Graph Reparameterizations for Enabling 1000+ Monte Carlo Iterations in Bayesian Deep Neural Networks

Jurijs Nazarovs[†], Ronak R. Mehta[†], Vishnu Suresh Lokhande[†], Vikas Singh[†]

[†] University of Wisconsin - Madison, USA

37th Conference on Uncertainty in Artificial Intelligence
July 27-30, 2021
Online
uai2021

Overview

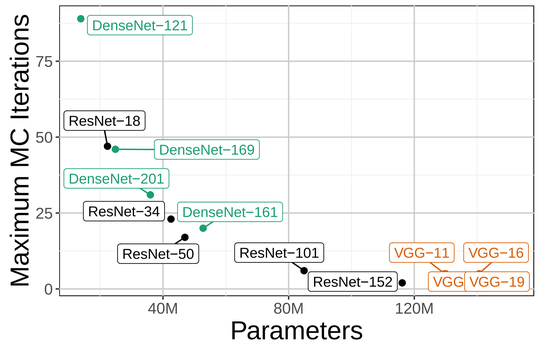
- New framework to run many more MC iterations in Bayesian Deep Neural networks (needed for more general priors). Significant decrease in GPU memory needs and improvements in runtime.
- Leads to smaller variances of the MC estimators, improving training convergence and final accuracy.

Motivation

- Together with ability to provide uncertainty, one of the features of BNNs is the flexibility of choosing posterior distribution q and prior p.
- This choice might significantly impact performance and numerical optimization.

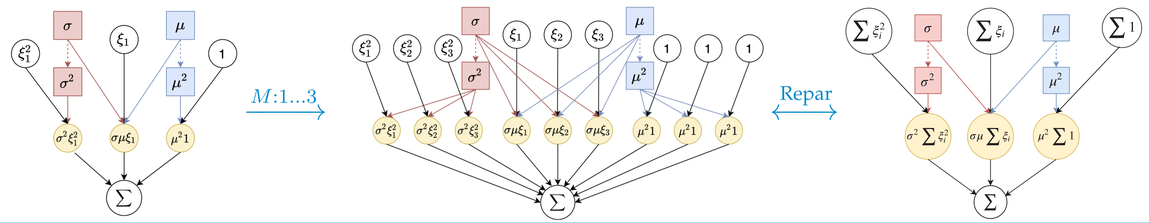
$$\boldsymbol{\theta}^* = \arg\min_{\theta} KL\left(q_\theta || p\right) - \mathbb{E}_{q_\theta}\left[\ln p(y|W,x)\right]$$

- Some choices lead to a closed form solution of KL. But others require iterative estimators (like MC).
- For direct implementation, the number of MC iterations for deep BNNs limited by GPU memory.



- Is there a way to surpass this limit, say by 1000×? In special cases, yes!

Methods and main results



- Loss minimization requires computing $\mathbb{E}_{q_\theta}\left[g(w)\right]$. However, for many q_θ, this quantity is intractable.
- Monte Carlo estimation to the rescue!

$$\mathbb{E}_{q_\theta}\left[g(w)\right] \approx \frac{1}{M}\sum_{i=1}^{M} g(w_i), \text{ where } w_i \sim Q_\theta.$$

It is **(a)** simple, **(b)** unbiased, **(c)** asymptotically exact.

- But Monte Carlo estimation also has issues!
- Consider a standard implementation for the MC approximation of $\mathbb{E}_{q_\theta}\left[w^2\right]$ and $W_\theta \sim N(\mu, \sigma^2)$.

```
for i in range(M):
    # sample 1 observation from N(0, 1)
    sample = sampler_normal.sample()
    w = mu * 1 + sigma * sample
    loss += w^2 / M
```

Increasing M results in GPU memory explosion.

- Reason: Computation Graphs (CG)
  The size of the CG (top Figure) grows linearly $O(M)$ with the number of MC iterations.
- Solution: Is there a CG reparameterization, such that size with $M = 3$ equals size with $M = 1$?.

- We describe a **parameterization tuple**: a way to measure size of CG created by MC.
- Provide **recipe**: how to identify distributions, where a CG reparameterization makes the size of CG independent of M (see summary in Table).
- Provide API to design your own BNN or use predefined BNN versions of Resnet, Densenet and VGG.

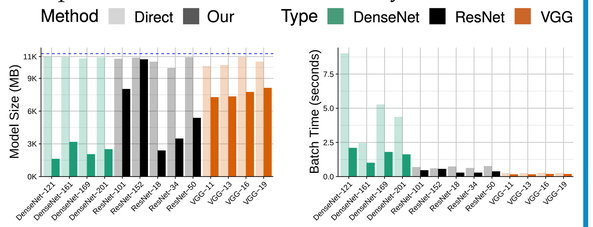
```
model = AlexNet(n_classes=10, n_channels=3,
                approx_post="Radial",
                kl_method="repar",
                n_mc_iter=1000)
```

Sampling: $W(\theta,\xi)$	Approximate Posterior p.d.f. q_θ		Prior p.d.f. $p(w)$		
Scaling property family: $W(\theta,\xi)=\theta\xi$	Exponential(θ) Rayleigh(θ) Erlang(k,θ) Error(a,θ,c) Inverse-Gamma(k,θ)	Standard Wald(θ) Weibull(k,θ) Gamma(k,θ) Log-Gamma(k,θ)	Exponential Dirichlet Inverse-Gamma Log-normal Inverse-Gaussian	Standard Wald Chi-squared Gamma Error Normal	Rayleigh Pareto Erlang Weibull
Location-Scale family: $W(\theta,\xi)=\mu+\sigma\xi$, $\theta=(\mu,\sigma)$	Normal(μ,σ) Logistic(μ,σ) Radial(μ,σ)	Laplace(μ,σ) Horseshoe(μ,σ)	Logistic	Exponential Laplace	Normal
			Normal variations, e.g., Horseshoe, Radial		
	Log-Normal(μ,σ)		Dirichlet	Pareto	

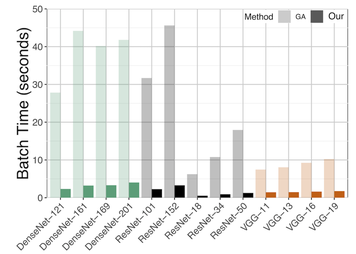
Research supported in part by NIH grants RF1 AG059312 and RF1 AG062336. RRM was supported in part by NIH Bio-Data Science Training Program T32 LM012413 grant to the University of Wisconsin Madison.

Experimental Results

- **CIFAR-10 Training**: For maximum possible number of MC iterations for a given model via the direct MC method, we show: Model size (dashed blue line indicates GPU capacity, 11GB) and training time. For some networks, our method occupies less than 25% of memory and 5× faster.



MC sampling is much slower using Gradient Accumulation (GA). Reparameterization reduces compute time by up to 14× for some networks.



- **CIFAR-10 Accuracy**: Confidence Set Accuracy and Confidence Sets for ResNet/DenseNet models with 100 MC iterations (not previously possible). Both ResNet and Densenet achieve accuracy of more than 90% with 100% confidence, but ResNet is 100% confident on almost 90% of the data.