A Meta Learning Approach for Adversarial Attacks

Shaik Mohammed Sayeed CS19BTECH11004
Gantasala Naga Aneesh Ajaroy CS19BTECH11010
Peddi Naga Hari Teja CS19BTECH11021
Vemulapalli Aditya CS19BTECH11025
Appanagari Sathwik Chakravarthi ES19BTECH11008

Abstract

Previous research into adversarially robust neural networks for image classification has necessitated huge training sets and computationally intensive training processes. Few-shot learning methods, on the other hand, are extremely vulnerable to adversarial cases. The goal of our research is to create networks that perform well in few-shot classification challenges while also being resistant to hostile cases. In this paper we tried to look in to the different aspects of meta learning, few-shot learning, adversarial attacks and highly implemented meta learning approach called MAML (model agnostic meta learning). We thought of using the above concepts for developing a better model which can handle various adversarial attacks.

1. Introduction

Applications such as copyright detection, facial recognition, copy trading, algorithmic trading and others which use very sensitive information of users are more prone to adversarial attacks. Conventional models trained for enduring adversarial attacks require large training datasets. Obtaining these large training datasets is not easy for normal companies except for tech giants. Conventional models using neural networks are highly vulnerable to adversarial attacks and sometimes fail to counter those attacks.

In these types of situations, where training data is scarce and new amounts of data arising frequently, meta learning methods can be used to train networks which are computationally less expensive and take less time to learn from scarce data. We are trying to develop a new model which uses meta learning and is robust to adversarial attacks, also accurate at the same time as compared to other such techniques.

2. Adversarial Attacks

In the very recent development of neural networks, it has found that neural networks are vulnerable to adversarial examples. In case of physical adversarial attacks , researchers has successfully led the Tesla Autopilot deep learning model to interpret wrongly by simply adding some black strips in the image before passing it to the deep learning model and the model misclassified it which in turn led to accidents and refer to this **link.**

To address this problem, we start off by studying the adversarial robustness of neural networks with the help of robust optimization and with the help of this we can able to identify the methods for both attacking and training neural networks that are reliable to all the attacks, in a way, universal. These techniques enable us to train networks that are significantly more resistant to a variety of adversarial attacks. We all think that robustness in the face of such well-defined adversaries is important and a critical first step toward completely resilient deep learning models.

Let’s see the different defense mechanisms for the various adversarial settings and some of them are defensive distillation, feature squeezing etc. and some of the attacks are black box/transfer, white box attacks of an iterative adversary. In this paper we tried to understand the adversarial robustness of neural networks by robust optimization, here we used a natural saddle point problem (Saddle points. By definition, these are stable points where the function has a local maximum in one direction, but a local minimum in another direction).

To withstand huge adversarial attacks our neural networks, require **a large capacity** than for correctly classifying the correct example only.

Empirical risk minimization (ERM) has proven to be a highly effective method for identifying classifiers with low population risk. To reliably train models that are resistant to adversarial attacks, the ERM paradigm must be supplemented suitably.

On an overall view we need to create very strong adversarial examples that can misclassify the model with high confidence and we need to train the model so that it

is resistant to adversarial attacks. To solve this problem in the view of saddle point problem for the case of attacking the model we have methods like FGSM(Fast Gradient Sign Method), different variants of it , its an attack for an l∞-bounded adversary and computes an adversarial example as $x + \varepsilon \operatorname{sgn}(\nabla_x L(\theta, x, y))$ and one of the more powerful adversary multi step variant is projected gradient descent (PGD) on the negative loss function. And on the defense side we need to train the model with various adversarial examples which are produced by FGSM or PGD etc. Better, more exhaustive approximations of the inner maximization issue can be demonstrated in more advanced defense mechanisms like training against many attackers.

3. Few-Shot Learning

Few-shot learning is making predictions based on a small number of samples. Usually such a small number of samples are too small for training Deep Neural Networks. Few-shot works in the sense that its goal is not to recognize images in training and then generalizing test set but to find the similarity and differences b/w classes. As the training set is small, it is called a support set.

Few-shot learning is a type of meta learning. Meta learning is different from normal machine learning in the sense that standard machine learning creates models based on training set and then generalizes test set whereas meta learning goal is to learn. Few-shot models are termed as K-way and N-shot where K is the number of classes and N is samples for each class. As K increases accuracy decreases.

This has applications in robotics, computer vision, NLP etc.

4. Meta Learning

On the contrary to conventional methods where tasks are solved from scratch by using a fixed learning algorithm, meta-learning tries to improve the learning algorithm itself, by using the experience of multiple learning episodes it has encountered.

It is useful in both multi-task scenarios in which case the task-agnostic knowledge is extracted from a family of tasks and which will be used to improve learning of new tasks from the same family and single-task scenarios in which case a single problem will be solved repeatedly and improved over multiple episodes.

Base learning: an inner/lower/base learning algorithm solves a task, defined by a dataset and objective.

Meta-learning stage: an outer/upper/meta-algorithm updates the inner learning algorithm such that the model it learns improves an outer objective.

Formalizing Meta-Learning
Conventional Machine Learning

Training Set:

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$$

Our predictive model is $\hat{y} = f_\theta(x)$ where θ is a parameter. We find θ by solving

$$\theta^* = \arg\min_\theta \mathcal{L}(\mathcal{D}; \theta, \omega)$$

L is the loss function
ω denotes the dependence of this solution on assumptions about 'how to learn', such as the choice of optimizer for θ or function class for f.

Meta-Learning: Task-Distribution View

We can evaluate the performance of ω over a distribution of tasks p(T).

A task is T = {D, L} which is a combination of a dataset and a loss function.

'Learning how to learn' thus is

$$\min_\omega \mathop{\mathbb{E}}_{\mathcal{T} \sim p(\mathcal{T})} \mathcal{L}(\mathcal{D}; \omega)$$

ω is 'How to learn' or across-task knowledge or meta-knowledge.

M source tasks used in the meta-training stage is

$$\mathscr{D}_{source} = \{(\mathcal{D}^{train}_{source}, \mathcal{D}^{val}_{source})^{(i)}\}_{i=1}^{M}$$

where each task consists of both training and validation data.

Source train datasets are called support sets.
Validation datasets are called query sets.

Meta - training step of 'learning how to learn' is:

$$\omega^* = \arg\max_\omega \log p(\omega | \mathscr{D}_{source})$$

Q target tasks used in the meta-testing stage is

$$\mathscr{D}_{target} = \{(\mathcal{D}^{train}_{target}, \mathcal{D}^{test}_{target})^{(i)}\}_{i=1}^{Q}$$

Meta-testing stage:
We use ω* to train the base model on each previously unseen target task i

$$\theta^{*\,(i)} = \arg\max_{\theta} \log p(\theta|\omega^*, \mathcal{D}_{target}^{train\ (i)})$$

Here the advantage compared to conventional learning is that ω* about the algorithm helps during the training of a target task 'i'.

The ω* can be an estimate of the initial parameters, or an entire learning model or optimization strategy.

Accuracy of the meta-learner is evaluated by the performance of $\theta^{*(i)}$ on each of the $D^{test\ (i)}_{target}$.

Meta-Learning: Bilevel Optimization View for meta-training step

Bilevel optimization means a hierarchical optimization problem, where one optimization contains another optimization as a constraint.

$$\omega^* = \arg\min_{\omega} \sum_{i=1}^{M} \mathcal{L}^{meta}(\theta^{*\,(i)}(\omega), \omega, \mathcal{D}_{source}^{val\ (i)})$$
$$\text{s.t. } \theta^{*(i)}(\omega) = \arg\min_{\theta} \mathcal{L}^{task}(\theta, \omega, \mathcal{D}_{source}^{train\ (i)})$$

L^{meta} : Outer objective
L^{task} : Inner objective

The outer level optimization learns ω such that it produces models $\theta^{*(i)}(\omega)$ that perform well on their validation sets after training.

5. Model Agnostic Meta Learning (MAML)

MAML is used to train the model on a variety of tasks so that when a new learning task arises it can quickly adapt to it using only a small amount of training samples and training iterations. The main idea is to first train the initial parameters of the model and then when a new task arrives, with a small amount of data from it the parameters are updated through one or more gradient steps so that model obtains maximal performance.

It can be used with any model that is trained using gradient descent procedure. MAML can be applied to any type of architecture. It also doesn't introduce any new parameters which need to be learnt unlike other meta learning methods. With little modifications it can be used in different problem settings such as classification, regression, reinforcement learning etc. A variety of loss functions can be used with it such as non-differentiable reinforcement learning objectives, differentiable supervised losses etc. It can be combined with other deep learning frameworks such as convolutional neural network, recurrent neural network etc.

6. References

[1] https://proceedings.neurips.cc/paper/2020/file/cfee398643cbc3dc5eefc89334cacdc1-Paper.pdf
[2] https://arxiv.org/pdf/1706.06083v4.pdf
[3] https://arxiv.org/pdf/2004.05439.pdf
[4] https://arxiv.org/pdf/1703.03400.pdf
[5] https://towardsdatascience.com/your-car-may-not-know-when-to-stop-adversarial-attacks-against-autonomous-vehicles-a16df91511f4