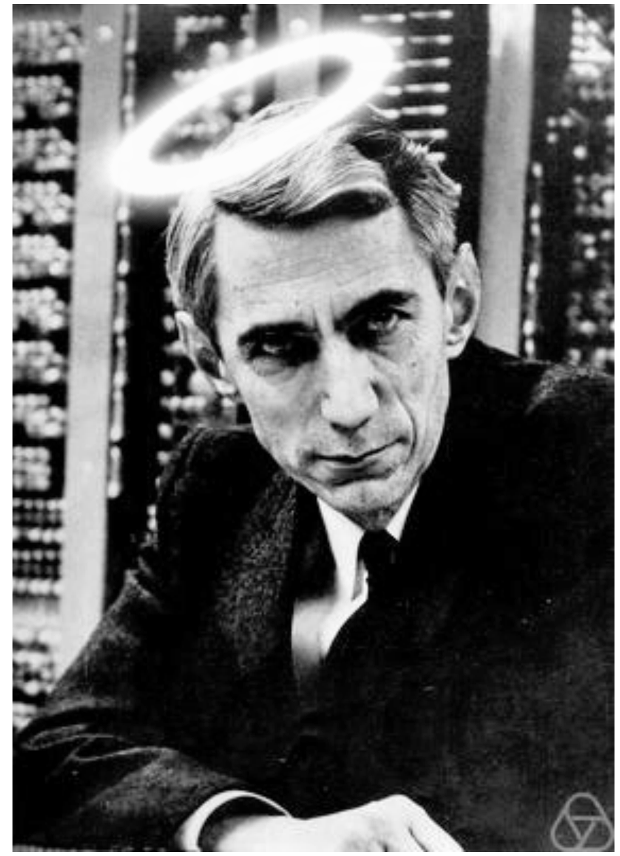
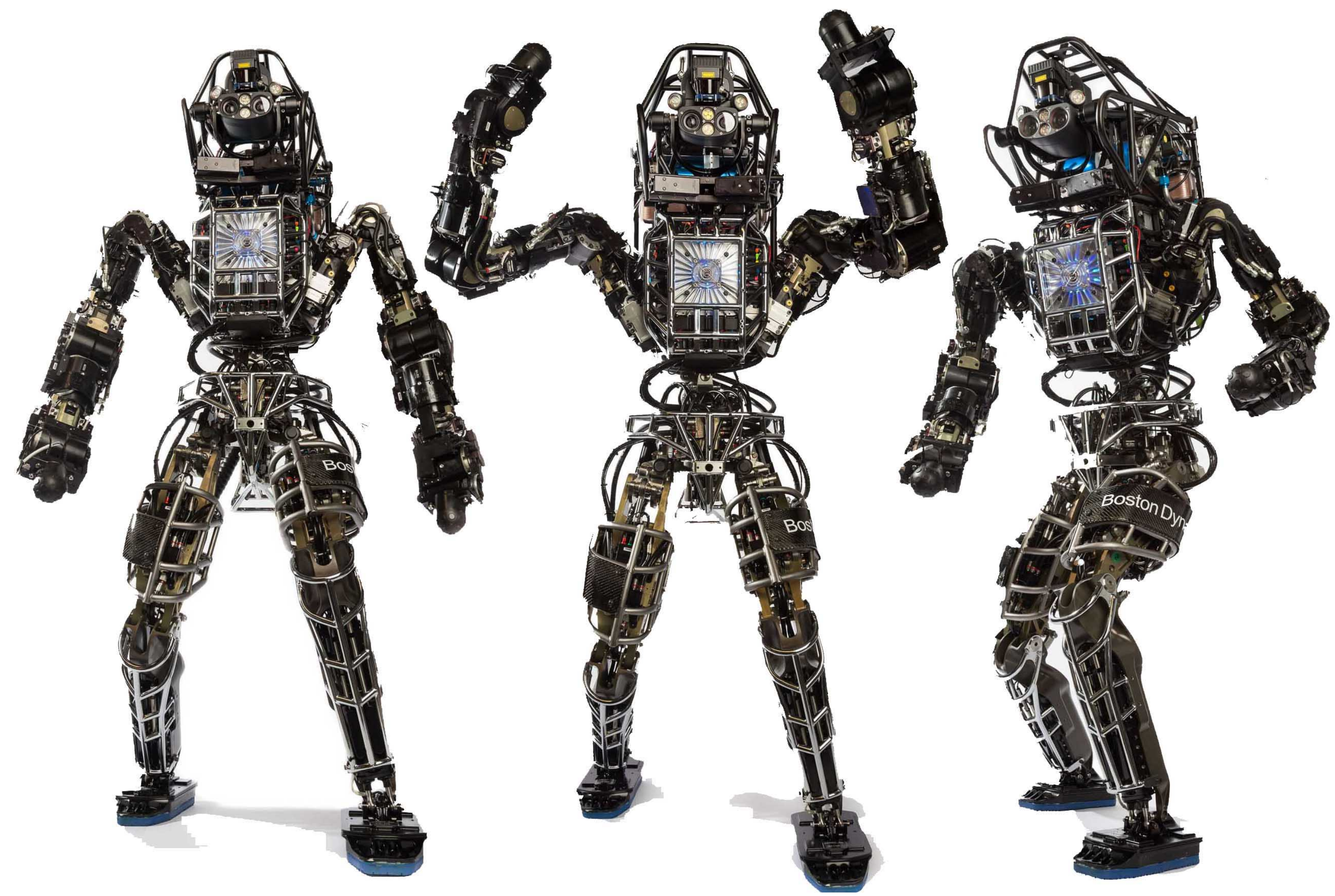


WHAT WOULD SHANNON DO? BAYESIAN COMPRESSION FOR DL

KAREN ULLRICH
UNIVERSITY OF AMSTERDAM

DEEP LEARNING AND REINFORCEMENT LEARNING
SUMMER SCHOOL MONTREAL 2017

Motivation



Motivation

- 1 Wh costs 0.0225 cent
- running a Titan X for 1h: 5.625 cent

- facebook has 1.86 billion active users
- VGG takes ~147ms/16 predictions

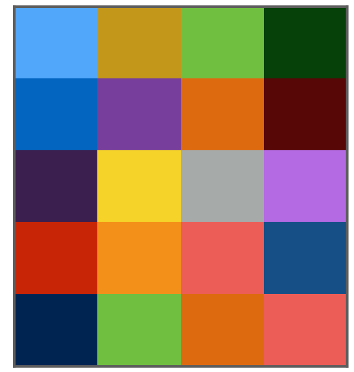
- making one prediction for all users costs 20 k€

Motivation - Summary

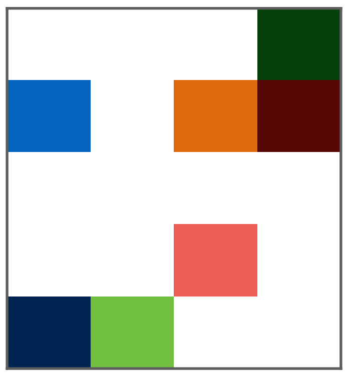
- mobile devices have **limited hardware**
- **energy costs** for predictions
- bandwidth **transmitting** models
- speeding up inference for **real time processing**
- relation to **privacy**

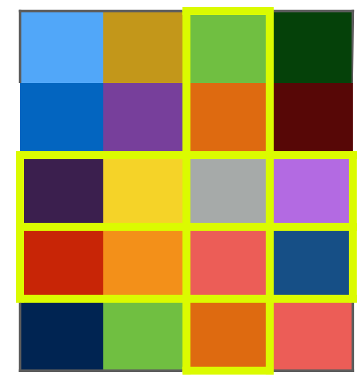
Practical view on compression

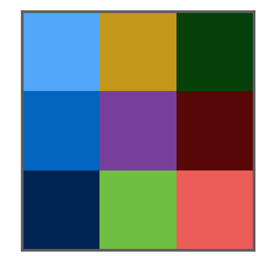
A : Sparsity learning



.size = $|\mathbf{w}|$







A = [].size = $|\mathbf{w}_{\neq 0}|$

IC = [0 1 3 0 1 2]

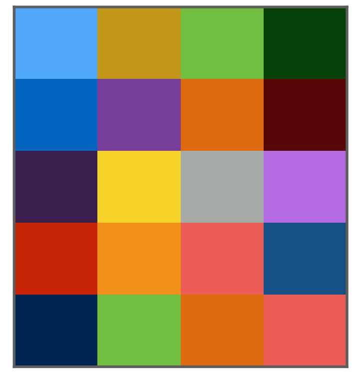
IR = [3, 0, 2, 3, …, 1]

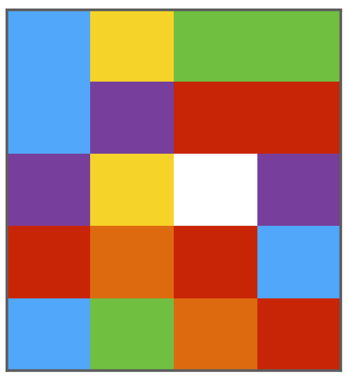
- (Unstructured) Pruning
- CR: $\approx \frac{|\mathbf{w}|}{2|\mathbf{w}_{\neq 0}|}$

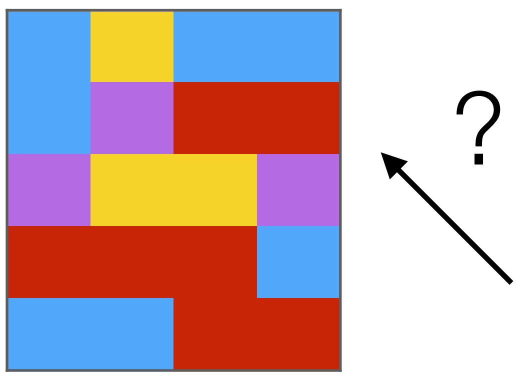
- Structured Pruning:
- CR: $\frac{|\mathbf{w}|}{|\mathbf{w}_{\neq 0}|}$

Practical view on compression

B : Bit per weight reduction







1	-3.5667
2	0.4545
3	0.8735
4	0.33546

- precision quantisation
- CR: 32/10 = 3
- PRO: fast inference
- CON: savings is not too big

- Set quantisation by clustering
- CR: 32/4 = 8
- PRO: extreme compressible with e.g. further Hoffman encoding
- CON: inference?

Practical view on compression

Summary - Properties

	Set quantisation	Bit quantisation
Unstructured pruning	- highest compression - flop and energy savings moderate	
Structured pruning		- lowest expected compression - BUT will save considerable amount of flops and thus energy

Practical view on compression

Summary - Applications

	Set quantisation	Bit quantisation
Unstructured pruning	-"ZIP"-format for NN - transmitting via limited channels - save millions of nets efficiently	
Structured pruning		- inference at scale - real time predictions - hardware limited devices

Variational lower bound

$$\log p(\mathcal{D}) \geq \mathcal{L}(q(\mathbf{w}), \mathbf{w})) = \mathbf{E}_{q(\mathbf{w})}[\log \frac{p(\mathcal{D}, \mathbf{w})}{q(\mathbf{w})}]$$
$$= \mathbf{E}_{q(\mathbf{w})}[\log p(\mathcal{D}|\mathbf{w})]] - KL(q(\mathbf{w})||p(\mathbf{w}))$$

Hinton, Geoffrey E., and Drew Van Camp. "Keeping the neural networks simple by minimizing the description length of the weights." *Proceedings of the sixth annual conference on Computational learning theory*. ACM, 1993.

MDL principle and Variational Learning

The best model is the one that compresses the data best. There are two costs, one for **transmitting a model** and one for reporting the **data misfit**.

Jorma Rissanen, 1978

Variational lower bound

$$\log p(\mathcal{D}) \geq \mathcal{L}(q(\mathbf{w}), \mathbf{w})) = \mathbf{E}_{q(\mathbf{w})}[\log \frac{p(\mathcal{D}, \mathbf{w})}{q(\mathbf{w})}]$$

$$= \mathbf{E}_{q(\mathbf{w})}[\log p(\mathcal{D}|\mathbf{w})]] - KL(q(\mathbf{w})||p(\mathbf{w}))$$

transmitting data misfit

transmitting the model

Hinton, Geoffrey E., and Drew Van Camp. "Keeping the neural networks simple by minimizing the description length of the weights." *Proceedings of the sixth annual conference on Computational learning theory*. ACM, 1993.

Variational lower bound

$$\log p(\mathcal{D}) \geq \mathcal{L}(q(\mathbf{w}), \mathbf{w})) = \mathbf{E}_{q(\mathbf{w})}[\log \frac{p(\mathcal{D}, \mathbf{w})}{q(\mathbf{w})}]$$

$$= \mathbf{E}_{q(\mathbf{w})}[\log p(\mathcal{D}|\mathbf{w})]] - KL(q(\mathbf{w})||p(\mathbf{w}))$$

$$p(\mathcal{D}|\mathbf{w}) = p(\mathbf{T}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{t}_n|\mathbf{x}_n, \mathbf{w})$$

$$\mathrm{KL}(q(\mathbf{w})||p(\mathbf{w})) = \mathbb{E}_{q(\mathbf{w})}[-\log p(\mathbf{w})] - H(q(\mathbf{w}))$$

$$H(q(\mathbf{w})) = -\int_{\Omega} q(\mathbf{w}) \log q(\mathbf{w})\, \mathrm{d}\mathbf{w} = -\int_{\mathbb{R}^I} \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma\mathbf{I}) \log \mathcal{N}(\mathbf{w}|\mathbf{0}, \sigma\mathbf{I}) = [\log(2\pi e \sigma^2)]^I.$$

Hinton, Geoffrey E., and Drew Van Camp. "Keeping the neural networks simple by minimizing the description length of the weights." *Proceedings of the sixth annual conference on Computational learning theory*. ACM, 1993.

Practical view on compression

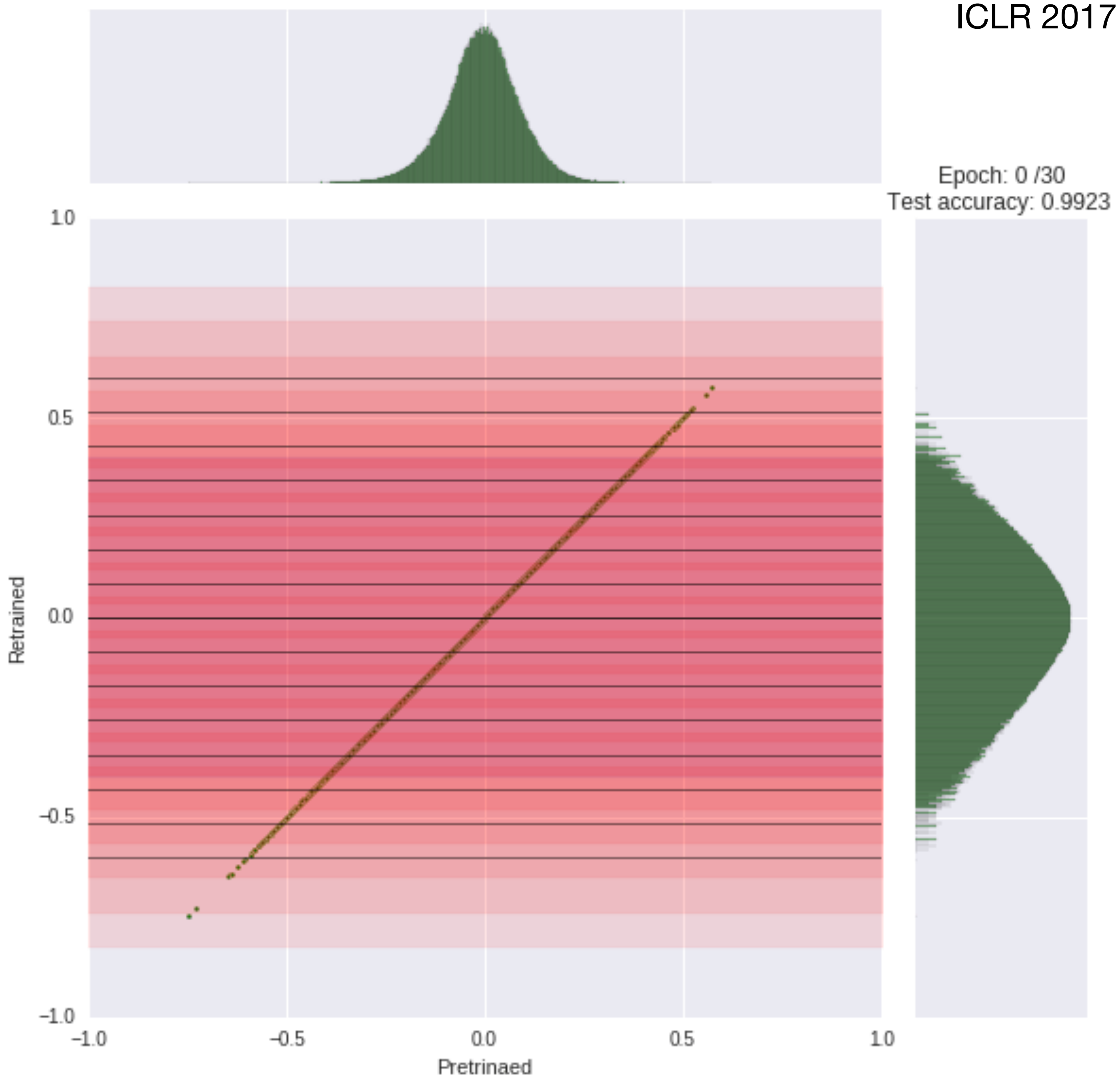
Summary - Properties

	Set quantisation	Bit quantisation
Unstructured pruning	- highest compression - flop and energy savings moderate	
Structured pruning		- lowest expected compression - BUT will save considerable amount of flops and thus energy

Soft weight-sharing for NN compression

KAREN ULLRICH, EDWARD MEED & MAX WELLING

ICLR 2017



- *Solution:* train a neural network with **gaussian mixture model prior**

$$q(\mathbf{w}) = \prod q(w_i) = \delta(w_i|\mu_i)$$

$$p(\mathbf{w}) = \prod_{i=1}^{I} \sum_{j=0}^{J} \pi_j \mathcal{N}(w_i|\mu_j, \sigma_j^2).$$

- Pruning by setting one component to zero with high mixing proportion

Nowlan, Steven J., and Geoffrey E. Hinton. "Simplifying neural networks by soft weight-sharing." *Neural computation* 4.4 (1992): 473-493.

Soft weight-sharing for NN compression

KAREN ULLRICH, EDWARD MEED & MAX WELLING

ICLR 2017

Model	Method	Top-1 Error[%]	Δ [%]	$\|\mathbf{W}\|[10^6]$	$\frac{\|\mathbf{W}_{\neq 0}\|}{\|\mathbf{W}\|}$ [%]	CR
LeNet-300-100	Han et al. (2015a)	$1.64 \rightarrow 1.58$	0.06	0.2	8.0	40
	Guo et al. (2016)	$2.28 \rightarrow 1.99$	-0.29		1.8	56
	Ours	$1.89 \rightarrow 1.94$	-0.05		4.3	**64**
LeNet-5-Caffe	Han et al. (2015a)	$0.80 \rightarrow 0.74$	-0.06	0.4	8.0	39
	Guo et al. (2016)	$0.91 \rightarrow 0.91$	0.00		0.9	108
	Ours	$0.88 \rightarrow 0.97$	0.09		0.5	**162**
ResNet (light)	Ours	$6.48 \rightarrow 8.50$	2.02	2.7	6.6	45

Practical view on compression

Summary - Properties

	Set quantisation	Bit quantisation
Unstructured pruning	- highest compression - flop and energy savings moderate	
Structured pruning		- lowest expected compression - BUT will save considerable amount of flops and thus energy

Bayesian Compression for Deep Learning

CHRISTOS LOUIZOS, KAREN ULLRICH & MAX WELLING

UNDER SUBMISSION NIPS 2017

- *Idea:* **use dropout to learn architecture**
- the variational version of dropout learns the dropout rate

- *Solution: Learn dropout rate for each weight structure, w*hen weights have a high dropout rate we can safely ignore them
- uncertainty in left over weights to compute bit precision

Kingma, Diederik P., Tim Salimans, and Max Welling. "Variational dropout and the local reparameterization trick." *NIPS*. 2015.
Molchanov, Dmitry, Arsenii Ashukha, and Dmitry Vetrov. "Variational Dropout Sparsifies Deep Neural Networks." *arXiv preprint arXiv:1701.05369* (2017).

Bayesian Compression for Deep Learning

CHRISTOS LOUIZOS, KAREN ULLRICH & MAX WELLING

UNDER SUBMISSION NIPS 2017

$$q(z) = \prod q(z_i) = \mathcal{N}(z_i | \mu_i^z, \alpha_i)$$

$$q(\mathbf{w}|z) = \prod q(w_i|z_i) = \mathcal{N}(w_i | z_i\mu_i, z_i^2\sigma_i^2)$$

force high dropout rates

push to zero for high dropout rates

Bayesian Compression for Deep Learning

CHRISTOS LOUIZOS, KAREN ULLRICH & MAX WELLING

UNDER SUBMISSION NIPS 2017

$$q(z) = \prod q(z_i) = \mathcal{N}(z_i|\mu_i^z, \alpha_i)$$

$$q(\mathbf{w}|z) = \prod q(w_i|z_i) = \mathcal{N}(w_i|z_i\mu_i, z_i^2\sigma_i^2)$$

$$p(w) = \int p(z)p(w|z)\mathrm{d}z$$

$$p(w) \propto \int \frac{1}{|z|}\mathcal{N}(w|0, z^2)dz = \frac{1}{|w|}$$

Bayesian Compression for Deep Learning

CHRISTOS LOUIZOS, KAREN ULLRICH & MAX WELLING

UNDER SUBMISSION NIPS 2017

Network & size	Method	Pruned architecture	Bit-precision
LeNet-300-100	Sparse VD	512-114-72	8-11-14
784-300-100	BC-GNJ	278-98-13	8-9-14
	BC-GHS	311-86-14	13-11-10
LeNet-5-Caffe	Sparse VD	14-19-242-131	13-10-8-12
	GD	7-13-208-16	-
20-50-800-500	GL	3-12-192-500	-
	BC-GNJ	8-13-88-13	18-10-7-9
	BC-GHS	5-10-76-16	10-10-14-13
VGG	BC-GNJ	63-64-128-128-245-155-63- -26-24-20-14-12-11-11-15	10-10-10-10-8-8-8- -5-5-5-5-5-6-7-11
(2× 64)-(2× 128)- -(3×256)-(8× 512)	BC-GHS	51-62-125-128-228-129-38- -13-9-6-5-6-6-6-20	11-12-9-14-10-8-5- -5-6-6-6-8-11-17-10

Bayesian Compression for Deep Learning

CHRISTOS LOUIZOS, KAREN ULLRICH & MAX WELLING

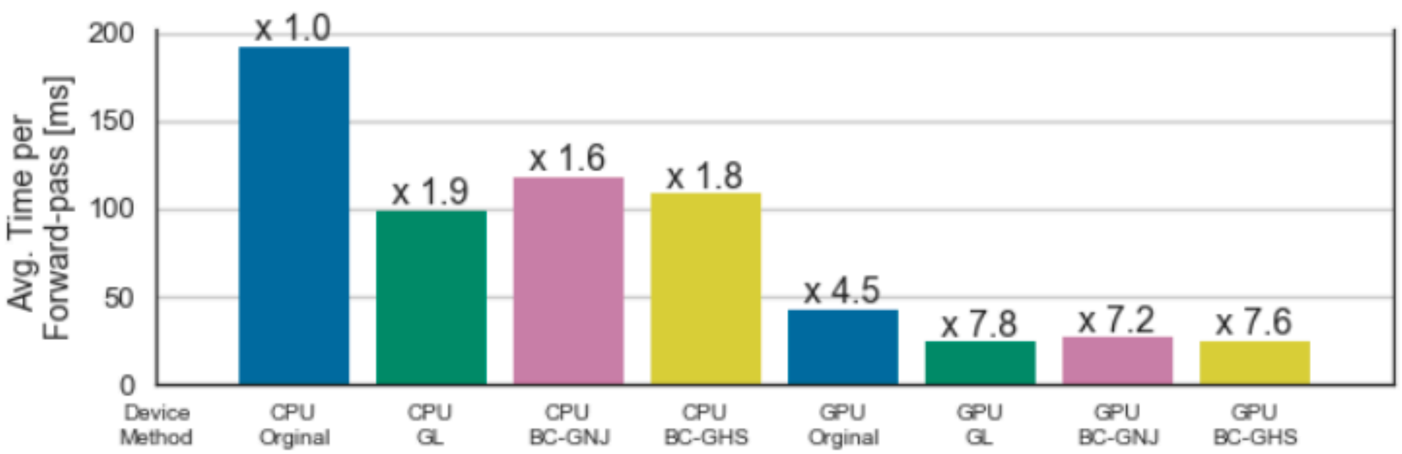
UNDER SUBMISSION NIPS 2017

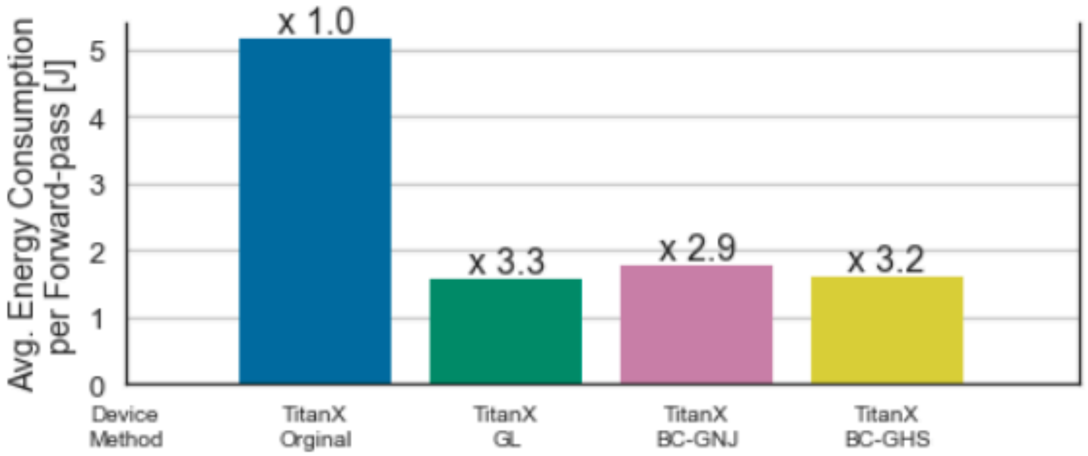
Model Original Error %	Method	$\frac{\|\mathbf{w}\neq 0\|}{\|\mathbf{w}\|}\%$	Compression Rates (Error %) Pruning	Fast Prediction	Maximum Compression
LeNet-300-100	DC	8.0	6 (1.6)	-	40 (1.6)
	DNS	1.8	28* (2.0)	-	-
1.6	SWS	4.3	12* (1.9)	-	64(1.9)
	Sparse VD	2.2	21(1.8)	84(1.8)	113 (1.8)
	BC-GNJ	10.8	9(1.8)	36(1.8)	58(1.8)
	BC-GHS	10.6	9(1.8)	23(1.9)	59(2.0)
LeNet-5-Caffe	DC	8.0	6*(0.7)	-	39(0.7)
	DNS	0.9	55*(0.9)	-	108(0.9)
0.9	SWS	0.5	100*(1.0)	-	162(1.0)
	Sparse VD	0.7	63(1.0)	228(1.0)	365(1.0)
	BC-GNJ	0.9	108(1.0)	361(1.0)	573(1.0)
	BC-GHS	0.6	156(1.0)	419(1.0)	771(1.0)
VGG	BC-GNJ	6.7	14(8.6)	56(8.8)	95(8.6)
8.4	BC-GHS	5.5	18(9.0)	59(9.0)	116(9.2)

Bayesian Compression for Deep Learning

CHRISTOS LOUIZOS, KAREN ULLRICH & MAX WELLING

UNDER SUBMISSION NIPS 2017





Warning: Don't be too enthusiastic!

These algorithms are merely proposals, little can be realised by common frameworks today.

- Architecture pruning
- Sparse matrix support (partially in big frameworks)
- Reduced bit precision (NVIDIA is starting)
- Clustering

Thank you for your attention.
Any questions?

KARENULLRICH.INFO
@KAREN_ULLRICH