Variational Inference for Neural Network Compression

Bayesian Compression for Deep Learning (Louizos et al, NIPS 2017)
Variational Network Quantization (Anonymous, under review for ICLR 2018)

Andrey Atanov, Polina Kirichenko
CS HSE

Outline

1. Neural networks compression
2. Recap: Bayesian neural networks
3. Sparse Variational Dropout
4. Structural sparsification (Bayesian Compression for Deep Neural Networks)
5. Quantization (Variational Network Quantization)

Neural Networks Compression

- Over-parametrized
- Store and run on hardware limited devices
- Acceleration
- Small representations

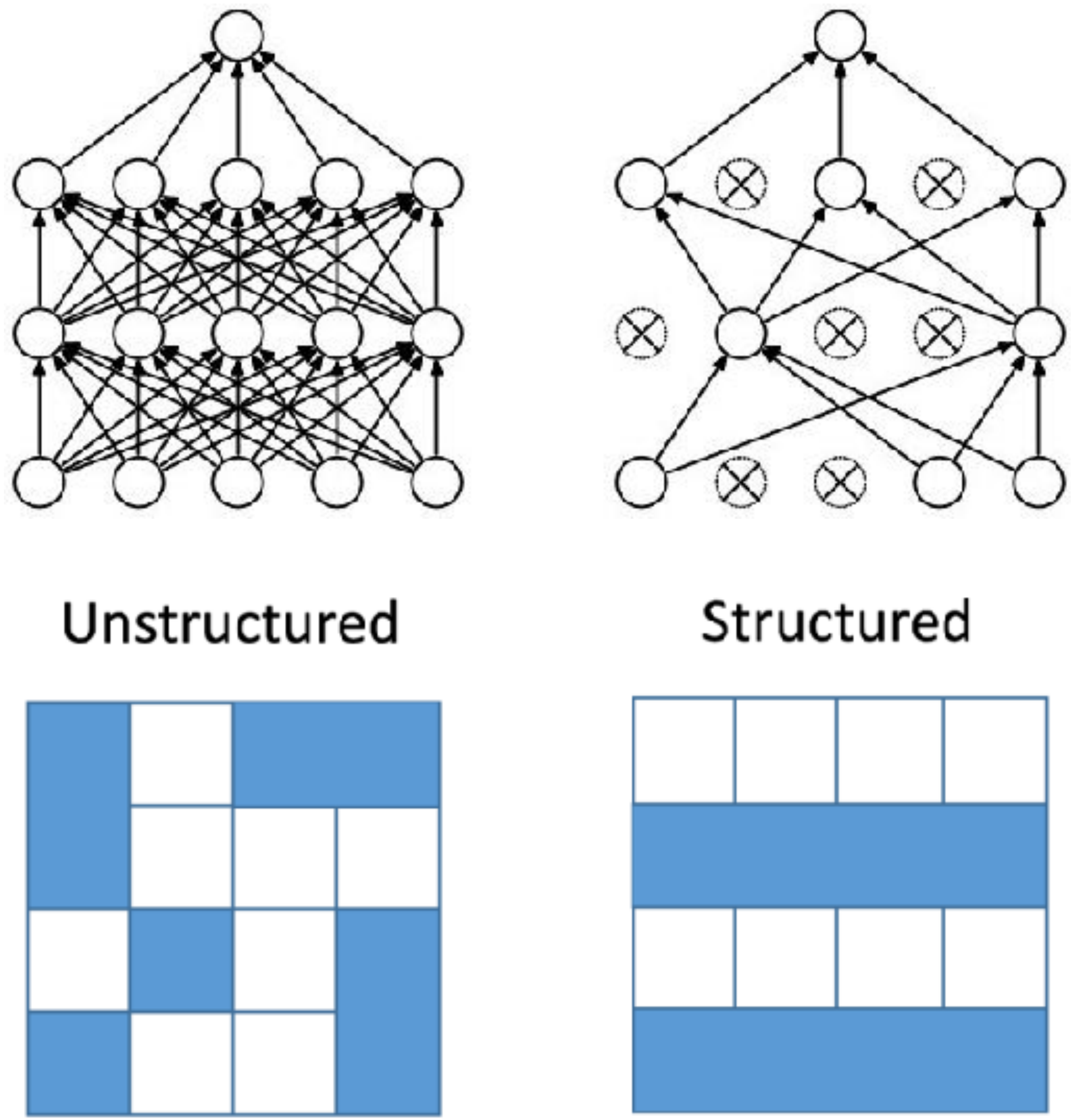
Neural Networks Compression

- simple or structural sparsification / pruning
- quantization
- low rank approximation for weight matrices
- ...

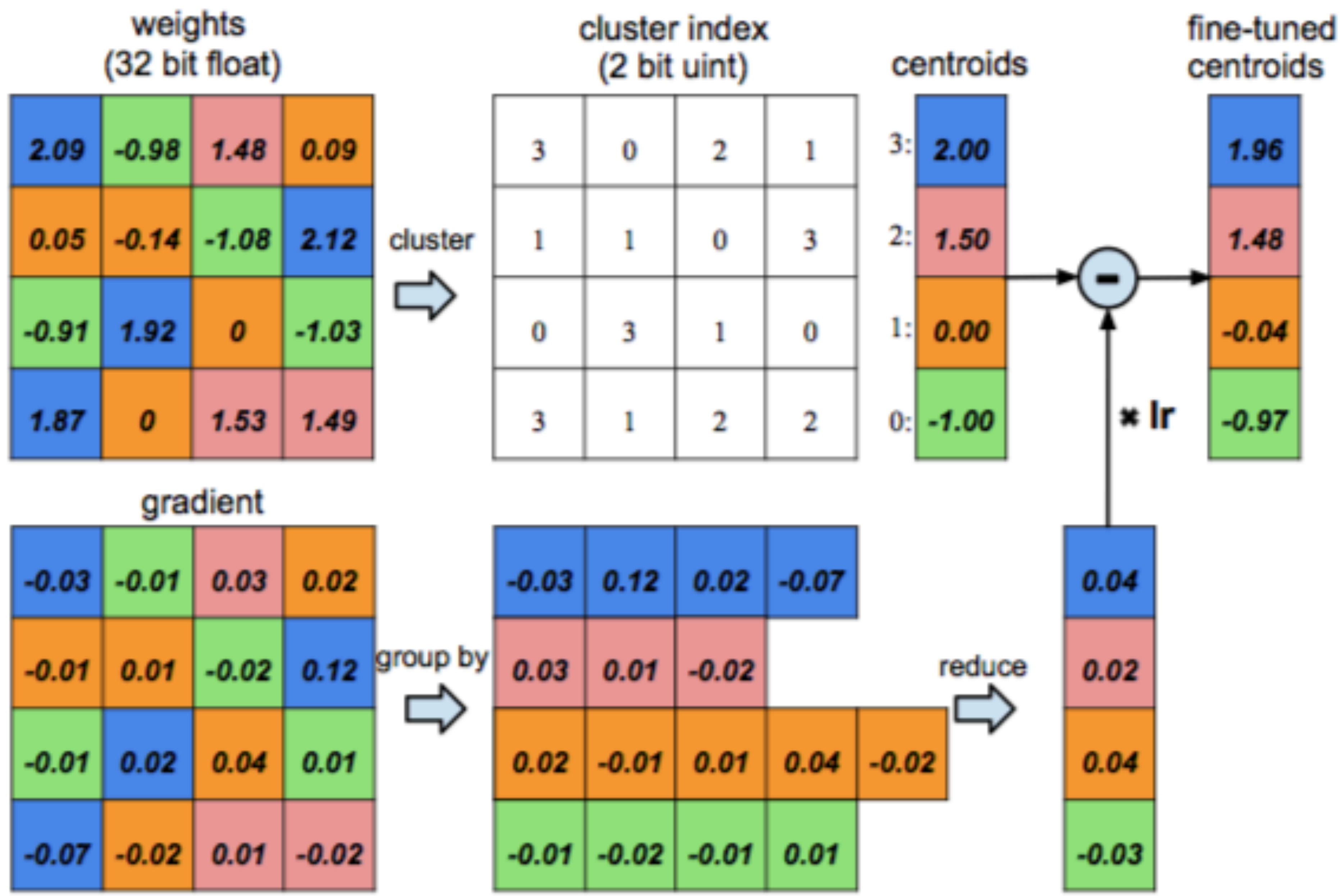
Neural Networks Compression

- simple or structural sparsification / pruning
- quantization
- low rank approximation for weight matrices
- ...

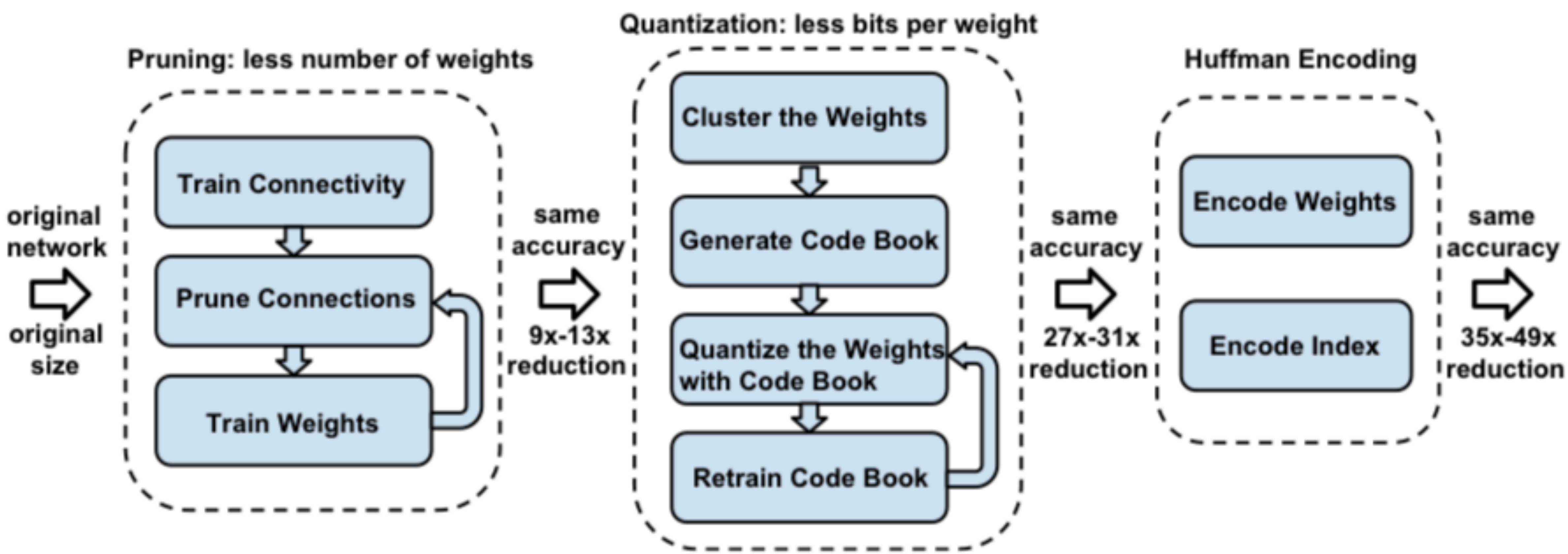
Structural Sparsity



Quantization



Neural Network Compression



Neural Network Compression

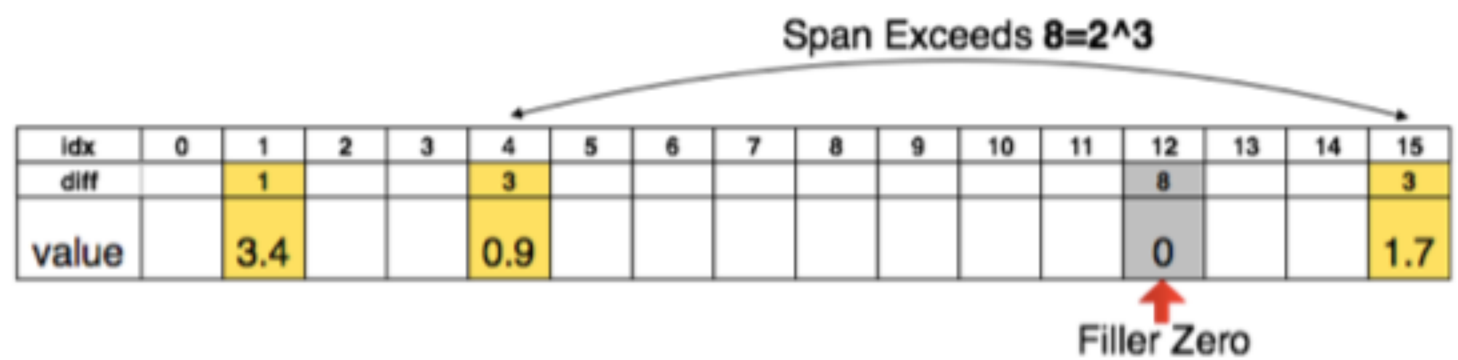


Figure 2: Representing the matrix sparsity with relative index. Padding filler zero to prevent overflow.


WE ARE THE BAYESIAN.
YOU WILL BE ASSIMILATED. YOUR TECHNOLOGICAL DISTINCTIVENESS WILL BE CONSIDERED A SPECIAL CASE OF OUR OWN. RESISTANCE IS FUTILE.
imgflip.com

Bayesian Inference

Given

$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ — training data

$p(w)$ — prior over the weights

Bayesian Inference

Given

$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ – training data

$p(w)$ – prior over the weights

Training:

$$p(w|\mathcal{D}) = \frac{p(\mathcal{D}|w)p(w)}{\int p(\mathcal{D}|w)p(w)dw}$$

Bayesian Inference

Given

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$$ — training data

$$p(w)$$ — prior over the weights

Training:

$$p(w|\mathcal{D}) = \frac{p(\mathcal{D}|w)p(w)}{\int p(\mathcal{D}|w)p(w)dw}$$

Inference:

$$p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, w)p(w|\mathcal{D})dw$$

Bayesian NNs

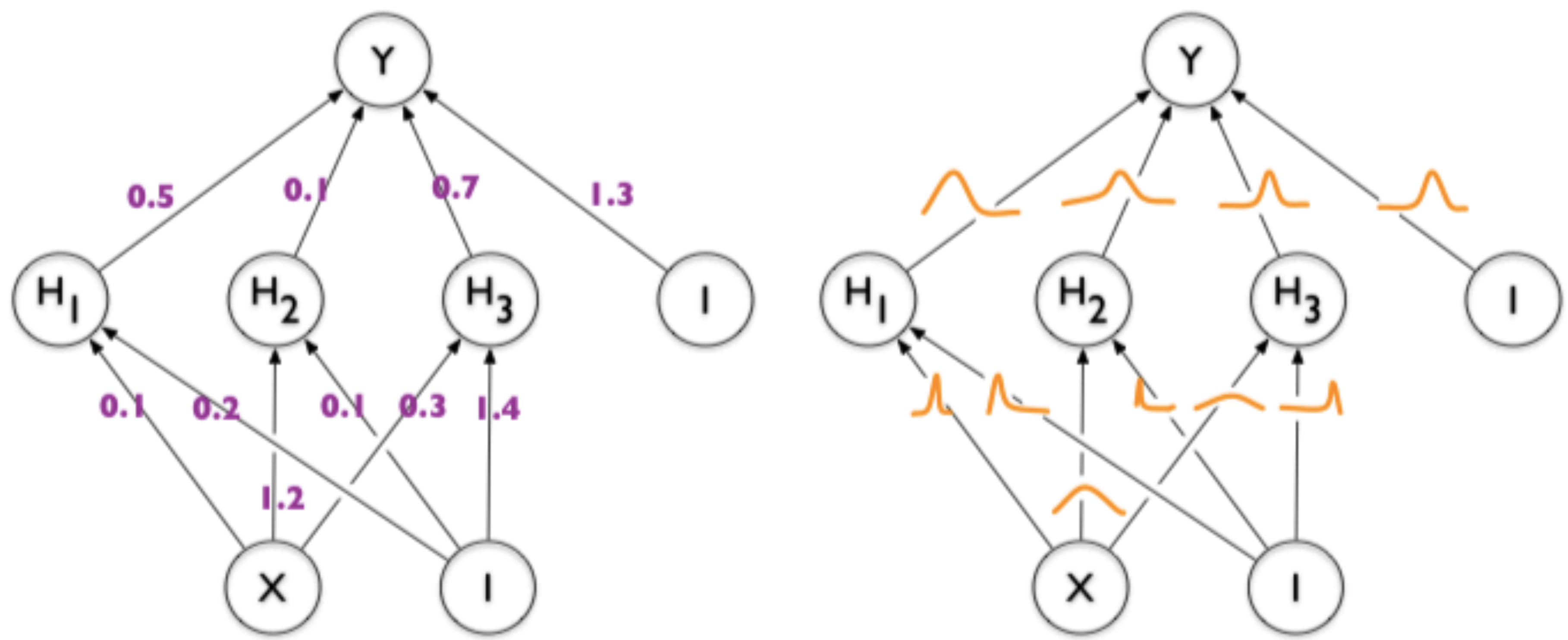
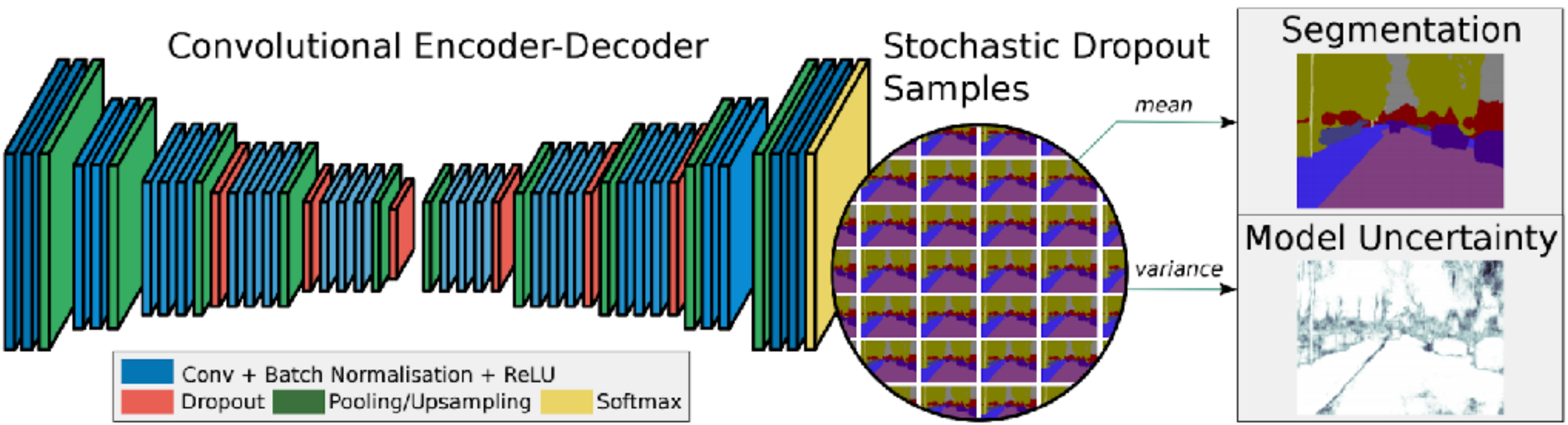


Figure 1. Left: each weight has a fixed value, as provided by classical backpropagation. Right: each weight is assigned a distribution, as provided by Bayes by Backprop.

Deep Bayesian NN



Stochastic Variational Inference

$$p(w|\mathcal{D}) \approx q_\phi(w)$$

Stochastic Variational Inference

$$p(w|\mathcal{D}) \approx q_\phi(w)$$

$$D_{\mathrm{KL}}(q_\phi(w)||p(w|\mathcal{D})) \longrightarrow \min_\phi$$

Stochastic Variational Inference

$$p(w|\mathcal{D}) \approx q_\phi(w)$$

$$D_{\mathrm{KL}}(q_\phi(w)||p(w|\mathcal{D})) \longrightarrow \min_\phi$$

$$D_{\mathrm{KL}}(q_\phi(w)||p(w|\mathcal{D})) = p(\mathcal{D}) - ELBO$$

Stochastic Variational Inference

$$p(w|\mathcal{D}) \approx q_\phi(w)$$

$$D_{\mathrm{KL}}(q_\phi(w)||p(w|\mathcal{D})) \longrightarrow \min_\phi \quad \Longleftrightarrow \quad ELBO \longrightarrow \max_\phi$$

$$D_{\mathrm{KL}}(q_\phi(w)||p(w|\mathcal{D})) = p(\mathcal{D}) - ELBO$$

Stochastic Variational Inference

$$p(w|\mathcal{D}) \approx q_\phi(w)$$

$$D_{\mathrm{KL}}(q_\phi(w)||p(w|\mathcal{D})) \longrightarrow \min_\phi \quad \Longleftrightarrow \quad ELBO \longrightarrow \max_\phi$$

$$D_{\mathrm{KL}}(q_\phi(w)||p(w|\mathcal{D})) = p(\mathcal{D}) - ELBO$$

$$ELBO = \underbrace{\sum_{n=1}^{N} \mathbb{E}_{q_\phi(w)} \log p(y_n|x_n, w)}_{\text{Data-term}} - \underbrace{D_{\mathrm{KL}}(q_\phi(w)||p(w))}_{\text{Regularization}}$$

Which prior to choose?

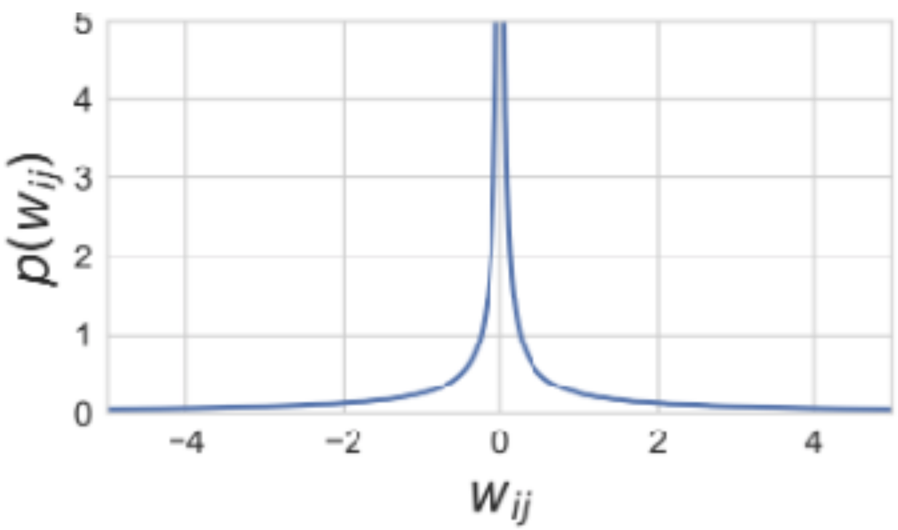
$$p(w) \text{ — } ?$$

Which prior to choose?

$$p(w) \text{ — } ?$$

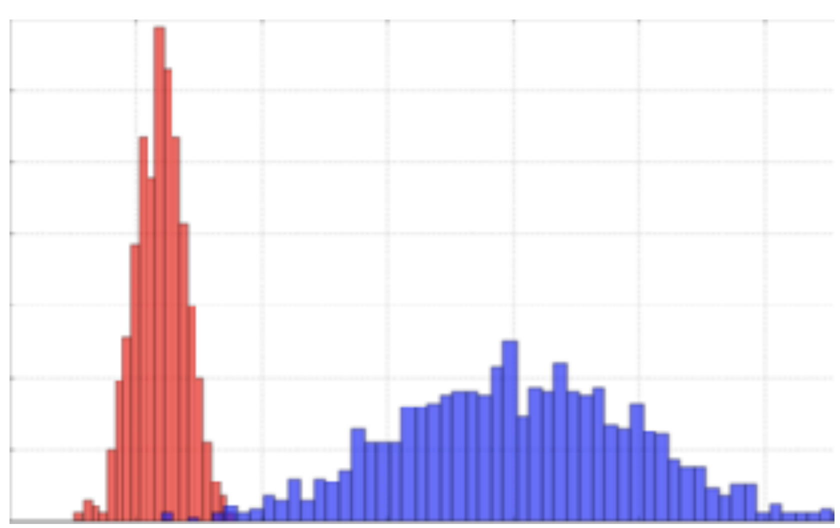
It depends…

Sparse Variational Dropout



$$p(w) \propto \frac{1}{|w|}$$

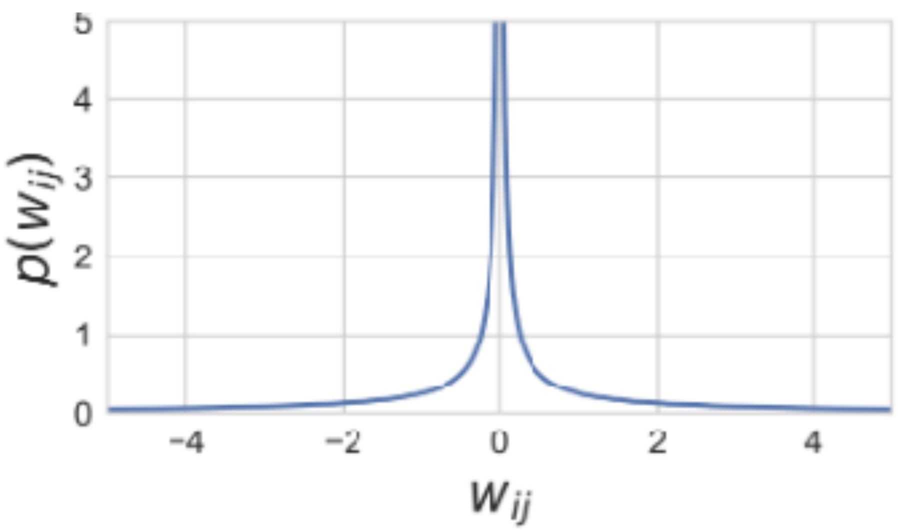
(improper) prior



$$q_\alpha(w) \sim \mathcal{N}(\theta, \alpha\theta^2)$$

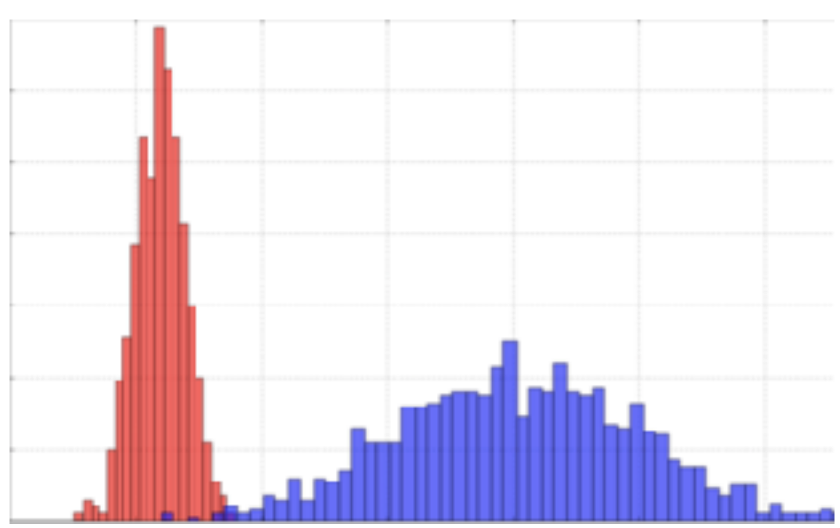
posterior approximation

Sparse Variational Dropout



$$p(w) \propto \frac{1}{|w|}$$

(improper) prior



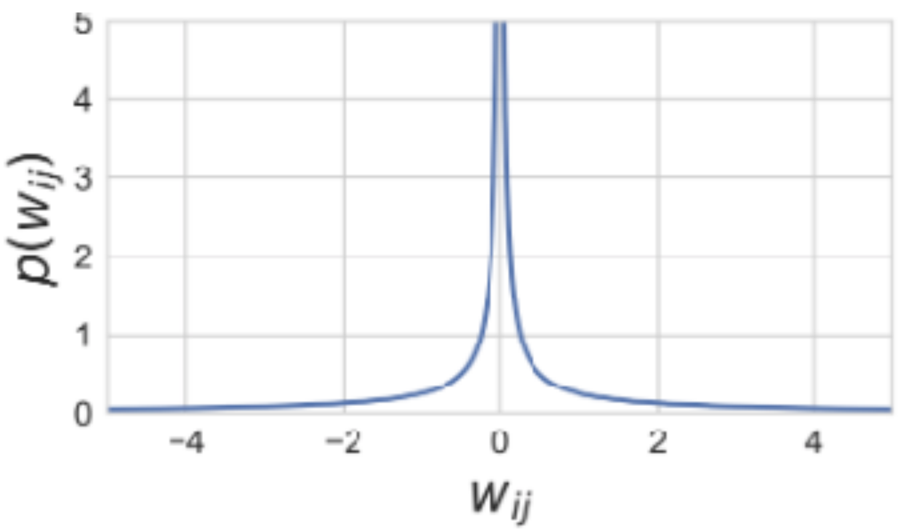
Gaussian dropout

$$\boxed{\theta \cdot \varepsilon = \theta \cdot \mathcal{N}(1, \alpha)}$$

$$q_\alpha(w) \sim \mathcal{N}(\theta, \alpha\theta^2)$$

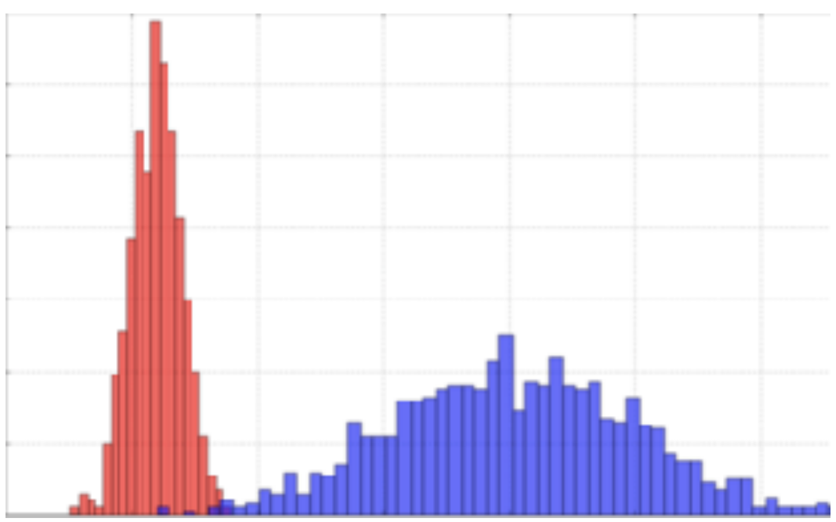
posterior approximation

Sparse Variational Dropout



$$p(w) \propto \frac{1}{|w|}$$

(improper) prior

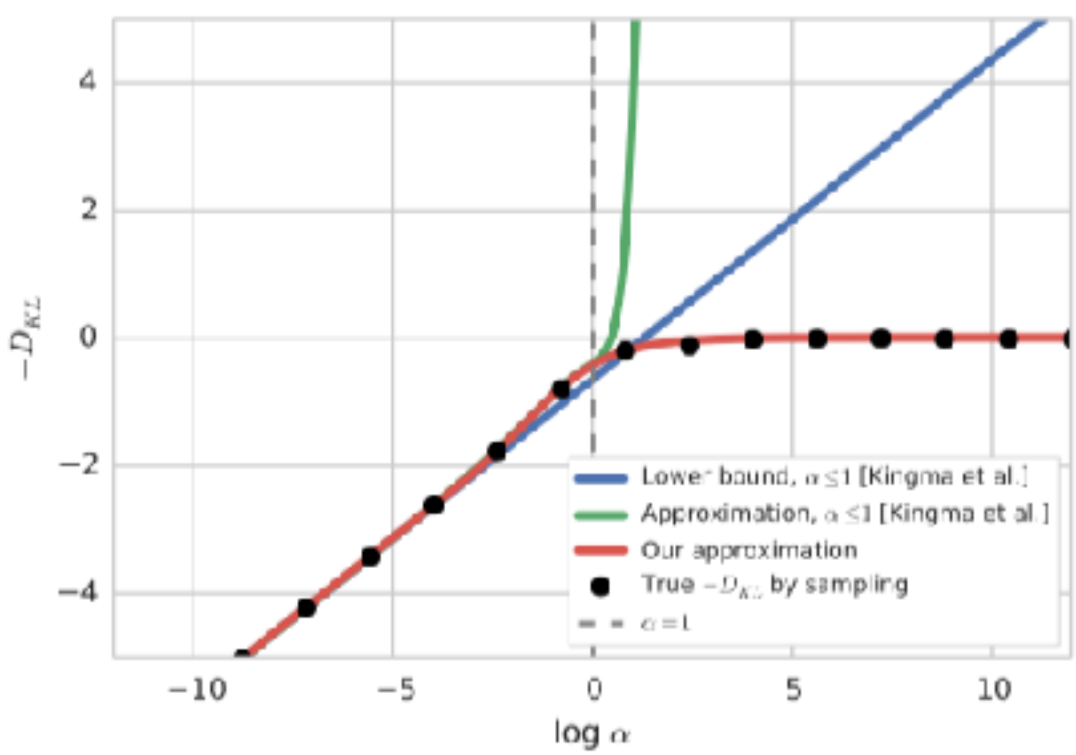


$$q_\alpha(w) \sim \mathcal{N}(\theta, \alpha\theta^2)$$

posterior approximation

Sparse Variational Dropout

$$ELBO = \sum_{n=1}^{N} \mathbb{E}_{q_\alpha(w)} \log p(y_n \,|\, x_n, w) \boxed{- D_{KL}(q_\alpha(w) \,||\, p(w))}$$



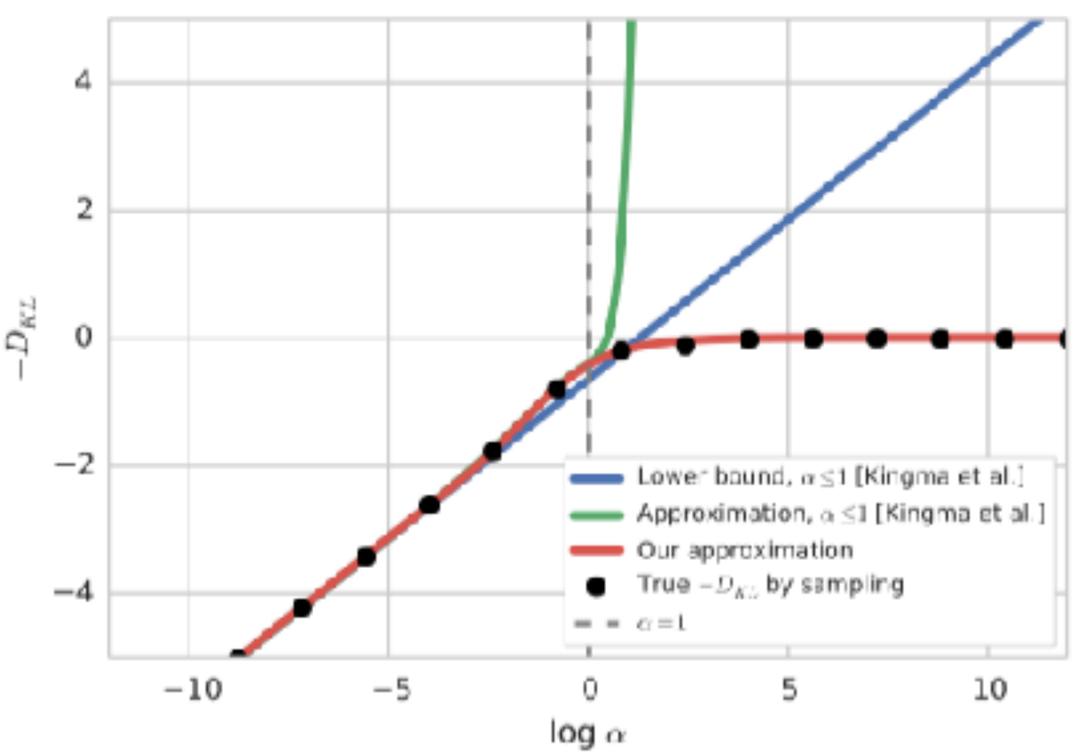
$$-D_{KL}(q_\alpha(w) \,||\, p(w)) =$$

$$= \frac{1}{2} \log \alpha_{ij} - \mathbb{E}_{\epsilon \sim \mathcal{N}(1, \alpha_{ij})} \log |\epsilon| + C$$

Molchanov et al «Variational Dropout Sparsifies Deep Neural Networks»

Sparse Variational Dropout

$$ELBO = \sum_{n=1}^{N} \mathbb{E}_{q_\alpha(w)} \log p(y_n \,|\, x_n, w) \boxed{- D_{KL}(q_\alpha(w) \,||\, p(w))}$$



$$-D_{KL}(q_\alpha(w) \,||\, p(w)) =$$

$$= \boxed{\frac{1}{2} \log \alpha_{ij}} - \mathbb{E}_{\epsilon \sim \mathcal{N}(1, \alpha_{ij})} \log |\epsilon| + C$$

Molchanov et al «Variational Dropout Sparsifies Deep Neural Networks»

Sparse Variational Dropout

$$\alpha_{ij} \to \infty$$

$$\Downarrow$$

$$\theta_{ij} \to 0, \quad \alpha_{ij}\theta_{ij}^2 \to 0$$

$$\Downarrow$$

$$q(w_{ij} \,|\, \theta_{ij}, \alpha_{ij}) \to \mathcal{N}(w_{ij} \,|\, 0, 0) = \delta(w_{ij})$$

Molchanov et al «Variational Dropout Sparsifies Deep Neural Networks»

Bayesian Compression for Deep Learning

Christos Louizos, Karen Ullrich, Max Welling

Hierarchical prior

Scale mixture of priors $w \sim \mathcal{N}(w \,|\, 0, z^2) \qquad z \sim p(z)$

$$p(w) = \int p(w \,|\, z) p(z) dz$$

a lot of well known sparsity inducing distributions are special cases

Normal-Jeffreys prior

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

Group sparsity:

$$p(W, z) \propto \prod_{i=1}^{A} \left[\frac{1}{|z_i|} \prod_{j}^{B} \mathcal{N}(w_{ij} \,|\, 0, z_i^2) \right]$$

$$q_\phi(W, z) = \prod_{i=1}^{A} \left[\mathcal{N}(z_i \,|\, \mu_{z_i}, \mu_{z_i}^2 \alpha_i) \prod_{j}^{B} \mathcal{N}(w_{ij} \,|\, z_i \mu_{ij}, z_i^2 \sigma_{ij}^2) \right]$$

Normal-Jeffreys prior

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

Group sparsity:

$$p(W, z) \propto \prod_{i=1}^{A} \left[\frac{1}{|z_i|} \prod_{j}^{B} \mathcal{N}(w_{ij} \mid 0, z_i^2) \right]$$

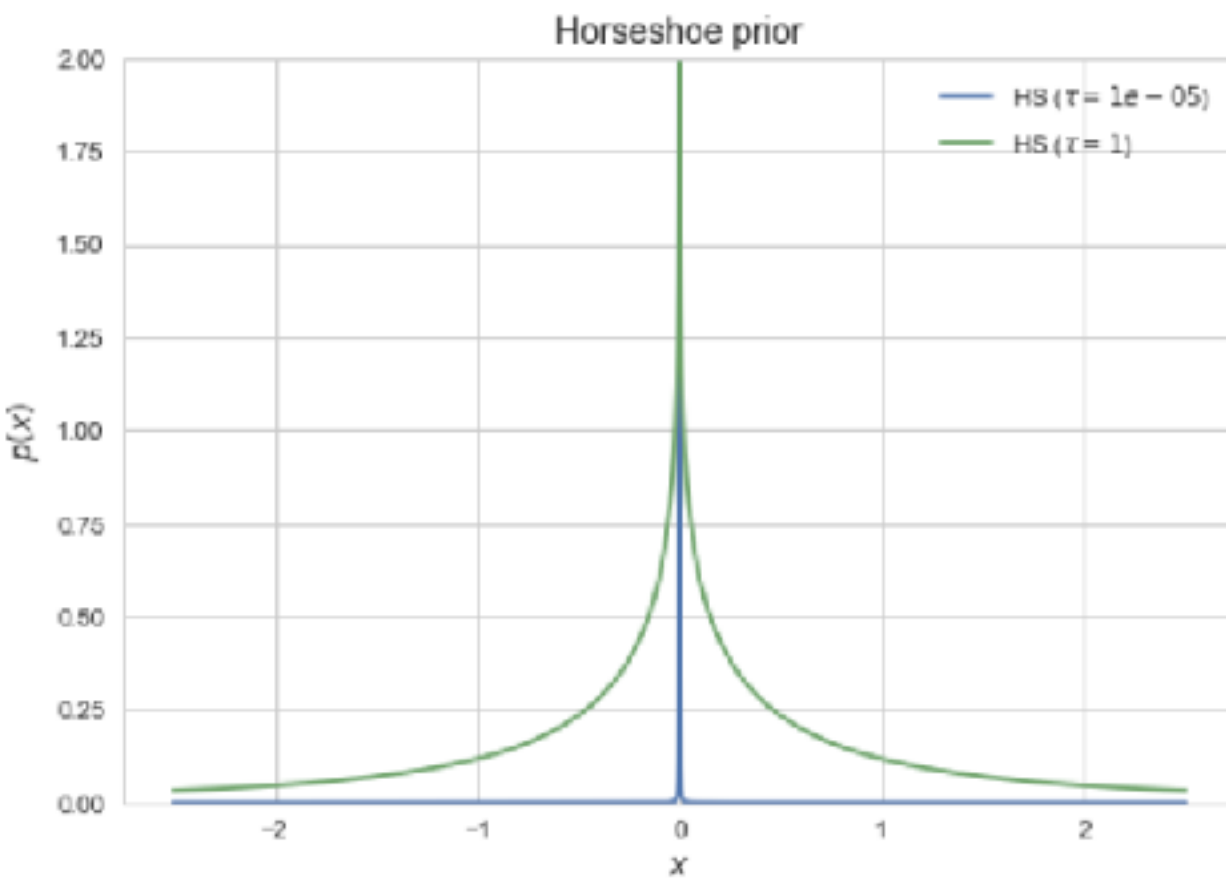
$$q_\phi(W, z) = \prod_{i=1}^{A} \left[\mathcal{N}(z_i \mid \mu_{z_i}, \mu_{z_i}^2 \boxed{\alpha_i)} \prod_{j}^{B} \mathcal{N}(w_{ij} \mid z_i \mu_{ij}, z_i^2 \sigma_{ij}^2) \right]$$

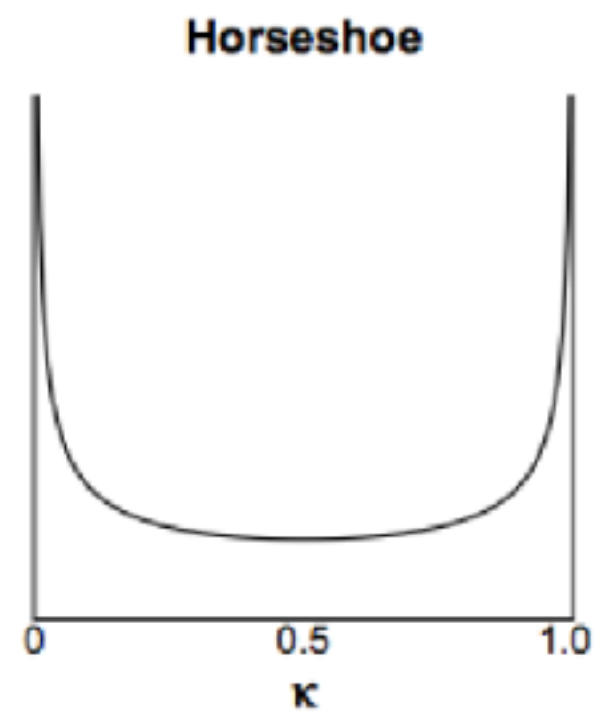
group dropout rate: $\log \alpha_i \geq t$

Horseshoe prior

$$p(z) = \mathcal{C}^+(0, s) = 2\left(s\pi(1 + (z/s)^2)\right)^{-1}$$ half-Caushy distribution

$p(w)$ horseshoe





$$\kappa = \frac{1}{1 + z^2}$$

shrinkage coefficient

Horseshoe prior

$$s \sim \mathcal{C}^+(0, \tau_0); \qquad \tilde{z}_i \sim \mathcal{C}^+(0,1); \qquad \tilde{w}_{ij} \sim \mathcal{N}(0,1); \qquad w_{ij} = \tilde{w}_{ij}\tilde{z}_i s,$$

global model sparsity

group sparsity

Inference:

$$q_\phi(s_b, s_a, \tilde{\boldsymbol{\beta}}) = \mathcal{LN}(s_b|\mu_{s_b}, \sigma^2_{s_b})\mathcal{LN}(s_a|\mu_{s_a}, \sigma^2_{s_a})\prod_i^A \mathcal{LN}(\tilde{\beta}_i|\mu_{\tilde{\beta}_i}, \sigma^2_{\tilde{\beta}_i})$$

$$q_\phi(\tilde{\boldsymbol{\alpha}}, \tilde{\mathbf{W}}) = \prod_i^A \mathcal{LN}(\tilde{\alpha}_i|\mu_{\tilde{\alpha}_i}, \sigma^2_{\tilde{\alpha}_i})\prod_{i,j}^{A,B} \mathcal{N}(\tilde{w}_{ij}|\mu_{\tilde{w}_{ij}}, \sigma^2_{\tilde{w}_{ij}}),$$

Learned Architecture

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

- **BC-GNJ: normal-Jeffreys**
- **BC-GHS: horseshoe**
- Sparse VD: Variational Dropout (not structural sparsity…)
- GL: Group Lasso
- GD: Generalized Dropout

Compression Rates

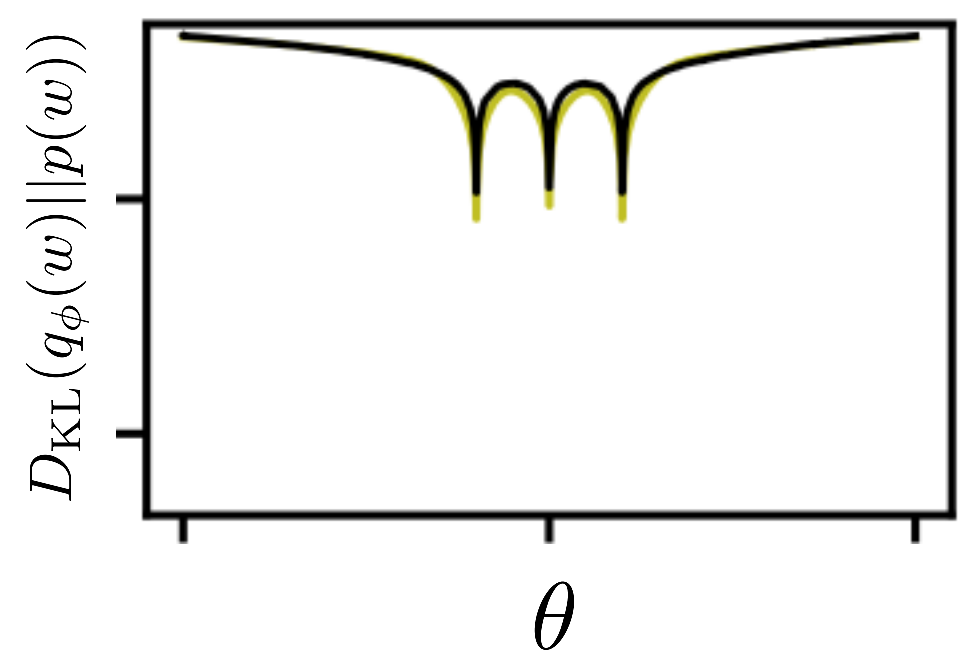
Model Original Error %	Method	$\frac{\lvert w \neq 0 \rvert}{\lvert w \rvert}$ %	Compression Rates (Error %) Pruning	Fast Prediction	Maximum Compression
LeNet-300-100	DC	8.0	6 (1.6)	-	40 (1.6)
	DNS	1.8	28* (2.0)	-	-
1.6	SWS	4.3	12* (1.9)	-	64(1.9)
	Sparse VD	2.2	21(1.8)	84(1.8)	113 (1.8)
	BC-GNJ	10.8	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)

Variational Nerwork Quantization

Anonymous authors

Variational Network Quantization

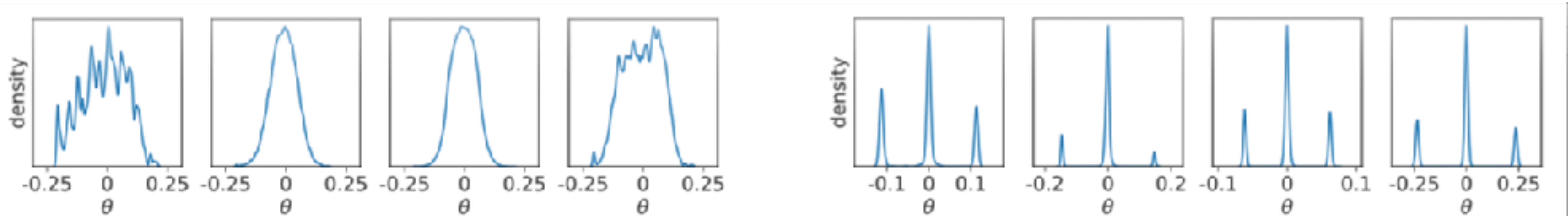
$$p(w_{ij}) \propto \sum_k a_k \frac{1}{|w_{ij} - c_k|}$$



posterior approximation

$$q_\alpha(w) \sim \mathcal{N}(\theta, \alpha\theta^2)$$

Variational Network Quantization



(a) Pre-trained network. No obvious clusters are visible in the network trained without VNQ. No regularization was used during pre-training.

(b) Soft-quantized network after VNQ training. Weights tightly cluster around the quantization target values.

Variational Network Quantization

MNIST

Method	**val. error** [%]	$\frac{\|w \neq 0\|}{\|w\|}$ [%]	bits
Original	0.8	100	32
VNQ (no P&Q)	0.67	100	32
VNQ + P&Q	0.73	28.3	2
VNQ + P&Q (random init.)	0.73	17.7	2
Deep Compression (P&Q)	0.74	8	$5-8$
Soft weight-sharing (P&Q)	0.97	0.5	3
Sparse VD (P)	0.75	0.7	-
Bayesian Comp. (P&Q)	1.0	0.6	$7-18$
Structured BP (P)	0.86	-	-

Variational Network Quantization

CIFAR10

Method	**val error** [%]	$\frac{\vert w \neq 0 \vert}{\vert w \vert}$ [%]	bits
Original	6.81	100	32
VNQ (no P&Q)	8.32	100	32
VNQ + P&Q (w/o 1)	8.78	46	2 (32)
VNQ + P&Q	8.83	46	2