Probabilistic Robustness Quantification of Neural Networks

Gopi Kishan [1] **and Apurva Narayan** [2]
[1] Indian Institute of Technology, Roorkee, India
[2] The University of British Columbia, Canada
gkishan@cs.iitr.ac.in, apurva.narayan@ubc.ca

Abstract

Safety properties of neural networks are critical to their application in safety-critical domains. Quantification of their robustness against uncertainties is an upcoming area of research. In this work, we propose an approach for providing probabilistic guarantees on the performance of a trained neural network.

We present two novel metrics for probabilistic verification on training data distribution and test dataset. Given a trained neural network, we quantify the probability of the model to make errors on a random sample drawn from the training data distribution. Second, from the output logits of a sample test point, we measure its p-value on the learned logit distribution to quantify the confidence of the model at this test point. We compare our results with softmax based metric using the black-box adversarial attacks on a simple CNN architecture trained for MNIST digit classification.

Introduction

Neural networks are increasingly becoming a crucial computational component of modern software. With their widespread adoption, it has become essential that we ensure (or at least gain confidence in) the correctness of neural networks, as we do with traditional programs. However, providing formal specifications of correctness is an even more challenging task for neural networks than for traditional programs, as neural networks are explicitly designed to learn patterns in training data that are not readily apparent to humans (Fazlyab, Morari, and Pappas 2019; Henriksen and Lomuscio 2019). In this paper, we step in the direction of probabilistic verification of learned models.

Background

We are going to construct our formulation of probabilistic verification with the notion of logits being random variables. Random variable is a function that assigns values to each of an experiment's outcomes. This function will be our Neural Network.

Studying the distribution of logits of various classes on MNIST dataset trained on a simple convolutional neural network, we found out these distributions to be Gaussian for both correct and false logit classes. Refer to Figure 1. This experiment motivates us to assume that a trained neural network learns Gaussian distributions in logit space that are independent which is justifiable in the paradigm of i.i.d training data. This assumption forms the basis of our method presented in the next section.

Method

We propose two metrics for probabilistic error quantification of the model hypothesis on training dataset and input test sample.

Through these metrics, first, we quantify how good is the hypothesis learnt on the dataset and then increase the reliability for out-of-distribution sample or adversarial example for the hypothesis to predict wrong class with high confidence.

Dataset Centric We can take any dataset and a trained model on it. The idea is when we draw a random sample from the dataset distribution, what is the probability that the hypothesis (trained model) will misclassify it. It is different from accuracy since accuracy is a discrete sample matching, but it is based on the overlap of the PDF of logit distribution. Formally, if Z is the logit vector, X be the correct logit element and Y be any other logit element, we want $P(X <= Y + \delta)$ where a δ can be included to ensure margin like maximum margin classifiers.

Note by our previous assumption, both random variables X and Y follow Gaussian distribution. So, $P(X - Y <= \delta) = \Phi[(\delta - \mu)/\sigma]$ is Gaussian CDF with mean $\mu = \mu_x - \mu_y$ and variance $\sigma^2 = \sigma_x^2 + \sigma_y^2$ where $X \sim N(\mu_x, \sigma_x)$ and $Y \sim N(\mu_y, \sigma_y)$.

Now for a k class logit vector Z, we have k-1 such comparisons, say $A_1, A_2, ...A_{k-1}$. We need to calculate $P(\cup_{i=1}^{k-1} A_i)$ which can be done using inclusion exclusion principle. For all the experiments in results section, $\delta = 0$.

Sample Centric The previous proposed method measures trust over entire dataset distribution which the neural network has learnt. However, sample centric approach can predict class and quantify model's confidence for its prediction on encountering a new sample point while testing. Currently in the literature, for classification, argmax over logits is performed to predict correct class or softmax to get the confi-

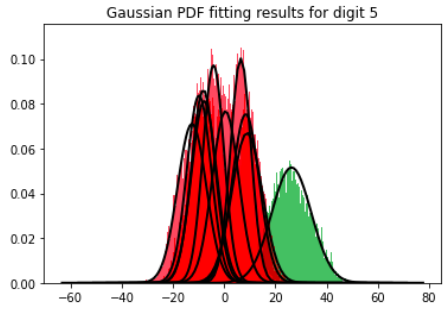

Figure 1: Shows distributions of logits as Gaussian PDF. Red PDF are for false logits and Green is for true logit. Overlap of red PDF with green shows error regions of hypothesis.

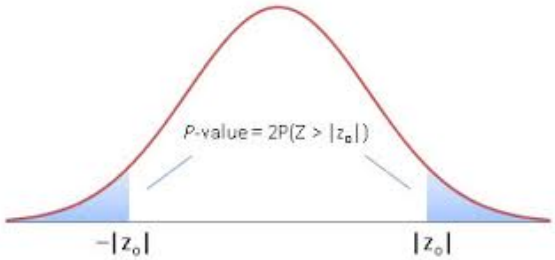

Figure 2: Shows learned PDF of logit Z and calculation of p-value for test logit output z_0.

dence of the prediction. We propose to use the 2-sided p-value measure for statistical significance of the input sample utilising the information of the learned distribution of logit vector while training.

Formally, when a data point X arrives, feed forward through the network to predict the logit vector z. We then compute p-value statistic of this vector z with our saved distribution parameter vector of correct μ_i and σ_i for all class i and return a vector whose elements are the p-value measure of each k-class distribution. Figure 2 shows the p-value calculation method used in experiments.

Results

We performed experiments on the MNIST dataset, where we trained a simple CNN architecture (380k parameters) to 99% test accuracy.

Using the first method, we evaluated this model to have 25.63% error (upper bound) at $\delta = 0$ data distribution. The logits PDF distribution is shown in Figure 2, and it validates our assumption of the independent Gaussian distribution of logits.

Then using the second method, we evaluated test samples and recovered 94.56% correct targets by taking argmax over p-value vector. Our method becomes very useful in case of a safety-critical system, where a false prediction with high confidence is undesirable. So, we compared the softmax prediction and p-value prediction on two adversarial attacks methods, *viz.* Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015) attack and Projected

Table 1: Fast Gradient Sign black-box untargeted attack

softmax err	4524	Total samples	10k
p-value err	6221	Total samples	10k
Metric	conf > 0.5	conf > 0.9	conf > 0.99
Softmax err	4402	2860	1682
p-value err	1640	303	27

Table 2: Projected Gradient Descent black-box untargeted attack

softmax err	5418	Total samples	10k
p-value err	5652	Total samples	10k
Metric	conf > 0.5	conf > 0.9	conf > 0.99
Softmax err	5402	4127	2566
p-value err	1170	208	24

Gradient Descent (PGD) (Madry et al. 2018) attack used in a black-box & untargeted fashion. We evaluated on 10k test-samples of MNIST and summarised the results of the two algorithm in Table 1 & 2 respectively.

In PGD attack, total errors are similar, but p-value gives significantly fewer sample errors with high confidence w.r.t softmax. Thus our metric measures can more reliably be trusted.

Conclusion

We proposed a two-level of quantification method of robustness. First, where we test the hypothesis which the model has learnt to represent data and second where once a hypothesis is given, we measure out of learned logit distribution.

The design of the probabilistic verification on both data centric and sample centric approach brings reliability to our deep learning systems. When a model encounters an out of distribution data or some adversarial example, it provides information as low confidence and could suggest human in the loop. Thus, increasing the reliability of DL systems.

We are actively working on making a probabilistic driven training approach based on our first method, where we will learn the desired confident representation automatically.

References

Fazlyab, M.; Morari, M.; and Pappas, G. J. 2019. Probabilistic Verification and Reachability Analysis of Neural Networks via Semidefinite Programming. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2726–2731.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. *CoRR* abs/1412.6572.

Henriksen, P.; and Lomuscio, A. 2019. *Efficient Neural Network Verification via Adaptive Refinement and Adversarial Search*. Ph.D. thesis, Imperial College London.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. *ArXiv* abs/1706.06083.