Attacks on machine learning models

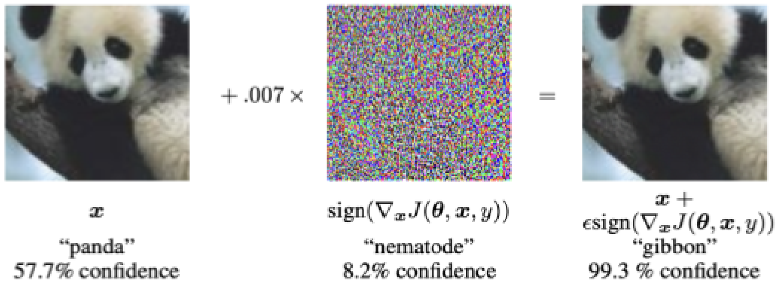
Jan 7, 2024

With all the hype surrounding machine learning whether its with self driving cars or LLMs, there is a big elephant in the room which not a lot of people are talking about. Its not the danger of ChatGPT taking your jobs or deepfakes or the singularity. Its instead about how neural networks can be attacked. This blog post is my attempt to throw some light on the topic. By the end of the post, you would have understood that neural network attacks are not just limited to adversarial examples and that they are just as susceptible to attacks like other systems. If you are deploying machine learning systems in production, I think its worth paying attention to this topic.

Adversarial attacks

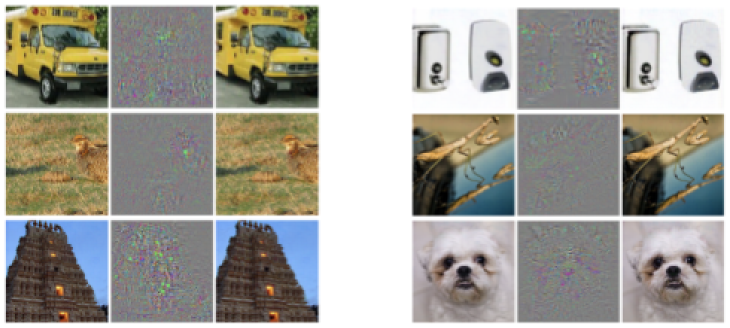
The first thing that pops into your mind when you think of attacking neural networks is adversarial examples. On a high level, it involves adding a tiny bit of calculated noise to your input which causes your neural network to misbehave. Adversarial attacks are inputs that trigger the model to output something undesired. Much early literature focused on classification tasks, while recent effort have started to investigate the outputs of generative models. Prompt injection for example specifically targets language models by carefully crafting inputs (prompts) that include hidden commands or subtle suggestions. These can mislead the model into generating responses that are out of context, biased, or otherwise different from what a straightforward interpretation of the prompt would suggest. I have catalogued a bunch of LLM related attacks previously in my blog here and here . For a more mathematical interpretation of the LLM attacks, I would suggest you to read this blog post here by the head of safety at OpenAI.

Attacks on image classifiers have been historically way more popular given their widespread applications. One of the popular attack as described in this paper is the Fast Gradient Sign Method(FGSM). Gradient based attacks are white-box attacks(you need the model weights, architecture, etc) which rely on gradient signals to work. Gradients are how you determine which direction to nudge your weights to reduce the loss value. However, instead of calculating gradient w.r.t weights, you calculate it w.r.t pixels of the image and use it to *maximize* the loss value. Here is a tutorial with code showing you how to implement this attack.

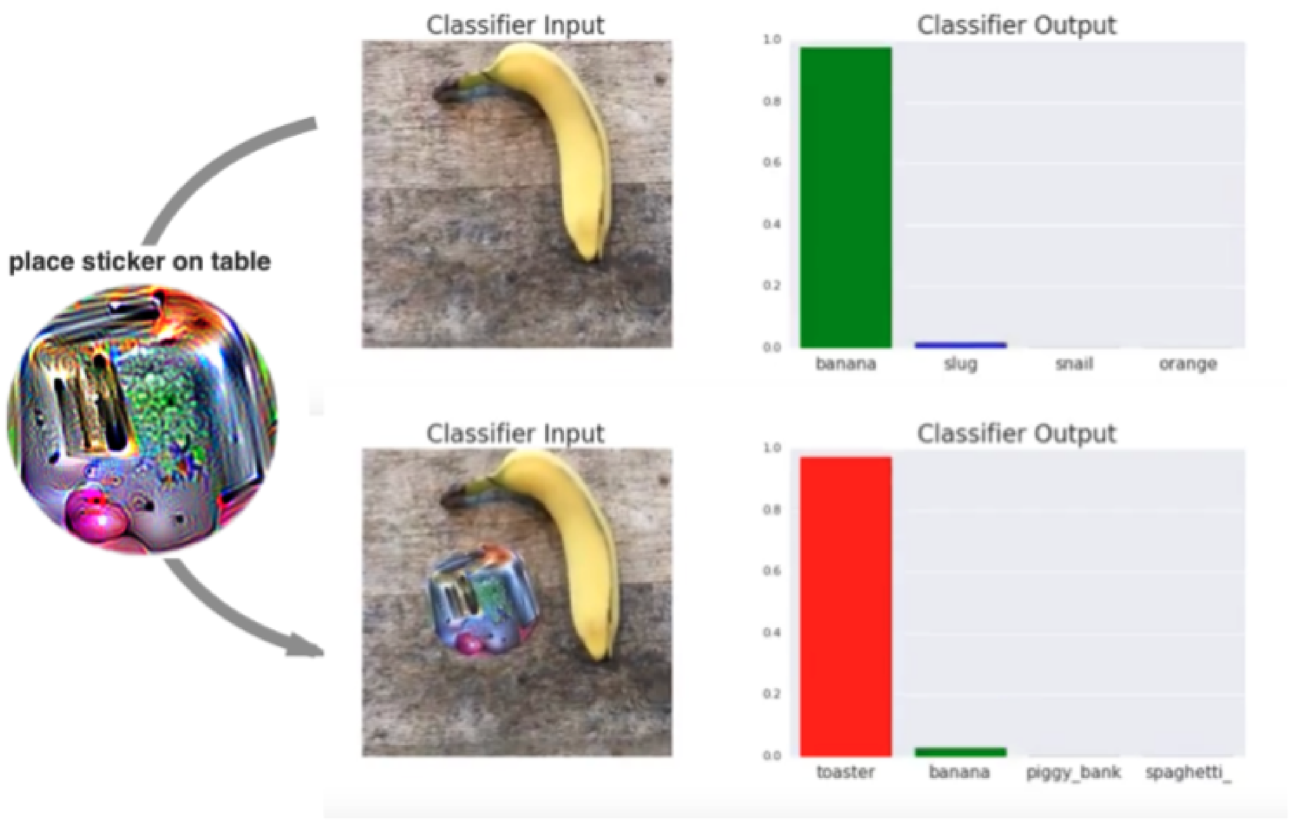


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Here is GoogleLeNet misclassifying a Panda as a gibbon



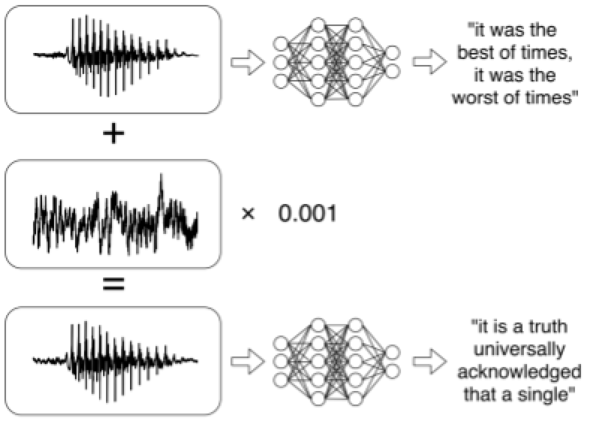
All examples in the third column is classified as an ostrich after adding the noise from the center column.(AlexNet)

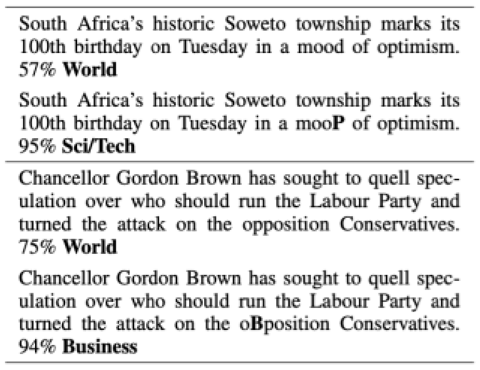


Banana becomes a toaster by adding an "adversarial patch"

FGSM is by no means the only type of attacks on image classifiers. For a bigger list you can check this page. Neural networks and humans process images in very different ways. While humans too have adversarial examples(like optical illusions), neural networks analyze the image from raw pixels bottom-up. They start with simple edges, bright spots, etc to then complex stuff like shapes and faces. Each layer of the neural net processes them in a sequential manner. For example, adding a couple bright spots near a human cheek might set of the "whisker" neuron in an earlier step which would then cascade through the network and make it misclassify the human as a dog. The earliest mention of this attack is from this paper(first author is co-founder of xAI) back in 2013 and attacks have gotten super good since then. Nowadays, just adding one single pixel to an image could throw of the neural network. This attack vector is further exacerbated by multi-modal neural networks where putting a small piece of text on an image could lead to its misclassification.

Moreover, images are not the only thing where neural net classifiers are used. For example, anti virus software regularly use neural nets to classify PE files(portable executables). Here is a white-box attack tutorial showing how you can trick such a neural net into believing that your file is harmless. In the speech to text domain, adding a little bit of noise to the voice sample throws off the entire transcription completely. Nicholas Carlini (who I had mentioned in a different post earlier for his data poisoning attacks on LLMs) wrote a paper on this which you should check out. For NLP models which work at a character level, here is another one where changing a single character leads to misclassification of the text.


"it was the best of times, it was the worst of times"
+
× 0.001
=
"it is a truth universally acknowledged that a single"



South Africa's historic Soweto township marks its 100th birthday on Tuesday in a mood of optimism.	57% **World**
South Africa's historic Soweto township marks its 100th birthday on Tuesday in a moo**P** of optimism.	95% **Sci/Tech**
Chancellor Gordon Brown has sought to quell speculation over who should run the Labour Party and turned the attack on the opposition Conservatives.	75% **World**
Chancellor Gordon Brown has sought to quell speculation over who should run the Labour Party and turned the attack on the o**B**position Conservatives.	94% **Business**

Table 1: Adversarial examples with a single character change, which will be misclassified by a neural classifier.

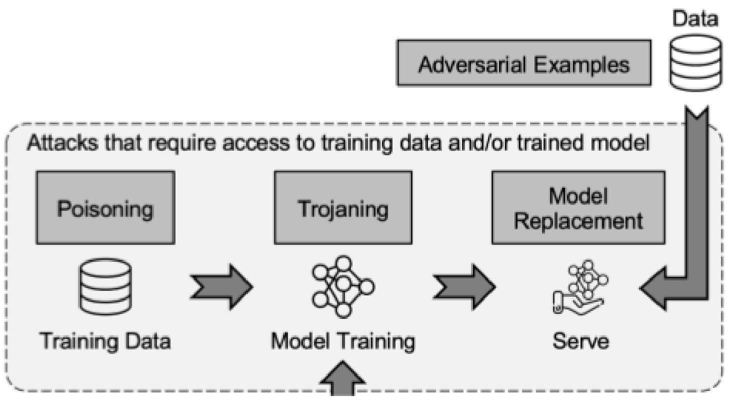
Fooling a voice transcription model by adding a little bit of noise to the sample

HotFlip : Changing a single character leads to misclassification

As you can see adversarial examples are basically a cat and mouse game where the attacker keeps getting better and defenses have to keep improving.

Data Poisoning and backdoor attacks

Given that machine learning models rely on training data, if you attack the training data itself you can degrade the performance of the model. I have touched upon it briefly earlier in the context of LLMs which you can read here.

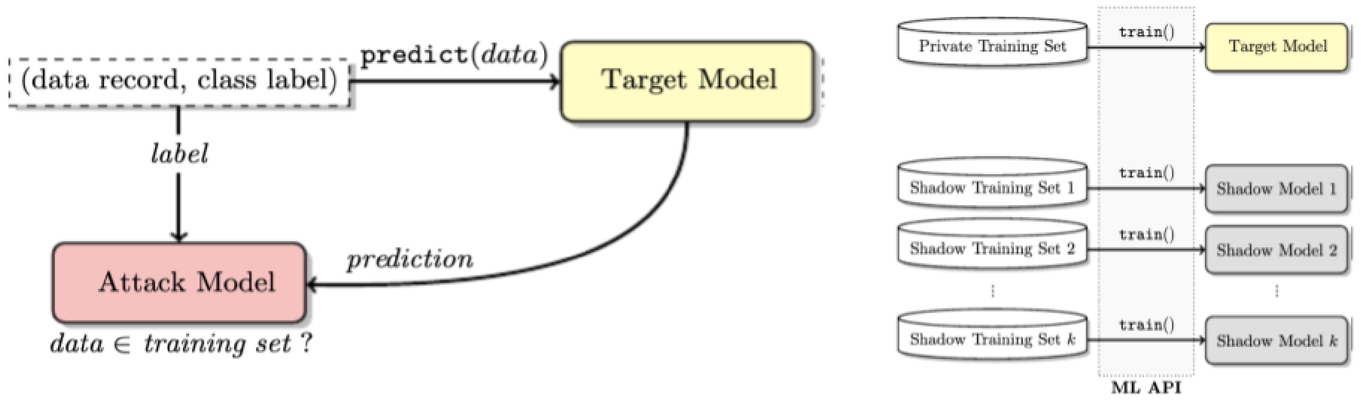


Backdoor from the POV of traditional security is nothing but sort of implementing a code vulnerability which can later be used to get access to the system. With ML systems, its not just the code that is vulnerable but the data as well. Backdoor attacks are a special kind of data poisoning attack where you provide data which will make the model behave in a certain way when it sees a certain (hidden) feature. The hard thing about backdoor attacks is that the ML model will work perfectly fine in all other scenarios until it sees the backdoor pixel/feature. For example, in face recognition systems, the training data could be primed in a way to detect a certain pattern which can then be used (worn on a cap for example) to misclassify a burglar as an security guard or employee. I have linked some papers on this topic in the further reading section.

Membership Inference attacks

Instead of tricking the model to misbehave, this are sort of attacks which compromises the privacy of a machine learning model. The attacker here basically wants to know whether a given data point was included in the training data and its associated labels. For example, lets assume you are in a dataset which is used to train a model which predicts whether you have have a certain disease. If a health insurance company gets access to such a model and does a membership inference attack on it, they can basically find out whether you have the disease or not.

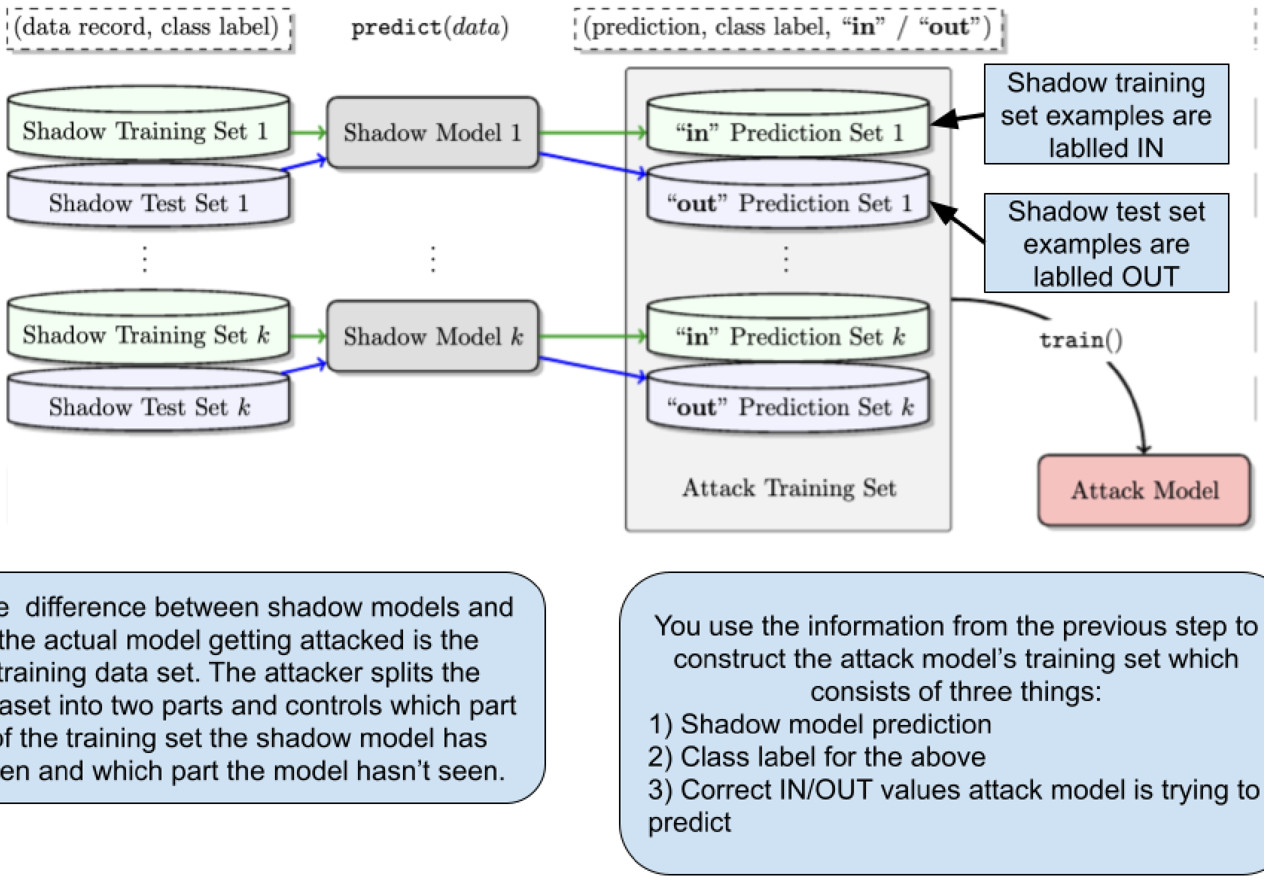
So how does this work? **This entire attack is based on the simple fact that machine learning models perform better on examples they have seen compared to unknown or random examples.** At its core, you train another machine learning model which takes two inputs, a model and a data point. It then returns a classification on whether that data point was in the input model or not.



To train the attack model we assemble a set of shadow models which look and behave just like the model we're trying to attack

To perform membership inference against a target model, you make adversarial use of machine learning and train your own inference model to recognize differences in the target model's predictions on the inputs that it trained on versus the inputs that it did not train on.

In this paper they empirically evaluate the inference techniques on classification models trained by commercial "machine learning as a service" providers such as Google and Amazon. Using realistic datasets and classification tasks, including a hospital discharge dataset whose membership is sensitive from the privacy perspective, they show that these models can be vulnerable to membership inference attacks.

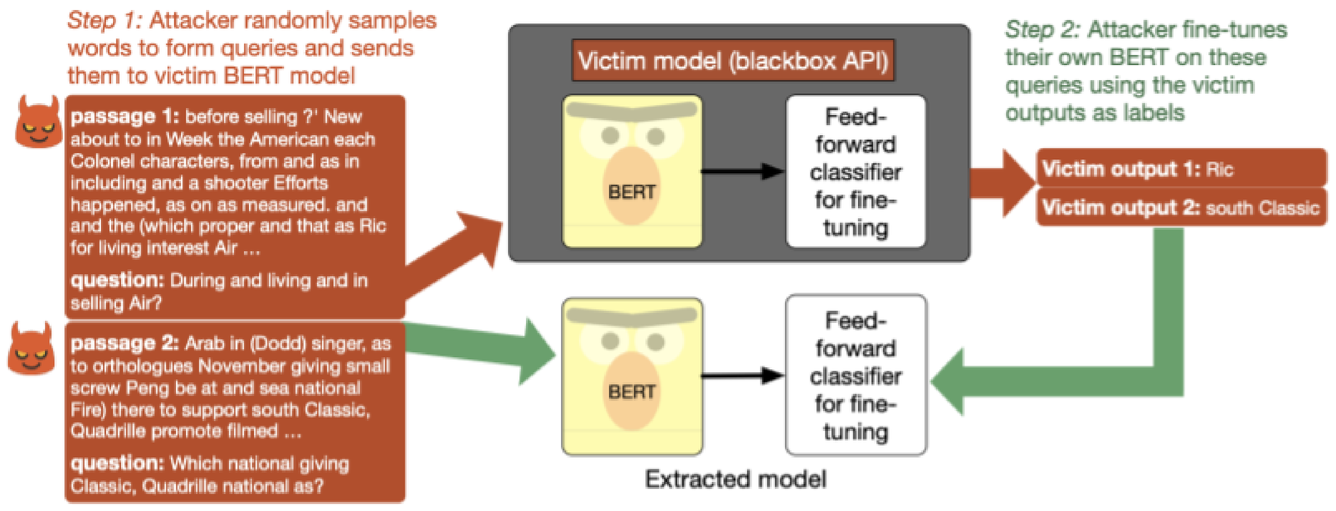

(data record, class label)
predict(data)
(prediction, class label, "in" / "out")
Shadow Training Set 1
Shadow Test Set 1
Shadow Model 1
"in" Prediction Set 1
"out" Prediction Set 1
Shadow training set examples are lablled IN
Shadow test set examples are lablled OUT
Shadow Training Set k
Shadow Test Set k
Shadow Model k
"in" Prediction Set k
"out" Prediction Set k
train()
Attack Training Set
Attack Model
The difference between shadow models and the actual model getting attacked is the training data set. The attacker splits the dataset into two parts and controls which part of the training set the shadow model has seen and which part the model hasn't seen.
You use the information from the previous step to construct the attack model's training set which consists of three things:
1) Shadow model prediction
2) Class label for the above
3) Correct IN/OUT values attack model is trying to predict

This attack basically uses machine learning models to attack another machine learning model. LLMs are also susceptible to this and I've linked some relevant papers in the further reading section.

Model Extraction attack

This is an attack on the model itself where the attacker is trying to steal the machine learning model from the owner. This can be pretty lucrative especially these days where the technical moat of certain $100B companies entirely depend on them having the best machine learning model.

This paper studies the attack in which an adversary with only query access to a victim model attempts to reconstruct a local copy. Assuming that both the adversary and victim model fine-tune a large pretrained language model such as BERT they show that the adversary does not need any real training data to successfully mount the attack.


Step 1: Attacker randomly samples words to form queries and sends them to victim BERT model
passage 1: before selling ?' New about to in Week the American each Colonel characters, from and as in including and a shooter Efforts happened, as on as measured. and and the (which proper and that as Ric for living interest Air ...
question: During and living and in selling Air?
passage 2: Arab in (Dodd) singer, as to orthologues November giving small screw Peng be at and sea national Fire) there to support south Classic, Quadrille promote filmed ...
question: Which national giving Classic, Quadrille national as?
Victim model (blackbox API)
BERT
Feed-forward classifier for fine-tuning
Step 2: Attacker fine-tunes their own BERT on these queries using the victim outputs as labels
Victim output 1: Ric
Victim output 2: south Classic
BERT
Feed-forward classifier for fine-tuning
Extracted model

An attacker first queries a victim BERT model(using random gibberish) and then uses its predicted answers to fine-tune their own BERT model.

In fact, the attacker need not even use grammatical or semantically meaningful queries: they show that random sequences of words coupled with task-specific heuristics form effective queries for model extraction on a diverse set of NLP tasks, including natural language inference and question answering.

Fairwashing

This kind of attack doesn't attack the model itself but targets the explanation methods.It refers to an attack where explanations are used to create the illusion of fairness in machine learning models, even when the models may still be biased or unfair. This term is a play on "whitewashing," implying that something undesirable (in this case, unfairness or bias) is being covered up. This is an attack on the domain of model interoperability where the entire focus of the field is to figure out explanations of model behavior. The attack tries to fool the statistical notion of fairness(like LIME and SHAP) but unfortunately the concepts were a bit too mathematical for for me to explain it here. In this paper, they propose a scaffolding technique that effectively hides the biases of any given classifier by allowing an adversarial entity to craft an arbitrary desired explanation. Apparently their approach can be used scaffold any biased classifier in a manner that its predictions on the inputs remain biased but post hoc explanations come across as fair.

Other attacks on ML models

- You can DoS a ML system by giving it certain sponge examples as part of your input. In this paper they find that you can increase the energy consumption(and thereby latency in responses) by 10x-200x by just crafting certain malicious sponge inputs which exploit certain GPU optimization techniques. This attack is particularly scary in the context of self driving cars. Imagine a sign board with such an example which causes a delay in response leading to life threating accidents.

- You can degrade a model performance by just changing the order in which you present the training data. In this paper they find that an attacker can either prevent the model from learning, or poison it to learn behaviors specified by the attacker. Apparently even a single adversarially-ordered training run can be enough to slow down model learning, or even to reset all of the learning progress.

Conclusion

- While ML systems are just like any other systems and are exploitable, they are extra hard to protect given there are both code vulnerabilities as well as data vulnerabilities.
- Current defenses against adversarial examples are whack-a-mole and real fixes might need massive changes to model development itself rather than pattern matching for attacks. As long as we are pattern matching, these attacks can never be truly prevented. You can't solve AI security problems with more AI
- High stake decisions and mission critical instances should involve human in the loop along with predictions from machine learning models

Further reading:

- LLM security content/research/papers/news
- Survey on practical adversarial examples for malware classifiers
- Blind backdoors in Deep Learning Models
- Hidden trigger backdoor attacks
- Security and Privacy Issues in Deep Learning
- Privacy in federated learning(survey paper)
- Membership inference in masked language models
- Extracting Training Data from Large Language Models
- Fairwashing: the risk of rationalization