LEARNING SPARSE NEURAL NETWORKS THROUGH L0 REGULARIZATION

Christos Louizos, Max Welling, Diederik P. Kingma

STA 4273 Paper Presentation

Daniel Flam-Shepherd, Armaan Farhadi & Zhaoyu Guo

March 2nd, 2018

Neural Networks: the good and the bad

Neural Networks : ...

1. are flexible function approximators that scale really well
2. are overparameterized and prone to overfitting and memorization

So what can we do about this?

Model compression and sparsification!

Consider the Empirical Risk minimization problem

$$\min_{\boldsymbol{\theta}} \; R(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} L(f(\mathbf{x}_i; \boldsymbol{\theta}), \mathbf{y}_i) + \lambda ||\boldsymbol{\theta}||_p$$

where

1. $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ is the iid dataset of input-output pairs
2. $f(\mathbf{x}; \boldsymbol{\theta})$ is the NN using parameters $\boldsymbol{\theta}$
3. $||\boldsymbol{\theta}||_p$ is the L^p norm
4. $L(\cdot)$ is the loss function

Lp Norms

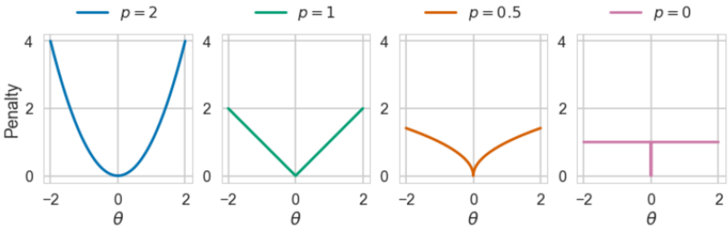


Figure: L_p norm penalties for parameter θ from Iousizos et al

The L_0 "norm" is just the number of nonzero parameters.

$$||\boldsymbol{\theta}||_0 = \sum_{j=1}^{|\boldsymbol{\theta}|} \mathbb{I}[\boldsymbol{\theta}_j \neq 0]$$

This does *not* impose shrinkage on large $\boldsymbol{\theta}_j$ rather it directly penalizes $|\boldsymbol{\theta}|$.

Reparameterizing

If we use the L_p norm $R(\boldsymbol{\theta})$ is non-differentiable at 0.

How can we relax this optimization and ensure $0 \in \boldsymbol{\theta}$?

First, Reparameterize by putting binary gates z_j on each θ_j.

$$\theta_j = \tilde{\theta}_j z_j, \quad z_j \in \{0,1\}, \quad \tilde{\theta}_j \neq 0, \quad \& \quad ||\boldsymbol{\theta}||_0 = \sum_{j=1}^{|\boldsymbol{\theta}|} z_j$$

let $z_j \sim \text{Ber}(\pi_j)$ with pmf $q(z_j|\pi_j)$ and we can formulate the problem as:

$$\min_{\tilde{\boldsymbol{\theta}},\boldsymbol{\pi}} R(\tilde{\boldsymbol{\theta}},\boldsymbol{\pi}) = \mathbb{E}_{q(\mathbf{z}|\boldsymbol{\pi})}\left[\frac{1}{N}\sum_{i=1}^{N} L(f(\mathbf{x}_i;\tilde{\boldsymbol{\theta}}\odot\mathbf{z}),\mathbf{y}_i)\right] + \lambda\sum_{j=1}^{|\boldsymbol{\theta}|}\pi_j$$

we cannot optimize the first term.

Smooth the objective so we can optimize it!

Let gates $\mathbf{z}$ be given by a hard-sigmoid rectification of $\mathbf{s}$, as follows

$$\mathbf{z} = g(\mathbf{s}) = \min(\mathbf{1}, \max(\mathbf{0}, \mathbf{s})), \quad \mathbf{s} \sim q_{\boldsymbol{\phi}}(\mathbf{s})$$

The probability of a gate being active is

$$q_{\boldsymbol{\phi}}(\mathbf{z} \neq 0) = \mathbf{1} - Q_{\boldsymbol{\phi}}(\mathbf{s} \leq \mathbf{0})$$

Then using the reparametrization trick on $\mathbf{s} = f(\boldsymbol{\phi}, \boldsymbol{\epsilon})$ so $\mathbf{z} = g(f(\boldsymbol{\phi}, \boldsymbol{\epsilon}))$

$$\min_{\tilde{\boldsymbol{\theta}}, \boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\frac{1}{N} \sum_{i=1}^{N} L(f(\mathbf{x}_i; \tilde{\boldsymbol{\theta}} \odot g(f(\boldsymbol{\phi}, \boldsymbol{\epsilon}))), \mathbf{y}_i) \right] + \lambda \sum_{j=1}^{|\boldsymbol{\theta}|} \mathbf{1} - Q_{\boldsymbol{\phi}}(s_j \leq 0)$$

Okay but which distribution $q_{\boldsymbol{\phi}}(\mathbf{s})$ should we use?

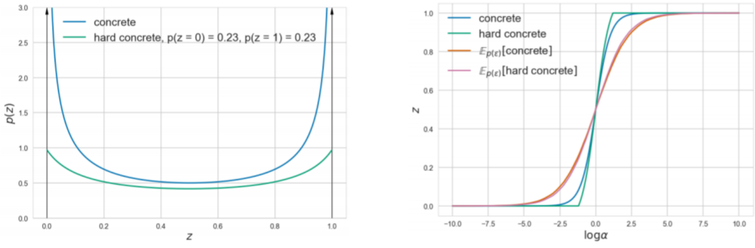
Hard Concrete Distribution

An appropriate smoothing distribution $q(\mathbf{s})$ is the binary concrete rv s :

$$u \sim U(0,1), \quad s = \text{Sigmoid}\left((\log u - \log(1-u) + \log \alpha)/\beta\right)$$

$$\overline{s} = s(\zeta - \gamma) + \gamma \text{ and } \; z = \min(1, \max(0, \overline{s}))$$

1. s is a concrete binary distributed
2. α is the location parameter, and
3. β is the temperature parameter
4. z is the hard concrete distribution.
5. we stretch $s \rightarrow \bar{s}$ into the range (γ, ζ) where $\zeta < 0$ and $\gamma > 1$.



(a) (b)

Figure 2: **(a)** The binary concrete distribution with location $\log\alpha = 0$ and temperature $\beta = 0.5$ and the hard concrete equivalent distribution obtained by stretching the concrete distribution to $(\gamma = -0.1, \zeta = 1.1)$ and then applying a hard-sigmoid. Under this specification the hard concrete distribution assigns, roughly, half of its mass to $\{0, 1\}$ and the rest to $(0, 1)$. **(b)** The expected value of the afforementioned concrete and hard concrete gate as a function of the location $\log\alpha$, obtained by averaging 10000 samples. We also added the value of the gates obtained by removing the noise entirely. We can see that the noise smooths the hard-sigmoid to a sigmoid on average.

Figure: Figure 2 from Iousizos et al

Hard Concrete Distribution

From earlier, we had $1 - Q_{\boldsymbol{\phi}}(\mathbf{s} \leq \mathbf{0})$ in L_0 complexity loss of the objective function. Now, if the random variable is hard concrete, then we can say:

$$1 - Q_{\boldsymbol{\phi}}(\mathbf{s} \leq \mathbf{0}) = \mathrm{Sigmoid}(\log \alpha - \beta \log \frac{-\gamma}{\zeta})$$

During test time, the authors use the following for the gate:

$$\hat{\mathbf{z}} = \min(\mathbf{1}, \max(\mathbf{0}, \mathrm{Sigmoid}(\log \boldsymbol{\alpha})(\zeta - \gamma) + \gamma)) \text{ and } \boldsymbol{\theta}^* = \tilde{\boldsymbol{\theta}}^* \odot \hat{\mathbf{z}}$$

Experiments - MNIST Classification and Sparsification

Network & size	Method	Pruned architecture	Error (%)
MLP 784-300-100	Sparse VD (Molchanov et al., 2017)	512-114-72	1.8
	BC-GNJ (Louizos et al., 2017)	278-98-13	1.8
	BC-GHS (Louizos et al., 2017)	311-86-14	1.8
	$L_{0_{hc}}, \lambda = 0.1/N$	219-214-100	1.4
	$L_{0_{hc}}, \lambda$ sep.	266-88-33	1.8
LeNet-5-Caffe 20-50-800-500	Sparse VD (Molchanov et al., 2017)	14-19-242-131	1.0
	GL (Wen et al., 2016)	3-12-192-500	1.0
	SBP (Neklyudov et al., 2017)	3-18-284-283	0.9
	BC-GNJ (Louizos et al., 2017)	8-13-88-13	1.0
	BC-GHS (Louizos et al., 2017)	5-10-76-16	1.0
	$L_{0_{hc}}, \lambda = 0.1/N$	20-25-45-462	0.9
	$L_{0_{hc}}, \lambda$ sep.	9-18-65-25	1.0

Experiments - MNIST Classification

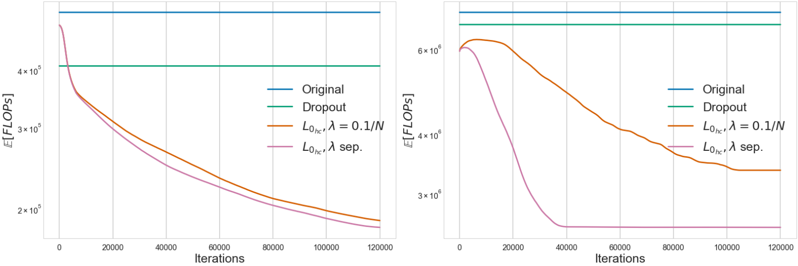


Figure: Expected FLOPs. Left is the MLP. Right is the LeNet-5

Experiments - CIFAR Classification

Network	CIFAR-10	CIFAR-100
original-ResNet-110 (He et al., 2016a)	6.43	25.16
pre-act-ResNet-110 (He et al., 2016b)	6.37	-
WRN-28-10 (Zagoruyko & Komodakis, 2016)	4.00	21.18
WRN-28-10-dropout (Zagoruyko & Komodakis, 2016)	3.89	18.85
WRN-28-10-$L_{0_{hc}}$, $\lambda = 0.001/N$	**3.83**	**18.75**
WRN-28-10-$L_{0_{hc}}$, $\lambda = 0.002/N$	3.93	19.04

Experiments - CIFAR Classification

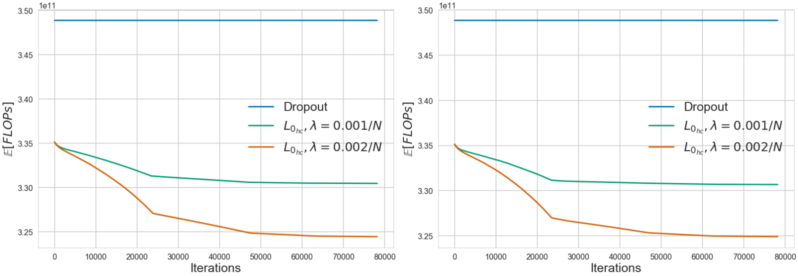


Figure: Expected FLOPs of WRN at CIFAR 10 (left) & 100 (right)

Discussion & Future Work

Discussion

1. L_0 penalty can save memory and computation
2. L_0 regularization lead to competitive predictive accuracy and stability

Future Work

1. Adopt a full Bayesian treatment over the parameter θ

Thank You ...