Compressing Neural Networks using the Variational Information Bottleneck

Bin Dai[1]; Chen Zhu[2]; Baining Guo[3]; David Wipf[3]

[1]Tsinghua University, [2]University of Maryland, [3]Microsoft Research

ABSTRACT

Neural networks can be compressed to reduce memory and computational requirements, or to increase accuracy by facilitating the use of a larger base architecture. In this work, we use the information bottleneck principle to inspire a novel energy function for compressing deep networks. Theoretical analysis supports the compression performance while empirical results show state-of-art compression rates across an array of datasets and network architectures.

CONTACT

Bin Dai:
daib13@mails.tsinghua.edu.cn
Chen Zhu:
chenzhu@umd.edu
David Wipf:
davidwip@microsoft.com

Code Available

Background

Neural networks are over-parameterized in terms of
- Number of parameters (r_W)
- Computational cost (FLOP)
- Memory footprint (r_N)

Previous compression strategies:
- Design more efficient network structures
- Quantize network weights
- Apply tensor/matrix decompositions
- Prune existing network structures, e.g.
  - Connections
  - Weight groups / activations
    - Bayesian approaches
    - Group Lasso
    - Smoothed l_0-norm approach

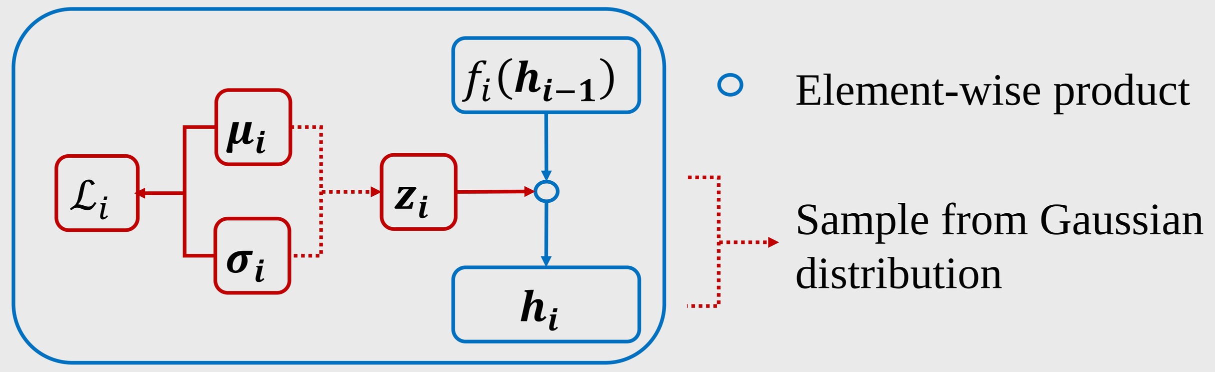


Figure 1. VIBNet layer/block structure

Method	r_W(%)	r_N(%)	Error(%)	Pruned Model
VD	25.28	58.95	1.8	512-114-72
BC-GNJ	10.76	32.85	1.8	278-98-13
BC-GHS	10.55	34.71	1.8	311-86-14
L0	26.02	45.02	**1.4**	219-214-100
L0-sep	10.01	32.69	1.8	266-88-33
DN	23.05	57.94	1.8	542-83-61
VIBNet	**3.59**	**16.98**	1.6	**97-71-33**

Table 1. LeNet300-100 on MNIST

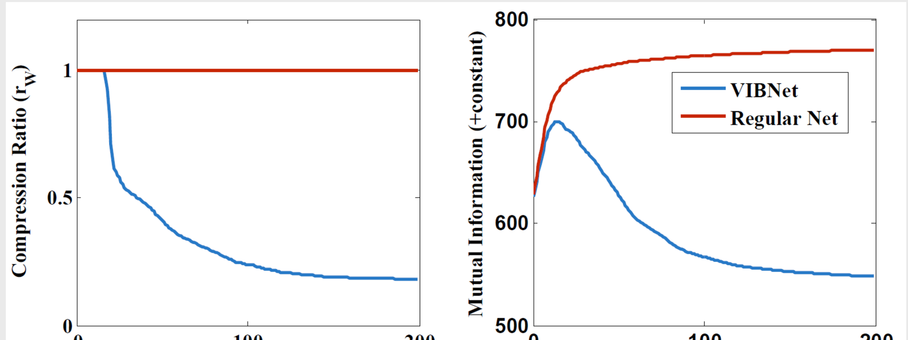


Figure 2. Compression ratio and mutual information between the first hidden layer and the input layer.

Method

We interpret a feedforward network as a Markov chain (All the layers are stochastic):

$$\boldsymbol{y} \to \boldsymbol{x} \to \boldsymbol{h}_1 \to \cdots \to \boldsymbol{h}_L \to \hat{\boldsymbol{y}}$$

❖ Intuition:
Minimize the mutual information $\boldsymbol{I}(\boldsymbol{h}_i;\boldsymbol{h}_{i-1})$ to remove inter-layer redundancy (*information bottleneck*)
Maximize the mutual information $\boldsymbol{I}(\boldsymbol{h}_i;\boldsymbol{y})$ to encourage accurate predictions of $\boldsymbol{y}$

(Tishby & Zaslavsky, arXiv 2015)

❖ Layer-wise objective: $\mathcal{L}_i = \gamma_i \boldsymbol{I}(\boldsymbol{h}_i;\boldsymbol{h}_{i-1}) - \boldsymbol{I}(\boldsymbol{h}_i;\boldsymbol{y})$

❖ Variational upper bound:

$$\tilde{\mathcal{L}}_i = \gamma_i \mathbb{E}_{\{\boldsymbol{x},\boldsymbol{y}\}\sim\mathcal{D},\boldsymbol{h}_{1:i-1}\sim p(\boldsymbol{h}_{1:i-1}|\boldsymbol{x})}\left[\mathbb{KL}[p(\boldsymbol{h}_i|\boldsymbol{h}_{i-1})||q(\boldsymbol{h}_i)]\right] - \mathbb{E}_{\{\boldsymbol{x},\boldsymbol{y}\}\sim\mathcal{D},\boldsymbol{h}\sim p(\boldsymbol{h}|\boldsymbol{x})}[\log q(\boldsymbol{y}|\boldsymbol{h}_L)]$$

❖ Parametric forms of $p(\boldsymbol{h}_i|\boldsymbol{h}_{i-1})$, $q(\boldsymbol{h}_i)$ and $q(\boldsymbol{y}|\boldsymbol{h}_L)$

$$p(\boldsymbol{h}_i|\boldsymbol{h}_{i-1}) = \mathcal{N}\left(\boldsymbol{h}_i|f_i(\boldsymbol{h}_{i-1})\odot\boldsymbol{\mu}_i, \operatorname{diag}\left[f_i(\boldsymbol{h}_{i-1})^2\odot\boldsymbol{\sigma}_i^2\right]\right)$$

$$q(\boldsymbol{h}_i) = \mathcal{N}(\boldsymbol{h}_i|\boldsymbol{0}, \operatorname{diag}[\boldsymbol{\xi}_i])$$

$q(\boldsymbol{y}|\boldsymbol{h}_L)$ is multinomial for classification and Gaussian for regression.

❖ Final variational IB-inspired objective function (*VIBNet*):

$$\boxed{\tilde{\mathcal{L}} = \underbrace{\sum_{i=1}^{L}\gamma_i\sum_{j=1}^{r_i}\log\left(1+\frac{\mu_{i,j}^2}{\sigma_{i,j}^2}\right)}_{\text{Regularization term}} - \underbrace{L\mathbb{E}_{\{\boldsymbol{x},\boldsymbol{y}\}\sim\mathcal{D},\boldsymbol{h}\sim p(\boldsymbol{h}|\boldsymbol{x})}[\log q(\boldsymbol{y}|\boldsymbol{h}_L)]}_{\text{Data-fit term}} \qquad (*)}$$

Method	r_W(%)	r_N(%)	FLOP(mil)	Error(%)
GD	1.38	32.00	0.250	1.1
GL	23.69	19.35	0.201	1.0
VD	9.29	60.78	0.660	1.0
SBP	19.66	21.15	0.213	**0.9**
BC-GNJ	0.95	35.03	0.283	1.0
BC-GHS	**0.64**	22.80	0.153	1.0
L0	8.92	85.82	1.113	**0.9**
L0-sep	1.08	40.36	0.389	1.0
VIBNet	0.83	**15.55**	**0.094**	1.0

Table 2. LeNet5-Caffe on MNIST

VD: Molchanov et al., ICML 2017
BC-GNJ / BC-GHS: Louizos et al., NIPS 2017
L0 / L0-sep: Louizos et al., ICLR 2016
DN: Pan et al., arXiv 2016
GD: Srinivas & Babu, arXiv 2016
GL: Wen et al., NIPS 2016

Method	r_W(%)	r_N(%)	FLOP(mil)	Error(%)
BC-GNJ	6.57	81.68	141.5	8.6
BC-GHS	5.40	74.82	121.9	9.0
VIBNet	**5.30**	**49.57**	**70.63**	8.8 (**8.5**)
PF	35.99	83.97	206.3	6.6
SBP	7.01	80.72	136.0	7.5
SBPa	5.78	66.46	99.20	9.0
VIBNet	5.45	57.86	86.82	6.5 (**6.1**)
NS-Single	11.50	-	195.5	6.2
NS-Best	8.60	-	147.0	5.9
VIBNet	**5.79**	**59.60**	**116.0**	6.2 (**5.8**)

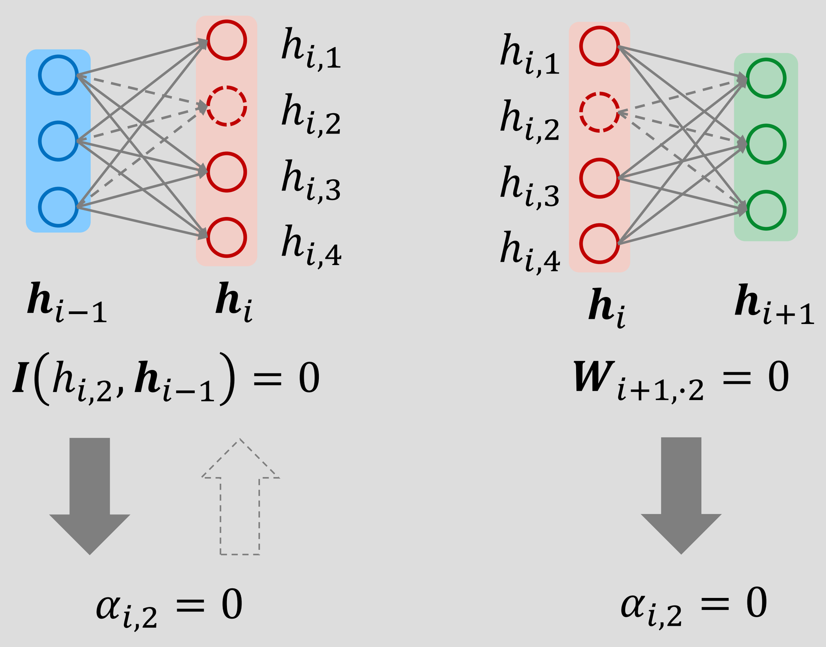
Table 3. VGG16 (modified) on CIFAR10

SBP / SBPa: Neklyu-dov et al., NIPS 2017
PF: Li et al., ICLR 2016
NS-Single / NS-Best: Liu et al., ICCV 2017
RNP: Lin et al., NIPS 2017

Theoretical Results

❖ Reduced redundancy via intrinsic sparsity:

Define $\alpha_{i,j} = \frac{\mu_{i,j}^2}{\sigma_{i,j}^2}$, at the global minimum of $(*)$

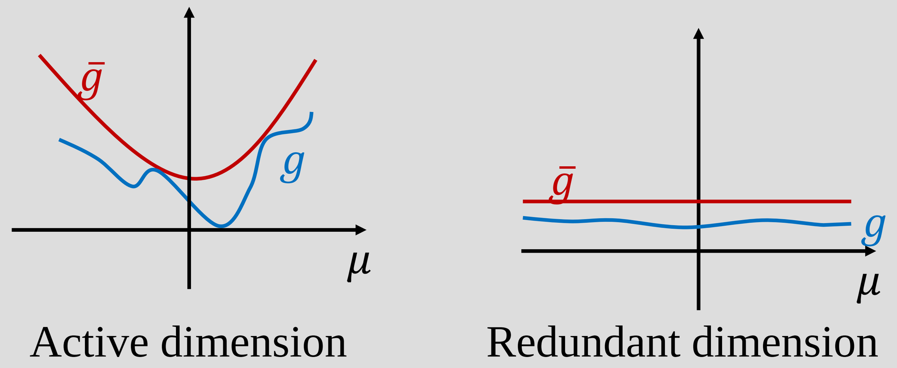


❖ Analysis of tractable upper bounds:
$-L\mathbb{E}_{\{\boldsymbol{x},\boldsymbol{y}\}\sim\mathcal{D},\boldsymbol{h}\sim p(\boldsymbol{h}|\boldsymbol{x})}[\log q(\boldsymbol{y}|\boldsymbol{h}_L)]$ can be written as

$$\int p(\boldsymbol{\epsilon})g(\boldsymbol{\epsilon};\boldsymbol{\theta},\boldsymbol{W})d\boldsymbol{\epsilon}$$

For fixed network weights $\boldsymbol{W}'$, replace $g(\boldsymbol{\epsilon};\boldsymbol{\theta},\boldsymbol{W})$ with a quadratic upper bound

$$\bar{g}(\boldsymbol{\epsilon};\boldsymbol{\theta}) \triangleq z(\boldsymbol{\epsilon};\boldsymbol{\theta})^T\boldsymbol{A}^T\boldsymbol{A}z(\boldsymbol{\epsilon};\boldsymbol{\theta}) + \boldsymbol{b}^T z(\boldsymbol{\epsilon};\boldsymbol{\theta}) + \mathrm{c}$$



$\boldsymbol{A}$ becomes low-rank if the network is over-parameterized. At any local minimum of the upper bound of $\tilde{\mathcal{L}}$, we have

$$\|\boldsymbol{\mu}^*\|_0 = \|\boldsymbol{\alpha}^*\|_0 \le rank[\boldsymbol{A}] + 1 \quad \Rightarrow \quad \text{sparse}$$

Method	r_W(%)	r_N(%)	FLOP(mil)	Error(%)
RNP	-	-	160	38.0
VIBNet	**22.75**	**59.80**	**133.6**	37.6 (**37.4**)
NS-Single	24.90	-	250.5	26.5
NS-Best	20.80	-	214.8	26.0
VIBNet	**15.08**	**73.80**	**203.1**	25.9 (**25.1**)

Table 4. VGG16 (modified) on CIFAR100