SE and Conditional Convolution for image classification

An-Yu Liu
EmailAddress@utdallas.edu

Abstract

Introduce the residual and identity concept which derives from the network in section 2-Related work, into a relatively standard ImageNet structure. Further improve it with SE and conditional convolution to see the performance of the network.

1. Introduction

The traditional way of increasing accuracy and performance of the xNN is scaling. However, scaling the network requires huge amount of resource to train the network which is often not feasible on mobile or embedded device. Recently introduced network such as MobileNet seires[2,5], MnasNet [3] etc provide a way to balance the performance and latency.

The ImageNet structure in this essay is relatively standard (Figure 1). First, we introduce a standard block with identity and residual parts. Second, improve the block with SE enhanced and Lastly further improve it with conditional convolution. Hyper parameters from the network list in Table 2. Training time and accuracy in Table 3 and per operation and MAC and coefficient counts in Table 4.

2. Related Work

The network in this paper was heavily inspired by SENet [1], MobileNetV2 [2], MnasNet [3], CondConv [4], MobileNetV3 [5] and EfficientNet [6].

MobileNetV2 [2] used inverted residuals to achieve or even beat the state of the art along multiple performance points without requiring large memory. Therefore, this is suitable for mobile device and embedded applications.

MnasNet [3] added squeeze and excite operations to the inverted residuals and used neural architecture search to optimize the network depth and width. A novel factorized hierarchical search space simplifies the search process and precludes layer diversity. In terms of accuracy vs inference latency, MnasNet is the best by the time this paper is published.

MobileNetV3 [5] replaced the swish nonlinearity with a hard swish nonlinearity, improved the design of the final encoder stages and used a new neural architecture search to optimize the depth and width. The tools for architecture search are Network Search and NetAdapt which allows fine-tuning of individual layers in a sequential manner.

EfficientNet [6] refined the mobile baseline network and focused on the combined scaling of network input resolution, depth and width to design larger networks. There exists certain relationship between network width and depth. This essay quantify the relationship between network width, depth, and resolution. Compound scaling method uniformly scales network width, depth, and resolution with a set of coefficients and these coefficients are determined by a grid search on the original small model.

A variant of EfficientNet replaced the convolution operations with conditional convolution operations. Conditional convolution operations increases the size and capacity of a network without sacrifice the efficient inference.

3. Design

It`s a relatively standard ImageNet structure in figure 1 and table 1 with the 1st 2 stride by 2 operations change to a stride by 1 operation as the image size has 1/4 the rows and cols of typical ImageNet images. The encode tail part is composed of 3x3 Convolution, Batch Norm and Relu, body part is 5 building blocks. The head is made of global avg pooling, Linear with bias.

The first one is the standard building block shown in Figure 2A. Residual combine with identity, while the residual first go through a 1x1 convolution, bath norm, Relu then FxF/S group convolution, batch norm, Relu and final one with 1x1 convolution and batch norm. Combining with identity becomes output of the block. SE is shown in Figure 2B, adding elements shown in Figure 3 into the standard building block. SE and conditional convolution enhanced building block (Figure 4 for details) introduces conditional convolution into the block. Conditional convolution uses 3D / 4D convolution weight tensors Wm from M experts combined to a single 3D / 4D weight tensor W via a learned input dependent weighted sum with fully grouped convolution / not fully grouped convolution.

This design has several differences with EfficientNet: this design only used 3x3 filters in the fully grouped convolution, modified the stem width, modified the depth and width of various blocks, modified the inverted residual expansion ratio, simplified the nonlinearity choice to ReLU (or Sigmoid where appropriate), only included 3 levels of down sampling as the cropped input is 3x56x56.

For implementations based on the SE enhanced building block, the internal rank reduction ratio R = 4.

For implementations based on SE and conditional convolution enhanced building blocks, the number of experts M = 4.

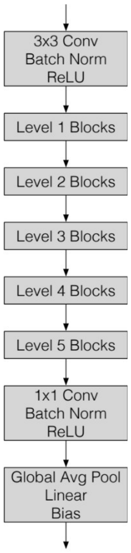


Figure 1: Network structure; the linear layer output dimension and bias dimension is the number of classes

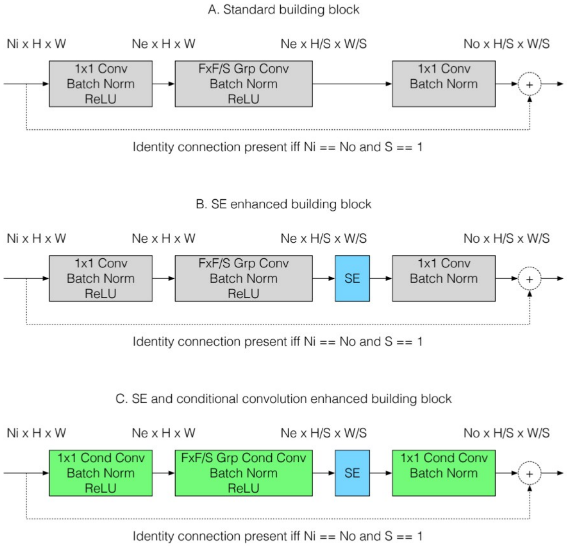


Figure 2: [A] Standard building block, [B] SE enhanced building block and [C] SE and conditional convolution enhanced building block

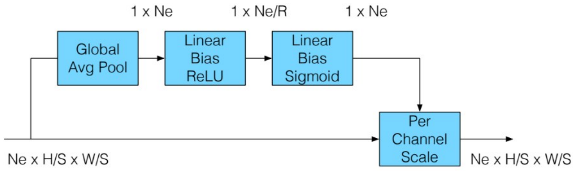


Figure 3: Squeeze and excite uses a learned input dependent per channel weighting to re weight input feature maps with internal rank reduction R

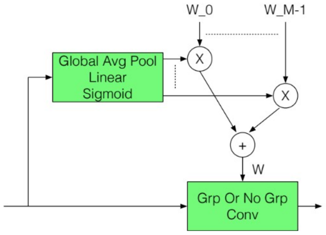


Figure 4: Conditional convolution uses 3D / 4D convolution weight tensors W_m from M experts combined to a single 3D / 4D weight tensor W via a learned input dependent weighted sum with fully grouped convolution / not fully grouped convolution

Re-peat	Input NixWxH	Operation
1	3x56x56	Conv (3x3/1), Batch Norm, ReLU
1	16x56x56	Block (Ne=4Ni, F=3, S=1, ID=True)
1	16x56x56	Block (Ne=4Ni, F=3, S=1, ID=False)
1	24x56x56	Block (Ne=4Ni, F=3, S=1, ID=True)
1	24x56x56	Block (Ne=4Ni, F=3, S=2, ID=False)
2	40x28x28	Block (Ne=4Ni, F=3, S=1, ID=True)
1	40x28x28	Block (Ne=4Ni, F=3, S=2, ID=False)
3	80x14x14	Block (Ne=4Ni, F=3, S=1, ID=True)

1	80x14x14	Block (Ne=4Ni, F=3, S=2, ID=False)
4	160x7x7	Block (Ne=4Ni, F=3, S=1, ID=True)
1	160x7x7	Block (Ne=4Ni, F=3, S=1, ID=False)
1	320x7x7	Conv (1x1/1), Batch Norm, ReLU
1	1280x7x7	Global Avg Pool, Linear, Bias
1	1x100	Output

Table 1: Network specification; block is either a [A] standard building block, [B] SE enhanced building block or [C] conditional convolution enhanced building block

4. Training

Table 2 includes a summary of all training hyper parameters. Note that this is a ~ generic ImageNet training routine such as you would find in RegNetX/Y [7]. Training routines that use more complex data augmentation, additional data, different train and test resolutions, more epochs, … can achieve higher accuracies.

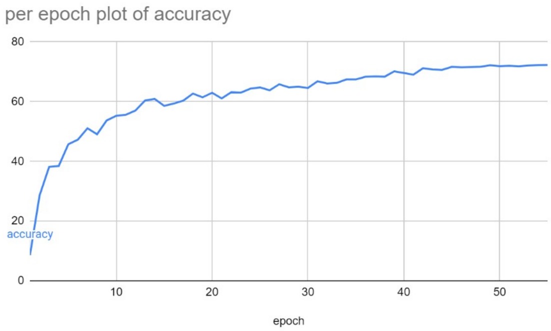
Table 3 includes final training results and figure 5 shows a plot of the per epoch accuracy and loss curves.

Parameter	Value
Learning rate	0.000200
epochs	55
Resolutions	56*56
...	...

Table 2: Training hyper parameters

Block	Training time	Accuracy
Standard	5 hours	72.26
SE enhanced	4 hours	74.04
SE and cond conv enhanced	Optional	Optional

Table 3: Training final results



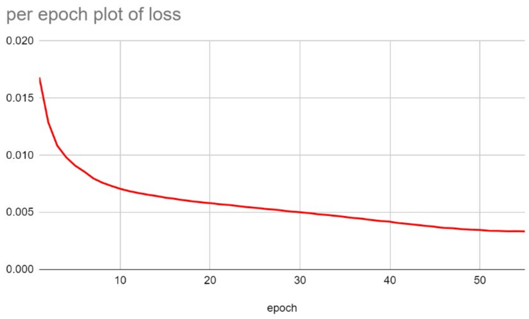


Figure 5: Training per epoch accuracy and loss curves

5. Implementation

Table 4 shows per operation MACs and number of filter coefficients for the stem convolution, convolutions in all standard blocks (taking into account repeats) and the head convolution and matrix multiplication, along with their sum for the full network.

Operation	Rep	MAC	Filter Cx
Conv (3x3/1), Batch Norm, ReLU	1	1354752	432
Block (Ne=4Ni, F=3, S=1, ID=True)	1	8228864	2624
Block (Ne=4Ni, F=3, S=1, ID=False)	1	9834496	3136
Block (Ne=4Ni, F=3, S=1, ID=True)	1	17160192	7008
Block (Ne=4Ni, F=3, S=2, ID=False)	1	10913280	5472
Block (Ne=4Ni, F=3, S=1, ID=True)	2	22328320	7008
Block (Ne=4Ni, F=3, S=2, ID=False)	1	7808640	28480
Block (Ne=4Ni, F=3, S=1, ID=True)	3	31799040	20640
Block (Ne=4Ni, F=3, S=2, ID=False)	1	7667520	162240
Block (Ne=4Ni, F=3, S=1, ID=True)	4	41269760	79680
Block (Ne=4Ni, F=3, S=1, ID=False)	1	15335040	842240
Conv (1x1/1), Batch Norm, ReLU	1	20070400	312960
Global Avg Pool, Linear, Bias	1	0	409600
Total	–	193770304	1874512

Table 4: Per operation and total MAC and filter coefficient counts for all trainable operations

6. Conclusion

This network structure is heavily influenced by the network listed in related work. Starting from the standard block for the network and gradually add SE and conditional convolution into the block. Hyper parameters, training time and accuracy are listed in Table 2 and 3 respectively. Lastly the Table 4 recorded the per operation, MAC, coefficient counts.

In the near future, I hope to modify the network with different filter sizes or change the scale of the network to see the training time vs accuracy performance. For individual block, try different SE rand reduction factors, different numbers of mixtures of experts. On top of that, using different block in the network design for future modifications.

References

[1] J. Hu et. al., "Squeeze-and-excitation networks," arXiv:1709.01507, 2017.

[2] M. Sandler et. al., "MobileNetV2: inverted residuals and linear bottlenecks," axXiv:1801.04381, 2018.

[3] M. Tan et. al., "MnasNet: platform-aware neural architecture search for mobile," arXiv:1807.11626, 2018.

[4] B. Yang et. al., "CondConv: conditionally parameterized convolutions for efficient inference," arXiv:1904.04971, 2019.

[5] A. Howard et. al., "Searching for MobileNetV3," arXiv:1905.02244, 2019.

[6] M. Tan and Q. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," arXiv:1905.11946, 2019.

[7] I. Radosavovic et. al., "Designing network design spaces," arXiv:2003.13678, 2020.