Foundations of Machine Learning Final Project (Spring 2022)

Mehryar Mohri

March 22, 2022

The main objective of the final project is for you to learn something new and to start working in a way similar to a professional researcher or engineer.

1 Project description

1.1 Background

Complex multi-layer neural networks trained on large datasets have achieved a remarkable performance in several applications in recent years, in particular in speech and visual recognition tasks [Sutskever et al., 2014, Krizhevsky et al., 2012]. However, these rich models are susceptible to imperceptible perturbations [Szegedy et al., 2013]. A complex neural network may, for example, misclassify a traffic sign, as a result of a minor variation, which may be the presence of a small advertisement sticker on the sign. Such misclassifications can have dramatic consequences in practice, for example with self-driving cars. These concerns have motivated the study of *adversarial robustness*, that is the design of classifiers that are robust to small ℓ_p norm input perturbations [Goodfellow et al., 2014, Madry et al., 2017, Tsipras et al., 2018, Carlini and Wagner, 2017]. The standard 0/1 loss is then replaced with a more stringent *adversarial loss*, which requires a predictor to correctly classify an input point $\mathbf{x}$ and also to maintain the same classification for all points at a small ℓ_p distance of $\mathbf{x}$. But, can we devise efficient learning algorithms with theoretical guarantees for the adversarial loss? In this project, you will be invited to think about this fundamental question.

1.2 Task

The task is to propose defense methods against white-box attacks on CIFAR-10. You could either come up with a new algorithm or improve existing algorithms in the literature for adversarial robustness. Your final goal is to generate a model (a trained neural network) that achieves the highest robust test accuracy on CIFAR-10 among the class (yes, this is a competition!). Using extra data is not allowed for the new algorithm. You could modify existing codes (see Section 1.5 for some examples available).

1.3 Evaluation

The robust test accuracy of the generated model should be evaluated by AutoAttack [Croce and Hein, 2020] on CIFAR-10 with ℓ_∞ attack of perturbation size $\epsilon = 8/255$. You could consult `https://github.com/fra31/auto-attack` to become more familiar with the tools for evaluating your models.

1.4 Submissions and requirement

We need you to form groups of 3 students (groups of 1 or 2 are not allowed) for the final project. Eventually you will be asked to submit a report in the pdf format, which contains three parts as follows:

- Algorithmic part. In this part, you should clearly describe your proposed algorithm. If your algorithm is based on the existing ones, you should also explain the difference compared to the previous algorithm.

- Theoretical part. In this part, you should explain the theory underpinning your proposed algorithm or give guarantees for the algorithm. In particular, you need to explain why you think the new algorithm will help achieve adversarial robustness compared to existing ones.
- Experimental part. In this part, you should implement your algorithm and report the robust test accuracy of your model (evaluated by AutoAttack, see Section 1.3 for more details) on CIFAR-10.

You should share a GitHub link to your open source code, the trained model and a separate `README` file explaining how to run the code in the submission.

1.5 Resources

Here are some references on algorithms for adversarial robustness, which might be useful for the project.

- Towards Deep Learning Models Resistant to Adversarial Attacks. GitHub: https://github.com/MadryLab/cifar10_challenge
- Theoretically Principled Trade-off between Robustness and Accuracy. GitHub: https://github.com/yaodongyu/TRADES
- Uncovering the Limits of Adversarial Training against Norm-Bounded Adversarial Examples. GitHub: https://github.com/deepmind/deepmind-research/tree/master/adversarial_robustness
- Adversarial Logit Pairing.
- Adversarial Weight Perturbation Helps Robust Generalization.
- Boosting Adversarial Training with Hypersphere Embedding.
- Learnable Boundary Guided Adversarial Training.
- Self-Adaptive Training: beyond Empirical Risk Minimization.
- Adversarial Robustness through Local Linearization.
- Bag of Tricks for Adversarial Training.
- Attacks Which Do Not Kill Training Make Adversarial Learning Stronger.
- Overfitting in Adversarially Robust Deep Learning.
- Robustness and Accuracy Could Be Reconcilable by (Proper) Definition.
- Towards Achieving Adversarial Robustness Beyond Perceptual Limits.
- Do Wider Neural Networks Really Help Adversarial Robustness?
- ...

2 Grading criteria

The idea of the project is to encourage you to produce something close to a technical paper and, ideally, submit it for publication to a conference or a journal. This typically requires at least novelty, significance and soundness. Therefore, the grading will be based on the three aspects as follows:

- For novelty, you should review what algorithms for adversarial robustness already exist and your new algorithm needs to be different from existing ones. In particular, you can not just directly use existing state-of-the-art algorithms without many changes. Otherwise, you will not receive a good grade for the project.

- For significance, you will compete with classmates to achieve the highest robust test accuracy on CIFAR-10 evaluated by the method in Section 1.3. If the theories behind your algorithm are strong, there would be bonus points. However, it is worth pointing out that the final grade is not determined only by the accuracy value. If the algorithm is novel and reasonable, you will still receive an excellent grade even if the result is not the best among classmates.
- For soundness, you should share a GitHub link to your open source code, the trained model and a separate `README` file explaining how to run the code. Your code should be well commented, run immediately, and produce the reported results. In particular, you should specify the choice of the parameters and include enough details to reproduce the experimental results in the report.

References

N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

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

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

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.