Assessing the Adversarial Robustness of Monte Carlo and Distillation Methods for Deep Bayesian Neural Network Classification

Meet P. Vadera[1], Satya Narayan Shukla[1], Brian Jalain[2], and Benjamin M. Marlin[1]
[1]University of Massachusetts Amherst, [2]US Army Research Laboratory

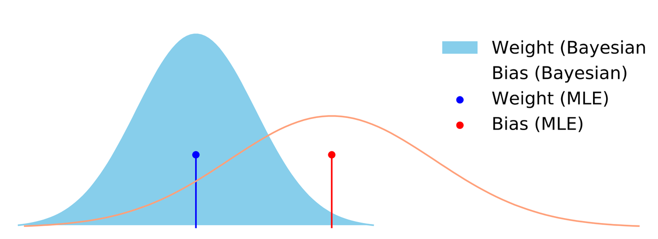
DEVCOM ARMY RESEARCH LABORATORY

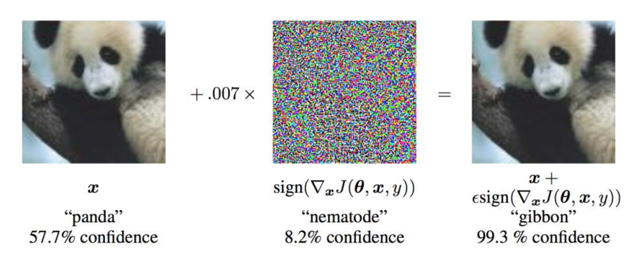
IoBT REIGN

UMassAmherst College of Information & Computer Sciences

Introduction

- Adversarial images, typically generated by modifying standard images with low norm perturbations can “fool” deep neural networks





- A potential reason for this is that decision boundaries of neural networks are unconstrained away from the training data manifold
- Bayesian neural networks (BNNs) can be better behaved away from training data due to the Bayesian model averaging effect.
- In this work, we analyze the adversarial robustness of Markov Chain Monte Carlo (MCMC) based BNNs and their distilled counterparts

Background

- Adversarial attacks are typically either white-box (access to the underlying objective function) or black-box (query access only)
- In our work, we focus on two popular kinds of white-box attacks:
  - **Fast Gradient Sign Method (FGSM)** (Goodfellow et al, 2014): Take an $\boldsymbol{\epsilon}$-step against the loss gradient in the L_∞ space

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon\ \mathrm{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\theta, \mathbf{x}, y))$$

  - **Projected Gradient Method (PGD)** (Madry et al., 2017): Iterative version of FGSM attack maintaining the L_∞ perturbations

$$\mathbf{x}^{t+1} = \Pi_{\mathbf{x}+S}(\mathbf{x}^t + \alpha\ \mathrm{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\theta, \mathbf{x}, y)))$$

- Classical Bayesian inference requires computing the posterior distribution over parameters, and subsequently marginalizing over the posterior to obtain the posterior predictive distribution

$$p(\theta|\mathcal{D}, \theta^0) = \frac{p(\mathcal{D}|\theta)p(\theta|\theta^0)}{\int p(\mathcal{D}|\theta)p(\theta|\theta^0)d\theta} \qquad p(y|\mathbf{x}, \mathcal{D}, \theta^0) = \int p(y|\mathbf{x}, \theta)p(\theta|\mathcal{D}, \theta^0)d\theta$$

- The posterior term is intractable for neural networks. Thus, approximations like variational inference (VI) or MCMC are used
- VI provides a biased but low-variance approximation, while MCMC methods provides an unbiased and high-variance approximation

Methods

- For implementing BNNs, we utilize Stochastic Gradient Langevin Dynamics (SGLD) (Welling & Teh, 2011), a SG-MCMC algorithm to sample from the approximate posterior distribution

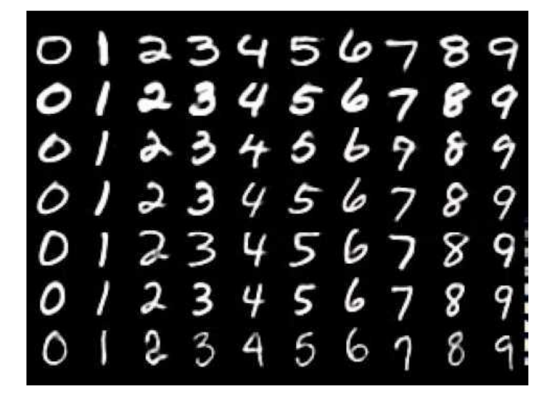
$$\Delta\theta_{t+1} = \frac{\eta_t}{2}\left(\nabla_\theta \log p(\theta|\lambda) + \frac{N}{M}\sum_{i \in \mathcal{S}} \nabla_\theta \log p(y_i|x_i, \theta_t)\right) + z_t \qquad z_t \sim \mathcal{N}(0, \eta_t I)$$

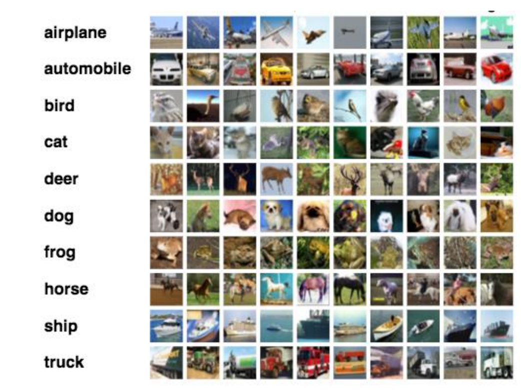
- However, MCMC based methods require storing parameter sets sampled from the posterior to be used during inference
- Bayesian Dark Knowledge (BDK) (Balan et al., 2015) proposes an online method of distilling the posterior predictive distribution of the Bayesian ensemble (teacher) into a smaller compact model (student). This is achieved by minimizing the KL-divergence between the teacher and the student
- Adversarial attacks on ensembles can present a challenge in terms of memory requirements. We circumvent this by sampling one model at a time and accumulating gradients as shown below.

$$\nabla_{\mathbf{x}} \mathcal{L}(y, \mathbf{x}; \theta_{1:K}) = \frac{1}{K}\sum_{i=1}^{K} \nabla_{\mathbf{x}} \mathcal{L}(\cdot; \theta_i)$$

Experimental Protocols

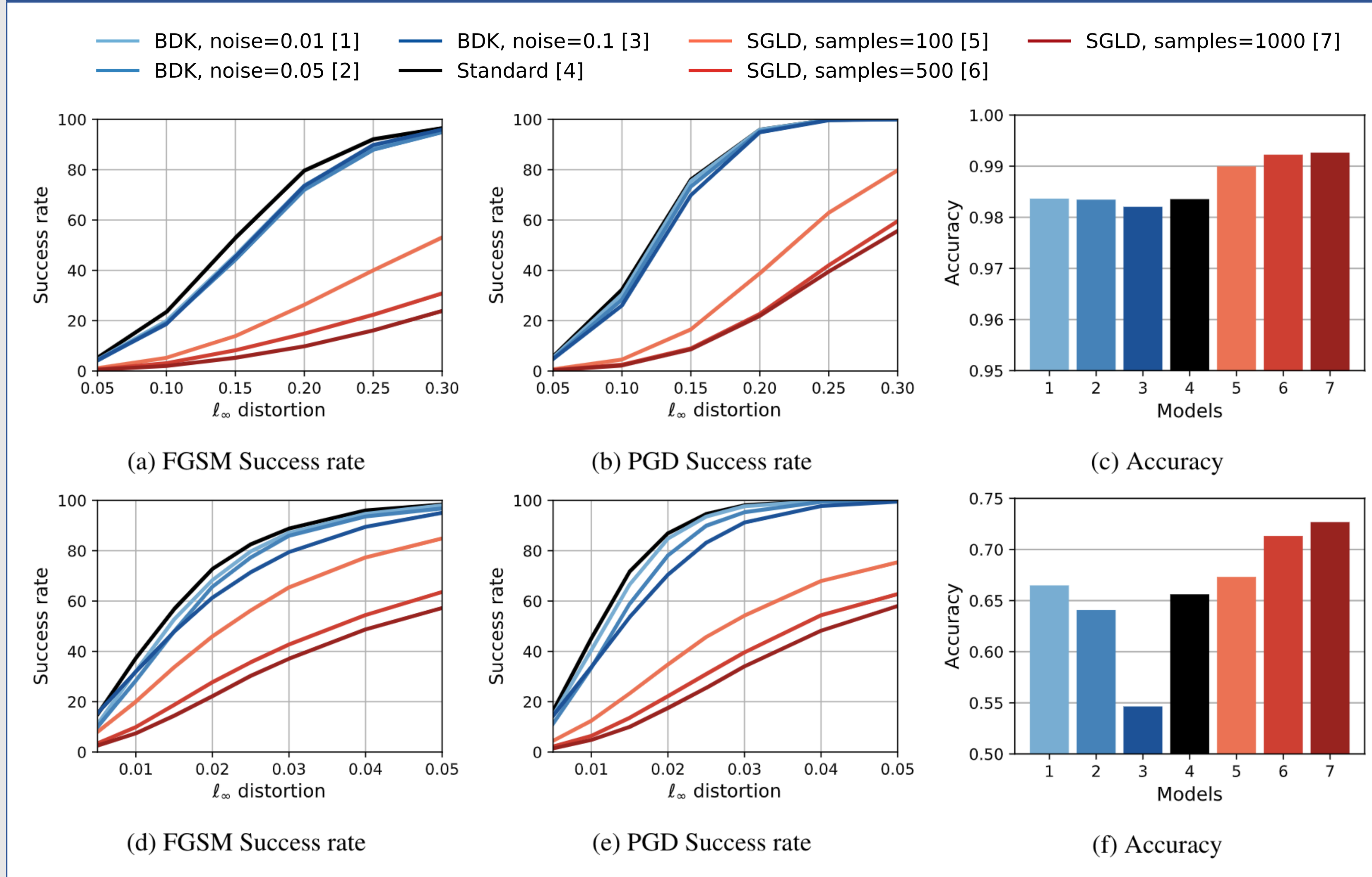
- **Datasets:** MNIST (60k training, 10k test) and CIFAR10 (50k training, 10k test)





- **Models:** 4-layer and 5-layer convolutional neural networks (CNNs) for MNIST and CIFAR10 respectively
  - **MNIST:** Input(1, (28,28)) - Conv(num_kernels=10, kernel_size=4, stride=1) - MaxPool(kernel_size=2) - Conv(num_kernels=20, kernel_size=4, stride=1) - MaxPool(kernel_size=2) - FC (80) - FC (output)
  - **CIFAR10:** Input(3, (32,32)) - Conv(num_kernels=16, kernel_size=5) - MaxPool(kernel_size=2) - Conv(num_kernels=32, kernel_size=5) - MaxPool(kernel_size=2) - FC(200) - FC (50) - FC (output)
- During BDK distillation, we apply a zero-mean and fixed variance Gaussian noise to the input training data. We also assess performance against variance levels.

Experimental Results



(a) FGSM Success rate (b) PGD Success rate (c) Accuracy

(d) FGSM Success rate (e) PGD Success rate (f) Accuracy

Top Row: MNIST, Bottom Row: CIFAR10

Discussion and Future Work

- MCMC based Bayesian ensembles show excellent robustness to adversarial attacks compared to standard point-estimated models
- Under BDK, the student models show improved robustness when compared to point-estimated models, but not at the level of full Bayesian ensembles using MCMC
- Increasing noise during BDK distillation helps improve adversarial robustness of the student, but comes at a cost of reduced accuracy on non-adversarial inputs
- Future work includes further investigation of BDK's posterior predictive distribution representation and a focus on improving adversarial robustness to match the level of full Bayesian ensembles

Contact email: {mvadera, snshukla, marlin}@cs.umass.edu, brian.jalain.ctr@mail.mil
This work was partially supported by the US Army Research Laboratory under cooperative agreement W911NF-17-2-0196.