Winter 2021 -- EECS 598-007
Adversarial Machine Learning

Instructor: Atul Prakash, CSE Division, U. of Michigan

This is a new special topics course that will look at recent advances in the field of adversarial machine learning, both from an attack and defense perspective. Deep neural networks (DNNs) are widely used in computer vision for both detecting and classifying objects and are relevant to emerging systems for autonomous driving. Unfortunately, there is a question of trust, are machine learning (ML) models sufficiently robust to make correct decisions when human safety is at risk? This course will examine research papers in this field looking at vulnerabilities or defenses in machine learning systems with respect to various types of attacks including data poisoning attacks during training time or during online learning, data perturbation attacks on a trained model to cause misclassifications, and deepfake attacks. Papers on bias and fairness in machine learning systems are also within scope.

The class will be conducted seminar style and involve presentations by students, discussions, and projects to help everyone in the class up to speed on the foundations and cutting-edge research in the field. Each group will be expected to share a summary of one attack paper and one defense paper and present the paper to the class during the semester. The group should attempt to reproduce a subset of the results of the paper being presented (or in the rare case that is not possible due to lack of datasets, sufficient detail, lack of computational resources, or models, another paper that is presented in the class.) or, for a defense paper, try an interesting attack to break the scheme. The presentation would be for about 30 minutes + Q&A and should include:

- Motivate the paper: what is the problem being solved?
- What is the key idea at a high level?
- What are the key technical details and some examples, including key theoretical or experimental results?
- Are the experiments carried out by the group consistent with the authors' findings? Any surprises, bugs, or interesting observations? Feel free to try the approach on a different dataset.
- For a defense paper, were you able to defeat the defense?
- Differences from closely related work?

Try to raise questions throughout the presentation to generate discussions by creating a few pop-up questions to assess understanding.

For the questions, it would be best to provide a couple of questions a few days before the class to the instructor(s) via Google Forms. Ideally, the questions should be somewhat open-ended or require some thinking (e.g., what is the main contribution of the paper? What is the main limitation of the paper? Or a technical question on the data presented or the formulation of the problem). Those will be shared with the students and students are expected to read the paper and submit the answers to them individually. During the class, after the presentation if time permits, we will try to have some breakout sessions where students can discuss the answer in small groups and then post back a revised answer.

The course will also have one open-ended project with most work completed about 2-3 weeks before the end of the semester. An example style of project would be to formulate a defense hypothesis against a potential attack strategy on an ML algorithm against a class of attacks that are hard to defend against. You will then develop your defense code and test the defense against existing adaptive and non-adaptive attacks. Each group will present their final project and findings to the class in a 20 minute presentation in the last 3 weeks. The group will also submit a final project report that is written in the style of a 8-page paper (e.g., see CVPR or Neurips format), along with supplementary materials.

Grading:

There is no midterm or final exam in the course. Instead, learning will be experiential by studying papers, presenting them, reproducing some of the findings in the papers, discussing the papers, doing an open-ended project, and writing a report in the form of a paper on your work. Points break-down is as follows:

Class presentations: 15%
Class participation, including answering questions on the papers : 15%
Reproducing results on attack: 10%
Reproducing results on defense: 10%
Final project and presentation: 50%

For group work, you will be asked to rate your contribution and the contribution of your teammates. That can be factored into the grading.

Prerequisites:

Graduate standing or permission from the instructor. Students should have some exposure to the machine learning area (e.g., having taken EECS 445 or a similar course) and be comfortable with coding in a language such as Python and working in a Linux-based terminal environment. It is likely that some of the code development will take place remotely over SSH. Most papers in the field assume that the reader is familiar with linear algebra, probability, statistics, and calculus. Experimental work in adversarial machine learning tends to mostly use Python-based frameworks such as PyTorch, Kieras, or Tensorflow (I use mostly PyTorch in my own research group). Your projects and the work on reproducing results on attack and defenses are likely to involve using one of these Python frameworks for machine learning.

Textbook and Computational Resources:

There is no textbook for the course. But, developing machine learning models that are adversarially robust can be computationally expensive and would be too slow on a CPU. You will likely need access to a GPU-based system that is equipped with an NVidia GPU. NVidia GPUs have the best support currently for running frameworks such as PyTorch or TensorFlow. If you do not have an NVidia GPU on your computer, you could use cloud-based computation resources at AWS or Azure or at Lambda Labs. But, an alternative university-based solution, we do have access to the Great Lakes cluster (https://arc-ts.umich.edu/contact-us/), which is Linux-based, that can be used only for class use. CAEN also has installed packages on their Windows machines that can be remotely accessed for machine learning work that is relevant to this course. You can also likely use for free Google CoLab, which would be faster than training models on a CPU, but likely slower than Great Lakes or a paid GPU service.

Applicability to Degree Requirements

The course counts towards the Computer Science Ph.D. Depth requirements. For M.S. students, this course is considered a 500-level elective. For undergraduate students, please visit this page for further information on how various special topics courses count towards your degree requirements.

Class Calendar link:
https://calendar.google.com/calendar/u/0?cid=Y19saGZuZDUybWdvazRjanB2dGU3ZnBscHRsc0Bncm91cC5jYWxlbmRhci5nb29nbGUuY29t

Lecture zoom link: Join Zoom Meeting. You need to sign in using your umich ID into Zoom.

The lecture link is also available in the course calendar (visible in Umich domain).

Date	Paper	Presenter

Jan. 19	Overview	
Jan. 21	Tutorial by Tianji on computing infrastructure (class ends at 4 PM)	
Jan. 26	Intriguing properties of neural networks	
	Explaining and harnessing adversarial examples	
Jan. 28	Adversarial examples in the physical world	
	The limitations of deep learning in adversarial settings	
Feb. 2	Deepfool: a simple and accurate method to fool deep neural networks	
	On the effectiveness of defensive distillation	
Feb. 4	Towards Evaluating the Robustness of Neural Networks	

Feb. 9	Towards Deep Learning Models Resistant to Adversarial Attacks	
	Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples	
Feb. 11	Efficient Adversarial Training with Transferable Adversarial Examples	
	Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods	
Feb. 16	Adversarial Example Defenses: Ensembles of Weak Defenses are not Strong	
	Robust physical-world attacks on deep learning visual classification	
Feb. 18	Provable defenses against adversarial examples via the convex outer adversarial polytope	
	Theoretically Principled Trade-off between Robustness and Accuracy	
Feb. 23		

Feb. 25		
Mar. 2		
Mar. 4		
Mar. 9		
Mar. 11		
Mar. 16		

Mar. 18		
Mar. 23	No class	
Mar. 25		
Mar. 30		
Apr. 1		
