1 Introduction

Neural Networks (NNs) have become the dominant approach to addressing computer vision (CV) [1, 2, 3], natural language processing (NLP) [4, 5, 6], speech recognition (ASR) [7, 8] and bioinformatics [9, 10] tasks. However, as observed by [11], they are susceptible to adversarial attacks - perturbations to the input which are almost imperceptible to humans, yet which drastically affect the predictions of the neural network. It was found that adversarial attacks are transferable [11, 12], that it is possible to craft adversarial attacks within the physical world [13] and that adversarial attacks are hard to defend against [14, 11, 15]. Currently, there are far more methods of successfully attacking networks than there are of defending networks [14, 12, 13, 16, 17, 18, 19, 20]. Altogether, this raises serious concerns about how safe it is to deploy neural networks for high-stakes applications.

In work done by [14] it was shown that adversarial attacks can be detected using a range of approaches. Unfortunately, it turns out that attacks can then be crafted to fool the proposed detection schemes. However, [14] singles out detection of adversarial attacks using uncertainty measures derived from Monte-Carlo dropout as being the most successful of the evaluated methods. Detection of adversarial attack using Monte-Carlo dropout was further investigated in [21]. In [21] adversarial attacks are interpreted as inputs which lie off the manifold of natural images - the stronger the adversarial perturbation, the further is the input from the manifold. Thus, adversarial samples can be seen as 'off-manifold' out-of-distribution inputs. This suggests that adversarial attacks can be detected using measures of model or distributional uncertainty [21] provided by approaches like Monte-Carlo dropout. Recently, [22] proposed Prior Networks - a new approach to modelling uncertainty which has been shown to outperform Monte-Carlo dropout on a range of tasks. Unlike approaches such as Monte-Carlo dropout, which indirectly specify a conditional distribution over output distributions, a Prior Network \( p(\pi|x^*; \theta) \) explicitly parametrizes a prior distribution over categorical output distributions.

Contributions. This work investigates the detection of Fast Gradient Sign Method (FGSM) [11], Basic Iterative Method (BIM) [13] and Momentum Iterative Method (MIM) [16] adversarial attacks using measures of model or distributional uncertainty derived from either a Monte-Carlo dropout derived ensemble or Prior Networks, respectively. Two threat models are assessed - adversarial attacks which have no knowledge of the detection scheme and detection-avoiding adversarial attacks which have full knowledge of the detection scheme. Results show that Prior Networks successfully detect both standard FGSM, BIM and MIM whitebox and blackbox adversarial attacks and also detection-avoiding whitebox and blackbox adversarial attacks.

2 Uncertainty Estimation

Bayesian approaches treat model parameters \( \theta \) as random variables and place a prior distribution \( p(\theta) \) over them to compute the posterior distribution \( p(\theta|D) \) via Bayes’ rule. Uncertainty in the model parameters induces a distribution over predictive distributions \( P(y|x^*; \theta) \) for each observation \( x^* \).
each set of model parameters parameterizes a conditional distribution over class labels. The expected predictive distribution $P(y|x^*, \mathcal{D})$ is obtained by marginalizing out the parameters. Unfortunately, both the marginalization and calculation of the model posterior $p(\theta|\mathcal{D})$ are intractable for neural networks. Typically the model posterior distribution is approximated using either an implicit or explicit variational approximation $q(\theta)$ and the integral is approximated via sampling (eq. 1), using approaches such as Monte-Carlo dropout [23].

$$
P(y|x^*, \mathcal{D}) = \int P(y|x^*, \theta)p(\theta|\mathcal{D})d\theta \approx \frac{1}{M} \sum_{i=1}^{M} P(y|x^*, \theta^{(i)}), \theta^{(i)} \sim q(\theta)$$

By selecting an appropriate approximate inference scheme and model prior $p(\theta)$ Bayesian approaches aim to craft a model posterior $p(\theta|\mathcal{D})$ such that the ensemble of distributions $\{P(\omega_i|x^*, \theta^{(i)})\}_{i=1}^{M}$ sampled from $q(\theta)$ is consistent in-domain and becomes increasingly diverse the further away $x^*$ is from the region of training data. The entropy of the expected distribution $P(\omega_i|x^*, \mathcal{D})$ will indicate the total uncertainty in predictions. Measures of the diversity of the ensemble, such as Mutual Information, assess uncertainty in predictions due to model uncertainty.

$$
\frac{\mathcal{MI}[y, \theta|x^*, \mathcal{D}]}{\text{Total Uncertainty}} = \frac{\mathcal{H}[\mathbb{E}_{p(\theta|\mathcal{D})}[P(y|x^*, \theta)]] - \mathbb{E}_{p(\theta|\mathcal{D})}[\mathcal{H}[P(y|x^*, \theta)]]}{\text{Expected Data Uncertainty}}
$$

In practice, however, for deep, distributed models with million of parameters, it is difficult to select an appropriate approximate inference scheme to craft a model posterior which induces a distribution over distributions with the desired properties. On the other hand, a Prior Network [22] $p(\pi|x^*, \tilde{\theta})$ directly parametrizes a prior distribution over categorical output distributions (in this work the Dirichlet distribution) and is explicitly trained to yield the desired behaviour of the distribution over distributions.

$$p(\pi|x^*, \tilde{\theta}) = \text{Dir}(\pi|\alpha), \alpha = f(x^*, \tilde{\theta})$$

The desired behaviors of the Prior Network can be visualized on a simplex (fig 1), where figure 1a describes confident behavior, figure 1b describes uncertainty due severe class overlap (data uncertainty) and figure 1c describes the behaviour for an out-of-distribution input.

![Figure 1: Desired Behaviors of a Dirichlet distribution over categorical distributions.](image)

A Prior Network is trained to display these behaviors by minimizing the KL-divergence between the model and target in-domain and out-of-domain Dirichlet distributions [22]. The target in-domain distribution is a sharp Dirichlet centered on the corner of the simplex corresponding to the target class (fig 1a). A flat Dirichlet is chosen as the out-of-distribution target distribution $p_{\text{out}}(\pi)$ (fig 1c). To train a Prior Network it is necessary to have out-of-distribution training data. For example, if a model is trained on CIFAR-10 [24], it is possible to use CIFAR-100 as the out-of-distribution dataset, as they don’t have overlapping classes. Given a trained Prior Network it is possible to calculate the Mutual Information using an expression similar to equation 2.

$$
\frac{\mathcal{MI}[y, \pi|x^*, \tilde{\theta}]}{\text{Total Uncertainty}} = \frac{\mathcal{H}[\mathbb{E}_{p(\pi|x^*, \tilde{\theta})}[P(y|\pi)]] - \mathbb{E}_{p(\pi|x^*, \tilde{\theta})}[\mathcal{H}[P(y|\pi)]]}{\text{Expected Data Uncertainty}}
$$

### 3 Detection-Avoiding Adversarial Attacks

If an adversarial attack is to avoid detection using measures of uncertainty then it must change a model’s prediction while leaving the measures of uncertainty unchanged. In the case of a DNN or

$^2$Where $\pi$ is a vector of probabilities: $[\pi_1, \ldots, \pi_K]^T = [P(y = \omega_1), \ldots, P(y = \omega_K)]^T$
Monte-Carlo dropout, one approach to do this is to simply permute the predicted distribution over classes so that the probability of the max class is assigned to the target class \( t \), and the probability of the target class \( t \) is assigned to the max class. The loss function minimized by the adversarial generation process will be the KL divergence between the predicted distribution over class labels \( P(y|x; \hat{\theta}) \) and the target permuted distribution \( P_t(y) \). For prior networks the equivalent approach would be to minimize KL divergence to the target permuted Dirichlet distribution.

\[
\mathcal{L}(P(y|x; \hat{\theta}), t) = D_{KL}(P_t(y)||P(y|x; \hat{\theta}))
\]

\[
\mathcal{L}(P(\pi|x; \hat{\theta}), t) = D_{KL}(P_t(\pi)||P(\pi|x; \hat{\theta}))
\]

\[ (5) \]

4 Results and Discussion

All models are trained on the CIFAR-10 data. Prior Networks are trained in two configurations PN and PN-ADV. PN is trained using CIFAR-10 as in-domain data and CIFAR-100 as ‘on-manifold’ out-of-distribution data; PN-ADV is trained using both CIFAR-100 and FGSM adversarial attacks as out-of-distribution training data. The idea of PN-ADV is to not only constrain the behavior of the predicted distribution over distribution on-manifold but also off-manifold. 'Standard’ BIM and MIM attacks are run for 10 iterations. Detection avoiding attacks are run for up to 100 iterations at a fixed perturbation of 40 to shown the computational complexity of the task.

<table>
<thead>
<tr>
<th>Model</th>
<th>AUPR</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max.P</td>
<td>Ent.</td>
<td>M.I.</td>
</tr>
<tr>
<td>DNN</td>
<td>48.7</td>
<td>47.1</td>
</tr>
<tr>
<td>MCDP</td>
<td>48.4</td>
<td>45.5</td>
</tr>
<tr>
<td>PN</td>
<td>52.7</td>
<td>51.0</td>
</tr>
<tr>
<td>PN-ADV</td>
<td>51.6</td>
<td>50.2</td>
</tr>
</tbody>
</table>

Table 1: Misclassification detection. 10 dropout samples were used with dropout probability of 0.5.

Table 2: CIFAR-10 out-of-domain detection

Figure 2 shows a summary of results for the most aggressive (MIM) adversarial attack. Figures 2a and 2b show that standard whitebox MIM attacks are successful given a high-enough perturbation, but are detectable using all approaches for small perturbations, and by PN-ADV for all perturbations. Figures 2c and 2d show that it is possible to craft successful detection-evading attacks against DNNS.
but it more difficult against MCDP or very hard and computationally expansive for Prior Networks. The experiments show that it is non-trivial to successfully construct whitebox adversarial attacks which yield the target class and not perturb any properties of the distribution over distributions for appropriately secured prior networks. This suggests that using measures of uncertainty derived from distributions over output distributions constrains the space of solutions to the adversarial optimization problem in a way which methods proposed in [25, 26, 27] do not. Furthermore, it is the explicit specification of the behaviour of a distribution over distributions both in-domain and out-of-domain both on- and off-manifold which greatly constrains the space of solutions where the attack both yields the target class and avoids changing properties distributions and distributions over distributions. These are encouraging results, however, further empirical evaluation on different datasets and adversarial attacks is necessary.
References


