Estimating Sparse Parameterization of Neural Networks

Ing. Lukáš Kulička

Stochastic and Physical Monitoring Systems Conference 2022
Supported from CTU Grant SVK 31/22/F4

June 23, 2022

Contents

Motivation

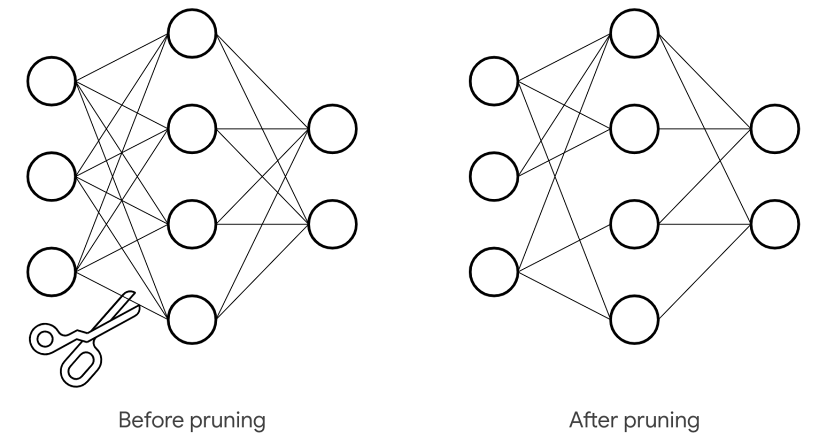

Figure 1: Example of pruning. Taken from: `https://blog.tensorflow.org/2019/05/tf-model-optimization-toolkit-pruning-API.html`.

Problem Formulation

- Sparse parameterization increases interpretability and reduces model complexity while preserving overall information (*pruning*).
- Probabilistic approach.
- A new approach to optimization using natural parameter distributions - a follow-up to Mohammad Khan's paper *Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam*.
- The goal: build methods that learn an arbitrarily complex model and find a sparse parameterization.
- `Julia 1.6.xx`, `Flux.jl` ML library.

Why is it Good to be Bayesian?

Bayes' theorem

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathcal{D}, \boldsymbol{\theta})d\boldsymbol{\theta}} \tag{1}$$

- $p(\mathcal{D}|\boldsymbol{\theta})$ - *likelihood*
- $p(\boldsymbol{\theta}|\mathcal{D})$ - *posterior*
- $p(\boldsymbol{\theta})$ - *prior*
- $p(\mathcal{D})$ - *evidence*

Artificial NN vs. Bayesian NN

$$\mathbf{h}^{(0)} = \mathbf{X},\ \mathbf{h}^{(l)} = a\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right),\ l = 1, \dots, L \\ \mathbf{y} = a^{(\text{out})}\left(\mathbf{W}^{(L+1)}\mathbf{h}^{(L)}\right) \tag{2}$$

ANN

- $\mathcal{D} = (\mathbf{y}, \mathbf{X}),\ \boldsymbol{\theta} = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L+1}$
- Likelihood, (L_1 regularization)
- Goal: $\underbrace{\mathbf{y} = f(\mathbf{X}, \boldsymbol{\theta})}_{\text{deterministic function}}$

BNN

- $\mathcal{D} = (\mathbf{y}, \mathbf{X}),\ \boldsymbol{\theta} = \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L+1}$
- Likelihood & prior
- Goal: $\underbrace{\mathbf{y} = f(\mathbf{X}, \boldsymbol{\theta})}_{\text{random function}},\ p(\boldsymbol{\theta}|\mathcal{D})$

Shrinkage Priors

- Model parameters (*weights*, *biases*) (mostly) in combination with model hyper-parameters $\boldsymbol{\alpha} \rightarrow$ hierarchical parameterization.
- Gaussian Scale Mixtures as marginal model parameters prior:

$$p(\boldsymbol{\theta}) = \int \mathcal{N}(\mathbf{0}, \boldsymbol{\alpha}^2\boldsymbol{\sigma}^2)p(\boldsymbol{\alpha})d\boldsymbol{\alpha} \tag{3}$$

Variance prior $p(\alpha_j^2)$	Marginal prior $p(\theta_j)$
Exponential	Laplace
Inverse-Gamma	Student-t
Bernoulli	Spike and Slab

Table 1: Variance priors and their corresponding marginal priors.

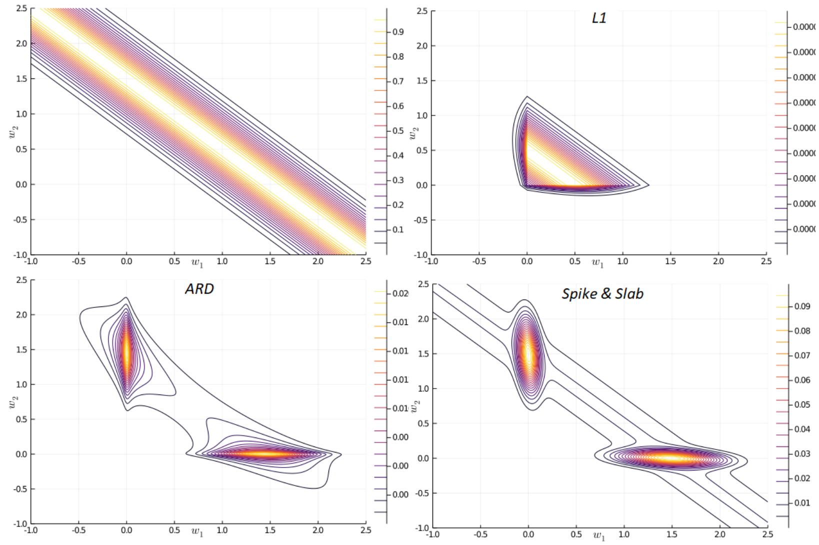

Figure 2: Gaussian likelihood with no prior vs. with Laplace (L_1), Student-t (ARD) and Spike and Slab prior. Taken from: [1].

Variational Inference

- Problem: (mostly) intractable integrals $\int p(\mathcal{D}, \mathbf{z}) d\mathbf{z}$ in Bayes' rule.
- Solution: find surrogate distribution $q(\mathbf{z}) \approx p(\mathbf{z}|\mathcal{D})$.

Evidence Lower Bound (ELBO) & KL Divergence

$$\log p(\mathcal{D}) = \mathcal{L}\big(q(\mathbf{z})\big) + \mathsf{KL}\big(q(\mathbf{z})||p(\mathbf{z}|\mathcal{D})\big) \tag{4}$$

Maximizing the ELBO

$$q_{\mathsf{opt}}(\mathbf{z}) = \arg\max_{q(\mathbf{z}) \in \mathcal{Q}} \mathcal{L}\big(q(\mathbf{z})\big) \tag{5}$$

Standard vs. Variational Optimization

Standard way

- Laplace prior as the L_1 regularization:

$$\text{argmax}_\theta \ \log p(\mathcal{D}|\boldsymbol{\theta}) + \lambda \sum_{j=1}^{J} |\theta_j| \tag{6}$$

Variational way

- Factorize the posterior $q(\mathbf{z}) \approx q(\boldsymbol{\theta})q(\boldsymbol{\psi})$.
- Analytical solution of the ELBO for the factor $q(\boldsymbol{\psi}|\boldsymbol{\gamma}, \boldsymbol{\delta})$.
- Factor $q(\boldsymbol{\theta})$ meets the VADAM conditions.

Variational ADAM with the ARD Prior

- We propose

$$f_{\text{obj}} = -\frac{1}{N}\sum_{n=1}^{N} \log p(\mathcal{D}_n|\boldsymbol{\theta}) - \sum_{j=1}^{J} \frac{1}{2}\theta_j^2\psi_j \tag{7}$$

1. **Initialize** prior parameters, learning rates in ADAM.
2. **Calculate** posterior parameters of Gamma factor $\gamma_{j,(t)}$, $\delta_{j,(t)}$.
3. $\psi_{j,(t)} \leftarrow \frac{\gamma_{j,(t)}}{\delta_{j,(t)}}$.
4. **Update** f_{obj} with VADAM.
5. **Update** $\gamma_{j,(t)}$, $\delta_{j,(t)}$
6. $(t+1) \leftarrow (t)$.

Sparse Logistic Regression

- Dataset: IRIS[1], $\mathcal{D} = (\mathbf{y}, \mathbf{X})$, where $\mathbf{y} \in \{\text{setosa}, \text{versicolor}, \text{virginica}\}^{150}$, $\mathbf{X} \in \mathbb{R}^{150\times 4}$.
- Model architecture: input layer of dimension 4, one hidden layer containing 8 neurons and ReLU activation, output layer of dimension 3 with softmax output activation $\rightarrow$ 67 trainable parameters.
- Goal: prune the network and obtain a sparse parameterization with minimal error increase on test data.
- Methods: L_1 regularization & Variational ADAM with the ARD prior.

[1] `https://archive.ics.uci.edu/ml/datasets/iris`

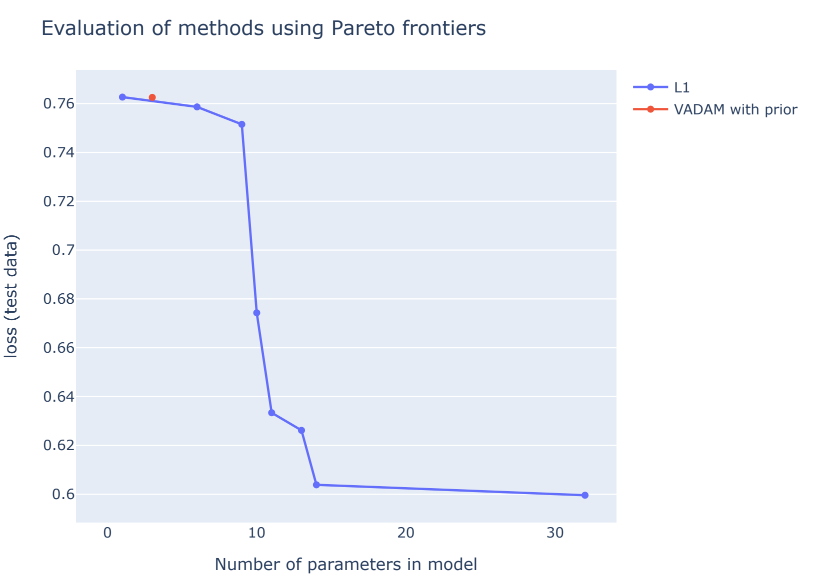

Figure 3: The evaluation of the methods used on the logistic regression problem. Taken from: [2].

Sparse Multi-Instance Learning

- Dataset: Musk[2] with 92 *bags* containing 476 instances each of dimension 166 and $\mathbf{y} \in \{0,1\}^{92}$.
- Model architecture: input layer of dimension 166, first hidden layer containing 10 neurons and $\tanh$ activation, pooling layer with MeanMax aggregation, second hidden layer containing 10 neurons and $\tanh$ activation, output layer of dimension 2 with sigmoid output activation $\rightarrow$ 1922 trainable parameters.
- Goal: prune the network and obtain a sparse parameterization with minimal error increase on test data.
- Methods: L_1 regularization & Variational ADAM with the ARD prior.

[2] `https://archive.ics.uci.edu/ml/datasets/Musk+(Version+2)`

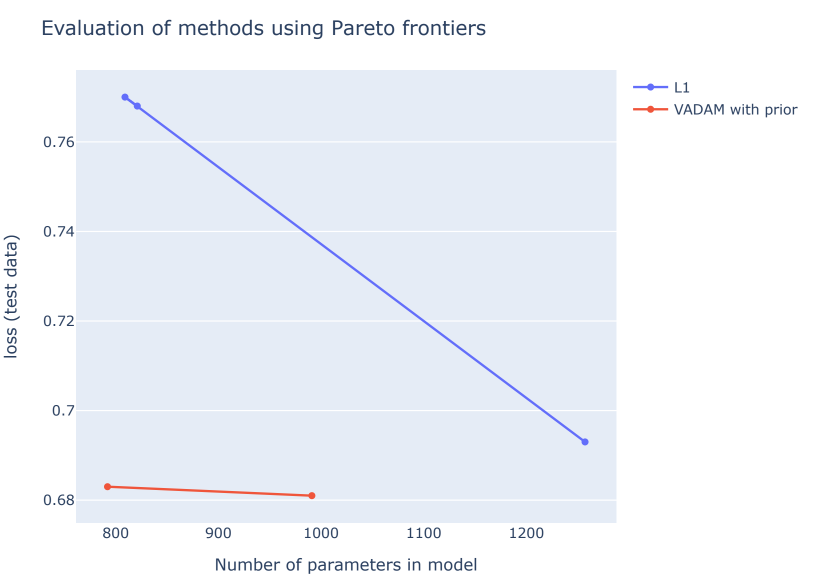

Figure 4: The evaluation of the methods used on the MIL problem. Taken from: [2].

Conclusion

- A new method was proposed to find a sparse parameterization of neural networks.
- Method was applied to a logistic regression model and a multi-instance learning model.
- Subsequently, it was tested and compared with the classical method of regularization in neural networks.

Thank you for your attention.

References

1 KULIČKA L. *Klasifikace dat popsaných stromovou strukturou*. Bachelor's Thesis. Czech Technical University in Prague. 2020.

2 KULIČKA L. *Estimating Sparse Parameterization of Neural Networks*. Master's Thesis. Czech Technical University in Prague. 2022.