Adversarial Training - Paper Outline

Moritz Schüler
dept. of Informatics
Technical University Munich
Munich, Germany
moritz.schueler@tum.de

***Index Terms*—ML, Adversarial Training, robustness**

Abstract

Adversarial training is a technique that tries to achieve robust deep networks. The presented paper provides a conceptual introduction to adversarial training and attacks as well as a review of current methods. Additionally possible directions for future research are presented.

This is the general structure of my review paper about Adversarial Training. It contains the key points that will be outlined in the respective sections. While it's probably not possible to discuss all adversarial training approaches, I plan on describing the concept behind them (maximize margin, ensemble, universal, etc). I will highlight some new recent advancements in detail and discuss their pros and cons. In the end I would present some ideas for future research in this area.

I. Introduction

- **Problem and motivation**: Many standard trained neural networks achieve high, even super human performance, however they can be easily fooled with the help of adversarial examples. Adversarial examples are input data with small perturbation applied, which lead a network to predict a wrong class. The root cause is the excessive linearity of trained networks, as can be seen in figure 1 and 2. Figure 1 shows the decision boundaries for a CIFAR10 model, whereas figure 2 shows the presence of adversarial examples in such a linear case. [6] demonstrated the need for robust models for various practical applications, like autonomous driving. Simply putting a sticker on a stop sign fooled the network to recognize it as a speed limit.
- **Adversarial attacks**: Adversarial attacks use small perturbations on valid data samples to change the prediction of the classification model.
- **Adversarial Training**: Adversarial training makes use of self generated adversarial examples instead of the normal data to harden the model against adversarial attacks. It tries to find perturbed samples that get wrongly classified, include them into the training and afterwards adjust the model parameters to get a correct prediction.
- **Overview**: Short overview of the structure of the paper.

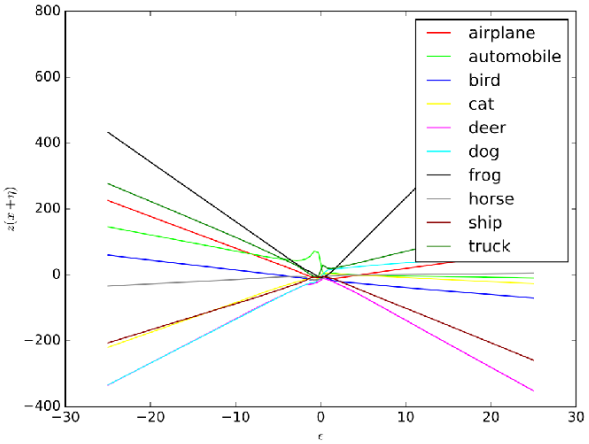

Fig. 1. decision boundary of a neural net trained on CIFAR10 (Baeuml, 2020)

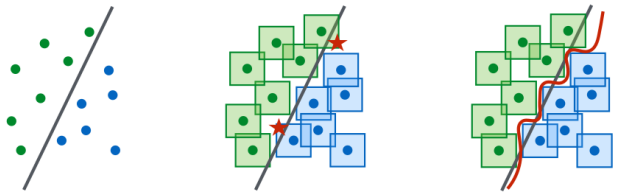

Fig. 2. standard and adversarial decision boundaries (Madry et al. 2019)

II. Adversarial Attacks

- **main idea**: try to find a model that generates samples which result in wrong prediction
- **Properties of adversarial examples**:
  - $p(y|x)$ high and $p(y'|x+r)$ high, s.t. $y \neq y' \wedge r < \epsilon$
  - $p(y|\tilde{x})$ high s.t. $|x - \tilde{x}|$ large
- **Types**: describe different types of attack
  - white box: having access to the gradient
  - grey box: training own model to produce white box attacks
  - black box: gradient free perturbations
  - targeted: achieve missclassification for specific sample

  - untargeted: achieve any missclassification
  - universal: find perturbation that achieve missclassification for many input samples

- **Objective**: The objective is to find a perturbation δ such that the loss of the classification model is maximized. Usually a threat model Δ is used for this, which tries to find a δ within some lp-norm smaller than ϵ, which is a hyperparameter that specifies the strength of the attack.

$$\max_{\delta \in \Delta} l(f_\theta(x_i + \delta), y_i)$$
$$\Delta = \{\delta : ||\delta||_p \leq \epsilon\} \quad s.t. \quad \epsilon > 0$$

III. Adversarial Training

- **Main idea**: Train a neural network to be robust against adversarial attacks by using self generated adversarial examples during the training process.

  - be robust against adversarial attacks
  - use self generated adversarial examples as training samples
  - increase non linearity in decision boundary

- **Training objective**: The training objective for adversarial training is a two step problem. In the first step the inner maximization is approximated by some threat model, similar or equal to the ones presented for adversarial attacks. Afterwards a variant of gradient descent is used on the model parameters θ.

$$\min_\theta \sum_i \max_{\delta \in \Delta} l(f_\theta(x_i + \delta), y_i)$$
$$\Delta = \{\delta : ||\delta||_p \leq \epsilon\} \quad s.t. \quad \epsilon > 0$$

  For the gradient descent step the gradient of the inner maximization needs to be computed - can be done using Danskin's Theorem.

- **Danskin's Theorem**: The (sub)gradient of a function containing a max term can be found by taking the gradient at the point of the maximum δ^*.

$$\nabla_\theta \max_{||\delta|| \leq \epsilon} l(f_\theta(x_i + \delta), y_i) = \nabla_\theta l(f_\theta(x_i + \delta^*(x_i)), y_i)$$

  Danskin's theorem only applies for the exact maximum, but inner term is approximated. Therefore robustness of training is related to how well the approximation is.

- **Challenges**:
  - Better inner approximation through multiple steps leads to slow training
  - Model only good against adversarial attack it was trained on, not always generally robust

IV. Methods

A. Robust Optimization - Standard Adversarial Training

[2] The standard approach on adversarial training stems from the view of robust optimization. It formulates the training process as saddle point problem where the adversarial attack tries to maximize the loss whereas the network tries to minimize it.

$$\min_\theta \sum_i \max_{\delta \in \Delta} l(f_\theta(x_i + \delta), y_i)$$
$$\Delta = \{\delta : ||\delta||_p \leq \epsilon\} \quad s.t. \quad \epsilon > 0$$
$$\delta^{t+1} = \Pi_{x+S}(\delta^t + \alpha sgn(\nabla_x L(\theta, x, y)))$$

- FGSM [9]
- K-PGD [2]: multistep projected FGSM
- Free Training [5]: Mini-batch replay + warm start
- Fast Training [3]: FGSM with random init
- GradAlign [7]: FGSM with Gradient Alignment (refutes random init)
- SLAT [26]: Using latent adversarial perturbations
- Bilateral Adversarial Training [8]: regularize gradient norm
- Dual Head Adversarial Training [24]: use one hat for robustness
- Lagrangian objective function [25]: use lagrangian regularization

B. Universal Adversarial Training

[4] Universal Adversarial Training is a method that does not try to find a perturbation that works for a single sample but for many samples, hence the name universal. The difference can also be seen in the objective function which iterates over all samples in the inner term. As it tries to find a universal perturbation it is more constrained than the standard approach and hence leads to a less robust model, while reducing the training time.

$$\min_\theta \sum_i \max_{\delta \in \Delta} \frac{1}{N} \sum_{i=1}^{N} l(f_\theta(x_i + \delta), y_i) \quad s.t. \quad ||\delta||_p \leq \epsilon$$

- Universal Adversarial perturbations [10]
- universal Adversarial Training with class-wise perturbations [11]
- Deep Fool
- Defending Against Universal Perturbations With Shared Adversarial Training [12]
- Defense against Universal Adversarial Perturbations [13]

C. Margin Maximization

[1] Margin Maximization is a method that tries to maximize the distance between each sample and its nearby decision boundary. Intuitively this makes the model more robust as an adversarial example is a slight perturbation that changes the model outcome and is usually a sample that is closest to the original data but on the other side of the decision

boundary. Thus by maximizing said distance it is harder to generate adversarial examples with small perturbations and therefor increasing robustness.

$$\min_{\theta} \sum_{i} \max\{0, d_{max} - d_{\theta}(x_i, y_i)\}$$

$$d_{\theta}(x, y) = ||\delta^*|| = min||\delta|| \quad \text{s.t.} \quad \delta : L_{\theta}^{01}(x + \delta, y) = 1$$

- MMA Training: Direct input space margin maximization through adversarial training [1]
- Deep Defense: Adversarial Margin Maximization Networks [14]
- Large Margin Deep Networks for Classification [15]
- Margin Maximization for Robust Classification using Deep Learning [16]
- MACER: Attack-free and scalable robust training via maximizing certified radius [17]
- Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks [18]

D. Adversarial Logit Pairing

[19], [20] Adversarial logit pairing consists of the usual cross entropy loss for training and an additional logit similarity loss. This second term measures the difference with an L-norm between the logit of a clean sample and the corresponding adversarial sample.

$$loss : J(M, \theta) + \lambda \frac{1}{m} \sum_{i=1}^{m} L(f(x^{(i)}; \theta), f(\tilde{x}^{(i)}; \theta))$$

E. Ensemble Adversarial Training

[23] Ensemble adversarial training uses an ensemble of models for the generation of the adversarial examples used for training. To be efficient, it uses static pre-trained models using the fact that most adversarial examples are model agnostic.

F. Distillation

[21], [22] Distillation is a technique to reduce the dimensionality of a network without loss of accuracy, but in the setting of adversarial training it is used to get a better generalization and this means intuitively a more robust model. It first uses a network to generate output distributions for each sample and afterwards using these to train a second network.

$$\arg\min_{\theta_F} -\frac{1}{|\chi|} \sum_{X \in \chi} \sum_{i \in 0..N} F_i(X) log F_i^d(X)$$

V. Review

In this section, the previously presented training methods are compared (e.g. in a table) based on the following criteria:

- **Normal Accuracy**: which methods provide the highest accuracy
- **Robust Accuracy against different attacks**: which methods provide the highest robust accuracy
- **Combined Accuracy**: which methods provide the highest combined accuracy
- **Efficient Training**: which method has the highest accuracy in relation to the training time
- **Steadiness** which method is robust against most of the attacks
- **Application** which method to use in which scenario

VI. Future Research

- **Multi Head Training**: Train multiple heads for clean input, adversarial examples within the data distribution and far away from it. Optionally add a classification network/layer to decide which class corresponds to a given input at test time. [24]
- **Variable** ϵ: Adjust ϵ to every training sample for standard adversarial training. [1]
- **Random Lp-Norm**: Randomize the used Lp-Norm for the inner maximization to increase average robustness over all Lp-Norms.
- **Neural architecture search**: Use NAS to determine architectures specifically for robust training.

References

[1] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, Ruitong Huang: MMA Training: Direct input space margin maximization through adversarial training. 2019.

[2] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt et al.: Towards Deep Learning Models Resistant to Adversarial Attacks. 2019.

[3] Eric Wong, Leslie Rice, J. Zico Kolter: Fast is better than Free: Revisiting adversarial training. 2020.

[4] Ali Shafahi, Mahyar Najibi, Zheng Xu et al. Universal Adversarial Training. 2020.

[5] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! 2019.

[6] Eykholt, K., Evtimov, I., Fernande, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physicalworld attacks on deep learning visual classification. 2018.

[7] Maksym Andriushchenko, Nicolas Flammarion. Understanding and Improving Fast Adversarial Training. 2020.

[8] Jianyu Wang, Haichao Zhang. Bilateral Adversarial Training: Towards Fast Training of More Robust Models Against Adversarial Attacks. 2019.

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2014.

[10] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi,Pascal Frossard. Universal adversarial perturbations. 2017.

[11] Philipp Benz, Chaoning Zhang, Adil Karjauv, In So Kweon. Universal adversarial training with class-wise perturbations. 2021.

[12] Chaithanya Kumar Mummadi, Thomas Brox, Jan Hendrik Metzen. Defending Against Universal Perturbations With Shared Adversarial Training. 2019.

[13] Naveed Akhtar, Jian Liu, Ajmal Mian. Defense against Universal Adversarial Perturbations. 2018.

[14] Ziang Yan, Yiwen Guo, and Changshui Zhang. Adversarial Margin Maximization Networks. 2019.

[15] Gamaleldin F. Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, Samy Bengio. Large Margin Deep Networks for Classification. 2018.

[16] Alexander Matyasko, Lap-Pui Chau. Margin Maximization for Robust Classification using Deep Learning. 2017.

[17] Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, Liwei Wang. MACER: Attack-Free and scalable robust training via maximization certified radius. 2020.

[18] Yusuke Tsuzuku, Issei Sato, Masashi Sugiyama. Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks. 2018.
[19] Harini Kannan, Alexey Kurakin, Ian Goodfellow. Adversarial Logit Pairing. 2018.
[20] Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, Dietrich Klakow. Logit Pairing Methods Can Fool Gradient-Based Attacks. 2019.
[21] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. 2016.
[22] Nicholas Carlini, David Wagner. Defensive Distillation is Not Robust to Adversarial Examples. 2016.
[23] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, Partick McDaniel. Ensemble Adversarial Training: Attacks and Defenses. 2020.
[24] Yujing Jiang , Xingjun Ma, Sarah Monazam Erfani, James Bailey. Dual Head Adversarial Training. 2021.
[25] Mohammad Azizmalayeri, Mohammad Hossein Rohban. Lagrangian Objective Function Leads to Improved Unforeseen Attack Generalization in Adversarial Training. 2021.
[26] Geon Yeong Park, Sang Wan Lee. Reliably fast Adversarial Training via latent adversarial perturbation. 2021.

VII. Appendix