📖

[논문 리뷰] SqueezeNet [2016]

Introduction

✨ **해당 논문은 three Advantage를 가진 samll CNN Architecture인 SqueezeNet을 소개합니다.**

Smaller CNN Architectures three Adavantage

- **CNN의 모델이 작을수록, 효율적인 분산 학습이 가능**합니다.
- **새 모델을 클라이언트에 내보낼 때**, 처리시간이 덜 걸립니다.
- **메모리가 제한된 기타 하드웨어 및 FPGAs에 배포하는데 적합**합니다.

> 💡 **SqueezeNet에 관한 연구는 CNN Architecture에 대한 접근법을 시사하며, 더 다양한 방식을 제공합니다.**

Related Work

Model Compression

가장 중요한 것은 **적은 parameter를 사용하되, 정확도를 보존**하는 것이다.
SVD는 사전 훈련 된 CNN 모델에 특이 값 분해 (SVD)를 적용하여 모델을 압축하는 것입니다.

CNN Microarchitecture

최근의 **CNN 구조에서**는, **각각의 layer에서 filter의 차원을 택하는 것이 매우 성가십니다.**
논문에서는 **GoogleNet과 관련하여 설명**을 하고 있으며, 마지막으로 **particular organization과 dimension of the individual models들을 CNN microarchitecture**라고 부릅니다.

CNN Macroarchitecture

여러 모듈을 **end-to-end CNN architecture**로 구성하는 **System Architecture인 CNN macroarchitecture를 정의**합니다.
ResNet 및 VGG등의 각각의 layer들의 연결방식과 관련하여 설명을 하고 있습니다.

Neural Network Design Space Exploration

CNN 거대한 구조를 가지고 있으며, 구조 내에는 microarchitecture, macroarchitecture, solvers, hyperparameters를 가지고 있습니다.

SqueezeNet Architecture

✨ **few parameter를 가진 CNN Architecture**에 대한 **Design Strategies를 설명으로 시작**됩니다.

Architectural Design Strategies

- **Replace 3x3 filters with 1x1 filters**

- **Given a budget of certain number of convolution filters,**
  3x3보다는 9배 적은 1x1을 선택합니다.

- **Decrease the number of input channels to 3x3 filters**
  - **parameter의 total quantity**는 (input * number of filters * 3*3)입니다.
    CNN에서 적은 parameter를 유지하기 위해서는, **filter뿐만 아니라 input도 줄여야 한다.**

- **DownSample late in the network so that convolution layers have large activation**
  - CNN의 각 layer들은 **Spatial resolution**에서
    최소 1x1에서 1x1 큰 output activation map을 생성합니다.
  - 저자들은 지연된 다운 샘플링의 큰 activation map이 다른 모든 맵들이 동일하게 유지될 때 더 높은 분류 정확도로 이어질 수 있다는 것이다.
    4개의 **서로 다른 CNN 구조에서 지연된 다운 샘플링을 적용**하였으며, 각각의 경우 **지연된 다운샘플링은 더 높은 분류 정확도**로 이어진다.

Fire Module

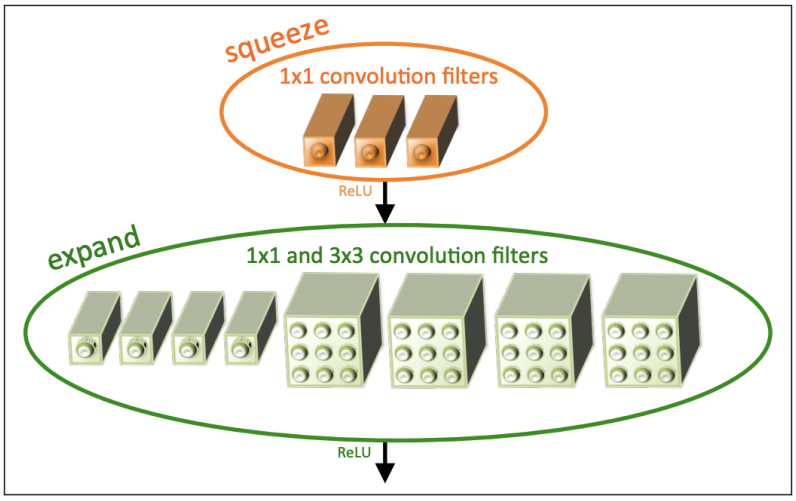


Figure 1: Microarchitectural view: Organization of convolution filters in the **Fire module**. In this example, $s_{1x1} = 3$, $e_{1x1} = 4$, and $e_{3x3} = 4$. We illustrate the convolution filters but not the activations.

Fire Module에서 s1x1, e1x1, e3x3의 3가지 조정 가능한 하이퍼 매개변수가 존재한다. fire module에는 s1x1의 squeeze 층의 필터, e1x1 expand층의 1x1 필터, e3x3는 확장 층의 3x3 필터이다.
Fire Module을 사용할 때는 s1x1을 (e1x1 + e3x3)미만으로 설정해야만 합니다.

Squeeze Architecture

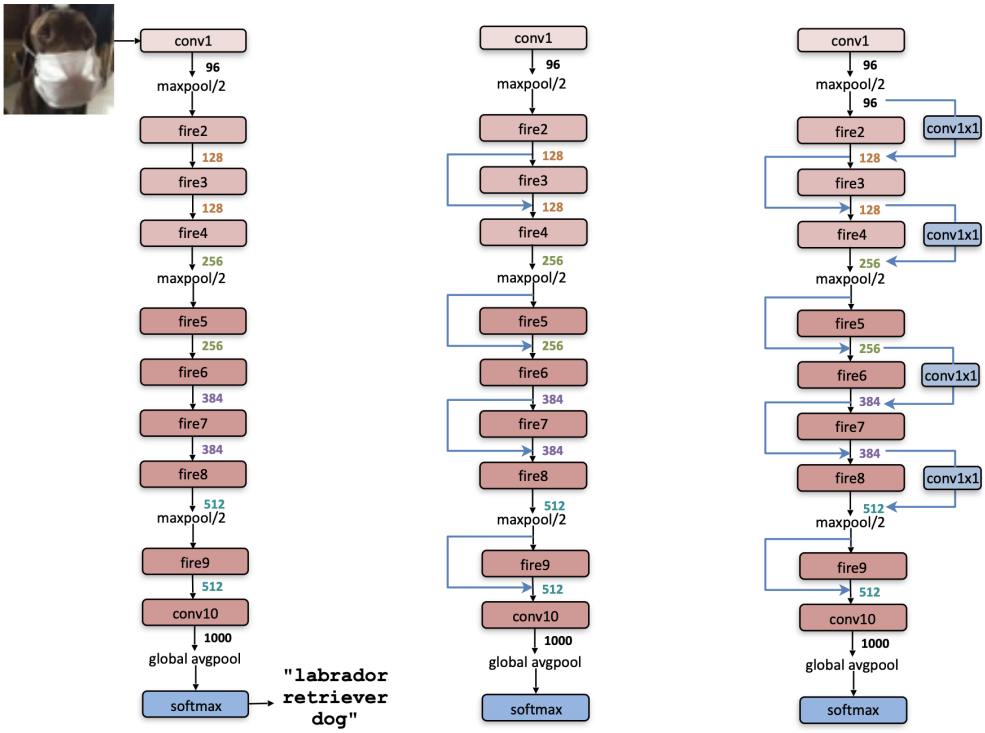


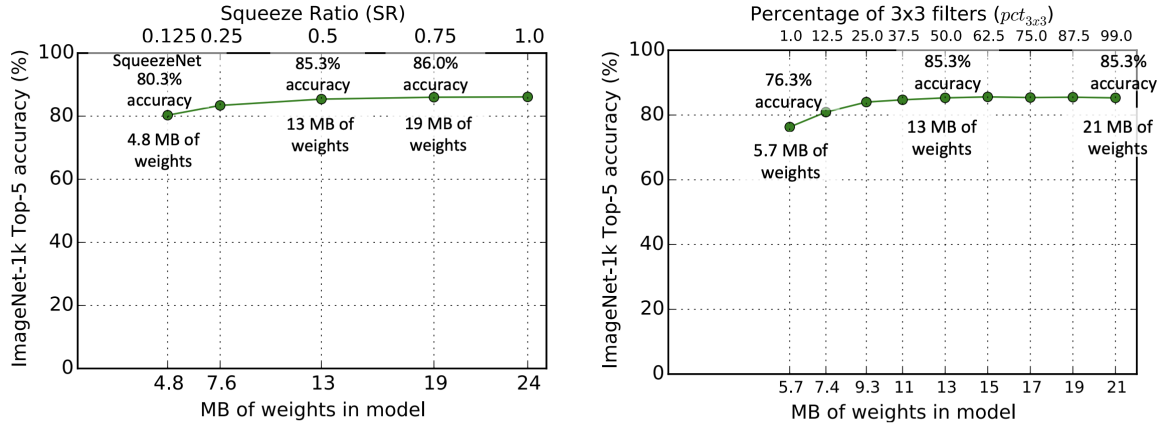
Figure 2: Macroarchitectural view of our SqueezeNet architecture. Left: SqueezeNet (Section 3.3); Middle: SqueezeNet with simple bypass (Section 6); Right: SqueezeNet with complex bypass (Section 6).

- 1x1 과 3x3 필터의 Output activation이 같기에,
  expand module의 3x3의 필터로 들어가는 데이터에 1-Pixel Zero Padding을 추가합니다.
- Squeeze expand layer 모두 ReLU 적용 및 Fire Module에는 Dropout 50% 이후 적용
- 초기 learning Rate 을 0.04로 설정한 후, 감소시킵니다.

Table 1: SqueezeNet architectural dimensions. (The formatting of this table was inspired by the Inception2 paper (Ioffe & Szegedy, 2015).)

layer name/type	output size	filter size / stride (if not a fire layer)	depth	s_{1x1} (#1x1 squeeze)	e_{1x1} (#1x1 expand)	e_{3x3} (#3x3 expand)	s_{1x1} sparsity	e_{1x1} sparsity	e_{3x3} sparsity	# bits	#parameter before pruning	#parameter after pruning
input image	224x224x3										-	-
conv1	111x111x96	7x7/2 (x96)	1				100% (7x7)			6bit	14,208	14,208
maxpool1	55x55x96	3x3/2	0									
fire2	55x55x128		2	16	64	64	100%	100%	**33%**	6bit	11,920	5,746
fire3	55x55x128		2	16	64	64	100%	100%	**33%**	6bit	12,432	6,258
fire4	55x55x256		2	32	128	128	100%	100%	**33%**	6bit	45,344	20,646
maxpool4	27x27x256	3x3/2	0									
fire5	27x27x256		2	32	128	128	100%	100%	**33%**	6bit	49,440	24,742
fire6	27x27x384		2	48	192	192	100%	**50%**	**33%**	6bit	104,880	44,700
fire7	27x27x384		2	48	192	192	**50%**	100%	**33%**	6bit	111,024	46,236
fire8	27x27x512		2	64	256	256	100%	**50%**	**33%**	6bit	188,992	77,581
maxpool8	13x12x512	3x3/2	0									
fire9	13x13x512		2	64	256	256	**50%**	100%	**30%**	6bit	197,184	77,581
conv10	13x13x1000	1x1/1 (x1000)	1				**20%** (3x3)			6bit	513,000	103,400
avgpool10	1x1x1000	13x13/1	0									
	activations	parameters					compression info				1,248,424 (total)	**421,098** (total)

Evaluation of SqueezeNet



(a) Exploring the impact of the squeeze ratio (SR) on model size and accuracy.

(b) Exploring the impact of the ratio of 3x3 filters in expand layers (pct_{3x3}) on model size and accuracy.

Figure 3: Microarchitectural design space exploration.

Conclusion

해당 논문에서는 CNN의 Design space에 관하여 체계적인 접근방식을 제안하였습니다.

본 저자들은 SqueezeNet이 다른 독자들이 CNN exploration을 보다 체계적인 방식으로 넓은 범위를 연구할 수 있다는 것을 시사하였습니다.

SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size

Recent research on deep neural networks has focused primarily on improving accuracy. For a given accuracy level, it is typically possible to identify multiple DNN architectures that achieve that accuracy level. With equivalent accuracy, smaller DNN architectures offer at least three

https://arxiv.org/abs/1602.07360