Bayesian Deep Learning and a Probabilistic Perspective of Generalization

Advay Koranne (ak845) and Duncan Jurman(dj383)

Cornell University CS 4782

May 2, 2024

Paper Introduction

1. Title: Bayesian Deep Learning and a Probabilistic Perspective of Generalization. [WI22]
2. Author: Andrew Gordon Wilson Pavel Izmailov (New York University).
3. Date: Submitted on 20 Feb 2020
4. Venue: Advances in Neural Information Processing Systems 33 (NeurIPS 2020).

Background and Motivation

1. Bayesian marginalization is used in deep neural networks to improve accuracy and calibration by considering multiple weight configurations rather than just one.
2. Deep ensembles are effective at approximating Bayesian marginalization, and the proposed method further enhances predictive distributions by marginalizing within attraction basins without significant extra cost.
3. SWAG (Stochastic Weight Averaging-Gaussian) extends the ensemble method by approximating the posterior distribution of weights with a Gaussian.

Method: Bayesian Model Averaging (BMA)

Equation

$$p(y \mid x, \mathcal{D}) = \int p(y \mid x, w) p(w \mid \mathcal{D}) \, dw$$

Variables

- Output (y): Output values (e.g., regression values, class labels)
- Inputs (x): Input data (e.g., spatial locations, images)
- Weights (w): Model parameters
- Data ($\mathcal{D}$): Training data

Explanation

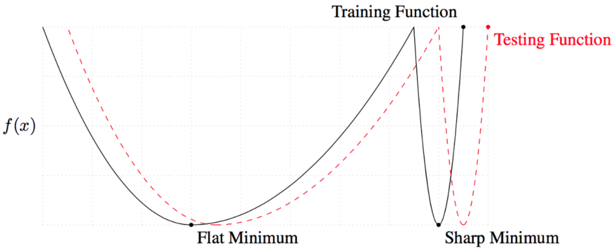
The equation above represents the Bayesian model average (BMA). Instead of relying on a single hypothesis, i.e., one set of parameters w, Bayesian inference uses all possible sets of parameters, each weighted by their posterior probability, to compute predictions.

Bayesian Marginalization Continued

In classical learning, the posterior is often approximated as $p(w \mid \mathcal{D}) \approx \delta(w = \hat{w})$, where $\hat{w} = \arg\max_w p(w \mid \mathcal{D})$. There are two cases:

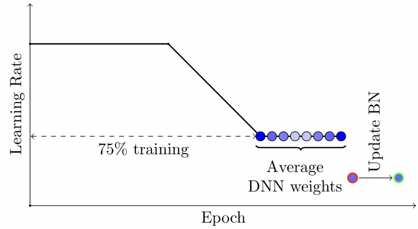
1. When $p(w|\mathcal{D})$ is sharply peaked, the conditional prediction $p(y|x, w)$ does not vary significantly across different values of w.
2. In modern neural networks, where data may be underspecified, various parameter settings can lead to a diverse array of plausible hypotheses for the data.

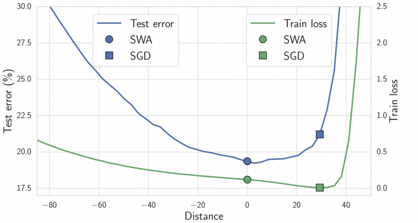
Stochastic Weight Averaging



Stochastic Gradient Descent will converge, but if the testing function is shifted, it may end up very wrong if it converged to a sharp minimum. [Izm+19]

Stochastic Weight Averaging

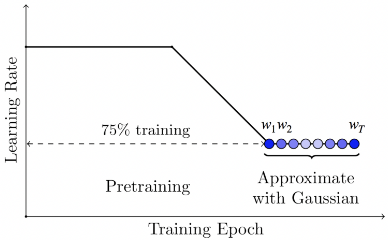




Gradient Descent tends to converge to sharp minima, often lying near the boundary of the loss function. Conversely, the Stochastic Weight Averaging (SWA) solution is centered in a wide region of low training loss.

Contribution: Stochastic Weight Averaging Gaussian (SWAG)

- **Compute Moments:** Calculate the first two moments (mean and variance) of the SGD trajectory. [Mad+19]
- **Gaussian Approximation:** Use these moments to construct a Gaussian approximation in the weight space.
- **Bayesian Model Averaging:** Sample from this Gaussian distribution.



Deep Ensembles vs. Bayesian Model Averaging

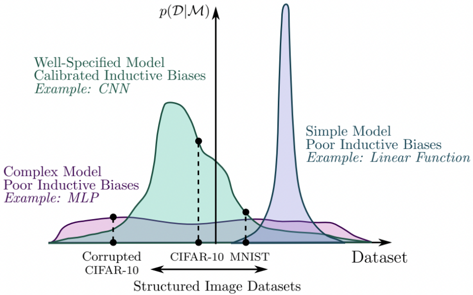
- **Deep Ensembles:**
  - Trains multiple models; averages predictions.
  - Offers robustness and improved generalization.
- **Bayesian Model Averaging (BMA):**
  - Weights predictions by model's posterior probabilities.
  - Captures full model uncertainty, more comprehensive.
- **Key Differences:**
  - *Aggregation:* Ensembles use simple average; BMA uses weighted average based on probability.

Multi Stochastic Weight Averaging Gaussian (MultiSWAG)

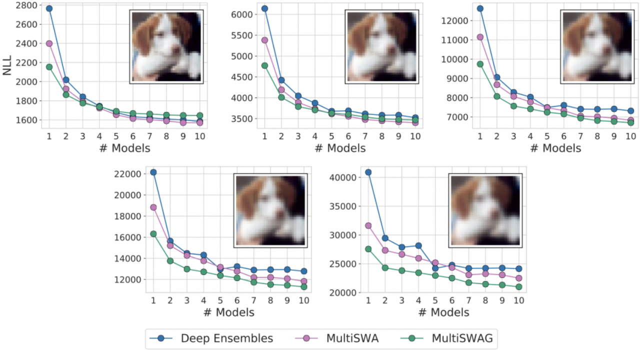
1. Deep Neural Networks often exhibit “basins” where the loss is minimized.
2. MultiSWAG constructs a Gaussian mixture posterior by aggregating multiple independent SWAG solutions.
3. MultiSWAG, like ensembling, achieves Bayesian model averaging by leveraging diverse models to capture and quantify uncertainty.
4. MultiSWAG provides a computationally efficient approach to explore the model space and obtain a more comprehensive estimation of uncertainty.

Bayesian Predictive Distribution and MultiSWAG

$$p(y \mid x, \mathcal{D}) = \underbrace{\int}_{\text{Sum over ensemble}} \underbrace{p(y \mid x, w)}_{\text{Model predictions}} \underbrace{p(w \mid \mathcal{D})\, dw}_{\text{SWAG Gaussian approximation}}$$



Result Re-implementation

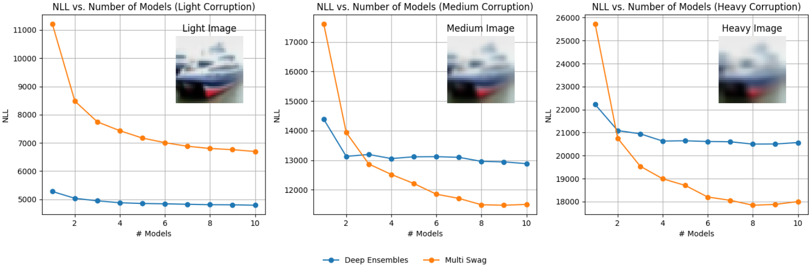


The negative log likelihood performance of Deep Ensembles, MultiSWAG, and MultiSWA was compared using a PreResNet-20 on CIFAR-10 under varying intensities of Gaussian blur corruption.

Re-implementation Approach

1. Utilized ResNet18 from torchvision.
2. Trained the model on CIFAR-10, a dataset comprising 60,000 32x32 color images across 10 classes, with 6,000 images per class. The dataset is split into 50,000 training images and 10,000 test images.
3. Evaluated performance on three levels of image blurring: light, medium, and heavy.
4. Ensembled 10 models and assessed their negative log likelihood.

Re-implementation Approach



Conclusion

1. Understanding Bayesian Deep Learning was a very steep learning curve.
2. We wanted to use libraries whenever possible to make it more efficient for example (CIRFAR-10, ResNet18, etc..)
3. Implementing SWAG was the most difficult part of the project.
4. The scaling for NLL is different than MultiSWAG – have to debug this.

[Izm+19] Pavel Izmailov et al. *Averaging Weights Leads to Wider Optima and Better Generalization*. 2019. arXiv: 1803.05407 [cs.LG].

[Mad+19] Wesley Maddox et al. *A Simple Baseline for Bayesian Uncertainty in Deep Learning*. 2019. arXiv: 1902.02476 [cs.LG].

[WI22] Andrew Gordon Wilson and Pavel Izmailov. *Bayesian Deep Learning and a Probabilistic Perspective of Generalization*. 2022. arXiv: 2002.08791 [cs.LG].