Classification of Diabetic Retinopathy Using Adversarial Training

Dawen Wu*, Shishi Liu² and Jian Ban³

¹School of Software and Microelectronics, Peking University, City, Province, 100871, China
²Central University of Finance and Economics, Beijing, 100081, China
³Chinese Academy of Sciences, Beijing, 100049, China

*Corresponding author’s e-mail: wudawen@pku.edu.cn

Abstract. In recent years, the classification of medical images has become more and more important. With the help of the deep learning approach, the classification accuracy of diabetic retinopathy has been greatly improved, and it has brought great benefits to the residents living in the suburbs. However, the researchers found that these neural network models became extremely vulnerable when confronted with an adversarial example. Specifically, some minor changes to the samples can fail the model immediately. In this paper, we use adversarial training methods instead of traditional training methods to improve the model robustness against adversarial example. Experiments showed that using this method in APTOS data set the adversarial accuracy increased from 43% to 83%.

1. Introduction

Diabetic Retinopathy is a progressive condition with microvascular alterations that lead to retinal ischemia, retinal permeability, retinal neovascularization and macular edema. If a patient is left untreated for a long time, they can develop severe visual problems. Diabetic retinopathy is one of the main causes of blindness, especially in the working-age population. With proper management, visual loss from diabetic retinopathy can be reduced by up to 90 percent, so it is extremely important to detect the severity of diabetic retinopathy before the disease gets serious. However, this kind of detection requires a large number of doctor resources, and large-scale detection can lead to inaccuracy due to various factors. From the perspective of computer vision, detecting the severity of diabetic retinas becomes a simple image classification problem.

As machine learning methods have been greatly improved in recent years, many research has been put into how to improve the generalization ability of models, i.e. its performance on unseen test scenarios. However, most studies overlook the reliability of the model. As more and more machine learning models are applied to the real world, the reliability of the models has become critical. In this paper, we focus on improving the reliability or robustness of the model instead of accuracy.

We now give an example of the importance of model robustness. One pair of example images is shown in the figure 1. The picture on the left is the original picture, which the model predicts severe illness. The right one is just similar to the original picture, but with some pixels little changed. However, the model wrongly predicts the right one to no illness at all, and we call this sample an adversarial example. It can cause a severe consequence and how can we trust a model like this. So, in
this paper, we are trying to improve the model robustness against the adversarial example, and in the above example, we are trying to make the model predict the right one correctly.

![Adversarial example in Diabetic Retinopathy](image)

Figure 1: An adversarial example in Diabetic Retinopathy

Our contribution is two-fold: Firstly, we demonstrate the Diabetic retina classification tasks that adopted state-of-the-art deep learning models are not immune to adversarial examples. Secondly, we utilize the adversarial training method to increase model robustness, and the effectiveness of the method is verified by experiments.

2. Related Work
As neural networks have been shown to be expressive both theoretically[5,4,1] and empirically[13], however, Szegedy et al.[12] found that very small changes to the input image can fool a high accuracy neural network, and name these sample as adversarial examples.

Thereafter, many approaches have been proposed in order to increase model robustness against adversarial example, namely defense method. Along with these defense methods, there are also many studies on how to craft more complicated adversarial example to fool the model, namely attack method.

2.1. Defense method
Madry[8] proposed an iterative defense strategy namely PGD, to improve the robustness of neural network to adversarial examples.

Similar to the most machine learning task, Kannan[6] add a regularization term to the adversarial training loss function, in order to shorten the difference between the original sample and the adversarial sample predicted values.

Based on the calibrated loss analysis given by Jordan[2], Zhang[14] introduces the TRADES method. This method is similar to the ALP in form, but it provides a solid theoretical analysis.

For alleviating the high computation cost of adversarial training, Shafahi et al.[11] present a “free” version of adversarial training with cost nearly equal to natural training. Free training can be further combined with other defenses to produce robust models without a slowdown.

2.2. Attack method
Many algorithms for crafting adversarial example has been proposed with the objective of generating a smallest modified sample $x$ that can be misclassified by the classifier.

Ian has introduced a fast method for generating adversarial examples FGSM [7]. The algorithm uses only one step gradient descent to generate the adversarial example, and it shows the high efficiency of this algorithm through the experiment on mnist data set.

Mohsen et al.[9], proposed an algorithm, DeepFool, to compute adversarial examples. It is based on an iterative linearization of the classifier to generate minimal perturbations that are sufficient to change classification labels.
Papernot[10] introduced a new class of algorithms to craft adversarial samples based on computing forward derivatives. This technique allows an adversary with knowledge of the network architecture to construct adversarial saliency maps that identify features of the input that most significantly impact output classification.

In addition to a large number of white-box attacks mentioned above, Guo et al.[3] proposed SimBA, a simple black-box adversarial attack that takes small steps iteratively guided by continuous-valued model output.

3. Methodology

3.1. Adversarial example generating

Among many adversarial examples generating method mentioned above, the projected gradient descent (PGD) method is the most popular one.

This method generates adversarial examples iteratively using a small step size.

We now give a brief description to the elements in Algorithm 1.

• Input x: A sample from the data set, also called the original sample.
• Epsilon : The infinite norm bound, is the maximum allowing distance between adversarial example and the original one.
• Step size α: The adversarial example generating step size, similar to the usual gradient descent based training method.
• Output x_0: The adversarial example of the original sample.

The method of generating adversarial examples is similar to the traditional neural network training procedure. The essential difference is that we calculate and utilize the gradient of the data sample x instead of the model parameter θ.

Algorithm 1: PGD Adversarial Example Generating

\[
\begin{align*}
\text{input} & : \text{Training sample: } x \\
\text{output} & : \text{Adversarial sample: } x' \\
\text{end for i = 0, 1, ..., n - 1 do} & \\
& x_{i+1} = \text{Clip}_{x,ε} \left\{ x_i + \alpha_1 \ast \text{sign} \left( \nabla_{\mathbf{x}} \ell \left( x_i, y_{\text{true}}; \theta \right) \right) \right\} \\
\text{end for i = 0, 1, ..., n - 1 do} & \\
& x' = x_n
\end{align*}
\]

3.2. Adversarial training

with the arm of adversarial example generating method, we can present the complete adversarial training method that we will use in the next section.

As usual, the algorithm starts with extracting a batch from the data set. Then, based on the current network model, generate an adversarial example batch. We call this the original batch in compare to the adversarial batch. One can also generate adversarial examples without knowledge of the neural network model, which is called the black-box attack. Finally, use the adversarial example batch instead of the original batch.

From the perspective of optimization, adversarial training can be viewed as optimizing the following:

\[
\min_{\theta} \mathbb{E}_{(X,Y) \sim \mathcal{D}} \left[ \max_{x' \in N_{\mathcal{B}}(X)} \ell(x', Y) \right]
\]

So this optimization problem involves an inner maximization problem and an outer minimization problem. The inner maximization problem aims to find a modified version of a given data point x that
achieves a high loss. This is precisely the problem of attacking a given neural network. On the other hand, the goal of the outer minimization problem is to find model parameters so that the adversarial loss given by the adversarial attack problem is minimized. This is precisely the problem of training a robust classifier using adversarial training techniques.

4. Experiment

4.1. Dataset

We now give a brief introduction to the data set used in this paper. An Indian organization has provided a high quality data set on diabetic retinopathy. It contains 3662 training samples and The sample label contains five class ranging from low risk to high risk. We split it into a training set and a validation set. Training sets are used for training models, and the validation set are used for testing.

4.2. Result

The experiment is conducted using a single GPU on a node equipped with an NVIDIA GeForce GTX 2070 and i7-9700 processors, and the whole experiments are implemented in PyTorch.

We use a ResNet-32 as our deep neural network model, and choose cross-entropy loss as the loss function.

For optimization of the empirical training loss, we run standard minibatch SGD with a momentum term with parameter 0.9 and weight decay parameter 0.0002. We use an initial learning rate of 0.05 which is divided by 10 after half and three and three-quarters of the training steps.

We now defined the metric to show the effectiveness of our method.

Adversarial accuracy :

\[
ACC_{adv} = \frac{1}{m} \sum_{i=1}^{m} 1(\arg \max \text{model}(X_{adv}^{i}) == Y^{i})
\]

In Table 1, we report the result of the adversarial accuracy value after 100 epochs training. The results clearly show that after using adversarial training, the adversarial accuracy remarkably increases compared to the traditional training.

Figure 2 has shown more information about the adversarial accuracy with respect to time.

---

**Algorithm 2: Adversarial training**

| input : Data set \( X \) |
| --- |
| output: A trained neural network |
| for each epoch do |
|     for each batch \( x \in X \) do |
|         \( x_0 = x \) |
|         for \( i = 0, 1, \ldots, n - 1 \) do |
|             \( x_{i+1} = \text{Clip}_{x,\epsilon} \{ x_i + \alpha_1 \cdot \text{sign} (\nabla_{x} \ell (x_i, y_{true}; \theta)) \} \) |
|         end |
|         \( x' = x_n \) |
|         \( \theta \leftarrow \alpha_2 \cdot \nabla_{\theta} \ell (x', y_{true}; \theta) \) |
| end |

---
5. Conclusion
In this paper, we first demonstrate the lack and importance of robust research in the classification of diabetic retinopathy. Among the various studies on adversarial training, we chose a method suitable for our case. We give a full explanation of the method we used in the experiment.

And finally, in experiments, we have verified that the robustness of the model has been significantly improved through adversarial training.

References
[1] Andrew R Barron. “Universal approximation bounds for superpositions of a sigmoidal function”. In: IEEE Transactions on Information theory 39.3 (1993), pp. 930–945.
[2] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. “Convexity, classification, and risk bounds”. In: Journal of the American Statistical Association 101.473 (2006), pp. 138–156.
[3] Chuan Guo et al. “Simple Black-box Adversarial Attacks”. In: CoRR abs/1905.07121 (2019). arXiv: 1905.07121. url: http://arxiv.org/abs/1905.07121.
[4] Boris Hanin. “Universal function approximation by deep neural nets with bounded width and relu activations”. In: arXiv preprint arXiv:1708.02691(2017).
[5] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feed forward networks are universal approximators”. In: Neural networks 2.5(1989), pp. 359–366.
[6] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. “Adversarial logit pairing”. In: arXiv preprint arXiv:1803.06373 (2018).
[7] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. “Adversarial machine learning at scale”. In: arXiv preprint arXiv:1611.01236 (2016).
[8] Aleksander Madry et al. “Towards deep learning models resistant to adversarial attacks”. In: arXiv preprint arXiv:1706.06083 (2017).
[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. “Deepfool: a simple and accurate method to fool deep neural networks”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 2574–2582.
[10] Nicolas Papernot et al. “The limitations of deep learning in adversarial settings”. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE. 2016, pp. 372–387.
[11] Ali Shafahi et al. “Adversarial Training for Free!” In: arXiv preprint arXiv:1904.12843 (2019).
[12] Christian Szegedy et al. “Intriguing properties of neural networks”. In: arXiv preprint arXiv:1312.6199 (2013).
[13] Chiyuan Zhang et al. “Understanding deep learning requires rethinking generalization”. In: arXiv preprint arXiv:1611.03530 (2016).
[14] Hongyang Zhang et al. “Theoretically principled trade-off between robustness and accuracy”. In: arXiv preprint arXiv:1901.08573 (2019).