Adversarial Distillation of Bayesian Neural Network Posteriors
Kuan-Chieh Wang†‡, Paul Vicol†‡, James Lucas†‡, Li Gu†, Roger Grosse†‡, Richard Zemel†‡
†University of Toronto  ‡Vector Institute

Motivation
• Bayesian neural networks (BNNs) are a principled way to reason about uncertainty.
• MCMC methods allow us to sample from the posterior, but have high storage cost.

Summary
• We introduce a framework called Adversarial Posterior Distillation (APD) that uses a Generative Adversarial Network (GAN) to model the BNN posterior.
• We show that APD performs as well as the original posterior samples in the following standard testbeds for BNNs while using less storage:
  • Anomaly detection
  • Active Learning (exploration)
  • Defense against adversarial attacks
• We analyze the suitability of using GANs for APD.

Background
• Stochastic Gradient Langevin Dynamics (SGLD) is an MCMC method that works with mini-batches:
  \[ \Delta \theta^t = \frac{\epsilon^t}{2} \left( \nabla \log p(\theta^t) + \frac{1}{n} \sum_{i=1}^{n} \nabla \log p(y_i | x_i, \theta^t) \right) + \eta^t \]
• GANs can sample from rich posterior distributions. We used the WGAN with gradient penalty (WGAN-GP).

Method

**Algorithm** Offline APD Distillation
1. Sample \( \{ \theta^t \}_{t=1}^{T} \) using MCMC updates, where \( T \) denotes the number of updates.
2. Optimize \( G \) with WGAN-GP loss using \( \{ \theta^t \}_{t=1}^{T} \) as real data.

• Online algorithm has sampling and GAN updates interleaved.

**Method (Cont.)**

\[ p(\theta|D) \sim p_{\text{SGLD}} \]
\[ z \sim \mathcal{N}(0, I) \]
\[ G(z) \]
\[ D \]
\[ 0, 1 \]

Distillation

\[ p(y|x, D) \approx \frac{1}{T} \sum_{t=1}^{T} p(y|x, \theta^t), \theta^t \sim p(\theta|D) \]
\[ p(y|x, D) \approx \frac{1}{T} \sum_{t=1}^{T} p(y|x, G(z^t)), z^t \sim \mathcal{N}(0, I) \]

Inference

SGLD
APD

**Toy Example**

• Problem Setup: Classify mixture of 2 Gaussians
• The deterministic network has a hard decision boundary, while SGLD is uncertain away from data.
• APD gradually learned to model SGLD.

Anomaly Detection

• Task: train only on in-distribution data (i.e. MNIST), and evaluate detection of out-of-distribution data.
• Model: fully connected neural network (784-400-400-10)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SGD</th>
<th>MC-Dropout</th>
<th>SGLD</th>
<th>APD (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>notMNIST</td>
<td>64.2</td>
<td>67.6</td>
<td>54.4</td>
<td>88.0</td>
</tr>
<tr>
<td>Omniglot</td>
<td>84.2</td>
<td>84.9</td>
<td>78.7</td>
<td>91.5</td>
</tr>
<tr>
<td>VR</td>
<td>56.9</td>
<td>61.4</td>
<td>52.2</td>
<td>90.1</td>
</tr>
<tr>
<td>GAUSSIAN</td>
<td>67.3</td>
<td>70.2</td>
<td>57.4</td>
<td>91.3</td>
</tr>
<tr>
<td>Uniform</td>
<td>85.4</td>
<td>80.7</td>
<td>85.8</td>
<td>93.6</td>
</tr>
</tbody>
</table>

• GANs can sample from rich posterior distributions. We used the WGAN with gradient penalty (WGAN-GP).

**Why GANs? / Storage Savings**

• Anomaly detection with increasing number of GMM components
• With APD, the storage cost (i.e., generator size) is fixed

**Active Learning**

• For BNNs, active learning using entropy was able to learn faster than random acquisition.

Adversarial Example Detection - MNIST

• We measured the AUROC for FGSM and PGD adversaries under each source model.

• 'Source' refers to the network used to generate attacks
• Here we used approximate model variance, \( U(x) \):

\[ U(x) = \frac{1}{T} \sum_{t=1}^{T} \left[ \left( \frac{1}{T} \sum_{i=1}^{T} \right) \frac{1}{T} \sum_{i=1}^{T} \right] \]

\[ \text{Source} \quad \text{Attack} \quad \text{MC Drop} \quad \text{SGLD} \quad \text{Ours} \]

\[ \text{MC-Drop} \quad \text{FGSM} \quad \text{PGD} \quad \text{FGSM} \quad \text{PGD} \quad \text{FGSM} \quad \text{PGD} \]

\[ \text{SGLD} \quad \text{FGSM} \quad \text{PGD} \quad \text{FGSM} \quad \text{PGD} \quad \text{FGSM} \quad \text{PGD} \]

\[ \text{Ours} \quad \text{FGSM} \quad \text{PGD} \quad \text{FGSM} \quad \text{PGD} \]

\[ \text{MC-Drop} \quad 89.53 \quad 94.01 \quad 91.70 \]

\[ \text{SGLD} \quad 54.99 \quad 81.76 \quad 75.98 \]

\[ \text{Ours} \quad 54.51 \quad 83.05 \quad 86.02 \]

\[ \text{MC-Drop} \quad 88.37 \quad 93.95 \quad 91.63 \]

\[ \text{SGLD} \quad 56.91 \quad 84.98 \quad 82.60 \]

\[ \text{Ours} \quad 54.98 \quad 88.01 \quad 93.15 \]

\[ U(x) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{T} \sum_{i=1}^{T} \]