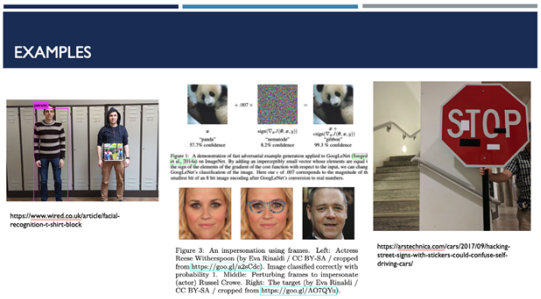
Matthew Todd Farrell, PhD
Secure Algorithms
Sandia National Labs
DHS S&T Brief

Evasion Attacks

Evasion Attacks are cases when an adversary is trying to hide something, or fool, a machine learning system. See reference here: Goodfellow, Shlens, Szegedy, Explaining and Harnessing Adversarial Examples. You can think of evasion attack as similar the little boy from the story, "A Wolf in Sheep's Clothing". The little boy appears as one thing (a wolf), but is actually just a little boy. Below are a few more examples that have appeared in literature and the media.



To make an attack (like those above) we have to assume the following:

1. **Existence:** To perform an evasion attack in practice, there needs to be a model (statistical, machine learning or deep learning based).
2. **Access:** The attacker has to have a way to get data into the model somehow.
3. **Goal:** Get your desired output without raising any alarms. If the attacker is caught then the attack is over, so assume that they tend to hide their existence by making small changes to the input.

Some intuition about the properties above are that the model will almost certainly have to exist if the attacker is going to be successful. Another important property of that same model is that it is fixed. By fixed in this case we mean that the attacker cannot, in any meaningful way, modify the training input or any other property of the model. The attacker has the freedom to do any of the following:

- Perform multiple queries on the model; they can query the model many times to collect the responses from the model to refine better evasion attempts.
- Depending on the situation, they can have white-box access or black-box access.
    - In the *black-box case* they can only observe the inputs and outputs of the model.
    - In the *white-box case* they observe all training data, parameters, and architecture.
    - In the *grey-box case* they can observe some subset of training data, parameters, and architecture.

There are typically two types of evasions – ones that are digital and others that are physical. You can think of physical attacks when an attacker attaches something (duct tape, a placard, etc) and

it tricks the model into classifying the object incorrectly. An example of this kind of attack would be the "stop sign" in the figure above. There the researchers attached duct-tape to the stop sign and caused it to convert a "stop sign" into a speed limit sign as judged by a neural network. Digital attacks are typically seen in face-detection algorithms when someone is trying to change their identity. These attacks, as seen above with the Reese Witherspoon example, take place by doctoring an image to fool a model into believing someone is Russell Crowe.

The takeaway here for the two types are:

1. manipulate digital input to the model, such as changing pixel values in a jpeg or png image before it is sent to the model for evaluation, or
2. manipulate physical attributes in the real world to cause an ML system to alter its expected behavior in some way.

LLMs and Foundational models are a relatively new attack surface. To my knowledge, there haven't been large public disclosures of attacks with LLMs being the target of an evasion attack. That said, they will be important as they get integrated, and folks figure out use-cases across the Government. OpenAI does maintain a set of usage-policies that indicate what attacks they might consider to be worthwhile (essentially ways around any security introduced by OpenAI to prevent these use-cases): https://openai.com/policies/usage-policies

Evasion Defenses

There are some defenses, and more being made all the time. For an overview of some current methods see: Yuan, et al., Adversarial Examples: Attacks and Defenses for Deep Learning – Secs. VI. The caveat to these is that sometimes they don't transfer well between different types of models and use-cases. Further, some are just basic cybersecurity defenses. A short list would be

- Restrict the number of queries on a model (api request limits)
- Network Distillation
- Adversarial Detection
- Adversarial (Re)Training

Aside from restricting queries to the API the other approaches are more data-oriented. The idea of Adversarial (Re)Training is to inject adversarial examples into the training process. This procedure in theory should allow the model to handle the perturbations on inputs but still classify correctly.

Defensive Distillation (https://arxiv.org/abs/1511.04508) tries to prevent a model from fitting too tightly to the data by using probabilities versus hard class labels. The idea is to train your neural network as usual and then train a second model that is trained from the probabilities of the first one.

Additional methods could be Feature Squeezing, where you reduce the degrees of freedom to construct adversarial examples by squeezing out unnecessary input features. If the distance is larger than a threshold, then the input sample is an adversarial example. See additional methods and references here: Carlini, Wagner, Adversarial Examples Are Not Easily Detected: Bypassing

Ten Detection Methods. 2017 and Tramer, Carlini, Brendel, Madry, On Adaptive Attacks to Adversarial Example Defenses. 2020.

Conclusion

The key things to note are that the attacker in an evasion attack is trying to make as small as possible a change to the data while causing the model to misclassify on the modified data. There are two major types of attacks; digital and physical. In general, there are defenses that are always being made, but they can be evaded very quickly. See Carlini, Wagner, Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods for more information.