SqueezeNet

crackhopper

Wed Jan 4 17:00:08 2017

Outline

SqueezeNet

Model Compression

Applying SVD E.L Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In NIPS, 2014.

Network Pruninge S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural networks. In NIPS, 2015b.

Quantization S. Han, H. Mao, and W. Dally. Deep compression: Compressing DNNs with pruning, trained quantization and huffman coding. arxiv:1510.00149v3, 2015a.

EIE Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. Eie: Efficient inference engine on compressed deep neural network. International Sympo- sium on Computer Architecture (ISCA), 2016a.

Microarchitecture

Kernel Size 7x7, 5x5, 3x3, 1x1

Inception modules comprised of a number of different dimensionalities of filters, usually including 1x1 and 3x3, plus sometimes 5x5, and sometimes 1x3 and 3x1.

Macroarchitecture

Depth VGG(12-19)

Bypass Connections Residual Networks, Highway Networks

Design Space Exploration

Bayesian Optimization J. Snoek, H. Larochelle, and R.P. Adams. Practical bayesian optimization of machine learning algorithms. In NIPS, 2012.

Simulated Annealing T.B. Ludermir, A. Yamazaki, and C. Zanchettin. An optimization methodology for neural network weights and architectures. IEEE Trans. Neural Networks, 2006.

Randomized Search J. Bergstra and Y. Bengio. An optimization methodology for neural network weights and architectures. JMLR, 2012.

Genetic Algorithms K.O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. Neurocomputing, 2002.

SqueezeNet

Fire Module

- a squeeze convolution layer (which has only 1x1 filters)
- an expand layer that has a mix of 1x1 and 3x3 convolution filters

Three tunable dimensions: s_{1x1} the number of filters in the squeeze layer, e_{1x1} the number of 1x1 filters in the expand layer and e_{3x3} the number of 3x3 filters in the expand layer.

Fire Module

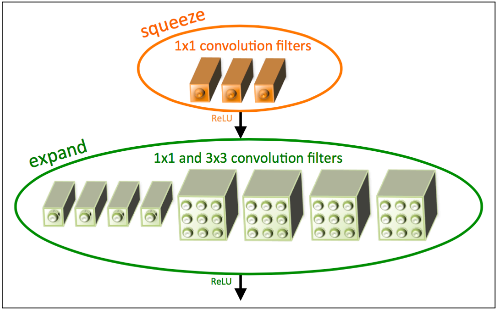


Figure 1: Microarchitectural view: Organization of convolution filters in the **Fire module**. In this example, $s_{1x1} = 3$, $e_{1x1} = 4$, and $e_{3x3} = 4$. We illustrate the convolution filters but not the activations.

Architectural Design Strategies

- Balance between 3x3 filters and 1x1 filters.
- Decrease the number of input channels to 3x3 filters.
- Pooling Later.

Architectural Design Strategies

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)

Other Squeezenet Details

- ReLU
- Dropout with a ratio of 50% is applied after the fire9 module.
- All Convolutional Structure
- Begin with a learning rate of 0.04, and we lin- early decrease the learning rate throughout training

Result Comparision

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	**363x**	57.5%	80.3%
SqueezeNet (ours)	Deep Compression	6 bit	4.8MB → 0.47MB	**510x**	57.5%	80.3%

SqueezeNet

CNN Microarchitecture Metaparameters

$base_e$ the number of expand filters in the first Fire module

$freq$ step that we increase the number of expand filters

$incr_e$ the number to increase when every $freq$ step

for module i, number of expand filters

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

CNN Microarchitecture Metaparameters

pct_{3x3} the percentage of expand filters that are 3x3

In other words, $e_{i,3x3} = e_i * pct_{3x3}$, and $e_{i,1x1} = e_i * (1pct_{3x3})$.

SR squeeze radio, $s_{i,1x1} = SR * e_i$

SqueezeNet has the following metaparameters:
$base_e = 128, incr_e = 128, pct_{3x3} = 0.5, freq = 2, andSR = 0.125$.

Result

SQUEEZE RATIO

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.

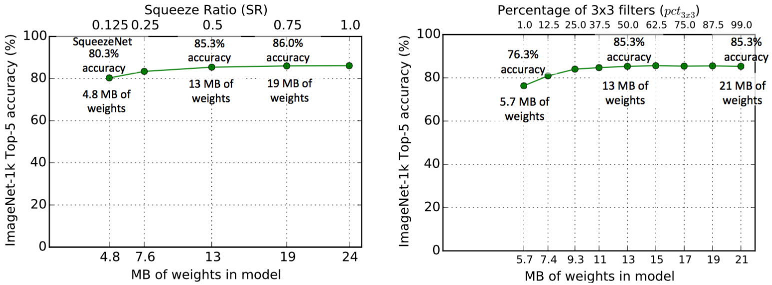
Accuracy plateaus SR=0.75 (a 19MB model).

TRADING OFF 1X1 AND 3X3 FILTERS

fix other metaparameters, vary pct_{3x3} from 1% to 99%.

Accuracy plateaus using 50% 3x3 filters

Result



(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.

CNN Macroarchitecture Design Space Exploration

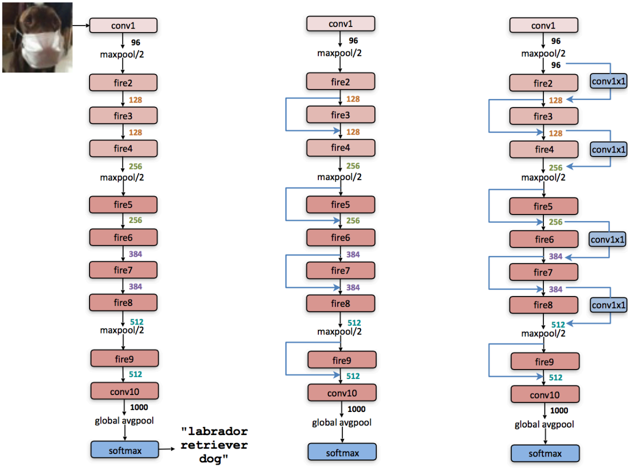


Figure 2: Macroarchitectural view of our SqueezeNet architecture. Left: SqueezeNet (Section 3.3); Middle: SqueezeNet with simple bypass (Section 6); Right: SqueezeNet with complex bypass (Section 6).

Result

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