Towards Reliable Learning for High Stakes Applications

Jinyang Gao,[1] **Junjie Yao,**[2] **Yingxia Shao**[3]*
[1]Alibaba Group, [2]ECNU, [3]BUPT,
jinyang.gjy@alibaba-inc.com, junjie.yao@sei.ecnu.edu.cn, shaoyx@bupt.edu.cn.

Abstract

In this paper, we focus on delivering reliable learning results for *high stakes applications* such as self-driving, financial investment and clinical diagnosis, where the accuracy of predictions is considered as a more crucial requirement than giving predictions for all query samples. We adopt the learning with reject option framework where the learning model only predict those samples which they convince to give the correct answer. However, for most prevailing deep learning predictors, the confidence estimated by the model themselves are far from reflecting the real generalization performance. To model the reliability of prediction concisely, we propose an exploratory solution called GALVE (Generative Adversarial Learning with Variance Expansion) which adopts generative adversarial learning to implicitly measure the region where the model achieve good generalization performance. By applying GALVE to measure the reliability of predictions, we achieved an error rate less than half of which straightforwardly measured by confidence in CIFAR10 and SVHN computer vision tasks.

Introduction

Nowadays, state-of-the-art AI solutions have achieved many successes across a wide range of real world applications. For most of the complex data analytics and decision making problems such as financial investment (Ding et al. 2015), medical treatment (Zheng and Gao 2017; Cai and Gao 2018) and self-driving system (Chen et al. 2015), learning algorithms, while in progressive development, are widely believed to be able to surpass human performance in the near future. Such emerging "high stakes" applications of AI raise high criteria on the reliability of deployed solutions. However, most of these applications rely on prevailing deep neural network models. While these models can provide high prediction accuracy in general cases, they may be vulnerable to unexpected egregious errors (Nguyen, Yosinski, and Clune 2015; Moosavi et al. 2016; Fawzi et al. 2016), particularly when being applied to data points that are not well-represented in the training set (e.g., Figure 1). In some above cases, the deep learning models act like doing random guesses on regions lack of training points, and predict with high confidence without even knowing that themselves are actually doing random guesses. For high stakes applications, every decision matters and such irresponsible actions are definitely prohibited. Unfortunately, most deep learning models act like a black-box without much explanation, and are hard to understand even for domain experts (Zeiler and Fergus 2014) It is not practical to prevent such failures by manually examining the inside logic. Consequently, developing a learning solution with reliable behaviour has attracted a great deal of interest.

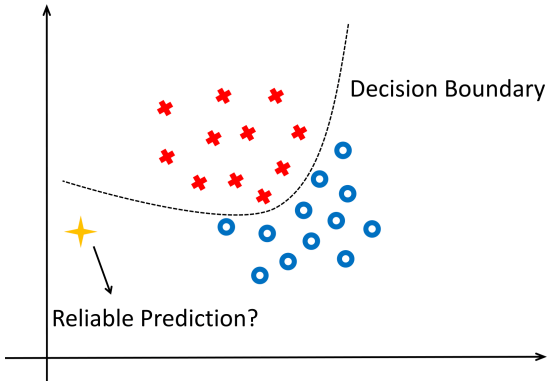

Figure 1: Prediction with high confidence could be non-reliable, especially when the query sample is far from the training distribution.

Learning with Reject Option (LRO)

To achieve reliable learning results, one possible solution specifically studied in this paper is to construct a learning model with reject option (Bartlett and Wegkamp 2008). Instead of optimizing the overall accuracy on the test samples, it aims at selecting the largest distribution (i.e. subset) from the test set, in which the averaged accuracy should be higher than a given threshold p. Learning with reject option can be applied to a wide range of *high stakes applications*, which have the following two properties: 1) be backed up by human or other default methods so that the applied model is not the only problem solving solution; 2) must perform better than a given requirement (e.g. better than human for financial investment etc.). Such reliable learning solutions are useful

*corresponding author

for many high stakes applications. For example, a model for CT image classification with 90% accuracy cannot be applied onto any clinical system, where the typical minimum requirement is 95%. As an alternative, by applying a model that provides 99% accuracy inside a *reliable region* which covers half of the patient images, the workload of radiologists could be effectively reduced by half.

A straight-forward approach for LRO is to only predict samples with high confidence (the probability of predicted class) or low entropy outputted by the model. However, as shown by previous works (Nguyen, Yosinski, and Clune 2015; Moosavi et al. 2016; Gu and Rigazio 2014; Goodfellow, Shlens, and Szegedy 2014), neural network models can be easily fooled and are vulnerable to adversarial samples – unexpected errors happened on nearly duplicated instance of training samples. Using the confidence or entropy by predictor to measure reliability of the prediction for a given query is far from satisfaction. Figure 1 gives an example where the query sample is far from training distribution hence its prediction is not reliable, while its distance to the decision boundary is very wide so that the confidence output by model could be very high.

We have done a quantitative illustrating experiments on ResNet (He et al. 2016; Shen and Gao 2018) image classification model with CIFAR10 dataset, where the average test error is 7.2%. Even for those samples where the confidence is higher than 99%, more than 4% of them are still misclassified. To achieve LRO, the confidence is not an ideal indicator as it can only avoid half of the failures by removing 30% most uncertain samples. There exists a huge performance gap between the error estimated by the predictor and its real performance.

Reliable Region

To estimate the real error probability for given query samples, we develop a concept called *reliable region*, in where the prediction model achieves good generalization performance. By measuring a reliable region for the predictor, unreliable predictions can be avoided by detecting that the query sample does not reside in the region. The main technical challenge for developing an effective LRO solution is to detect the reliable region for the applied model. There is no general guarantee for the performance of instances which do not belong to the training set. Models which overfit can only generalize well to a small region near each training sample, while some other robust models could generalize to the whole space at the same level of performance in the training set. In essence, the reliable region could be related to the latent training distribution generalized by the applied model where its loss is minimized.

Generative Adversarial Approach

Inspired by the aforementioned fact, we devise a novel approach to model the latent training distribution via generative adversarial networks (GANs) (Goodfellow et al. 2014), which is developed to generate new samples from the latent data distribution of a dataset. GANs consist of two models trained simultaneously: a generative model $\mathcal{G}$ generating samples from the latent data distribution, and a discriminative model (i.e. adversarial network) $\mathcal{D}$ detecting whether a sample belongs to the original dataset. GANs can be used as a generative model of the latent data distribution for given datasets under neural network representation. Its discriminative model $\mathcal{D}$ is an ideal classifier to detect whether the input is from the latent data distribution learned by GANs. Our key insight here is that *by assigning the discriminative model $\mathcal{D}$ the same computation architecture used in the applied prediction model, they should have almost the same generalization ability so that the underlying latent data distribution estimated by them should also be pretty similar*. Therefore, by training GANs using the same source of training data paired with an appropriate generative network, the discriminative model $\mathcal{D}$ could provide useful information estimating the reliable region.

We realize that most existing architecture of GANs are focusing on generating high quality samples rather than training high quality discriminator. *Generative Adversarial Learning with Variance Expansion* (GALVE) is proposed to address the above challenge, where the generator is obtained via the prevailing variant WGAN-GP (Gulrajani et al. 2017), and the discriminator is further fine-tuned using samples generated with a higher variance of initialization. Thus, the discriminator can benefit from negative samples with a wider coverage and more diversity.

We empirically evaluate GALVE on CIFAR10 and SVHN dataset. The results demonstrate that the errors happened in samples with highest reliability measured by GALVE is only 40-45% of the errors happened in samples with highest reliability measured by confidence of predictor.

The contribution of this paper are three fold:

1. We showed that for high stakes applications, reliable results could be obtained via learning with reject option framework.
2. We analyzed that there is a strong connection between the estimation of generalization performance and the out put by the discriminator of generative adversarial training.
3. We found that simply applying generative adversarial training may not work well since most GANs focusing on the performance of the generator rather than discriminator, and proposed a general method GALVE to address this issue.

Methodology

Most learning models can be defined as a *Risk Minimization* problem. We use $\mathbf{x} = \langle data, label \rangle$ to denote the data instance, $\mathcal{F}_{\mathbf{w}}$ to denote a model hypothesis with parameter $\mathbf{w}$, and $\mathcal{L}$ to denote the loss function where the model aim to minimize. Let p_{real} be the real distribution of data from where all the samples are generated, the risk associated with model $\mathcal{F}_{\mathbf{w}}$ is defined as the expectation of the loss for the potential data distribution:

$$\mathbf{E}_{\mathbf{x} \sim p_{real}} \mathcal{L}(\mathcal{F}_{\mathbf{w}}(\mathbf{x})) \tag{1}$$

The goal is to find the model $\mathcal{F}_{\mathbf{w}}$ that minimizes the risk.

In general, the risk cannot be directly minimized since the exact latent data distribution p_{real} is unknown. Instead, the

common way is to use the distribution of training set $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ to approximate p_{real}. Therefore, the *Empirical Risk* is used as the optimization target:

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{X}}\mathcal{L}(\mathcal{F}_{\mathbf{w}}(\mathbf{x})) \triangleq \sum_i \frac{1}{n}\mathcal{L}(\mathcal{F}_{\mathbf{w}}(\mathbf{x_i})) \tag{2}$$

However, the empirical risk may not reflect real risk. The difference could be large when the model hypothesis $\mathcal{F}_{\mathbf{w}}$ is complex while the training data is not sufficient. To prevent such over-fitting problem, the minimization target $\mathcal{L}oss(\mathbf{w}, \mathcal{X})$ is often defined as the empirical risk plus a regularization term $\rho(\mathbf{w})$ to penalize the complexity of model $\mathcal{F}_{\mathbf{w}}$:

$$\mathcal{L}oss(\mathbf{w}, \mathcal{X}) \triangleq \mathbf{E}_{\mathbf{x}\sim\mathcal{X}}\mathcal{L}(\mathcal{F}_{\mathbf{w}}(\mathbf{x})) + \rho(\mathbf{w}) \tag{3}$$

By applying such deep learning model to a test data distribution p_{test}, the expectation or error:

$$\mathbf{E}_{\mathbf{x}\sim p_{test}}\mathcal{L}(\mathcal{F}_{\mathbf{w}}(\mathbf{x})) \tag{4}$$

is often theoretically unbounded. Even when $p_{test} = p_{real}$, the bound obtained by VC-dimension theory is usually too weak and meaningless due to the huge parameter size in prevailing deep learning models. As criticized by (Nguyen, Yosinski, and Clune 2015), deep learning models are easily fooled so that they predict most test samples wrongly with high confidence. (Gu and Rigazio 2014) has also shown that deep learning models are vulnerable to adversarial attacks: misclassify examples that are only slightly different from training examples. All these evidences suggest that the performance of deep learning model is far from reliable – even for the case $\mathbf{x} \sim p_{real}$, unexpected failure still happens.

Though there is almost no bound for their generalization error, deep learning models have achieved the state-of-the-art performance in lots of common tasks in computer vision, natural language processing and time series prediction area. Recently, Generative Adversarial Networks (GANs) show that deep learning architectures can actually learn a latent data distribution and generate new samples with good quality (e.g. meaningful images). An theoretical analysis (Arora et al. 2017) supported by experiments (Arora and Zhang 2017) on GANs show that although the discriminative model of GANs is designed to validate if a given sample is from the training set, it fails to detect near duplicate samples even with a model size that can remember all the training instances. All the results could lead to another widely accepted conclusion: Deep learning models have much stronger generalization ability than what is analyzed by VC-dimension or criticized by adversarial examples. Their reliable region could cover most of the spaces in p_{real} or p_{test}, since they achieve general good performance on test set. How to detect whether query samples reside in reliable region or where unexpected errors happen and adversarial examples reside in, remains to be a challenging problem. We aim to detect the reliable region and use such information to build classifier with reject option, where the model only predict samples within reliable region.

Latent Data Distribution Modeled by GANs

Generative Adversarial Networks (GANs) are neural networks consist of two networks competing with each other. The two networks namely generator $\mathcal{G}$ – to generate data set and discriminator $\mathcal{D}$ – to validate the data set. The goal is to generate data points that are similar to the data points in the training set.

GANs are trained via the following min-max game. In each step, the generator $\mathcal{G}$ produces an example $\mathcal{G}(\mathbf{z})$ from random noise $\mathbf{z} \sim p(\mathbf{z})$ that has the potential to fool the discriminator $\mathcal{D}$. The discriminator $\mathcal{D}$ is then presented real data examples $\mathbf{x} \sim \mathcal{X}$, together with the examples $\mathcal{G}(\mathbf{z})$ produced by the generator, and its task is to distinguish those artificially generated samples $\mathcal{G}(\mathbf{z})$. $\mathcal{D}(\mathbf{x})$ predicts the probability that $\mathbf{x}$ came from the data $\mathcal{X}$ rather than $\mathcal{G}(\mathbf{z})$. Afterwards, the discriminator is rewarded for correct classifications and the generator is rewarded when $\mathcal{G}(\mathbf{z})$ is misclassified as real samples by the discriminator. The GANs objective is thus defined as following:

$$\min_{\mathcal{D}}\max_{\mathcal{G}} \mathbf{E}_{\mathbf{x}\sim\mathcal{X}}\log\mathcal{D}(\mathbf{x}) + \mathbf{E}_{\mathbf{z}\sim p(\mathbf{z})}\log(1-\mathcal{D}(\mathcal{G}(\mathbf{z}))) \tag{5}$$

i.e. $\mathcal{D}$ minimize:

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{X}}\log\mathcal{D}(\mathbf{x}) + \mathbf{E}_{\mathbf{z}\sim p(\mathbf{z})}\log(1-\mathcal{D}(\mathcal{G}(\mathbf{z}))) \tag{6}$$

while $\mathcal{G}$ maximize:

$$\mathbf{E}_{\mathbf{z}\sim p(\mathbf{z})}\log(1-\mathcal{D}(\mathcal{G}(\mathbf{z}))) \tag{7}$$

As shown by (Goodfellow et al. 2014), GANs estimate the latent data distribution p_{data} where the training samples $\mathcal{X}$ are generated from. Using $p_{\mathcal{G}}$ to denote the distribution of $\mathcal{G}(\mathbf{z})$, the discriminator $\mathcal{D}$ will converge at:

$$\mathcal{D}(\mathbf{x}) = \frac{\alpha p_{data}(\mathbf{x})}{\alpha p_{data}(\mathbf{x}) + p_{\mathcal{G}}(\mathbf{x})} \tag{8}$$

where $\alpha : 1$ is the ratio between real samples and generated samples. And $\mathcal{G}$ will converge at $p_{\mathcal{G}} = p_{data}$ when $\mathcal{D}(\mathbf{x})$ is an oracle classifier with unlimited capacity (Arora et al. 2017).

Most researches of GANs target at learning to generate new samples based on a given dataset by applying the generator $\mathcal{G}$. In this paper, we revisit the potential of leveraging the discriminator $\mathcal{D}(\mathbf{x})$, where the latent data distribution p_{data} is implicitly modeled. p_{data} could an *ideal factor* to measure the reliable region. This is because when the discriminator model $\mathcal{D}(\mathbf{x})$ share the same neural network architecture with the prediction model $\mathcal{F}_{\mathbf{w}}(\mathbf{x})$, they should have similar generalization ability, i.e. the p_{data} modeled by $\mathcal{D}(\mathbf{x})$ could be similar to the distribution where $\mathcal{L}(\mathcal{F}_{\mathbf{w}}(\mathbf{x}))$ is minimized.

$$\mathcal{L}oss(\mathbf{w}, \mathcal{X}) \approx \mathbf{E}_{\mathbf{x}\sim p_{data}}\mathcal{L}(\mathcal{F}_{\mathbf{w}}(\mathbf{x})) \tag{9}$$

Though there is no theoretical result to guarantee that neural networks with the same architecture trained by the same dataset lead to the same latent data distribution, it is widely accepted that they should be closely related. Evidences have even been found that neural networks with different architectures still share a lot of common adversarial samples (Papernot, McDaniel, and Goodfellow 2016) (i.e. misclassified samples which are very close to the training samples), suggesting that their latent data distributions share similar weaknesses when modeling the real data distribution p_{real}. By defining a background distribution as p_{bg}, and a

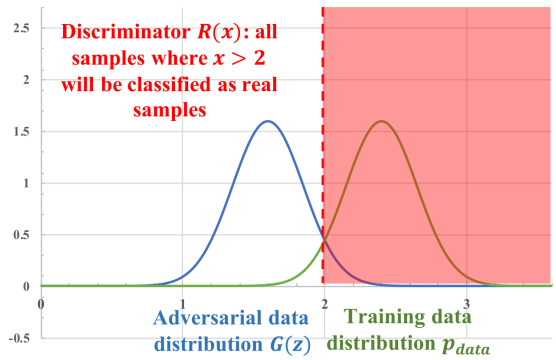

Figure 2: The discriminator of GAN $\mathcal{D}(\mathbf{x})$ can only differentiate $\mathcal{G}(\mathbf{z})$ from p_{data}, but fails to differentiate other outliers (e.g. samples $x > 3$) from p_{data}.

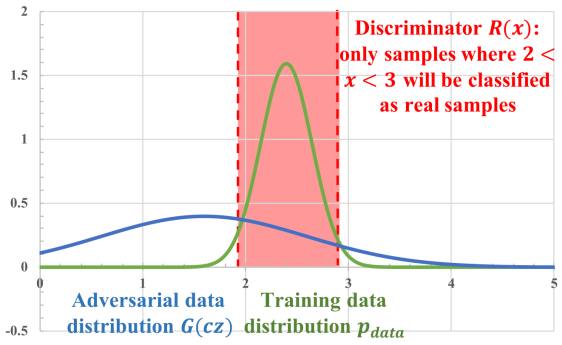

Figure 3: GALVE: $\mathcal{G}(c\mathbf{z})$ has a wider range of coverage so that $\mathcal{R}(\mathbf{x})$ which learnt to differentiate $\mathcal{G}(c\mathbf{z})$ from p_{data} could be a practical classifier implicitly modeling p_{data}.

sample ratio between p_{data} and p_{bg} as $\alpha' : 1$, we can model the samples in p_{test} as either generated from p_{data} or p_{bg}. Given a sample $\mathbf{x}$, its probability to be generated from p_{data} can be modeled as $\mathcal{R}(\mathbf{x})$ where:

$$\mathcal{R}(\mathbf{x}) = \frac{\alpha' p_{data}(\mathbf{x})}{\alpha' p_{data}(\mathbf{x}) + p_{bg}(\mathbf{x})} \tag{10}$$

Naturally, samples generated from p_{data} are in the reliable region. Therefore, $\mathcal{R}(\mathbf{x})$ could be an effective indicator on whether the prediction $\mathcal{F}_{\mathbf{w}}(\mathbf{x})$ on $\mathbf{x}$ is reliable or not.

From Equation 8 and Equation 10, we can observe that $\mathcal{R}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x})$ are closely related. A natural question raises: it is possible to use GANs to build a solution where $\mathcal{D}(\mathbf{x})$ can be used to approximate $\mathcal{R}(\mathbf{x})$?

Why directly Applying GAN does not Work?

The major insight of this paper is that the discriminators of GANs actually implicitly model the latent data distribution (i.e. Eq 8). Thus we asked if the discriminator can be directly used to measure the reliability score (could be viewed as outlier detection problem for query samples). A straight-forward solution is to use $\mathcal{D}(\mathbf{x})$ from any conventional GANs model as the approximation of $\mathcal{R}(\mathbf{x})$. However, based on the experiments, we found that WGAN-GP or other GANs cannot achieve our goal (see experiments in Sec). The distribution $p_{\mathcal{G}}$ could be far from the distribution p_{bg}, which we aim to differentiate with p_{data}. Ideally, a perfect discriminator could detect every difference between samples in p_{data} and samples elsewhere. However, such discriminator is unlikely to be obtained via the training where only $\mathcal{G}(\mathbf{z})$ are used as negative samples. Most variants of GANs aim at generating high quality new samples $\mathcal{G}(\mathbf{z})$, instead of training a perfect discriminator model $\mathcal{D}(\mathbf{x})$. Therefore, $\mathcal{G}(\mathbf{z})$ often collapses to a simpler distribution than p_{data} so that high quality samples can be stably generated.

In essence, the discriminator is not a generalized detector of weird things. It is trying to tell whether a sample came from the real data or one specific distribution: the generator. Figure 2 shows an example: the discriminator learnt by GAN could be far from a general classifier telling whether a given sample is from real data distribution or not. Typical GANs focus on generating high quality samples. Based on our problem, we obviously desire to differentiate the real data distribution from a high diversity distribution rather than a high quality distribution.

GALVE

In our context, p_{bg} is expected to be a more general distribution where the prediction model $\mathcal{F}_{\mathbf{w}}(\mathbf{x})$ cannot generalize to and unexpected failures happen at. We focus on the performance of discriminator model $\mathcal{D}(\mathbf{x})$ rather than the performance of generator $\mathcal{G}(\mathbf{z})$, i.e. $\mathcal{G}(\mathbf{z})$ is expected to generate samples with high diversity to cover as much unexpected cases as possible. However, when $\mathcal{G}(\mathbf{z})$ cannot produce high quality samples, $\mathcal{D}(\mathbf{x})$ is not likely to be well trained since most of samples from $\mathcal{G}(\mathbf{z})$ are too far from p_{data} such that they can be easily detected. We have to trade off between a fine-tuned classifier on inaccurate target and an unbiased classifier with over-easy training samples.

We propose *GALVE*, i.e. Generative Adversarial Learning with Variance Expansion to resolve the aforementioned challenge. The central idea of GALVE is to learn the discriminator $\mathcal{R}(\mathbf{x})$ to differentiate p_{data} not from adversarial distribution $\mathcal{G}(\mathbf{z})$, but a generated data distribution with a higher variance. Figure 3 is an intuitive example explaining why GALVE works: by increasing the variance of $\mathcal{G}(\mathbf{z})$ by a factor of c, the discriminator $\mathcal{R}(\mathbf{x})$ is much general and could be useful to differentiate p_{data} from all other distributions. Suppose that p_{data} is a simple Gaussian distribution $\mathcal{N}(0, \sigma^2)$, by generating $\mathcal{G}(\mathbf{z}) \sim \mathcal{N}(0, c^2\sigma^2)$ where $c > 1$, we have:

$$\mathcal{R}(x) = \frac{\alpha \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{(\sigma)^2}}}{\alpha \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{(\sigma)^2}} + \frac{1}{c\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{(c\sigma)^2}}} \tag{11}$$

$$= \frac{\alpha c}{\alpha c + \mathrm{e}^{\frac{(c^2-1)x^2}{(c\sigma)^2}}} \tag{12}$$

Therefore, $\mathcal{R}$ learns a smooth decision boundary based on the norm of x/σ, which is a desired discriminator for

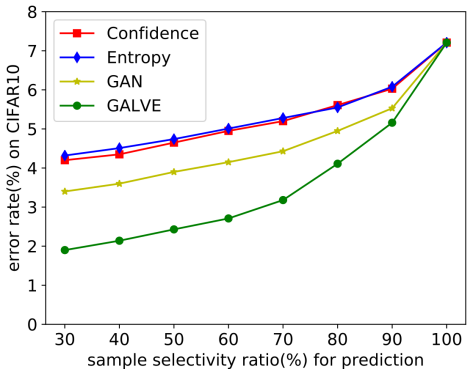

Figure 4: Error Rate(%) on CIFAR10

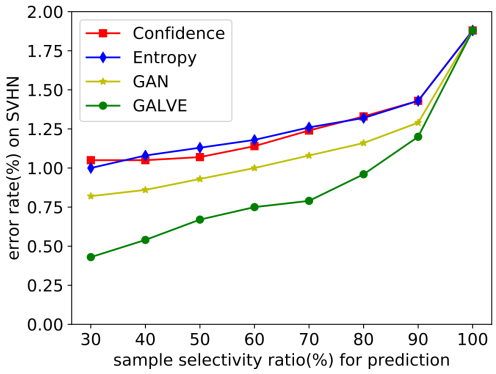

Figure 5: Error Rate(%) on SVHN

$\mathcal{N}(0, \sigma^2)$.

In GALVE, the generator $\mathcal{G}(\mathbf{z})$ and the original discriminator $\mathcal{D}(\mathbf{x})$ are trained using the current prevailing training method WGAN-GP (Gulrajani et al. 2017). GALVE can also be trained using any GANs variants where the $\mathcal{G}(\mathbf{z})$ is trained to approximate p_{data}. In a high dimensional space, $\mathcal{G}(\mathbf{z})$ and p_{data} are very complex distribution than the simple Gaussian in the above example. However, the random seed of samples, i.e. $\mathbf{z}$, are usually initialized using a Gaussian distribution. Therefore, we propose to train $\mathcal{R}(\mathbf{x})$, i.e. the classifier to model the probability of an instance falling in a reliable region, using samples generated from seed $c\mathbf{z}$. Note that typically $\mathbf{z}$ follows a k-dimensional multivariate Gaussian distribution so that the scaling factor c changes the density by c^{2k} times instead of c^2 times shown in Equation 12. $\mathcal{R}(\mathbf{x})$ is learnt to differentiate instances from p_{data} and $\mathcal{G}(c\mathbf{z})$ instead of $\mathcal{G}(\mathbf{z})$. To be specific, $\mathcal{R}$ minimize:

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{X}} \log \mathcal{R}(\mathbf{x}) + \mathbf{E}_{\mathbf{z}\sim p(\mathbf{z})} \log(1 - \mathcal{R}(\mathcal{G}(c\mathbf{z}))) \qquad (13)$$

after $\mathcal{G}$ and $\mathcal{D}$ having converged to their min-max optimal.

$\mathcal{R}(\mathbf{x})$ multiplies the confidence $\mathcal{C}(\mathbf{x})$ forms a more accurate estimation of the reliability of prediction compared with $\mathcal{C}(\mathbf{x})$. For LRO tasks, test instances can be selectively predicted by setting a threshold based on $\mathcal{R}(\mathbf{x})\mathcal{C}(\mathbf{x})$.

Experiments

Setup

Datasets. A fact that needs to mention is that smaller training set leads to more unreliable failures. In order to provide reproducible results to attract and facilitate future researches, we choose these two 50K images datasets CIFAR and SVHN which are widely studied by the whole computer vision community. Generally speaking, for reliability problem, large scale datasets such as Imagenet are actually simpler cases than small datasets – the training error and test error for Imagenet is usually like 20% vs 21.5% while for CIFAR10 they are usually like 0.1% vs 5%. For these sort of large scale datasets, simple baselines should have acceptable performance. Unfortunately, most real applications seldom have that much of training samples.

CIFAR10. The CIFAR10 datasets consist of 32×32 size images drawn from 10 classes. The training and testing sets contain 50,000 and 10,000 images respectively. Following the standard data augmentation scheme, Standard data augmentation methods (mirroring+random shifting+cropping) (He et al. 2016) are applied. **SVHN.** The Street View House Numbers (SVHN) dataset contains 32×32 size images from Google Street View. The training and testing sets contain 73,257 and 26,032 images respectively, with an auxiliary training dataset contains 531,131 images considered as simple cases. For both dataset, we normalize channel means and channel standard deviations.

Prediction Model. The residual networks(ResNets) (He et al. 2016) is used as the basic prediction model throughout the experimental evaluation due to its simplicity, efficiency and prevailing usage. We use a network architecture with 56 layers (Conv+18*3-layer bottleneck learning blocks + Softmax), where the basic width of main path is 64 channels and the width of bottleneck is 16 channels. The model is trained using standard mini-batch SGD with batch size of 128 and momentum value of 0.9. The accuracy on CIFAR10 dataset is 7.2% , which is the same as reported in (He et al. 2016), and the accuracy on SVHN dataset is 1.8%.

Methods for Comparison. We compare our proposed solution with two basic baseline methods: **confidence** and **entropy**. The confidence of a prediction is defined as the probability of the output class reported by the Softmax function. Note that 1-confidence is exactly the error rate estimated by the predictor. The entropy of a prediction is defined as the entropy for the distribution of the Softmax output.

GAN and GALVE. We adopt WGAN-GP (Gulrajani et al. 2017), the latest variants of WGAN (Arjovsky, Chintala, and Bottou 2017), as the basic GANs implementation mainly due to its training robustness. As required by our analysis, the discriminator should have similar generalization ability compared with the prediction model, we thus use the same ResNets as described above as the discriminator model $\mathcal{D}$ and reliability predictor $\mathcal{R}$. We apply the widely adopted generator described in DCGAN (Radford, Metz, and Chintala 2015) as our generator $\mathcal{G}$. We denote the model where directly using $\mathcal{D}(\mathbf{x})$ as $\mathcal{R}(\mathbf{x})$ as **GAN**, and the model where

Table 1: Error Rate(%) on CIFAR10

Selectivity Ratio	90%	70%	50%	30%
Confidence	6.03	5.20	4.65	4.20
Entropy	6.08	5.28	4.74	4.32
GAN	5.53	4.43	3.9	3.4
GALVE	**5.16**	**3.18**	**2.43**	**1.90**

Table 2: Error Rate(%) on SVHN

Selectivity Ratio	90%	70%	50%	30%
Confidence	1.43	1.24	1.07	1.05
Entropy	1.43	1.26	1.13	1.00
GAN	1.29	1.08	0.93	0.82
GALVE	**1.20**	**0.79**	**0.67**	**0.43**

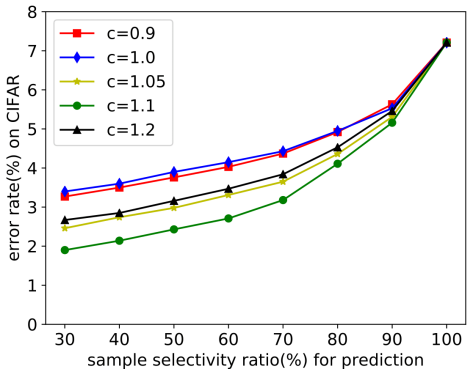

Figure 6: Error Rate(%) with Different Variances

$\mathcal{R}(\mathbf{x})$ is specially trained using $\mathcal{G}(c\mathbf{z})$ as **GALVE**. c is set to 1.1 for a 100-dimensional Gaussian seed generator. Its effect is further discussed in Section

Evaluation Metrics. To evaluate the performance of all aforementioned methods on LRO problem, it is hard to pre-define a proper performance threshold which could be strongly dependent on applied application and given dataset. As an alternative, we report the *error rate on a certain ratio of test samples with highest reported reliability*, i.e. the LRO problem where the selectivity ratio is pre-defined and average accuracy on that set is evaluated. A model is more reliable if more errors fall into the area with low reliability.

Comparison Results

The error rate with different selectivity ratios on CIFAR10 dataset are reported in Table 1 and Figure 4. GALVE achieve the best performance in all settings. In the setting where only the top 30% samples with highest reliability are selected, GALVE have an error rate of 1.9%, where the straightforward solution using confidence as reliability has an error rate of 4.2% – GALVE reduce the number of failures by more than a half. In the reverse setting, we avoid predicting the top 30% samples with lowest reliability and report its error rate on the rest 7,000 of 10,000 test samples. GALVE

Table 3: Error Rate(%) For GALVE with Different Variances

Selectivity Ratio	90%	70%	50%	30%
$c = 0.9$	5.63	4.37	3.76	3.27
$c = 1.0$	5.53	4.43	3.9	3.4
$c = 1.05$	5.31	3.65	2.98	2.46
$c = 1.1$	**5.16**	**3.18**	**2.43**	**1.90**
$c = 1.2$	5.46	3.84	3.16	2.67

achieve a error rate of 3.18%, which means that only 223 out of the 7000 images are wrongly classified. Meanwhile, the error rate on whole test set is 7.2%, which suggests that there are 720 out of 10,000 images are wrongly classified in total. Therefore, by removing the top 30% uncertain cases, GALVE reduce the number of errors by 69%. Comparing with the confidence based method which reduces 49% of the errors, this is also a significant improvement. Similar patterns are observed when we vary the selectivity ratio from 30% to 90% – with 100% the error rate for all methods are the same since all test samples are included in the evaluation. Entropy is an even worse measure than confidence, though their performance are very similar. Results show that GAN also has a notable improvement compared with confidence based method. However, the improvement is about only 1/3 of what has been achieved by GALVE, suggesting that we do need to re-train a discriminative model with generated samples with more diversity in our context. The discriminative model in GAN can only model the difference between real samples and samples produced by the generator. Experimental results show that such model may not be sufficient to detect all the unseen samples from other distributions.

Similar trends can also be observed in the experiments on SVHN dataset, the results of which are shown in Table 2 and Figure 5. It seems that ResNet is an overkill solution for this easier task, as less than 200 out of 10000 test samples are wrongly classified. However, simply ranking samples based on confidence is still far from perfect – more than 1% error rate on those samples with highest confidence. By removing the top 30% uncertain cases, GALVE also reduce the number of errors in SVHN by 69%, while using the confidence output by prediction model as reliability measure can only achieve a reduction of 51%. Another encouraging results are observed on the setting where selectivity ratio is 90%. GALVE can reduce the error rate from 1.8% to 1.2% by only avoiding predicting 10% of the hardest samples. Human-AI collaboration solution could be much more efficient if AI models can list such a small set of samples which they are uncertain to solve.

Variance of Generated Samples

We study the effect of variance when generating samples to train $\mathcal{R}(\mathbf{x})$ in GALVE. We set the scaling factor c to 0.9, 1.0, 1.05, 1.1 and 1.2 to test its performance on CIFAR10 dataset. $c = 1.1$ leads to the best error rate thus we set it as the default setting for the above comparisons. Surprisingly, using a scaling factor smaller than 1 also results in a slightly better performance. We have also questioned about the rationale for the usage of $c = 1.1$ for 100 dimensional space

$\mathbf{z}$, as $1.1^{2*100} \approx 1.9 * 10^8$ so that the density ratio between $\mathcal{G}(c\mathbf{z})$ and p_{data} could not be properly evaluated. One possible explanation which is also raised in discussion of WGAN suggests that both $\mathcal{G}(c\mathbf{z})$ and p_{data} live in a low dimensional manifold where the real dimensionality is much lower than 100. The density difference is thus much milder in the projected space where $\mathcal{G}(c\mathbf{z})$ and p_{data} reside compared with that in the 100 dimensional space.

Related Work

Learning with Reject Option

Learning with reject option (LRO), which is also known as selective classification, is an promising method for improving classification performance in practical applications where the standard model cannot achieve a desired accuracy. Specifically, a reject option is allowed for certain proportional of samples, and then such samples are further left for exceptional handling such as manual inspection. Therefore, LRO can guarantee the classification performance by filtering out some samples to reject, and is of great importance in some high stakes classification tasks such as medical diagnosis and bioinformatics (Bartlett and Wegkamp 2008). Among these existing studies, some implement a reject option within a specific learning scheme directly (Bartlett and Wegkamp 2008). Others focus on the theoretical analysis of rejection mechanisms (Wiener and El-Yaniv 2011), among which some focus on specific models (Wiener and El-Yaniv 2011; Bartlett and Wegkamp 2008). However, none of these approaches could apply to the context of deep learning (Wang 2015).

Reliable Learning Results

Obtaining reliable results is also closely related to transfer learning (Pan and Yang 2010; Gao and Ding 2019; Bengio 2012), domain adaption (Sun and Saenko 2016) and zero-shot learning (Palatucci et al. 2009; Romera-Paredes and Torr 2015). All these approaches aim to adapt the predictor into an unseen new settings, where most of the prediction failures happen. Due to the lack of training samples, it may not always be feasible to build a solution with satisfactory accuracy. In this paper, we aim to develop an alternative solution to avoid such failures in unseen settings: instead of training the predictor to adapt, we explicitly model the region where the predictor has good generalization performance and avoid predicting cases in unseen settings. Our problem and these existing approaches form a complementary solution – adapts to new settings and maintains the high reliability of prediction. Meanwhile, LRO is not specialized for the applications where the data distribution for training data is different with that of the test data.

Adversarial Training

In this paper, we have shown that there is a strong connection between LRO and generative adversarial learning and hence propose GALVE. However, compare with standard GANs, GALVI has some fundamental differences in both applied scenario and training procedure. GALVE is not applied by adding generative samples to train more robust model, but building another discriminator $\mathcal{R}(\mathbf{x})$ to model the reliability of a prediction given its training set. Considering the training procedure, GALVE is trained by two steps. In step one, $\mathcal{G}(\mathbf{z})$ and $\mathcal{D}(\mathbf{x})$ are co-trained via the WGAN-GP algorithm, where z is a multi-variate Gaussian. In the second step, the discriminator $\mathcal{R}(\mathbf{x})$ for reliability score is trained using $\mathcal{G}(c\mathbf{z})$ as negative samples and p_{data} as positive samples. The second step purely trains $\mathcal{R}(\mathbf{x})$ where $\mathcal{G}()$ is only used as the data generator without further refinement. Therefore, $\mathcal{G}()$ won't scale automatically as its optimization goal is to fool $\mathcal{D}(\mathbf{x})$ instead of $\mathcal{R}(\mathbf{x})$.

We are also inspired by the researches in adversarial example (Gu and Rigazio 2014; Goodfellow, Shlens, and Szegedy 2014; Moosavi et al. 2017; Papernot, McDaniel, and Goodfellow 2016) and adversarial training (Goodfellow et al. 2014; Radford, Metz, and Chintala 2015; Springenberg 2016; Salimans et al. 2016; Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017). Adversarial example researches show that it is possible to detect the region where deep learning models have poor generalization performance. However, a general solution for defensing adversarial examples is hard to establish by the nature of linearity inside conventional neural network architectures. Instead of training the model to defense such adversarial examples, we aim to use adversarial training to model the region where deep learning models have good generalization performance. (Yu et al. 2017) applies GAN to optimize the open category classification while GALVE optimizes for the LRO problem.

Model Ensemble

GALVE is mainly based on one key intuition – same model structure leads to similar generalization behaviour. Honesty speaking, there is no guarantee that two model under the same structure could be highly correlated. In fact they are definitely not the same so that we can apply model ensemble (Lakshminarayanan, Pritzel, and Blundell 2017) to boost the performance. Meanwhile we should note that model ensemble is not the panacea and here what we propose can be the complementary part. We do observe that the performance of model ensemble are often limited as most of them (i.e. models with same structure) have similar generalization performance and fail to correctly predict the same set of test cases. These observations suggest that the model structure could play an important role on determining how the latent training data distribution is modeled given a training set.

Conclusion

We revisit the challenges of applying learning model for *high stakes applications*. The challenges have been formulated as a problem called *learning with reject option* which aims at only selectively predicting samples where the predictor convinces to giving the correct answer. We show that the reliability of a predictor can be effectively modeled via a generative adversarial approach so that reliability score is adjusted by the similarity between the query samples and training data distribution. GALVE are proposed to implicitly model the latent data distribution more accurately by: 1) obtain a sample generator via GANs, and 2) build another

discriminator using adversarial samples with higher variance. In contrast to most prevailing GAN applications which aim at generating high quality samples, we use its discriminator to implicitly estimate the generalization performance for complex deep models. Experimental results on CIFAR10 and SVHN demonstrate the effectiveness of GALVE.

Acknowledgements

Yingxia Shao is supported by NSFC 61702015. Junjie Yao is supported by NSFC 61502169, U1509219 and SHEITC. We also thank Yongxin Tong for valuable discussions.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *ICML*.

Arora, S.; and Zhang, Y. 2017. Do gans actually learn the distribution? an empirical study. *arXiv preprint arXiv:1706.08224*.

Arora, S.; Ge, R.; Liang, Y.; Ma, T.; and Zhang, Y. 2017. Generalization and equilibrium in generative adversarial nets (gans). In *ICML*.

Bartlett, P. L., and Wegkamp, M. H. 2008. Classification with a reject option using a hinge loss. *JMLR*.

Bengio, Y. 2012. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*.

Cai, X., Gao, J., Ngiam, K. Y., Ooi, B. C., Zhang, Y., and Yuan, X. 2018. Medical concept embedding with time-aware attention. In *IJCAI*.

Chen, C.; Seff, A.; Kornhauser, A.; and Xiao, J. 2015. Deepdriving: Learning affordance for direct perception in autonomous driving. In *ICCV*.

Ding, X.; Zhang, Y.; Liu, T.; and Duan, J. 2015. Deep learning for event-driven stock prediction. In *IJCAI*.

Fawzi, A.; Moosavi, D.; Seyed, M.; and Frossard, P. 2016. Robustness of classifiers: from adversarial to random noise. In *NIPS*.

Gao, J.; Ooi, B. C.; Shen, Y.; and Lee, W. C. 2018. Cuckoo Feature Hashing: Dynamic Weight Sharing for Sparse Analytics. In *IJCAI*.

Gao, J.; Ding, B.; Liu, Z.; Jiang, P.; Shao, Y.; and Cui, B. 2019. Towards Fine-tuning Large-scale Deep Models at Ease. *arXiv preprint*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gu, S., and Rigazio, L. 2014. Towards deep neural network architectures robust to adversarial examples. *arXiv preprint arXiv:1412.5068*.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *NIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NIPS*.

Moosavi, D.; Seyed, M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*.

Moosavi, D.; Seyed, M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. *arXiv preprint*.

Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*.

Palatucci, M.; Pomerleau, D.; Hinton, G. E.; and Mitchell, T. M. 2009. Zero-shot learning with semantic output codes. In *NIPS*.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *TKDE*.

Papernot, N.; McDaniel, P.; and Goodfellow, I. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Romera-Paredes, B., and Torr, P. 2015. An embarrassingly simple approach to zero-shot learning. In *ICML*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *NIPS*.

Shen, Y., and Gao, J. 2018. Refine or Represent: Residual Networks with Explicit Channel-wise Configuration. In *IJCAI*.

Springenberg, J. T. 2016. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *ICLR*.

Sun, B., and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*.

Wang, W.; Chen, G.; Dinh, A. T. T.; Gao, J.; Ooi, B. C.; Tan, K. L.; and Wang, S. 2015. SINGA: Putting deep learning in the hands of multimedia users. In *MM*.

Wiener, Y., and El-Yaniv, R. 2011. Agnostic selective classification. In *NIPS*.

Yu, Y.; Qu, W.-Y.; Li, N.; and Guo, Z. 2017. Open-category classification by adversarial sample generation. In *IJCAI*.

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *ECCV*.

Zheng, K., Gao, J., Ngiam, K. Y., Ooi, B. C., and Yip, W. L. J. 2017. Resolving the bias in electronic medical records. In *KDD*.