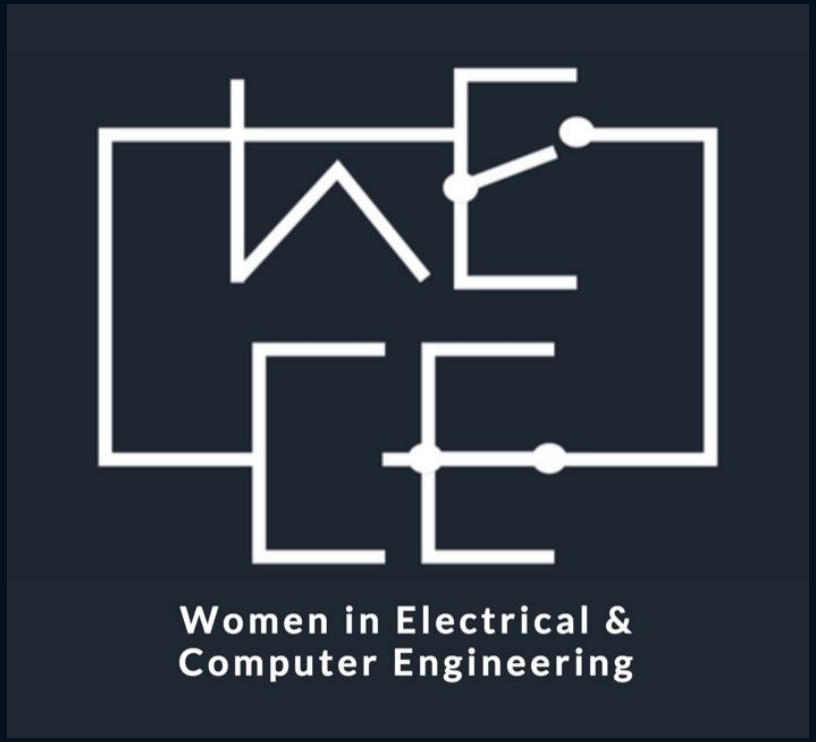


Machine Learning Workshop

On Hacking Machine Learning

Machine Learning - Can it be hacked?

Machine Learning - Can it be hacked?

Yes!

Machine Learning - Can it be hacked?

Short answer: Yes!

Long answer: Yes, but there are various attack vectors and a broad range of definitions for what it really means to “hack” an ML model

Why do we care? *(Reasons to hack)*

- Clearview AI: facial recognition company, gathered training data through web scraping, unsuspecting individuals at apartment buildings, etc.
- Facial recognition being used to identify protestors this summer, likely at future protests
- Facial recognition being used on college campuses, proctoring software

Those are the things we know about - but what about the ones we don’t yet know?

Why do we care? *(Reasons to secure)*

- Autonomous vehicles or drones: we want cars to drive safely, drones to deliver our packages as expected

- Facial recognition could be used to verify things such as financial transactions, do not want people to easily hack in such applications

- We don’t want potentially-sensitive training data, such as medical data, to be leaked

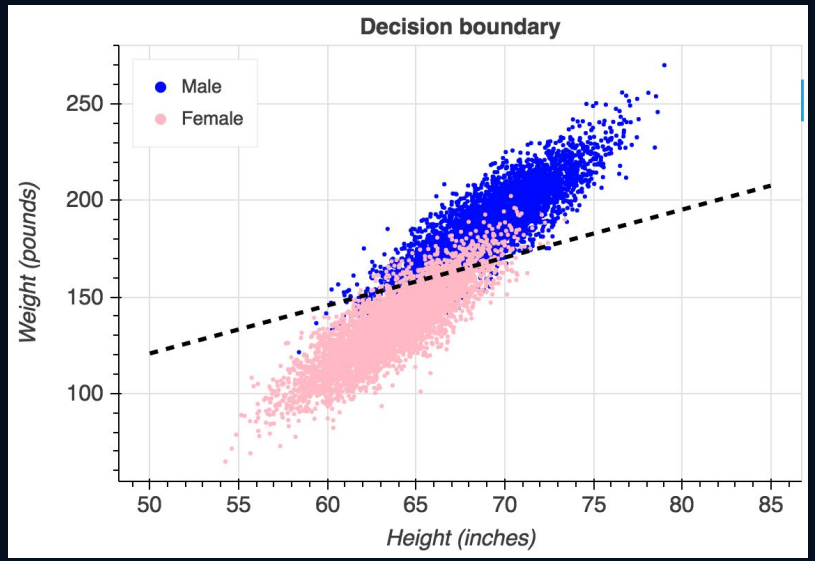
A Broad Dichotomy of ML Hacks

- Blackbox vs. whitebox attacks
- Data-related attack vectors
  - Data poisoning
  - Adversarial inputs
- Fundamental model problems
  - Model recreation
  - Adversarial inputs: a feature, not a bug

First - *how does classification work?*

Training data goes into model → Weights are determined

- Weights are then used to classify another dataset such as a “validation” set or real data



Source:
https://towardsdatascience.com/understanding-logistic-regression-step-by-step-704a78be7e0a

First - *how does classification work?*

Weights are then used to classify another dataset such as a “validation” set or real data

- So, accuracy of classification can rely on many factors
  - Quality of data input
  - How real-world or test data compares to training inputs
  - Model architecture: number of layers, layer types, loss function, etc.
  - Training techniques such as no technique vs. SGD vs. ADAM
  - Hyperparameter tuning: learning rate, epoch size, number batches, etc.

Second - *how does object detection work?*

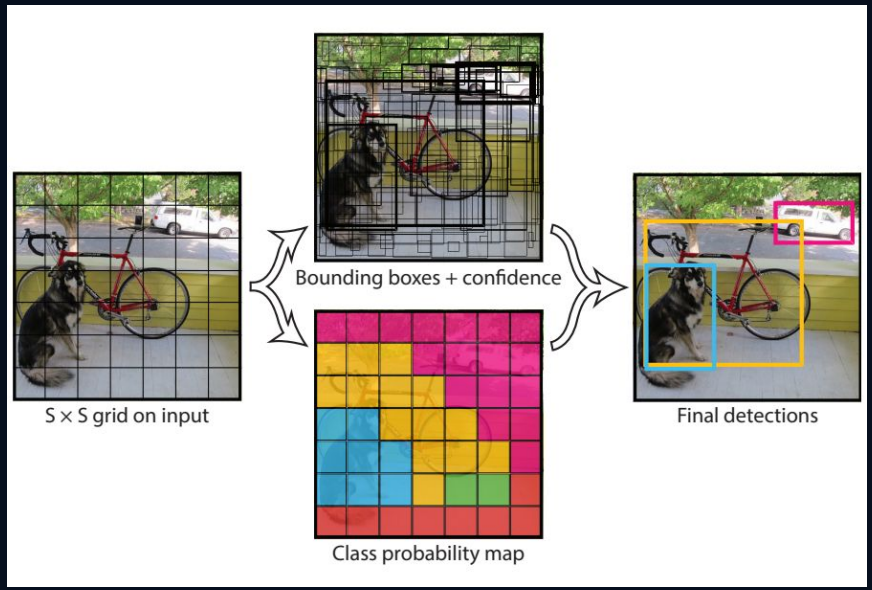
Object detection is a bit harder than classification!

- The same overall idea holds for classification with regards to things that affect quality of output
- However, typically requires techniques such as deep learning to prove effective (whereas a binary or even 10-class classifier performs pretty well without it)
- Whereas classifiers always know inputs can only come from a fixed class, eg. cats or dogs, object detection frameworks must decide whether or not an object exists in the frame in addition to classifying it

Second - *how does object detection work?*

Many different object detection frameworks out there with different techniques for getting the job done.

Convolutional neural nets (CNNs) are pretty common.



Source: https://pjreddie.com/media/files/papers/yolo_1.pdf

Blackbox and Whitebox Attacks

Blackbox: underlying model weights are not known

Whitebox: underlying model weights ARE known

- Knowing the underlying model makes attacking things easier
- But, with enough queries, blackbox models can easily become whitebox…

Example: can be done on NLP models even with random word queries, see https://arxiv.org/pdf/1910.12366.pdf

Data Poisoning

Generally, weights are determined by training input.

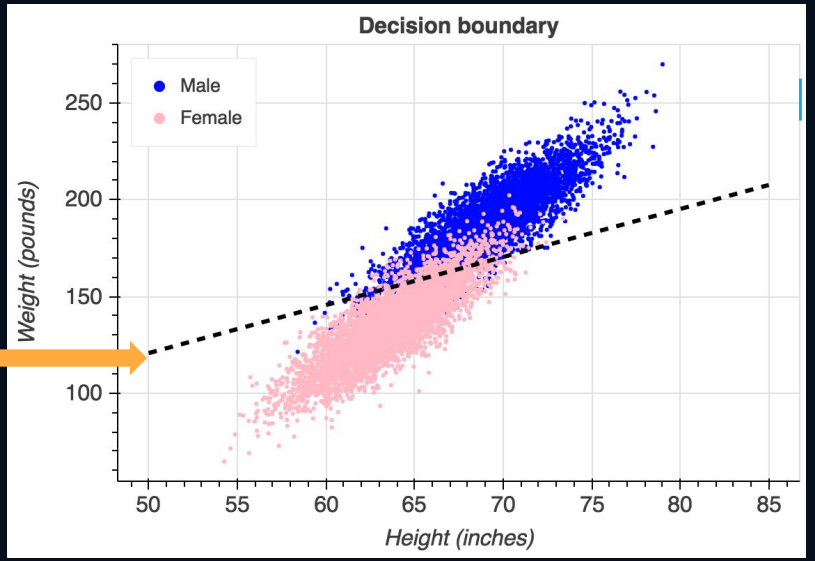
So, the output of models could be significantly altered depending on what data is used.

Data poisoning seeks to intentionally train with specific data that changes the model weights, possibly to generate a known backdoor classification

Data Poisoning

In other words, attempt to shift the decision boundary line by injecting malicious input.

(make this line move somewhere else)



Data Poisoning

Not necessarily something new: likely began in early 2000s with evading spam filters

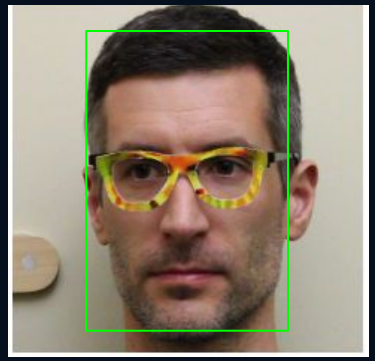
See also: Tay, Microsoft’s twitter bot who started out nice, ended up with… questionable tweets

Adversarial Inputs

But, to poison a dataset, we may need adversarial inputs to help us do so!

- Generally, any kind of intentionally confusing input
- Many fun visual examples
- Can either be generated with low-tech methods or direct attacks on whitebox models

Adversarial Inputs: can lead to interesting fashion



Classified as

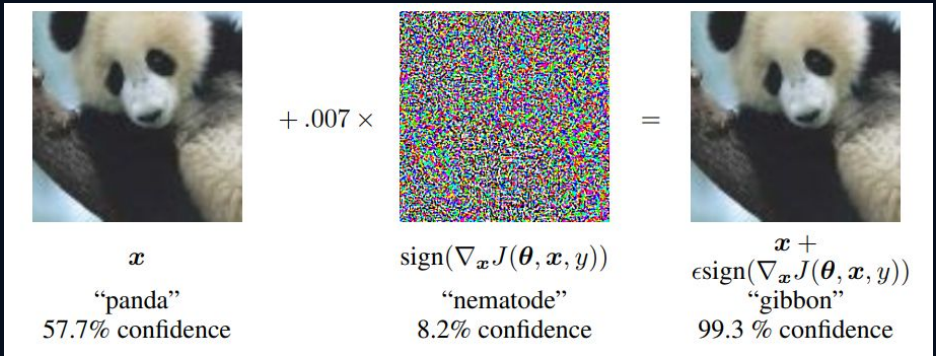
Source: https://users.ece.cmu.edu/~lbauer/papers/2016/ccs2016-face-recognition.pdf

Source:
https://www.technologyreview.com/2019/08/15/65421/a-new-clothing-line-confuses-automated-license-plate-readers/

Adversarial Inputs

How do we generate adversarial inputs?

- For facial recognition: could wear the right masks/hats/sunglasses, IR lighting
- More generally: apply whitebox attacks such as Fast Gradient Sign Attack (FGSM)
  - FGSM is possibly best-known attack, but definitely not the only one!



Source: https://arxiv.org/pdf/1412.6572.pdf

Source: https://arxiv.org/pdf/1707.08945.pdf

Possible Solutions

- Try to detect noisy/adversarial inputs
- Simulate attacks yourself to try and fix them
    - Many packages which can help you do this, such as Adversarial Robustness Toolkit!
- Detect potential backdoors
- Adversarial training
    - This means training on adversarial examples to improve robustness

Interested in learning more about this topic?

Since this is just a 1-hour workshop, we glossed over a lot of fundamentals...

- Start with ML/AI coursework
  - At UIUC: CS/ECE 440 Artificial Intelligence, CS 446 Machine Learning, CS 498 Applied Machine Learning, CS 498 Deep Learning, CS 498 Trustworthy Machine Learning
  - Outside of UIUC: Andrew Ng's Machine Learning Coursera, Yann LeCun's Deep Learning coursework notes, insert other famous ML people here, I'm sure they all offer something

- Get involved in ML/AI projects
  - Could do internships in the area
  - Could do research
  - Could read papers at top conferences such as NeurIPS, CVPR
  - Could read Medium blog posts