An Analysis of Deep Neural Network Models for Practical Applications

brAIns Paper Study

안윤표[1]

[1]UNIST - MLV Lab
brAIns - brew AI neo scientists

Sep 17, 2021

Table of Contents

Table of Contents

ImageNet Challenge

Figure: ImageNet Challenge

Source: `https://paperswithcode.com/dataset/imagenet`

ImageNet Challenge

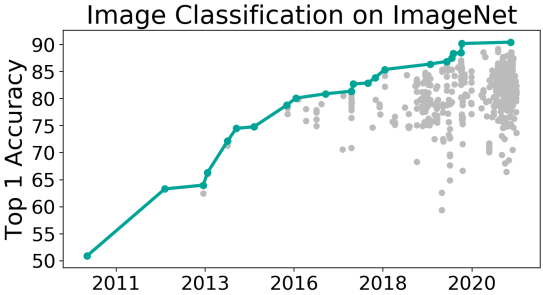

Figure: ImageNet Challenge

Source: `https://paperswithcode.com/dataset/imagenet`

ImageNet Challenge의 한계점

오직 정확도(Accuracy)만 측정한다!

ImageNet Challenge의 한계점

당신의 선택은?

- 정확도 100%를 가지지만 한 결과에 72시간 걸리는 모델
- 정확도 80%를 가지고 1분이면 결과가 나오는 모델

평가요소

- Accuracy
- Memory Footprint
- Parameters
- Operations Count
- Inference Time
- Power Consumption

Operations Count

Operation Per second (OPs)

OPs := # of Operations / second

GOPs := Giga OPs = OPs $\times 10^9$

Table of Contents

Models

- AlexNet [KSH17]
- batch normalized AlexNet
- batch normalized Network in Network [LCY14]
- ENet [PCKC16]
- GoogLeNet [SLJ+14]
- VGG-16 and -19 [SZ15]
- ResNet-18, -34, -50, -101 and -152 [HZRS15]
- Inception-v3 [SVI+15]
- Inception-v4 [SIVA16]

AlexNet

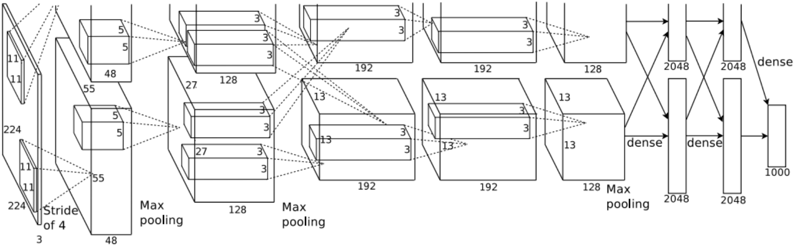

Figure: AlexNet 핵심 구조 [KSH17]

Network In Network

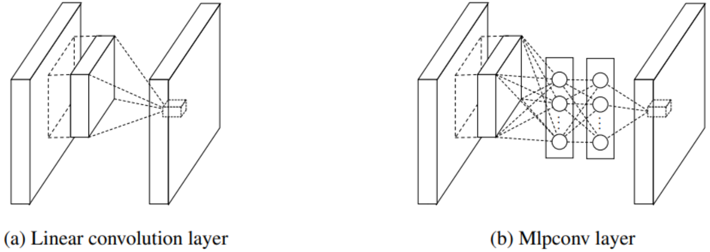

Figure: Linear Convolution Layer vs Mlpconv layer [LCY14]

Network In Network

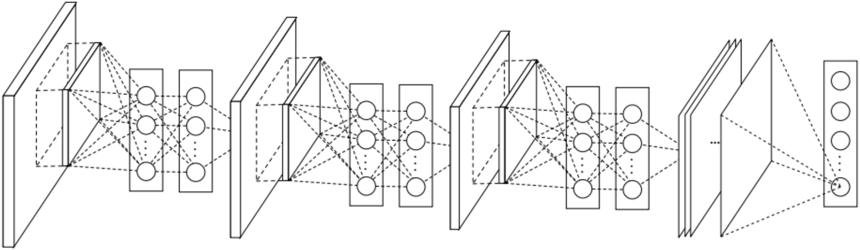

Figure: Network in Network 전체 구조 [LCY14]

ENet

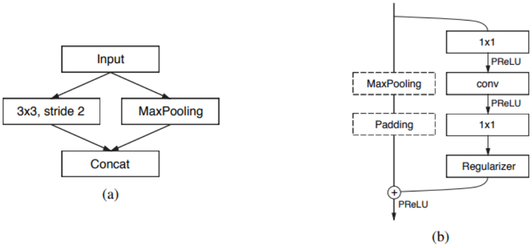

Figure: ENet 구조 [PCKC16]

GoogLeNet

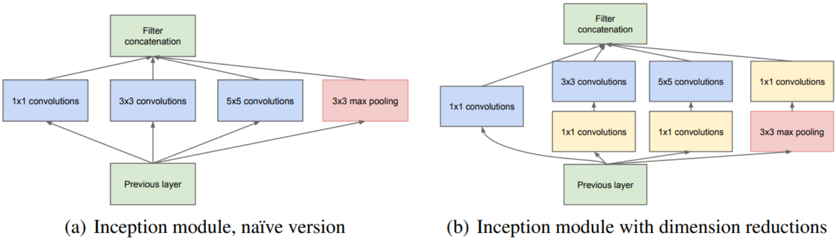

Figure: GoogLeNet 모듈둘 [SLJ+14]

GoogLeNet (a.k.a. Inception v1)

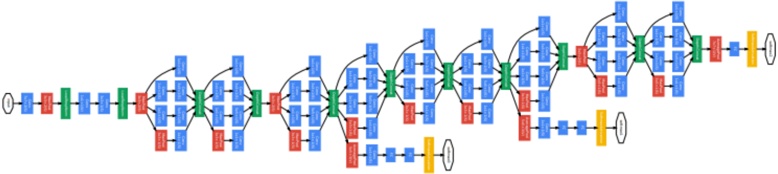

Figure: GoogLeNet 전체 구조 [SLJ+14]

VGGNet

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure: VGGNet 전체 구조 [SZ15]

ResNet

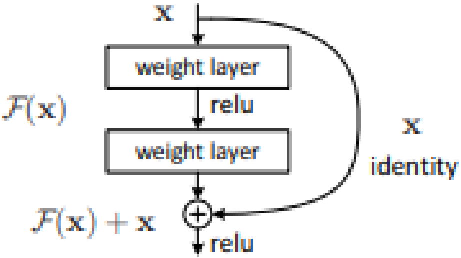

Figure: Residual Block [HZRS15]

ResNet

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				
		[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	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

Figure: ResNet 전체 구조 [HZRS15]

Table of Contents

Machine

- Torch7 with cuDNN-v5 and CUDA-v8 backend
- Experiment on JetPack-2.3 NVIDIA Jetson TX1 board: 64-bit ARM A57 CPU, a 1 T-Flop/s 256-core NVIDIA Maxwell GPU and 4GB shared RAM
- Measuring power consumtion by Keysight MSO-X 2024A 200MHz digital oscilloscope

Accuracy

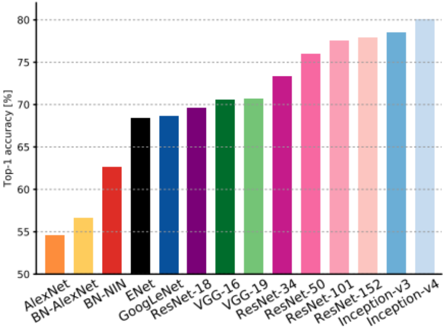

Figure: Model / Accuracy [CPC17]

Accuracy

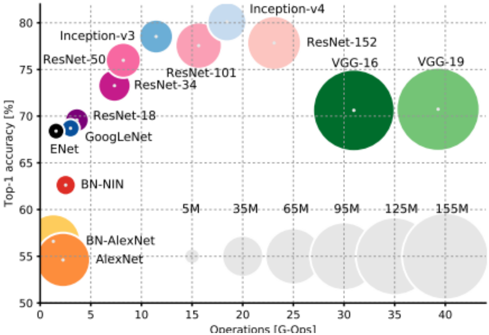

Figure: Operation / Accuracy + Parameters [CPC17]

Inference Time

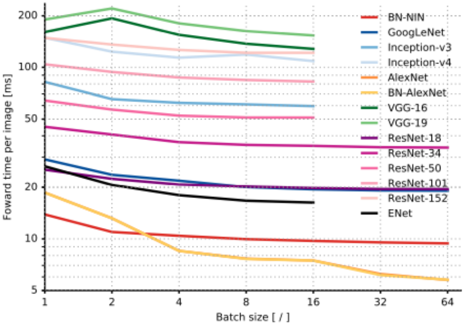

Figure: Batch Size / Inference Time [CPC17]

Power

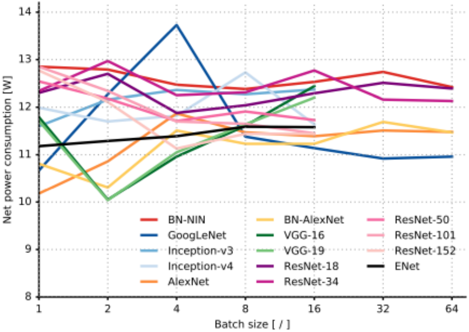

Figure: Batch Size / Power [CPC17]

Memory

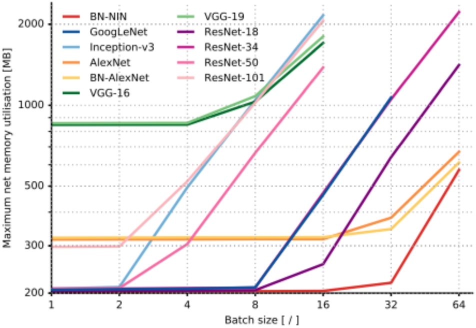

Figure: Batch Size / Memory [CPC17]

Memory

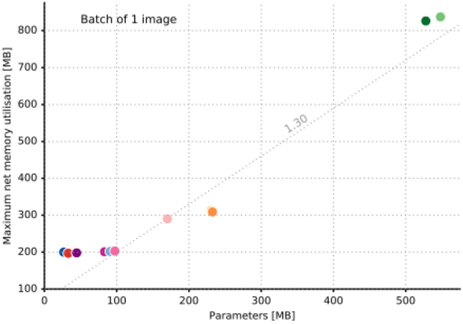

Figure: Parameters / Memory [CPC17]

Operations

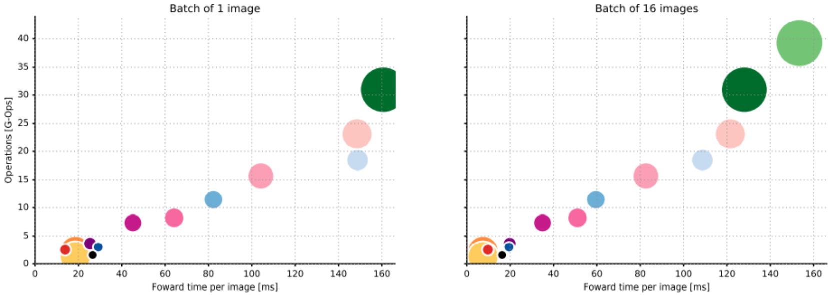

Figure: Inference Time / Operations Left: batch size 1 Right: batch size 16 [CPC17]

Operations and Power

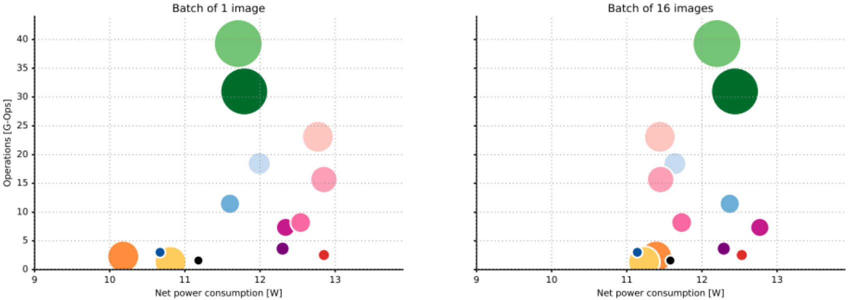

Figure: Power / Operations Left: batch size 1 Right: batch size 16 [CPC17]

Accuracy and Throughput

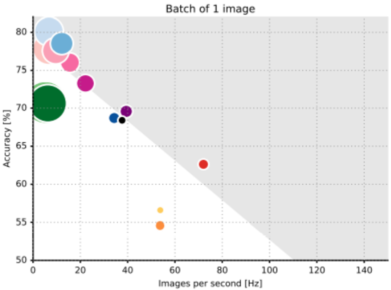

Figure: Inference per Second / Operations Left: batch size 1 Right: batch size 16 [CPC17]

Parameters Utilization

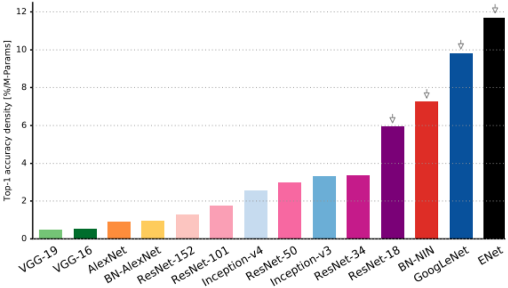

Figure: Accuracy per Parameter Left: batch size 1 Right: batch size 16 [CPC17]

Table of Contents

Conclusion

- 한정된 자원 아래에서 ImageNet 문제는 ENet이 가장 효과적이다.
- Accuracy와 Inference Time은 Hyperbolic 관계이다.
- Operation Count를 통하여 효과적으로 Inference Time을 유추할 수 있다.
- Power Consumption은 최대 정확도와 모델의 복잡도에 의해 결정된다.
- Power Consumption은 Batch size와 Architecture과는 무관하다.

Pros and Cons

장점

- 여러 모델에 대해 여러한 요소들을 고려하여 실험을 진행함.
- 관계를 통하여 현재 모델에 어떠한 한계점이 존재하는지, 어떠한 trade-off를 얻을 수 있는지 설명함.
- 2017년에 나온 논문이지만, 현재 모델의 평가 방법에 참고할 수 있다.

단점

- 상관관계가 "왜" 일어났는지에 대한 설명이 부족함. 상관관계의 원인에 대한 분석이 요구된다.
- ImageNet에 분석이 한정되어 있다. 다른 데이터셋에 대한 추가적인 분석이 필요하다.

References I

- Alfredo Canziani, Adam Paszke, and Eugenio Culurciello, *An analysis of deep neural network models for practical applications*, 2017.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, *Deep residual learning for image recognition*, 2015.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, *Imagenet classification with deep convolutional neural networks*, Commun. ACM **60** (2017), no. 6, 84–90.
- Min Lin, Qiang Chen, and Shuicheng Yan, *Network in network*, 2014.
- Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello, *Enet: A deep neural network architecture for real-time semantic segmentation*, 2016.

References II

- Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi, *Inception-v4, inception-resnet and the impact of residual connections on learning*, 2016.
- Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, *Going deeper with convolutions*, 2014.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna, *Rethinking the inception architecture for computer vision*, 2015.
- Karen Simonyan and Andrew Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015.

Questions?

Fin