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

Authors: Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer

Presented by: Mingyang Zheng (mz2594), Lingyu Zhang (lz2494)

Outline

- Goal
- Related Work
- Architectural Design Strategies
- Fire Module and SqueezeNet Architecture
- Microarchitecture Design
- Microarchitecture Design

Goal

Identify a model that has very few parameters while preserving accuracy

Advantages:

- Smaller CNNs require less communication across servers during distributed training.
- Smaller CNNs require less bandwidth to export a new model from the cloud to an autonomous car.
- Smaller CNNs are more feasible to deploy on FPGAs and other hardware with limited memory

Related Work

- **MODEL COMPRESSION**
  - Apply singular value decomposition (SVD) to a pretrained CNN model (Denton et al., 2014)
  - Network Pruning: pretrained model, threshold with zeros to form a sparse matrix, iterations sparse CNN (Han et al., 2015b).
  - Deep Compression: Network Pruning + quantization + huffman encoding
- **CNN MICROARCHITECTURE**
  - With the trend of designing very deep CNNs, it becomes cumbersome to manually select filter dimensions
  - Module: comprised of multiple convolution layers with a specific fixed organization
- **DESIGN SPACE EXPLORATION**
  - Depth
  - The choice of connections across multiple layers
  - Developing automated approaches for finding NN architectures

Related Work

- LeNet-5 (1998)
- AlexNet (2012)
- OverFeat (2013)
- VGG-16, VGG-19 (2014)
- GoogLeNet (2014)
- PReLUnet (2015)
- ResNet-50, ResNet-101, ResNet-152 (2015)
- SqueezeNet (2016)
- Stochastic Depth (2016)
- ResNet-200, ResNet-1001 (2016)

Architectural Design Strategies

- **Strategy 1:**

  Replace 3x3 filters with 1x1 filters:

  - 9X fewer parameters

- **Strategy 2:**

  Decrease the number of input channels to 3x3 filters

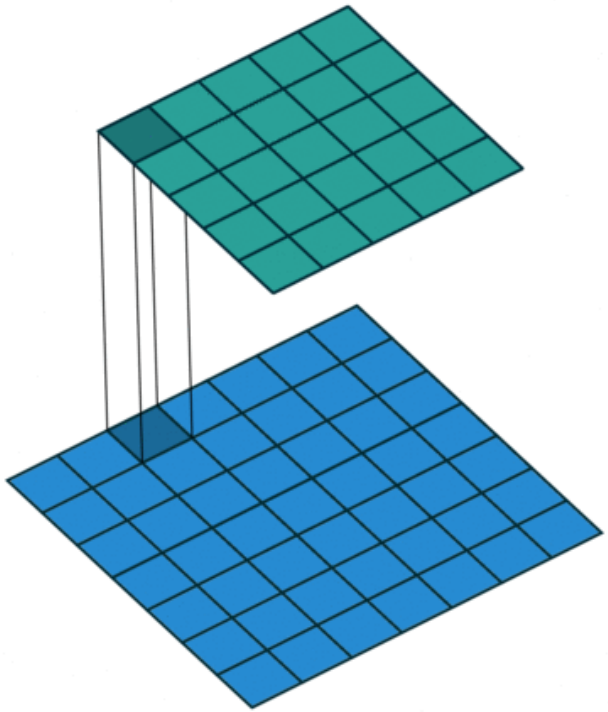
  - (number of input channels) * (number of filters) * (3*3).

- **Strategy 3:**

  Downsample late in the network so that convolution layers have large activation maps

  - Intuition: delayed downsampling => large activation maps => higher classification accuracy, with all else held equal

Architectural Design Strategies: strategy 1x1



- 1x1 convolution acts like coordinate-dependent transformation in the filter space. It is important to note here that this transformation is strictly linear, but in most of application of 1x1 convolution, it is succeeded by a non-linear activation layer like ReLU. This transformation is learned through the (stochastic) gradient descent. But an important distinction is that it suffers with less over-fitting due to smaller kernel size (1x1).
- Combination of 1x1 (x F) convolution is mathematically equivalent to a multi-layer perceptron

Fire module

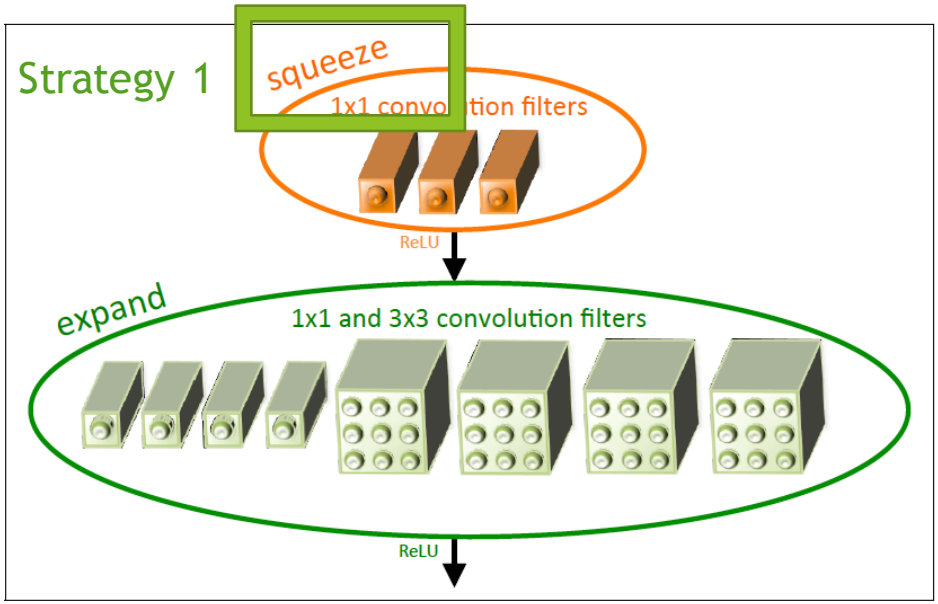


Figure 1: Microarchitectural view: Organization of convolution filters in the **Fire module**. In this example, $s_{1x1} = 3$, $e_{1x1} = 4$, and $e_{3x3} = 4$. We illustrate the convolution filters but not the activations.

s1x1 <(e1x1 + e3x3), Strategy 2

SqueezeNet Architecture

Strategy 3

SqueezeNet Architecture: dimensions

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)

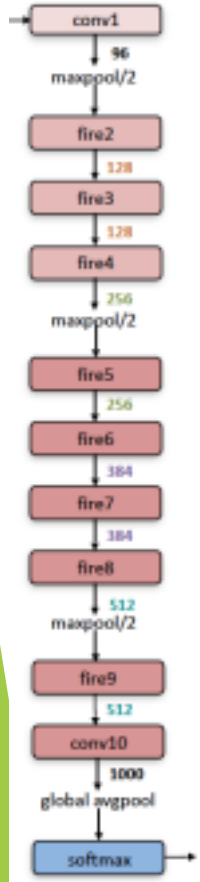
SqueezeNet Architecture:
Comparing SqueezeNet to Model Compression Approaches

Table 2: Comparing SqueezeNet to model compression approaches. By *model size*, we mean the number of bytes required to store all of the parameters in the trained model.

CNN architecture	Compression Approach	Data Type	Original → Compressed Model Size	Reduction in Model Size vs. AlexNet	Top-1 ImageNet Accuracy	Top-5 ImageNet Accuracy
AlexNet	None (baseline)	32 bit	240MB	1x	57.2%	80.3%
AlexNet	SVD (Denton et al., 2014)	32 bit	240MB → 48MB	5x	56.0%	79.4%
AlexNet	Network Pruning (Han et al., 2015b)	32 bit	240MB → 27MB	9x	57.2%	80.3%
AlexNet	Deep Compression (Han et al., 2015a)	5-8 bit	240MB → 6.9MB	35x	57.2%	80.3%
SqueezeNet (ours)	None	32 bit	4.8MB	**50x**	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	8 bit	4.8MB → 0.66MB	**[illegible]**	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	6 bit	4.8MB → 0.47MB	**510x**	57.5%	80.3%

In summary:
by combining CNN architectural innovation (SqueezeNet) with state-of-the-art compression techniques (Deep Compression), they achieved a 510 reduction in model size with no decrease in accuracy compared to the baseline.

Design Space Exploration: Microarchitecture-- metaparameters



- In SqueezeNet, each Fire module has three dimensional hyperparameters : s1x1 , e1x1 , and e3x3 . SqueezeNet has 8 Fire modules with a total of 24 dimensional hyperparameters.

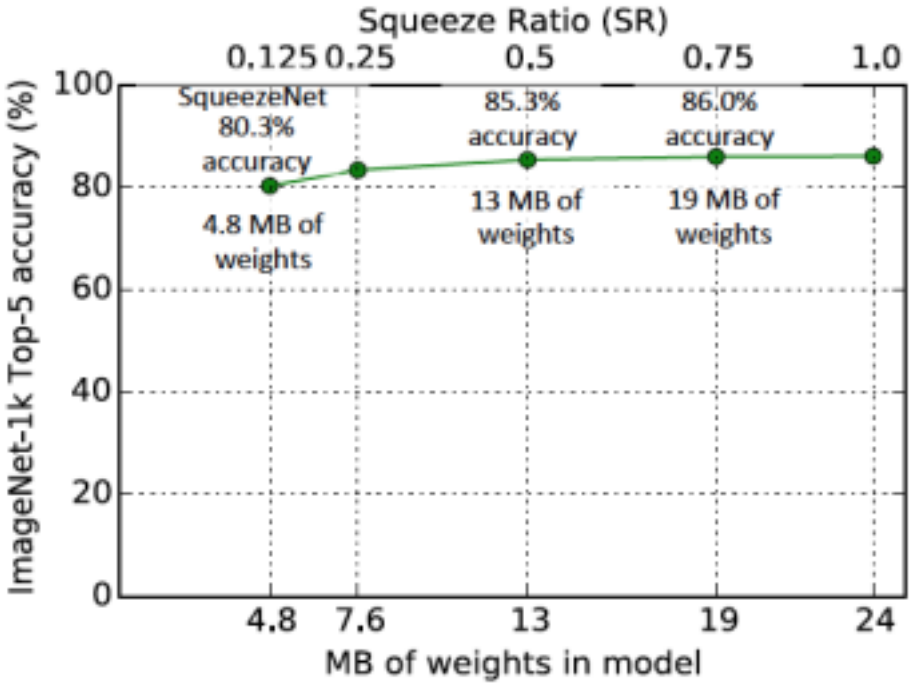
To do broad sweeps of the design space of SqueezeNet-like architectures, metaparameters: control the dimensions of all Fire modules in a CNN.

$$e_i = base_e + (incr_e * \left\lfloor \frac{i}{freq} \right\rfloor)$$

$$s_{i,1x1} = SR * (e_{i,1x1} + e_{i,3x3})$$

$base_e = 128$, $incr_e = 128$, $pct_{3x3} = 0.5$, $freq = 2$, and $SR = 0.125$.

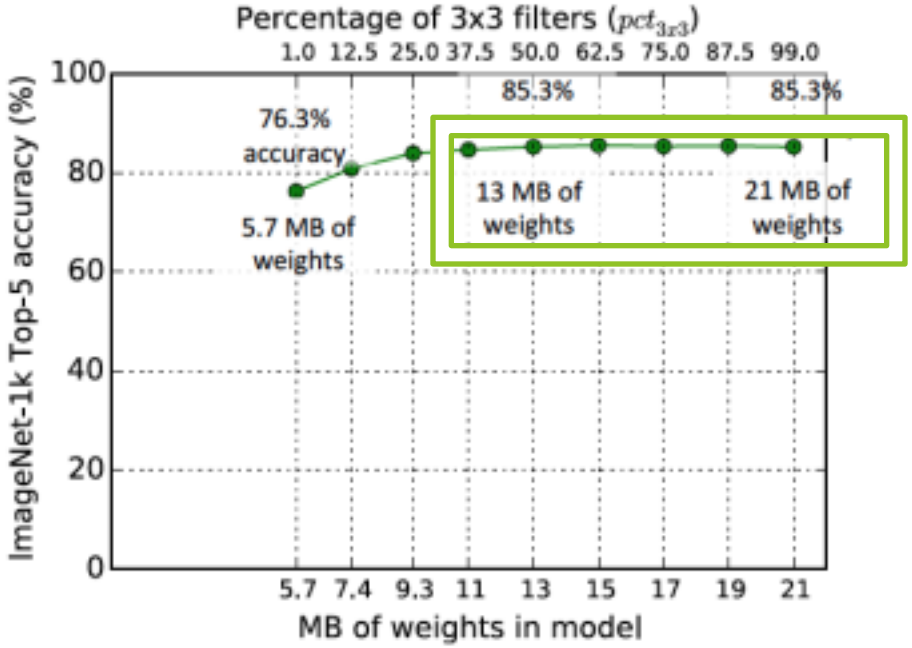
Design Space Exploration: Microarchitecture – squeeze ratio



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

Conclusion: increasing SR beyond 0.125 can further increase ImageNet top-5 accuracy from 80.3% (i.e. AlexNet-level) with a 4.8MB model to 86.0% with a 19MB model.

Design Space Exploration: Microarchitecture – 1x1 & 3x3

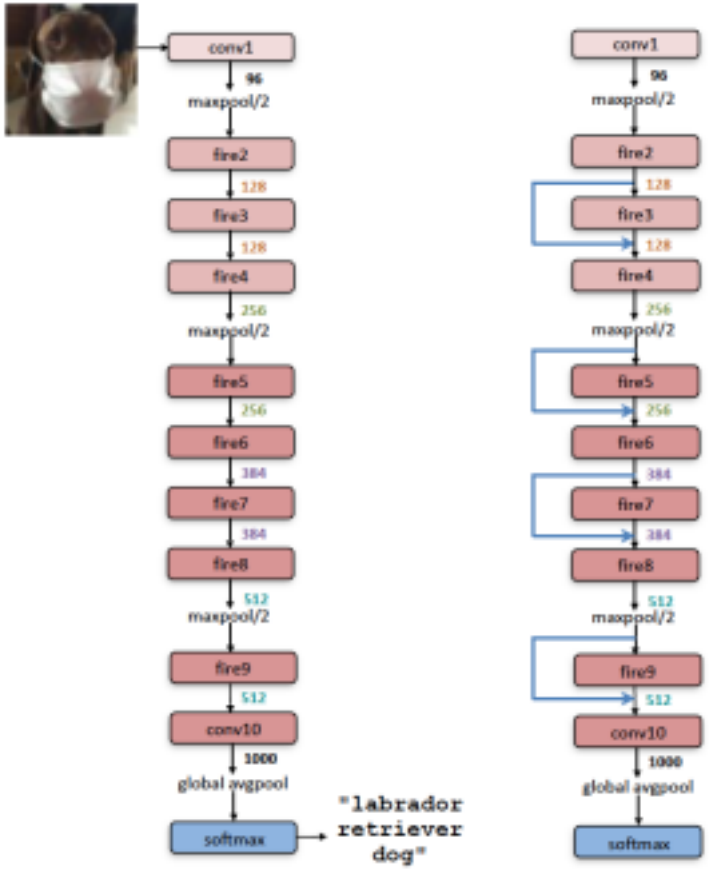


(b) Exploring the impact of the ratio of 3x3 filters in expand layers (pct_{3x3}) on model size and accuracy.

Conclusion:

the top-5 accuracy plateaus at 85.6% using 50% 3x3 filters, and further increasing the percentage of 3x3 filters leads to a larger model size but provides no improvement in accuracy on ImageNet

Design Space Exploration: Macroarchitecture – simple bypass



In SqueezeNet, the squeeze ratio (SR) is 0.125, meaning that every squeeze layer has 8x fewer output channels than the accompanying expand layer.

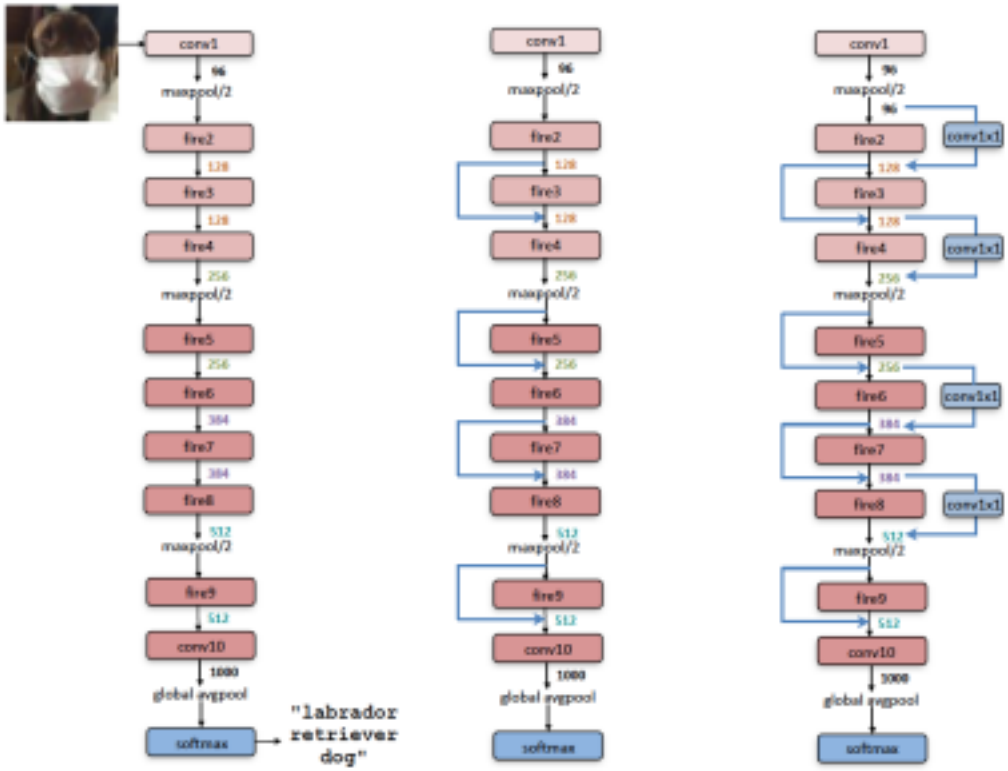
Due to this severe dimensionality reduction, a limited amount of information can pass through squeeze layers. -- bottleneck

$$e_i = base_e + (incr_e * \left\lfloor \frac{i}{freq} \right\rfloor)$$

$$freq = 2,$$

only half of the Fire modules can have simple bypass connections

Design Space Exploration: Macroarchitecture – complex bypass



complex bypass: includes a 1x1 convolution layer with the number of filters set equal to the number of output channels that are needed

Design Space Exploration: Macroarchitecture – results

Table 3: SqueezeNet accuracy and model size using different macroarchitecture configurations

Architecture	Top-1 Accuracy	Top-5 Accuracy	Model Size
Vanilla SqueezeNet	57.5%	80.3%	4.8MB
SqueezeNet + Simple Bypass	**60.4%**	**82.5%**	4.8MB
SqueezeNet + Complex Bypass	58.8%	82.0%	7.7MB

Interestingly, the simple bypass enabled a higher accuracy accuracy improvement than complex bypass.

Summary

- **Goal**
  - Fewer parameters while preserving accuracy
- **Related Work**
  - Singular value decomposition (SVD)
  - Network pruning
  - Deep Compression
- **Architectural Design Strategies**
  - Filter size
  - Input channels
  - Downsample
- **Fire Module and SqueezeNet Architecture**
- **Design Space Exploration**
  - **Microarchitecture**
    - **Squeeze ratio**
    - **trade off: 1x1 vs 3x3**
  - **Macroarchitecture**
    - **Simple bypass**
    - **Complex bypass**