Prior Networks for Detection of Adversarial Attacks

Andrey Malinin
Department of Engineering
University of Cambridge
am969@cam.ac.uk

Mark Gales
Department of Engineering
University of Cambridge
mjfg@eng.cam.ac.uk

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 bio-informatics [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*[1] 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 $\mathtt{p}(\boldsymbol{\pi}|\boldsymbol{x}^*;\hat{\boldsymbol{\theta}})$ *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-evading whitebox and blackbox adversarial attacks.

2 Uncertainty Estimation

Bayesian approaches treat model parameters $\boldsymbol{\theta}$ as random variables and place a prior distribution $\mathtt{p}(\boldsymbol{\theta})$ over them to compute the posterior distribution $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$ via Bayes' rule. Uncertainty in the model parameters induces a *distribution over predictive distributions* $\mathtt{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})$ for each observation $\boldsymbol{x}^*$ -

[1] *Distributional uncertainty* arises when the test data is 'out-of-distribution' relative to the training data. Bayesian approaches model *distributional uncertainty* through *model uncertainty* [22].

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

each set of model parameters parameterizes a conditional distribution over class labels. The expected predictive distribution $\mathtt{P}(y|\boldsymbol{x}^*, \mathcal{D})$ is obtained by marginalizing out the parameters. Unfortunately, both the marginalization and calculation of the model posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$ are intractable for neural networks. Typically the model posterior distribution is approximated using either an implicit or explicit variational approximation $\mathtt{q}(\boldsymbol{\theta})$ and the integral is approximated via sampling (eq. 1), using approaches such as Monte-Carlo dropout [23].

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

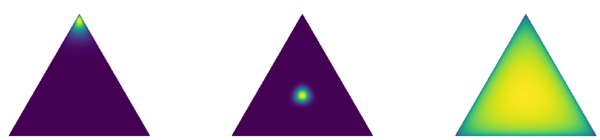
By selecting an appropriate approximate inference scheme and model prior $\mathtt{p}(\boldsymbol{\theta})$ Bayesian approaches aim to craft a model posterior $\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})$ such that the ensemble of distributions $\{\mathtt{P}(\omega_c|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})\}_{i=1}^{M}$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

$$\underbrace{\mathcal{MI}[y, \boldsymbol{\theta}|\boldsymbol{x}^*, \mathcal{D}]}_{Model\ Uncertainty} = \underbrace{\mathcal{H}[\mathtt{E}_{\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathtt{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]]}_{Total\ Uncertainty} - \underbrace{\mathtt{E}_{\mathtt{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathcal{H}[\mathtt{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]]}_{Expected\ Data\ Uncertainty} \tag{2}$$

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] $\mathtt{p}(\boldsymbol{\pi}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})$ [2] 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.

$$\mathtt{p}(\boldsymbol{\pi}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) = \mathtt{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}),\ \boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) \tag{3}$$

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



(a) Low uncertainty (b) Data uncertainty (c) Out-of-distribution

Figure 1: Desired Behaviors of a Dirichlet distribution over categorical distributions.

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. 1:a). A flat Dirichlet is chosen as the out-of-distribution target distribution $\mathtt{p}_{\mathtt{out}}(\boldsymbol{\pi})$ (fig. 1:c). 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:

$$\underbrace{\mathcal{MI}[y, \boldsymbol{\pi}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}]}_{Distributional\ Uncertainty} = \underbrace{\mathcal{H}[\mathtt{E}_{\mathtt{p}(\boldsymbol{\pi}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}[\mathtt{P}(y|\boldsymbol{\pi})]]}_{Total\ Uncertainty} - \underbrace{\mathtt{E}_{\mathtt{p}(\boldsymbol{\pi}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}[\mathcal{H}[\mathtt{P}(y|\boldsymbol{\pi})]]}_{Expected\ Data\ Uncertainty} \tag{4}$$

3 Detection-Avoiding Adversarial Attacks

If an adversarial attack is to avoiding 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 $\boldsymbol{\pi}$ is a vector of probabilities: $\begin{bmatrix}\pi_1, & \cdots, & \pi_K\end{bmatrix}^T = \begin{bmatrix}\mathtt{P}(y = \omega_1), & \cdots, & \mathtt{P}(y = \omega_K)\end{bmatrix}^T$

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 $\mathtt{P}(y|\tilde{\boldsymbol{x}};\hat{\boldsymbol{\theta}})$ and the target permuted distribution $\mathtt{P_t}(y)$. For prior networks the equivalent approach would be to minimize KL divergence to the target permuted Dirichlet distribution.

$$\mathcal{L}\big(\mathtt{P}(y|\tilde{\boldsymbol{x}};\hat{\boldsymbol{\theta}}),t\big) = D_{KL}(\mathtt{P_t}(y)||\mathtt{P}(y|\tilde{\boldsymbol{x}};\hat{\boldsymbol{\theta}})),\ \mathcal{L}\big(\mathtt{P}(\boldsymbol{\pi}|\tilde{\boldsymbol{x}};\hat{\boldsymbol{\theta}}),t\big) = D_{KL}(\mathtt{p_t}(\boldsymbol{\pi})||\mathtt{p}(\boldsymbol{\pi}|\tilde{\boldsymbol{x}};\hat{\boldsymbol{\theta}})) \quad (5)$$

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.

Model	AUPR Max.P	AUPR Ent.	AUPR M.I.	% Error
DNN	48.7	47.1	-	**8.0**
MCDP	48.4	45.5	37.6	**8.0**
PN	**52.7**	51.0	51.0	8.5
PN-ADV	51.6	50.2	50.2	8.2

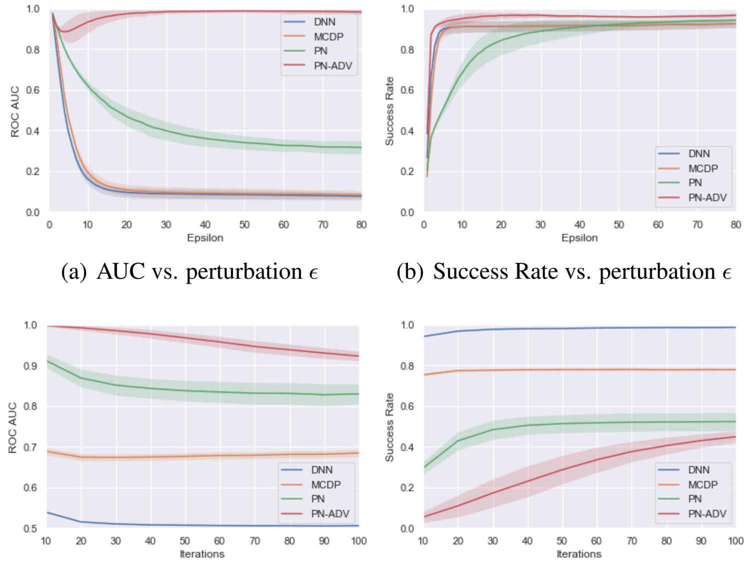
Table 1: Misclassification detection. 10 dropout samples were used with dropout probability of 0.5 .

Table 1 Shows the misclassification detection performance and classification error rate of the four model considered in this work. Misclassification detection performance is assessed using area under a precision-recall curve with misclassifications as the positive class. Results show that using adversarial examples as additional *off-manifold* training data for a Prior Network does not degrade classification performance. Furthermore, it does not significantly affect misclassification detection performance. Table 2 shows the out-of-distribution detection performance of the aforementioned models, with the SVHN, LSUN and TinyImageNet datasets used as out-of-distribution data. Performance is assessed using area under an ROC curve (ROC AUC). The results show that PN-ADV does not suffer from significant drops in OOD detection performance, and in fact outperforms a standard Prior Network on LSUN and TinyImageNet.

OOD Data	Model	ROC AUC Max.P	ROC AUC Ent.	ROC AUC M.I.
SVHN	DNN	90.1	90.8	-
	MCDP	89.6	90.6	83.7
	PN	98.1	**98.2**	**98.2**
	PN+ADV	98.0	98.1	98.1
LSUN	DNN	89.8	91.4	-
	MCDP	89.1	90.9	89.3
	PN	94.4	94.4	94.4
	PN+ADV	94.8	**94.9**	**94.9**
TIM	DNN	87.5	88.7	-
	MCDP	87.6	89.2	86.9
	PN	94.3	94.3	94.3
	PN+ADV	94.6	**94.6**	**94.6**

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 is possible to craft successful detection-evading attacks against DNNS,



(a) AUC vs. perturbation ϵ (b) Success Rate vs. perturbation ϵ

(c) AUC vs. Iter (d) Success Rate vs. Iter

Figure 2: Plots of ROC AUC and Success Rate vs. perturbation (A and B) or iterations (C and D) against whitebox standard (A and B) and detection evading (C and D) MIM attacks.

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

[1] Ross Girshick, “Fast R-CNN,” in *Proc. 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[2] Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[3] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee, “Learning to Generate Long-term Future via Hierarchical Prediction,” in *Proc. International Conference on Machine Learning (ICML)*, 2017.

[4] Tomas Mikolov et al., “Linguistic Regularities in Continuous Space Word Representations,” in *Proc. NAACL-HLT*, 2013.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space,” 2013, arXiv:1301.3781.

[6] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, “Recurrent Neural Network Based Language Model,” in *Proc. INTERSPEECH*, 2010.

[7] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, “Deep neural networks for acoustic modeling in speech recognition,” *Signal Processing Magazine*, 2012.

[8] Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng, “Deep speech: Scaling up end-to-end speech recognition,” 2014, arXiv:1412.5567.

[9] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad, “Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission,” in *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2015, KDD ’15, pp. 1721–1730, ACM.

[10] Babak Alipanahi, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey, “Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning,” *Nature Biotechnology*, vol. 33, no. 8, pp. 831–838, July 2015.

[11] Christian Szegedy, Alexander Toshev, and Dumitru Erhan, “Deep neural networks for object detection,” in *Advances in Neural Information Processing Systems*, 2013.

[12] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy, “Explaining and harnessing adversarial examples,” in *International Conference on Learning Representations*, 2015.

[13] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio, “Adversarial examples in the physical world,” 2016, vol. abs/1607.02533.

[14] Nicholas Carlini and David A. Wagner, “Adversarial examples are not easily detected: Bypassing ten detection methods,” *CoRR*, 2017.

[15] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, 2016, pp. 582–597.

[16] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li, “Boosting adversarial attacks with momentum,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] Nicholas Carlini and David A. Wagner, “Towards evaluating the robustness of neural networks,” *CoRR*, 2016.

[18] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami, “Practical black-box attacks against deep learning systems using adversarial examples,” *CoRR*, vol. abs/1602.02697, 2016.

[19] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami, “The limitations of deep learning in adversarial settings,” in *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, 2016, pp. 372–387.

[20] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song, "Delving into transferable adversarial examples and black-box attacks," *CoRR*, vol. abs/1611.02770, 2016.

[21] L. Smith and Y. Gal, "Understanding Measures of Uncertainty for Adversarial Example Detection," in *UAI*, 2018.

[22] Andrey Malinin and Mark JF Gales, "Predictive uncertainty estimation via prior networks," 2018.

[23] Yarin Gal and Zoubin Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. 33rd International Conference on Machine Learning (ICML-16)*, 2016.

[24] Alex Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[25] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff, "On detecting adversarial perturbations," in *Proceedings of 5th International Conference on Learning Representations (ICLR)*, 2017.

[26] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku, "Adversarial and clean data are not twins," *CoRR*, vol. abs/1704.04960, 2017.

[27] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick D. McDaniel, "On the (statistical) detection of adversarial examples," *CoRR*, vol. abs/1702.06280, 2017.