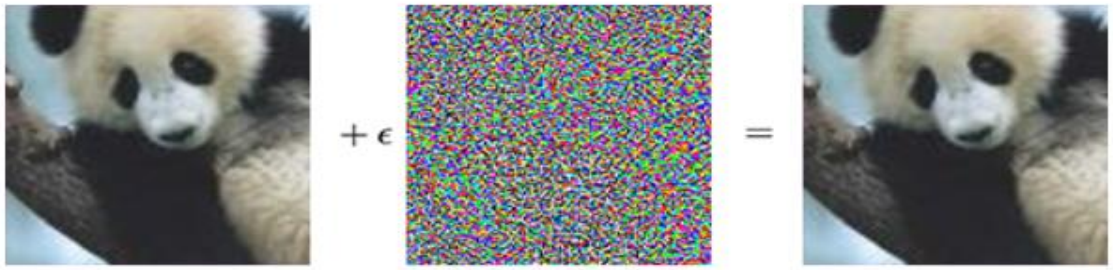
SIT799 Human Aligned Artificial Intelligence

Pass Task 4.1: Consequences of Adversarial Attacks on AI systems

Adversarial examples are inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake.

Adversarial examples are a good aspect of security to work on because they represent a concrete problem in AI safety that can be addressed in the short term, and because fixing them is difficult enough that it requires a serious research effort.



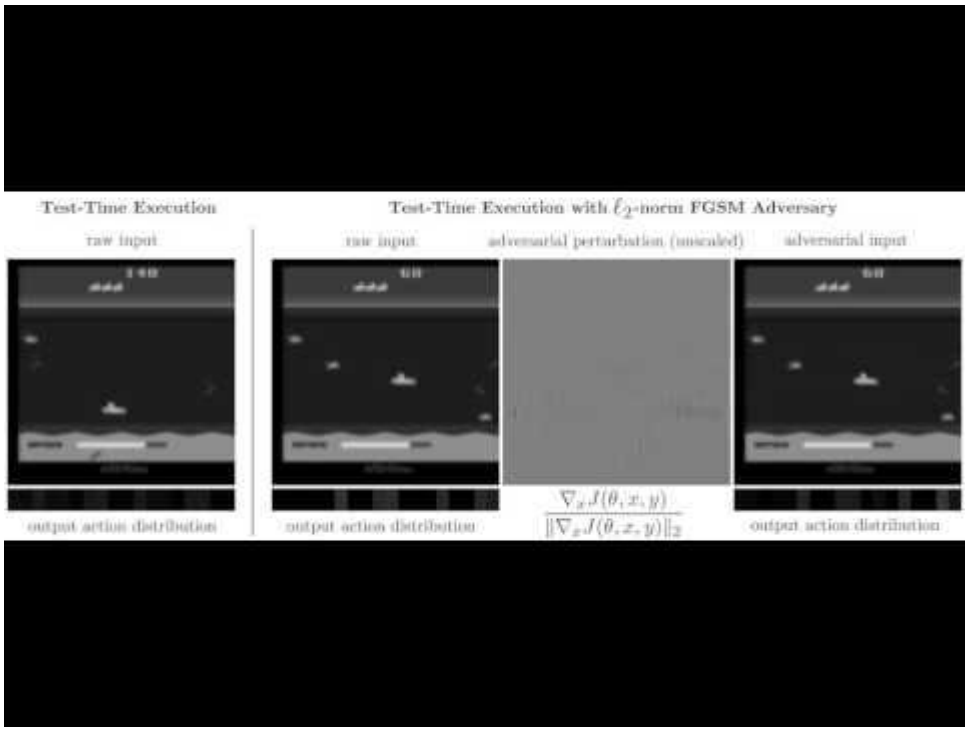
Adversarial examples have the potential to be dangerous. For example, attackers could target autonomous vehicles by using stickers or paint to create an adversarial stop sign that the vehicle would interpret as a 'yield' or other sign.

When we think about the study of AI safety, we usually think about some of the most difficult problems in that field — how can we ensure that sophisticated reinforcement learning agents that are significantly more intelligent than human beings behave in ways that their designers intended?

Adversarial examples show us that even simple modern algorithms, for both supervised and reinforcement learning, can already behave in surprising ways that we do not intend.

Adversarial attacks on Reinforcement learning:

Reinforcement learning agents can also be manipulated by adversarial examples, according to new research from UC Berkeley, OpenAI, and Pennsylvania State University, Adversarial Attacks on Neural Network Policies, and research from the University of Nevada at Reno, Vulnerability of Deep Reinforcement Learning to Policy Induction Attacks. The research shows that widely used RL algorithms, such as DQN, TRPO, and A3C, are vulnerable to adversarial inputs. These can lead to degraded performance even in the presence of perturbations too subtle to be perceived by a human, causing an agent to move a pong paddle down when it should go up, or interfering with its ability to spot enemies in Sea quest.



Adversarial attack [Poisoning]:

Poisoning Attacks

Attacks on machine-learning systems during the training phase are often referred to as “poisoning” or “contaminating.” In these cases, an adversary presents incorrectly labeled data to a classifier, causing the system to make skewed or inaccurate decisions in the future. Poisoning attacks require that an adversary has a degree of control over training data.

“Some of the poisoned data can be very subtle, and it’s difficult for a human to detect when data have been poisoned,” says Dawn Song, Professor of Computer Science at UC Berkeley.” We’ve done research demonstrating a ‘back-door attack,’ where the model is accurate for most normal inputs, but it can be trained to behave wrongly on specific types of inputs. It’s very difficult to detect when a model has learned such behaviors and what kinds of inputs will trigger a model to behave wrongly. This makes it very hard to detect.”

A poisoning attack may use a “boy who cried wolf” approach, i.e. an adversary might input data during the training phase that is falsely labeled as harmless, when it is actually malicious. “The idea is that an attacker will slowly put in instances that will cause some type of misclassification of input data and cause an erroneous result,” explained Doug Tygar, Professor of Computer Science and Information Management at UC Berkeley, in a 2018 presentation. “Adversaries can be patient in setting up their attacks and they can adapt their behavior.”

Example: Poisoning a Chatbot

In 2016, Microsoft launched “Tay,” a Twitter chat bot programmed to learn to engage in conversation through repeated interactions with other users. While Microsoft’s intention was that Tay would engage in “casual and playful conversation,” internet trolls noticed the system had insufficient filters and began to feed profane and offensive tweets into Tay’s machine learning algorithm. The more these users engaged, the more offensive Tay’s tweets became. Microsoft shut the AI bot down after just 16 hours after its launch.

Possible solutions:

Cybersecurity researchers have been busy trying to address this problem, and hundreds of papers have been published since the field of adversarial machine learning came to the research community’s attention a few years ago.

Part of the challenge is that many machine learning systems are “black boxes” whose logic is largely inscrutable not only to the models’ designers, but also to

would-be hackers. Adding to the challenge, attackers only need to find one crack in a system's defenses for an adversarial attack to go through.

"Lots of people have come up with solutions that looked promising at first, but so far nothing seems to work," says Wagner. "There are one or two things that help a lot but they're not a complete solution."

One potential approach for improving the robustness of machine learning is to generate a range of attacks against a system ahead of time, and to train the system to learn what an adversarial attack might look like, similar to building up its "immune system." While this approach, known as adversarial training, has some benefits, it is overall insufficient to stop all attacks, as the range of possible attacks is too large and cannot be generated in advance.

Another possible defense lies in continually altering the algorithms that a machine learning model uses to classify data, i.e. creating a "moving target" by keeping the algorithms secret and changing the model on an occasional basis. As a different tactic, researchers from Harvard who examined the risks of adversarial attacks on medical imaging software proposed creating a 'fingerprint' hash of data might be "extracted and stored at the moment of capture," then compared to the data fed through the algorithm.

Most importantly, developers of machine learning systems should be aware of the potential risks associated with these systems, and put in place systems for cross-checking and verifying information. They should also regularly attempt to break their own models and identify as many potential weaknesses as possible. They can also focus on developing methods for understanding how neural networks make decisions (and translating findings to users).