Estimating Model Uncertainty of Neural Networks in Sparse Information Form

Jongseok Lee, Matthias Humt, Jianxiang Feng and Rudolph Triebel

Institute of Robotics and Mechatronics, German Aerospace Center (DLR)

Main Idea:

A dual representation of Gaussian distribution:

$$\begin{aligned}\bar{x} &\propto \exp\left\{-\frac{1}{2}(\bar{x}-\mu)^T\Sigma^{-1}(\bar{x}-\mu)\right\}\\ &= \exp\left\{-\frac{1}{2}\bar{x}^T\Sigma^{-1}\bar{x}+\mu^T\Sigma^{-1}\bar{x}\right\}\\ &= \exp\left\{-\frac{1}{2}\bar{x}^T\boldsymbol{I}\bar{x}+\mu^{IV}\bar{x}\right\} \text{ or } \bar{x}\sim\mathcal{N}^{-1}(\mu^{IV},\boldsymbol{I})\end{aligned}$$

Our main idea is to estimate the posterior distribution of neural networks with this formulation, also called information form. Key benefits are:

- Approximate Bayesian inference can be simplified to scalable Laplace Approximation with provable guarantees.
- Sparsity can be exploited as information content on each parameters tends to be sparse with over-parameterization.

Bayesian Neural Networks

Bayesian framework of Neural Networks by inferring probability distributions over models weights:

$$p(\theta|x,y)=\frac{p(y|x,\theta)p(\theta)}{p(y|\theta)} \quad (1)$$

Marginalization over possible values of parameters for predictive uncertainty:

$$p(y^*|x^*,x,y)=\int p(y^*|x^*,\theta)p(\theta|x,y)d\theta \quad (2)$$

One of the main challenge is scalability!

Important Results:

Bayesian Neural Network with an expressive variational family can scale to large deep neural networks and data-sets (e.g. Dense121 on ImageNet classification task), while the space complexity remains similar to mean-field approximations.

Approximate Inference

- Use Laplace Approximation:

$$p(\theta|D)\sim\mathcal{N}(\theta_{\text{map}},\boldsymbol{I}^{-1}) \text{ or } \sim\mathcal{N}^{-1}(\theta^{IV}_{\text{map}},\boldsymbol{I}).$$

Only need estimates of the Hessian of neural networks, e.g. information matrix $\boldsymbol{I}$.

- Diagonals are known and easy to compute:

$$\boldsymbol{I}=\mathbb{E}[\delta\theta\delta\theta^T] \text{ and } \boldsymbol{I}_{ii}=\mathbb{E}[\delta\theta_i^2]$$

This is not true for the covariance matrix.

- Kronecker Eigen-decomposition plus diagonal:

$$\boldsymbol{I}_{\text{inf}}=(U_A\otimes U_G)\Lambda(U_A\otimes U_G)^T+D$$

Eigenvalues Λ and diagonal matrix D are computed using the known diagonals of information matrix. Leads to a guarantee that the information matrix is more accurate than the state-of-the-art Hessian approximators, e.g. KFAC, w.r.t a norm.

Low Rank Sampling Computations

- Predictive uncertainty requires sampling:

$$p(y^*|x^*)\approx\frac{1}{T}\sum_{t=1}^{T}y^*(x^*,\theta_t^s) \quad \text{for } \theta_t^s\sim\mathcal{N}^{-1}(\theta^{IV}_{MAP},\boldsymbol{I}_{\text{inf}}).$$

This requires an expensive inversion!

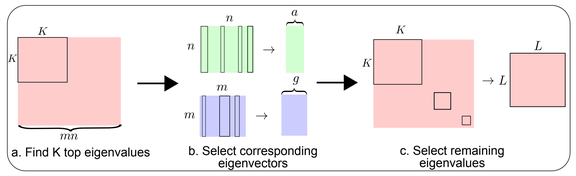
- The devised cheaper computation routine:

1. Low rank approximation while saving Kronecker products in eigenvectors.

$$\boldsymbol{I}_{\text{inf}}\approx\hat{\boldsymbol{I}}_{\text{inf}}=(U_a\otimes U_g)\Lambda_{1:L}(U_a\otimes U_g)^T+D$$

2.The samples can be drawn from the information form, in which we need an inversion over only L by L matrix rather than the whole parameter space.

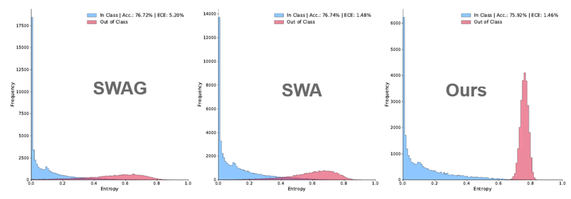
Sparsification Algorithm



- Conventional low rank approximation cannot preserve the Kronecker structure!
- We propose a simple Kronecker-based sparsification algorithm (shown in figure above).

Results

- ImageNet out-of-distribution detection tasks.

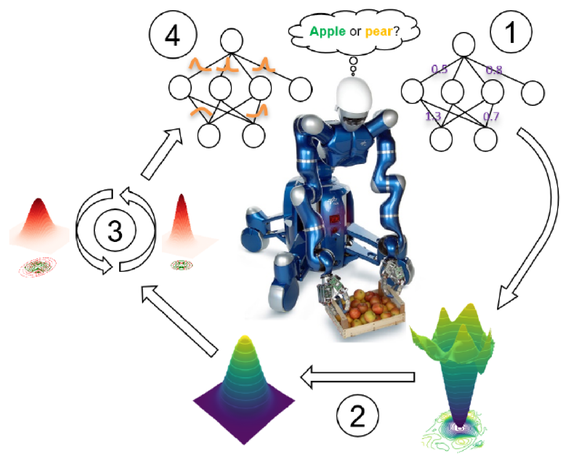


- Space complexity comparisons.

	Diag	KFAC	EFB	INF
Model	Size	Size	Size	Size
ResNet18	*44.6*	362.4	407.0	**47.0**
ResNet50	*97.3*	586.9	684.2	**105.3**
ResNet152	*229.0*	1485.9	1714.9	**250.1**
DenseNet121	*30.1*	393.3	423.4	**37.0**
DenseNet161	*108.6*	1446.2	1554.7	**123.3**

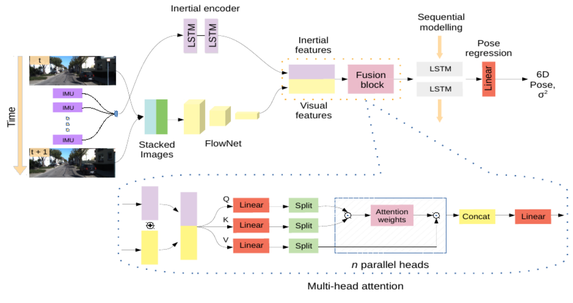
- Our approach can scale to large data-sets and architectures such as ImageNet set-ups.
- Expressive posterior family can be made as tractable as mean-field approximations.
- Other experiments can be found in our ICML 2020 paper. Link.

Extension 1



- On how Bayesian Optimization can be used to further improve the performance, and ease the hyperparameter searches. Link.

Extension 2



- On how our method can be used for the task of learning the motion of a car using IMU and monocular cameras. Link.

Open-source Software

- Curvature Library at DLR-RM github