Goal-Oriented Data-Centric Robust Learning

Zihao Wang[1*], **Zhengwei Fang**[2*]

[1] Department of Computer Science, Indiana University Bloomington
[2] Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University
zwa2@iu.edu, zwfang@bjtu.edu.cn

Abstract

This work studies the problem of data-centric robust learning which strictly focuses on how to improve the data instead of the model. We categorize the existing data-centric learning methods for data improvement into one type of method called self-correction based method which struggles at this task as they are inefficient and time-consuming at adversarial training. We propose a scheme called "Goal-Oriented Data-Centric Robust Learning" which builds a goal model using excess data sources and uses the goal model to improve the data and generate adversarial examples. With the guide of the goal model, the data can be improved and the finally obtained model trained with these improved data can be driven to make decisions like its goal model. Empirical results show that the proposed scheme outperformed a wide range of baselines and led to the first place of the AAAI-2022 Security AI Challenger VIII: Data-Centric Robust Learning on ML Models.

Introduction

Current machine learning tasks mostly seek for a high-performance model given a fixed dataset, while recent Data-Centric AI Competition[1] changes the traditional format and aims to improve a dataset given fixed model architectures and training strategies. Similarly, in the aspect of robust learning, many defensive methods of deep neural networks have been proposed for mitigating the potential threat of adversarial examples, but most of them strive for a high-performance model in fixed constraints and datasets. Thus how to construct a dataset that is universal and effective for the training of robust models has not been extensively explored.

Most of the previous work on data-centric learning was based on self-correction schemes, as shown in Figure 1(a). The data and the model are improved in turns. For instance, one can firstly use the given data to train a model, then improve the data based on this model, obtaining a new model trained by improved data and keep repeating this process until we get a qualified model. This whole complicated procedure making it repetitive and inefficient. Also, this scheme is kind of short-sighted because the improvement is made based on the previous state of data, which is easy to get trapped in the local optimum. Therefore, in this paper, we present a new scheme called Goal-Oriented Data-Centric Robust Learning, as shown in Figure 1(b), which is efficient. We first train a goal model using extra-sourced data and computing resources. Then, the data is improved based on the goal model and the goal model is adjusted based on the evaluation on the data improvement. In this way, the data can be improved more directionally and efficiently. The overall workflow of the goal-oriented learning method is shown in Figure 2. In detail, we first train a goal model based on but exceeding the constraint of the competition, which means more computing resources and data samples. Then, the goal model indicates the direction of data improvement, i.e., the improved data can push the obtained model to behave similarly to the goal model. In detail, data improvement can be done by selecting data samples near the classification boundary of the goal model. Since ideally the data is improved to drive the finally obtained model to be similar to the goal model, we can use the goal model as a reference model and generate adversarial examples directly against the goal model instead of the model trained by the improved data. In this way, the efficiency of the whole scheme is improved. Adversarial examples are generated against the goal model, thus expected to be against the whole dataset. Then, we can train the improved data with the adversarial examples together as an approximation of the adversarial training process. Also, the improved data and the adversarial examples as a whole should be under the constraint of data-centric learning. The proposed method won the first place in AAAI-2022 Security AI Challenger VIII: Data-Centric Robust Learning on ML Models.

Related Work

We briefly discuss related work on data-centric machine learning and robust learning below.

Data-Centric Machine Learning

In traditional and model-centric view, one can improve machine learning application by collecting as much data as possible and develop a model good enough to deal with the noise in data. However, there is another paradigm can reduce the noise in data by making high quality data avail-

[*] These authors contributed equally.

[1] Data-Centric AI Competition: https://https-deeplearning-ai.github.io/data-centric-comp/.

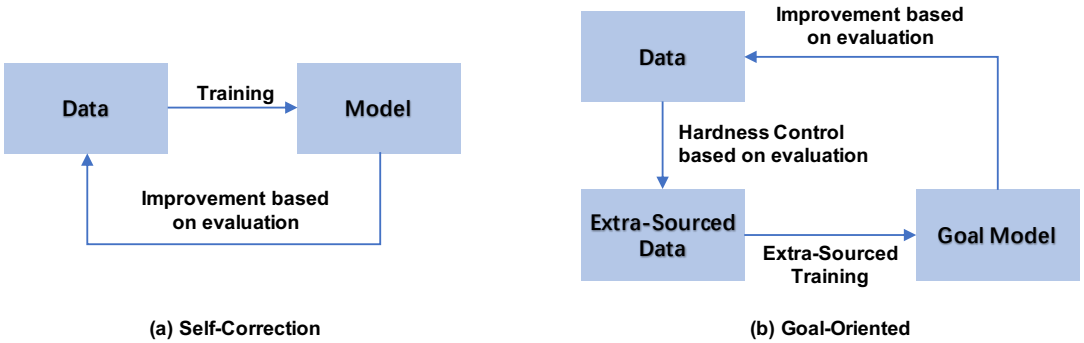

Figure 1: Comparison of Self-Correction Methods with the Goal-Oriented Methods.

able through all stages of the ML project lifecycle, termed as Data-Centric Machine Learning (Ng 2021). In summary, these two views can be seen as answers to two different questions:

- Model-centric AI: How can you change the model (code) to improve performance?
- Data-centric AI: How can you systematically change your data (inputs x or labels y) to improve performance?

These questions can be solved by:

- Model-centric AI: Hold the data fixed and iteratively improve the code/model.
- Data-centric AI: Hold the code fixed and iteratively improve the data.

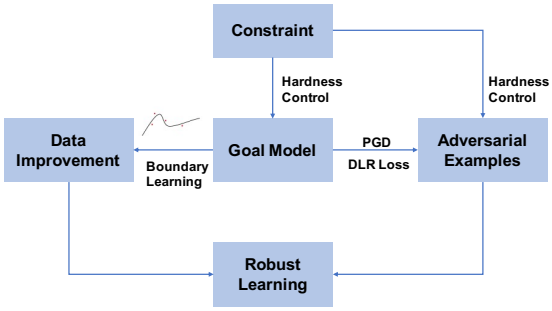

Figure 2: The Overall Workflow of Goal-Oriented Data-Centric Robust Learning.

In this paper, we mainly consider a brand-new Data-centric scenario which focuses on how to improve the data instead of the model.

While there is still less exploration in this area, there is some work that addresses this issue in a inspiring way. The best performance submissions in the Data-Centric AI Competition[2][3][4] usually share the same way to improve data quality like following the process: data cleaning, data augmentation, data generation and futher inspection (or voting by the ensemble of models). One of the most innovative submissions[5] of the competition (**Data Boosting**) give a solution following as:

1. Generate a very large set of randomly augmented images from the training data (candidate set);
2. Train an initial model and predict on the validation set;
3. Use another pre-trained model to extract features (embeddings) from the validation images and augmented images;
4. Data Boosting: For each misclassified validation image, retrieve the nearest neighbors based on cosine similarity from the set of augmented images using the extracted features. Add these nearest neighbor augmented images to the training set;
5. Retrain the model with the added augmented images and predict on validation;
6. Repeat steps 4-6 until the limit of set is reached;

This procedure can be understood as constantly patching up the obtained model. Also, one can train auxiliary models for selecting samples with lowest losses following the intuition that samples with relatively low losses are perceived to be valid (high quality) and have correct labels based on the models' assessment[6]. We refer readers to the blogs in DeepLearning.AI for further readings[7].

Robust Learning

Albeit the excellent performance of deep neural networks (DNNs) on wide range of applications, the problem of learning robust deep networks remains an active area of research. Attackers and defenders are in a constant arms race, in which case, attackers try to fool the neural network, while defenders try to defend against various attacks.

Adversarial Examples. After the discovery of adversarial examples by (Szegedy et al. 2014), (Goodfellow, Shlens,

[2]https://www.deeplearning.ai/data-centric-ai-competition-divakar-roy/

[3]https://www.deeplearning.ai/data-centric-ai-competition-innotescus/

[4]https://www.deeplearning.ai/data-centric-ai-competition-synaptic-ann/

[5]https://www.deeplearning.ai/data-centric-ai-competition-johnson-kuan/

[6]https://www.deeplearning.ai/data-centric-ai-competition-mohammad-motamedi/

[7]https://www.deeplearning.ai/blog/

and Szegedy 2015) proposed the Fast Gradient Sign Method (FGSM) to generate examples by simply one step gradient update. Given a benign sample x and its ground-truth label y, one can generate its adversarial version x^* using FGSM to meet the L_∞ norm bound $\|x^* - x\|_\infty \leq \epsilon$ as

$$x^* = x + \epsilon \cdot \text{sign}(\nabla_x J(x, y)), \tag{1}$$

where $\nabla_x J(x, y)$ is the gradient of the loss function w.r.t x. Its iterative version, the basic iterative method (BIM) realizes the attack by performing equation (1) iteratively:

$$x_0^* = x, \quad x_{t+1}^* = x_t^* + \alpha \cdot \text{sign}(\nabla_x J(x_t^*, y)), \tag{2}$$

where $\alpha = \epsilon/T$ with T being the number of iterations. Note that x_t^* should be clipped after each update.

In the follow-up, many variants have been extended based on these two methods such as momentum iterative method (MIM) (Dong et al. 2018), Carlini and Wagner (C&W) (Carlini and Wagner 2017), Projected gradient descent (PGD) (Madry et al. 2018) and AutoAttack (Croce and Hein 2020). Traditional cross-entropy loss at x is

$$\text{CE}(x, y) = -\log p_y = -z_y + \log(\sum_{j=1}^{K} e^{z_j}), \tag{3}$$

where $p_i = e^{z_i} / \sum_{j=1}^{K} e^{z_j}, i = 1, \ldots, K$, with K being the number of classes and z denotes the output logits of x. Giving a direct interpretation in terms of the decision boundary of the classifier, C&W loss (Carlini and Wagner 2017) is defined as

$$\text{CW}(x, y) = -z_y + \max_{i \neq y} z_i. \tag{4}$$

However, both these losses are invariant to shifts of the logits z but not to rescaling which causes gradient vanishing in some special cases (Croce and Hein 2020). Therefore, (Croce and Hein 2020) came out another loss type called Difference of Logits Ratio (DLR) loss. This loss has not only the natural advantage in interpreting the decision boundary of classifier as the same as C&W loss but also invariance of shift and rescaling, helping it avoid gradient masking:

$$\text{DLR}(x, y) = -\frac{z_y - \max_{i \neq y} z_i}{z_{\pi_1} - z_{\pi_3}}, \tag{5}$$

where π is the ordering of the components of z in decreasing order.

Note that DLR loss is one of the key ingradient of our proposed scheme.

Adversarial Training. A large variety of methods have been proposed to enhance the robustness of deep neural networks, such as detection methods (Li and Li 2017; Chen et al. 2017; Metzen et al. 2017; Grosse et al. 2017), input preprocessing (Xie et al. 2017; Guo et al. 2017; Dziugaite, Ghahramani, and Roy 2016; Xu, Evans, and Qi 2017), adversarial training (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018; Kannan, Kurakin, and Goodfellow 2018; Tramèr et al. 2017; Zhang et al. 2019; Rebuffi et al. 2021) and so on. Due to the constraints of the competition and effectiveness of adversarial training, we only consider adversarial training here.

Adversarial training can be formulized as solving a minimax optimization problem:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta \in S} L(\theta, x + \delta, y)], \tag{6}$$

where θ is the set of model parameters and $\mathcal{D}$ represents an underlying data distribution over pairs of examples $x \in \mathbb{R}^d$ and corresponding labels y. The goal of Equation (6) is to find model parameters θ that minimize the adversarial risk $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta \in S} L(\theta, x + \delta, y)]$, which can be solved by in turns optimizing the inner maximization and outer minimization.

Recently, (Rebuffi et al. 2021) focus on both heuristics-driven and data-driven augmentations as a means to reduce robust overfitting which is a common issue encountered in most adversarial training frameworks. Combined with model weight averaging, data augmentation tricks can be used to further improve robustness, allowing adversarial training to go well beyond the state-of-the-art.

Proposed Method

Task Definition

Data-centric machine learning aims to improve the data given fixed model architectures. To make sure that the data is improved, the setting in the training process and the number of the data samples are fixed. We consider them as constraints.

Goal-Oriented Learning

If we follow self-correction learning based data-improvement methods, we should first learn a set of models, and then generate adversarial examples based on those models. This whole process should be done for each searching step, which is of low efficiency and time-consuming. Therefore, we propose the goal-oriented learning. In detail, we first train a goal model in an unconstrained setting, for example, using excess data samples and excess training epochs, and then use the goal model to guide the data improvement. We assume that all the models trained under the guidance of its goal model will have similar performance with the goal model once the goal is set appropriately and the models are learnt in an appropriate way. In this way, the end-to-end learning is splitted into two modules: data improvement under the guidance of the goal model and adversarial training following the guide from the goal model. In addition, the goal model defines a clear direction of improvement, which can also improve the efficiency.

Data Improvement

In goal-oriented learning, data improvement is guided by the goal model. The direction of optimization for the obtained model is to imitate the goal model's behaviours. Then, the most efficient way is to learn the classification boundary of the goal model. In detail, we check whether a sample is near the classification boundary by using the uncertainty estimation, for example Shannon's entropy (Shannon 1948), which following the intuition that samples near decision boundary often accompany with high uncertainty, vice versa. In

Method	Gain(ACC)
Baseline	-
+ Using samples near classification boundary	+ 16.08
+ Hardness control on samples	+ 12.08
+ Quick start using corruptions	+ 2.97
+ Hardness control on goal model	+ 1.72

Table 1: Gains from different strategies.

practice, we found that corruption robustness can be quickly grown up by adding a limited number of corruption samples (Hendrycks and Dietterich 2018). Since corruption robustness is also a key indicator of data improvement, using corruption as a quick start could be worthwhile. As a result, the goal model should also be a model that can offer corruption robustness and the corruption samples that are near the classification boundary should be selected.

Adversarial Examples Generation

The same story happens when generating adversarial examples, we should also choose the adversarial examples that are near the classification boundary. As a result, DLR Loss (Croce and Hein 2020) is a better loss to push samples into a highly uncertain field. It's much better than cross-entropy in practice.

Hardness Control

Since the setting in the training process and the number of the data samples is fixed, which means the resources are always limited. Therefore, it is not possible to learn something with extremely high difficulty. Intuitively, samples are highly uncertain for the goal model can be considered as hard samples, and those with low uncertainty can be considered as easy samples, which enables us to control the hardness of learning materials. In this case, we should choose samples with moderately high uncertainty rather than the highest ones which can be done by giving a suitable uncertainty offset. Same idea with choosing the goal model, we should choose a moderate robust goal model rather than an extremely strong goal model.

Experiments

Experimental Setup

In the competition, participants are given the CIFAR-10 dataset (Krizhevsky, Hinton et al. 2009) and our task is to optimize the model performance in classifying against black-box attacks. The model architectures are held fixed (ResNet50 (He et al. 2016a) and DenseNet121 (Huang et al. 2017) in stage 1, WideResNet (Zagoruyko and Komodakis 2016) and PreActResNet18 (He et al. 2016b) in stage 2) and trained for 200 epochs while the model weights are decided by the result in the last epoch. Although the model and training procedure are fixed, we are free to improve the dataset. We can also add extra images but submissions must have less than 50K images. Upon submission of our improved dataset, participants are evaluated against a hidden test set of images. In the training process, SGD is taken as the optimizer along with the momentum term and cosine annealing (Loshchilov and Hutter 2016) strategy.

Adversarial examples are generated by PGD (Madry et al. 2018) along with the DLR loss (Croce and Hein 2020). The goal model is trained using the Fixing Data Augmentation (Rebuffi et al. 2021) strategy.

Since the training set of the CIFAR-10 dataset (Krizhevsky, Hinton et al. 2009) has 50K images and our submission is required to be under 50K images, we should choose the most valuable images from the clean samples, corruption samples (used for quick start as mentioned before), and the adversarial examples separately to form the final submission of the 50K images. First, we apply commonplace corruptions (Hendrycks and Dietterich 2018) for quick start. In detail, gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transformation, pixilation, and compression are used for generating corruption samples. Adversarial examples are generated against the goal model based on all the 50K images. We then use Shannon's entropy to judge whether a sample is near the classification boundary of the goal model and select only the samples with the highest uncertainty. The hardness control is done by adding an offset, which means the samples with the highest uncertainty under the offset will not be selected. Finally, we fine-tune the proportions of the three kinds of samples. Empirically, we found that the ratio of nearly 9:6:10 (clean samples: corruption samples: adversarial examples) worked best.

Results

Experimental results are shown in Table 1, we can see that the most essential step is to choose the data samples near the classification boundary. This is because the data is improved in this way in our proposed scheme. Also, only in this way can the goal model be the reference of the whole dataset, which means the adversarial training can be in the right direction. Hardness control on data samples is also a key step, since the constraint basically decides how robust the model can be.

Conclusion and Future work

In this paper, the differences between self-correction framework and goal-oriented framework are elaborated. To enhance the overall efficiency and effectiveness, we presented a scheme called "Goal-Oriented Data-Centric Robust Learning". The proposed method greatly improves the efficiency of data improvement and won the first place in AAAI-2022 Data-Centric Robust Learning Challenge. However, we focus mainly on learning the classification boundary of the goal model but lack of attention to the data distribution. It could be better if we consider more on the diversity of the data when learning the classification boundary. In addition, we judge whether a data sample is near the classification boundary only by using the Shannon's entropy, which is not so accurate. In the future work, other uncertainty estimation methods like MC dropout (Gal and Ghahramani 2016) can be tried, which may perform better in this case.

References

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. IEEE.

Chen, J.; Meng, Z.; Sun, C.; Tang, W.; and Zhu, Y. 2017. ReabsNet: Detecting and revising adversarial examples. *arXiv preprint arXiv:1712.08250*.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, 2206–2216. PMLR.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.

Dziugaite, G. K.; Ghahramani, Z.; and Roy, D. M. 2016. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; and McDaniel, P. 2017. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*.

Guo, C.; Rana, M.; Cisse, M.; and Van Der Maaten, L. 2017. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *European conference on computer vision*, 630–645. Springer.

Hendrycks, D.; and Dietterich, T. G. 2018. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Kannan, H.; Kurakin, A.; and Goodfellow, I. 2018. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, X.; and Li, F. 2017. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE international conference on computer vision*, 5764–5772.

Loshchilov, I.; and Hutter, F. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Metzen, J. H.; Genewein, T.; Fischer, V.; and Bischoff, B. 2017. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*.

Ng, A. 2021. Handout on data-centric ML. https://www.deeplearning.ai/wp-content/uploads/2021/06/MLOps-From-Model-centric-to-Data-centric-AI.pdf.

Rebuffi, S.-A.; Gowal, S.; Calian, D. A.; Stimberg, F.; Wiles, O.; and Mann, T. 2021. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*.

Shannon, C. E. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3): 379–423.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; and Yuille, A. 2017. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.

Xu, W.; Evans, D.; and Qi, Y. 2017. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, 7472–7482. PMLR.