụng phương pháp mở rộng quy theo hai bước:

+ Bước 1: đặt φ = 1, giả sử có thêm gấp đôi tài nguyên và thực hiện tìm kiếm lưới cho α, β, γ. Các giá trị tốt nhất cho EfficientNet-B0 là α = 1,2, β = 1,1, γ = 1,15, với ràng buộc α · β2 · γ2 ≈ 2

+ Bước 2: đặt α, β, γ thành các hằng số và mở rộng quy mô mạng cơ sở với các φ khác nhau để thu được EfficientNet-B1 đến B7.

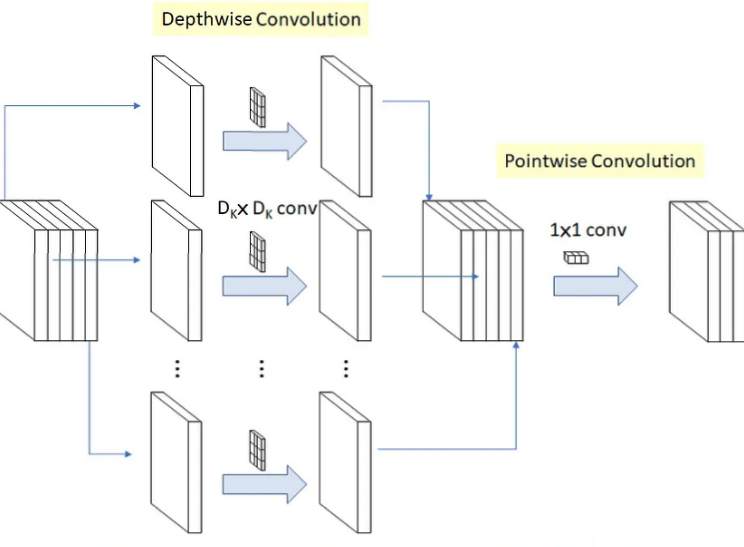
Đáng chú ý, có thể đạt được hiệu suất thậm chí còn tốt hơn bằng cách tìm kiếm trực tiếp α, β, γ xung quanh một mô hình lớn, nhưng chi phí tìm kiếm trở nên đắt hơn rất nhiều trên các mô hình lớn hơn.

* 1. **MBConv**

MBConv bắt nguồn từ MobileNetV2, đôi khi gọi là Inverted Residual Block bao gồm Depth-Wise Separable Convolution.

**2.4 Depthwise seperable convolution (MobileNetV1)**

Depthwise seperable convolution gồm 2 thành phần là depthwise convolution và pointwise convolution.



**Depthwise convolution**

Trong depthwise convolution, mỗi filter chỉ có một channel và nó sẽ tương tác với duy nhất một channel trên input. Số filters sẽ bằng với số channels của input.

**Pointwise convolution**

Pointwise convolution chính là Conv layer 1x1.

**Lợi ích của Depthwise seperable convolution**

Input: feature map Df x Df x M

Output: feature map Df x Df x N

Trong đó:

+ Df là chiều rộng và chiều cao của feature map có đầu vào là vuông.

+ M là số channel input

+ N là số channel output.

+ K là kernel Dk x Dk x M x N.

Tổng chi phí tính toán cho standard convolution:

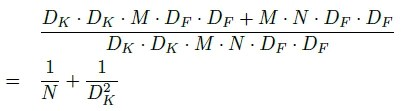


Tổng chi phí tính toán cho Depthwise Separable Convolution:



Depthwise Convolution (trái), Pointwise Convolution (phải)

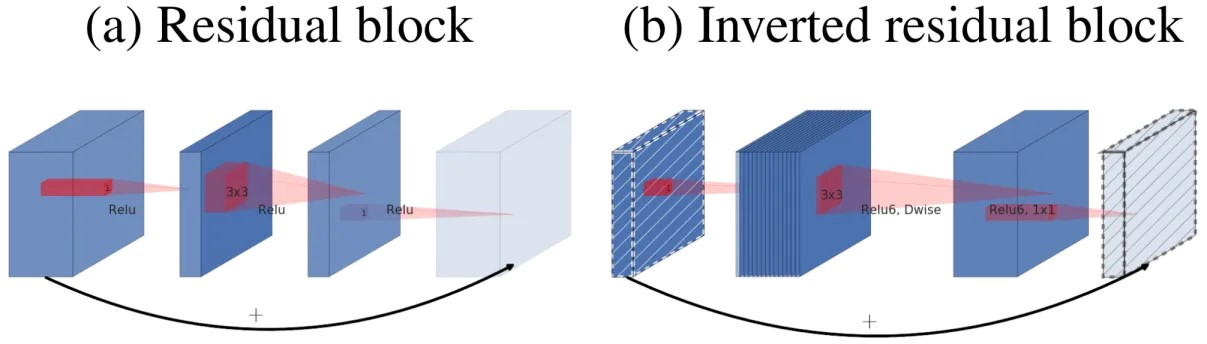
Chi phí tính toán của Depthwise Separable Convolution so với standard convolution:



Khi DK×DK là 3×3, có thể đạt được khả năng tính toán ít hơn từ 8 đến 9 lần, nhưng độ chính xác chỉ giảm một chút.

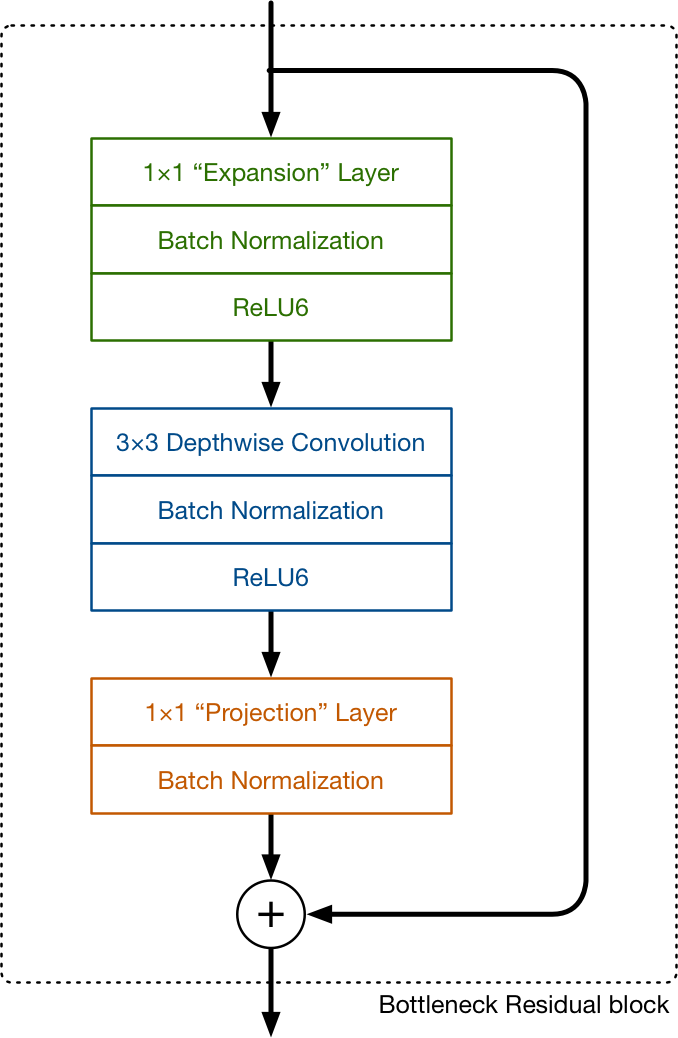
Do đó, Depthwise Separable Convolution giảm chi phí tính toán đáng kể so với standard convolution nhưng lại có hiệu suất gần như nhau, thuận lợi cho triển khai các model yêu cầu tốc độ cao hoặc trên các edge devices.

**2.5 Inverted residual block**

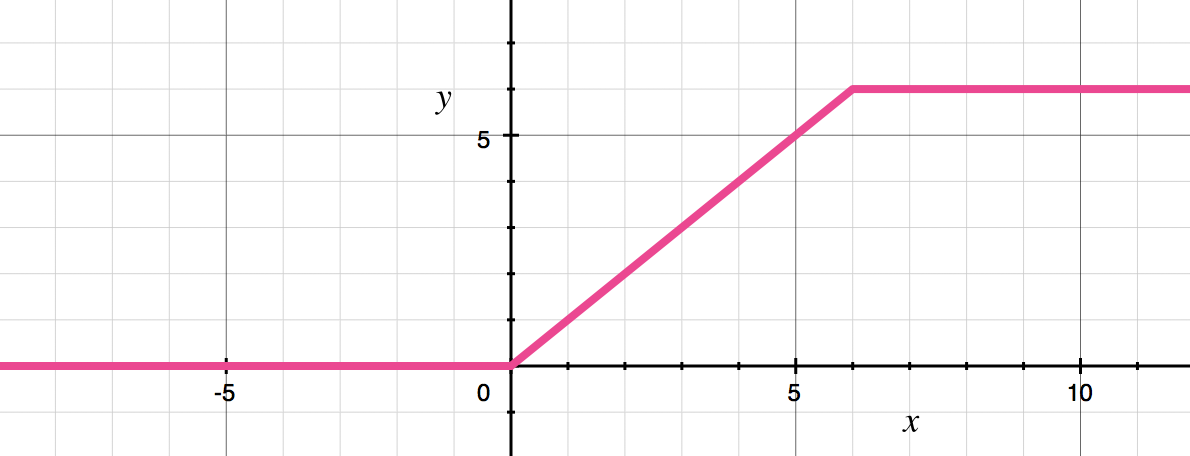


Residual block tuân theo cách tiếp cận rộng > hẹp> rộng liên quan đến số lượng kênh. Input có số lượng kênh cao, được nén với tích chập 1x1 làm cho tích chập 3x3 có ít tham số hơn. Sau đó sử dụng tích chập 1x1 khác để thực hiện skip connection input với output.

Inverted residual block tuân theo cách tiếp cận hẹp>rộng>hẹp. Bước đầu tiên mở rộng số channels input bằng cách sử dụng tích chập 1x1. Số channels được mở rộng được xác định thông qua expansion factor - đây là một hyperparamter. Mặc định expansion factor = 6. Sau đó, đi qua tích chập 3x3 làm giảm đáng kể số lượng tham số và cuối cùng là một tích chập 1x1 khác để thực hiện skip connection input với output. Ngoài ra, trước khi thực hiện skip connection, cần áp dụng một hàm tuyến tính để ngăn chặn sự phi tuyến tính phá hủy quá nhiều thông tin.



Trong thực tế, Inverted residual block còn có thêm Batch Normalization và ReLU 6



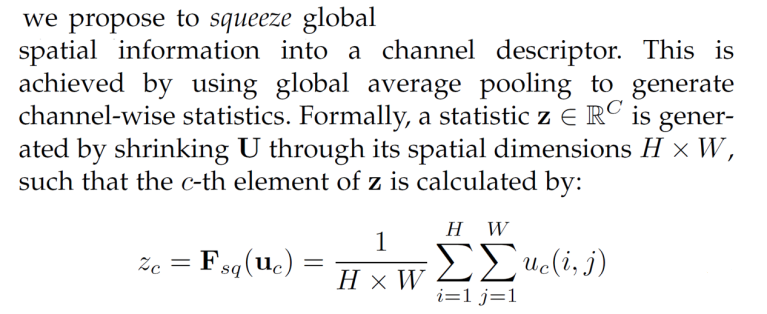
ReLU6

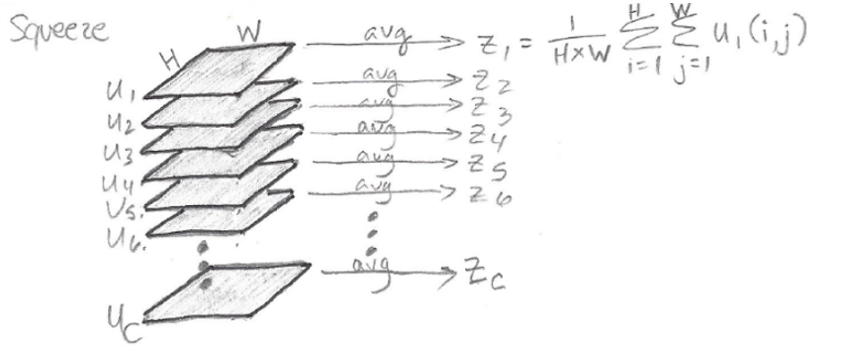
**2.6 Squeeze and Excitation (SE) Block**

SE là một building block dành cho CNNs để cải thiện sự phụ thuộc lẫn nhau giữa các kênh bằng cách thực hiện hiệu chỉnh lại tính năng động theo từng kênh, điều này có nghĩa là thay vì đặt trọng số bằng nhau cho tất cả các kênh, mạng sẽ tự động gán trọng số cao cho các kênh quan trọng nhất.

**Squeeze: Global Information Embedding**

Sử dụng global average pooling để tổng hợp các feature map trên các kích thước không gian H x W của chúng để tạo ra bộ mô tả kênh.



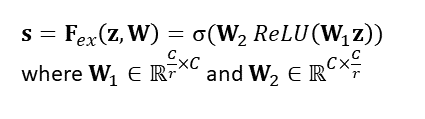


Các đại lượng vô hướng [z 1, z 2,..., z C] kết hợp để tạo ra một vectơ độ dài z có độ dài C, sẽ được sử dụng trong giai đoạn excitation.

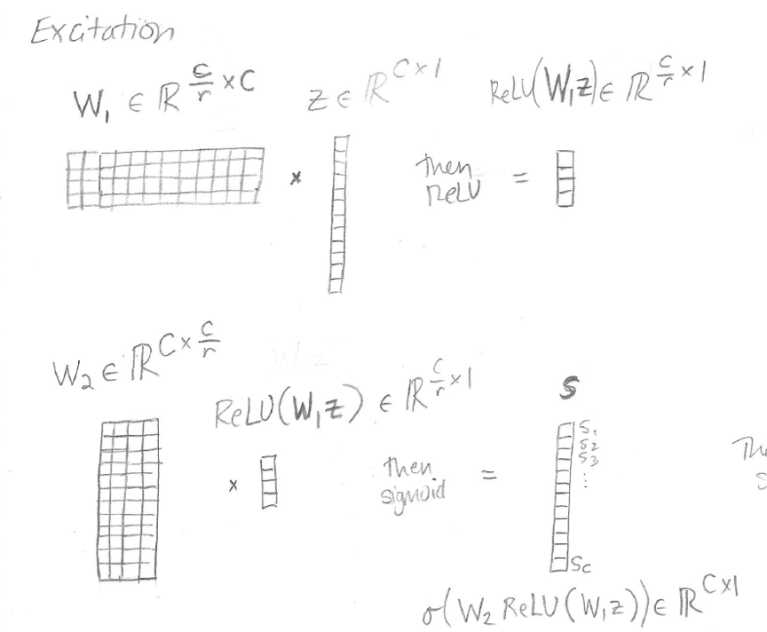
**Excitation: Adaptive Recalibration**

Bước này được thiết kế để nắm bắt hoàn toàn các phụ thuộc kênh cụ thể. Chuyển đổi đầu ra của bước squeeze (vector z) thành một vector kích hoạt s, sau đó được sử dụng để thay đổi tỷ lệ bản đồ đặc trưng.

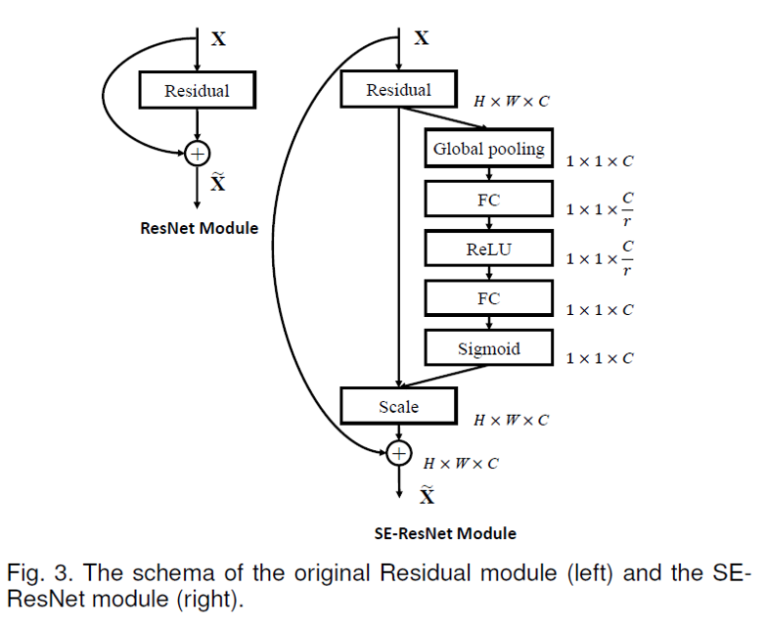
Vector s được xác định bằng cách sử dụng hai lớp fully-connected để xuống kích thước C/r:

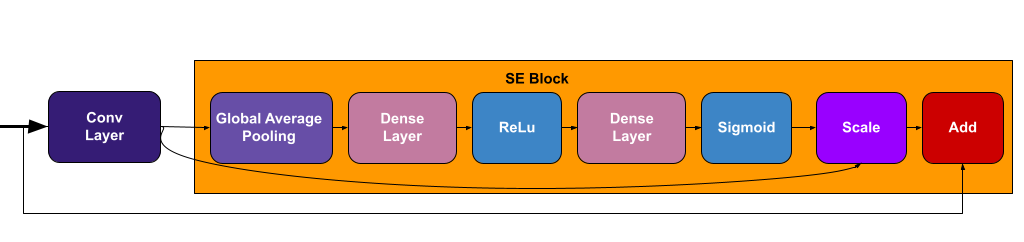


"reduction ratio " là tên được đặt cho siêu tham số r. Biểu diễn trung gian co lại khi r lớn hơn. Mục đích của việc thu nhỏ biểu diễn thành C/r và sau đó mở rộng nó trở lại C là để giảm độ phức tạp của mô hình và cải thiện khả năng khái quát hóa.

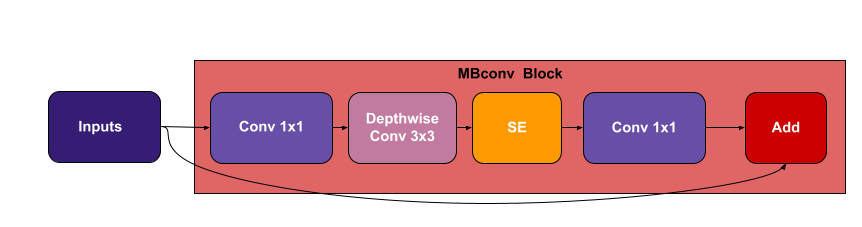


**Squeeze and Excitation Block**





* 1. **MBConv block cùng với SE block được sử dụng trong EfficientNet**

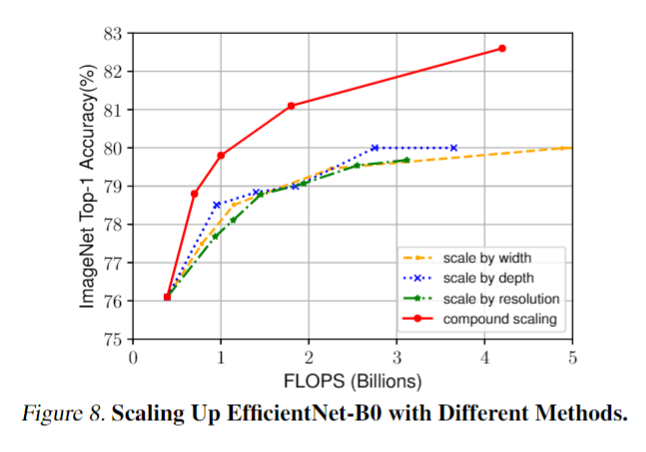


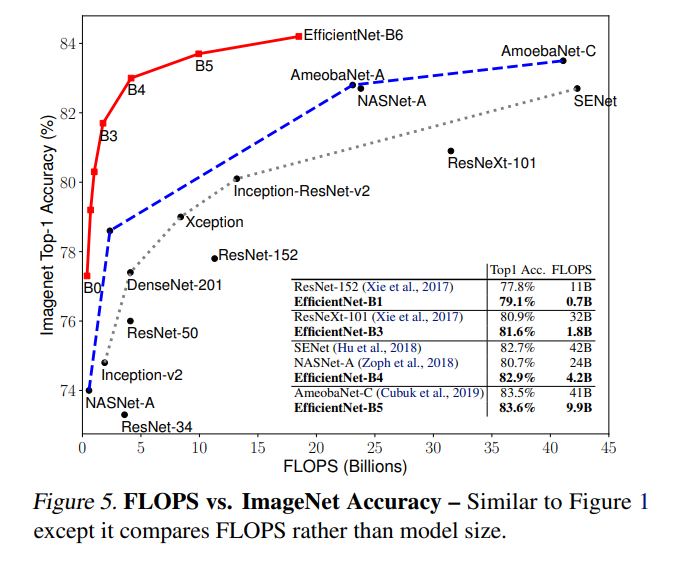
1. **Metric**

Top-1 error rate: tỷ lệ phần trăm của số lượng ảnh trong tập kiểm tra bị phân loại sai vào class có xác suất dự đoán cao nhất.

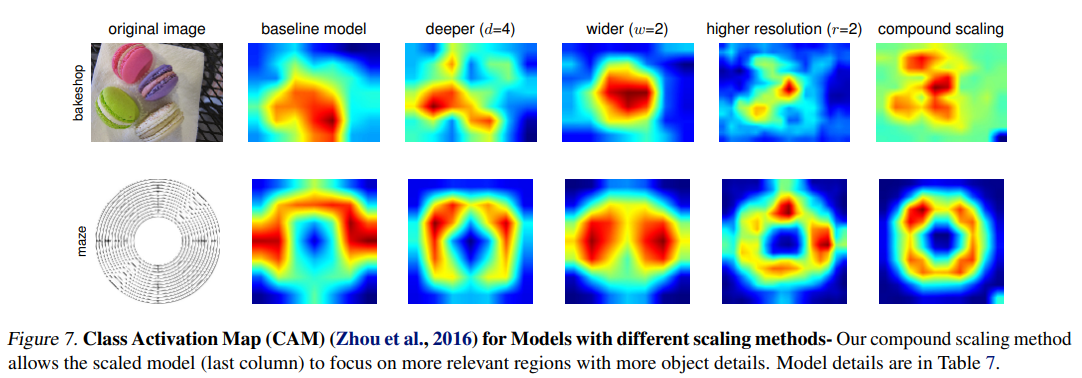
Top-5 error rate : tỷ lệ phần trăm của số lượng ảnh trong tập kiểm tra bị phân loại sai nhưng nhãn đúng của chúng xuất hiện trong top 5 kết quả phân loại cao nhất của model.

1. **Kết quả**

****

****

Compound scaling method vượt trội so với các phương pháp khác và EfficientNet đã cho thấy sự vượt trội so với các mô hình hiện đại về hiệu quả và độ chính xác.



Mô hình có sử dụng compound scaling method có xu hướng tập trung vào các vùng phù hợp hơn với nhiều chi tiết đối tượng hơn, trong khi các mô hình khác thiếu chi tiết đối tượng hoặc không thể chụp được tất cả các đối tượng trong ảnh.

1. **Note**

- Hiệu quả của việc mở rộng quy mô mô hình phụ thuộc rất nhiều vào mạng cơ sở.

Ưu điểm của EfficientNets:

+ Cải thiện hiệu suất đồng thời tăng hiệu quả tính toán bằng cách giảm số lượng tham số và FLOP

+ Cung cấp phương pháp chia tỷ lệ mới có thể áp dụng cho các mạng khác nhau để cải thiện hiệu suất.

Nhược điểm:

+ Đối với các mô hình EfficientNets như B6, B7 vẫn cần nhiều tài nguyên để tính toán.