The Effects of Skew on Convolutional Bayesian Neural Network Performance

William Watson
Johns Hopkins University
billwatson@jhu.edu

Abstract

We will examine the effect of a skewed classification task on Bayesian Neural Networks. We will attempt to perform parameter learning via Bayesian approaches to approximate parameter posterior distributions and compare the model's robustness to skew with a regular network.

1 Introduction

Consider the simplest neural network, consisting of a single feature $x \in \mathbb{R}$ and prediction $y \in \mathbb{R}$ as follows:

$$y = w \cdot x + b \tag{1}$$

Usually, we would use frequentist approaches to solve for a point estimate of the feature weight w and bias term b, either through stochastic gradient descent or ordinary least squares. However, lets consider a Bayesian Neural Network with Gaussian priors:

$$\begin{aligned} w &\sim \mathcal{N}\left(\mu_w,\ \sigma_w^2\right) \\ b &\sim \mathcal{N}\left(\mu_b,\ \sigma_b^2\right) \\ y &= w \cdot x + b \\ y &\sim \mathcal{N}\left(x \cdot \mu_w + \mu_b,\ x^2 \cdot \sigma_w^2 + \sigma_b^2\right) \end{aligned} \tag{2}$$

We can use a Bayesian approach to yield a posterior belief about our parameters w and b, which allows us to ascribe an uncertainty to our predictions with confidence intervals. We can also sample models and aggregate predictions to form a more robust predictor given the data. However, the derivation of the posterior distribution can be intractable, and hence we can use variational inference methods and Monte Carlo sampling as an approximation to the true posterior.

2 Data and Task

We plan to use the Rijkmuseum Dataset [3] to perform artist classification. However, we will explore how skew will effect our model's generalization to a test set. The Rijkmuseum Dataset is heavily skewed, with 90 artists having over 200 pieces, Rembrant alone with 1,384 pieces, out of a total collection with 6,629 artists. Hence this dataset will easily provide a skewed setting for experimentation.

We will constrain the classification problem to fewer artists, filtering out lesser-known artists and controlling the skew manually. We will train a model using several methods, and perform an inference task on predicting the artists.

3 Methods

We will train our network using 2 Bayesian methods and compare it with a regular neural network as our baseline.

3.1 Stochastic Variational Inference (SVI)

As detailed in Hoffman et. al [1], we can approximate the posterior distributions by using *Stochastic Variational Inference*. Here, the task is transformed from a complex inference problem into a stochastic optimization problem that follows noisy estimates of the gradient to maximize the *evidence lower bound* by repeatedly subsampling the data.

3.2 Markov Chain Monte Carlo (MCMC)

We will attempt a *Markov Chain Monte Carlo* approach using the No-U Turn Sampler (NUTS) [2] to provide a sampling-based method to extract the posterior distributions of our parameters. We can compare both Bayesian approaches in their choice of posterior distributions for our parameters.

3.3 Maximum Likelihood Estimation (MLE)

Our baseline model will consist of an equivalent CNN trained in the normal setting, providing a point estimate through *Maximum Likelihood Estimation*. This will give us an idea on how skew effects point estimate learning dynamics.

4 Software

This project will make use of Pyro[1], a probabilistic programming language using PyTorch[2] to enable a unified framework for building neural networks and Bayesian modeling.

5 Proposed Analysis

For this project, we expect to analyze several results:

5.1 Skew Performance

For each level of skew, we will evaluate the performance in terms of accuracy, and attempt to ascribe confidence intervals on our Bayesian models for predictions.

5.2 Parameter Posterior Estimates

Since we are working with CNNs, we can sample kernels that may provide interesting insight into the distribution of feature extraction.

References

[1] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[2] M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *arXiv e-prints*, page arXiv:1111.4246, Nov 2011.

[3] T. E. J. Mensink and J. C. van Gemert. The rijksmuseum challenge: Museum-centered visual recognition. In *ACM International Conference on Multimedia Retrieval*, 2014.

[1] http://pyro.ai
[2] https://pytorch.org