Bayesian Compression for Deep Learning

Vadym Kalashnykov

Higher School of Economics

11 марта 2020 г.

Overview

Deep learning model compression

Despite the fact that DL models are extremely successful it's still difficult to apply them in many real world scenarios.

- large scale = energy cost
- often far from real-time
- hardware limited devices

Even next two objectives are not perfectly aligned:

- model size
- required computational resources

Deep learning model compression

Two major options:

- Optimize architecture (pruning)
- Compress weights (selecting effective fixed point precision for each weight)

Both can be done from a Bayesian perspective: set prior over model weights. Let's employ sparsity inducing priors for groups of weights corresponding to hidden units (neurons, conv. kernels). Using sparsity inducing priors on individual weights leads to complicated and inefficient coding schemes.
Use posterior uncertainty to assess which bits are significant.

Minimum Description Length Principle

MDL defines the best hypothesis to be the one that minimizes the sum of the model (complexity cost $\mathcal{L}^C$) and the data misfit (error cost $\mathcal{L}^E$) with the minimum number of bits.
Consider the following probability model:

- dataset: $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$
- model: $p(\mathcal{D}|\mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w})$
- posterior over weights: $p(\mathbf{w}|\mathcal{D}) = p(\mathcal{D}|\mathbf{w})p(\mathbf{w})/p(\mathcal{D})$ is intractable
- approximate with $q_\phi(\mathbf{w})$
- optimize with respect to
$$\mathcal{L}(\phi) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{w})}[\log p(\mathcal{D}|\mathbf{w})]}_{\mathcal{L}^E} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{w})}[\log p(\mathbf{w})] + \mathcal{H}(q_\phi(\mathbf{w}))}_{\mathcal{L}^C}$$

Minimum Description Length Principle

Objective:

$$\mathcal{L}(\phi) = \underbrace{\mathbb{E}_{q_\phi(\mathbf{w})}[\log p(\mathcal{D}|\mathbf{w})]}_{\mathcal{L}^E} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{w})}[\log p(\mathbf{w})] + \mathcal{H}\left(q_\phi(\mathbf{w})\right)}_{\mathcal{L}^C}$$

Sparsity inducing priors for groups of weights prune hidden units.
Remove infinitely precise weights to the entropy term: $\mathcal{H}(\delta(\mathbf{w})) = -\infty$
Reparametrization trick:

$$\mathcal{L}(\phi) = \mathbb{E}_{p(\epsilon)}[\log p(\mathcal{D}|f(\phi, \epsilon))] + \mathbb{E}_{q_\phi(\mathbf{w})}[\log p(\mathbf{w})] + \mathcal{H}\left(q_\phi(\mathbf{w})\right)$$

allows us to obtain unbiased stochastic gradients with respect to the variational parameters .

Scale mixtures of normals

Set prior over parameter w

- $z \sim p(z)$, represents weights' scale as random variable
- $w \sim \mathcal{N}\left(w; 0, z^2\right)$
- select $p(z)$ in a way that marginal prior over the parameters has heavier tails and more mass at zero

Improper log-uniform prior

Let $p(z)$ be the improper log-uniform prior: $p(z) \propto |z|^{-1}$
It's convenient because of

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

Couple the scales of weights within the same group simply sharing the scale variable z in the joint prior:

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

where A is the dimensionality of the input and B the dimensionality of the output.

Improper log-uniform prior

Consider performing variational inference with a joint approximate posterior parametrized as follows:

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

Here α_i is the dropout rate of the corresponding group. Parametrize as $\sigma_{z_i}^2 = \mu_{z_i}^2 \alpha_i$ to reduce objective's gradients' variance. Objective becomes:

$$\mathbb{E}_{q_\phi(\mathbf{z}) q_\phi(\mathbf{W}|\mathbf{z})}[\log p(\mathcal{D}|\mathbf{W})] - \mathbb{E}_{q_\phi(\mathbf{z})}\left[KL\left(q_\phi(\mathbf{W}|\mathbf{z}) \| p(\mathbf{W}|\mathbf{z})\right)\right] - KL\left(q_\phi(\mathbf{z}) \| p(\mathbf{z})\right)$$

Improper log-uniform prior

Under this parametrization the KL-divergence from the conditional prior $p(\mathbf{W}|\mathbf{z})$ to the approximate posterior $q_\phi(\mathbf{W}|\mathbf{z})$ is independent of z:

$$KL\left(q_\phi(\mathbf{W}|\mathbf{z})\|p(\mathbf{W}|\mathbf{z})\right) = \frac{1}{2}\sum_{i,j}^{A,B}\left(\log\frac{z_i^2}{z_i^2\sigma_{ij}^2} + \frac{z_i^2\sigma_{ij}^2}{z_i^Z} + \frac{z_i^2\mu_{ij}^2}{z_i^2} - 1\right)$$

Improper log-uniform prior

KL-divergence from the normal-Jeffreys scale prior $p(\mathbf{z})$ to the Gaussian variational posterior $q_\phi(\mathbf{z})$ depends only on the "implied" dropout rate $\alpha_i = \sigma^2_{z_i}/\mu^2_{z_i}$:

$$-KL\left(q_\phi(\mathbf{z})\|p(\mathbf{z})\right) \approx \sum_i^A \left(k_1\sigma\left(k_2 + k_3 \log \alpha_i\right) - 0.5m\left(-\log \alpha_i\right) - k_1\right)$$

$\sigma(\cdot), m(\cdot)$ are the sigmoid and softplus functions $and \mathrm{k}_1, k_2, k_3$ are constant terms. To prune groups of parameters we simply specify a threshold for the dropout rate $\log \alpha_i = \left(\log \sigma^2_{z_i} - \log \mu^2_{z_i}\right) \geq t$.

To asess the bit precision of each weight we use

$$\mathbb{V}\left(w_{ij}\right)_{NJ} = \sigma^2_{z_i}\left(\sigma^2_{ij} + \mu^2_{ij}\right) + \sigma^2_{ij}\mu^2_{z_i}$$

.

Half-Cauchy scale prior

Another option for $p(z)$ is a half-Cauchy distribution:

- $\mathcal{C}^+(0, s) = 2\left(s\pi\left(1 + (z/s)^2\right)\right)^{-1}$
- $s \sim \mathcal{C}^+(0, \tau_0)$, τ_0 is a free parameter and can be tuned
- $\tilde{z}_i \sim \mathcal{C}^+(0, 1)$
- $\tilde{w}_{ij} \sim \mathcal{N}(0, 1)$
- $w_{ij} = \tilde{w}_{ij}\tilde{z}_i s$

In this case calculations differ from log-normal prior case, still the logic is quite the same.

Experiments

Run Bayesian Compression (BC) on

- LeNet-300-100, LeNet-5-Caffe on MNIST
- VGG 10 on CIFAR 10

The groups of parameters were constructed by coupling the scale variables for each:

- filter for the convolutional layers
- input neuron for the fully connected layers

Experiments : Learned architectures

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

Experiments : Compression Rates

Model Original Error %	Method	$\frac{\lvert\mathbf{w}\neq 0\rvert}{\lvert\mathbf{w}\rvert}\%$	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)

Sources

- Christos Louizos, Karen Ullrich, Max Welling "Bayesian Compression for Deep Learning"preprint arXiv:1705.08665 (2017).
- Diederik P. Kingma, Tim Salimans, Max Welling "Variational Dropout and the Local Reparameterization Trick"arXiv:1506.02557 (2015)

Questions

- Как связано вычисление приближенного апостериорного распределения на веса модели с процедурой сжатия?
- Опишите два подхода к сжатию модели в ситуации, когда зафиксирована оптимальная архитектура.
- Какое семейство распределений подразумевается под scale mixtures of normals? Опишите две параметризации рассмотренные в статье.