Targeted Adversarial Examples for Black Box Audio Systems

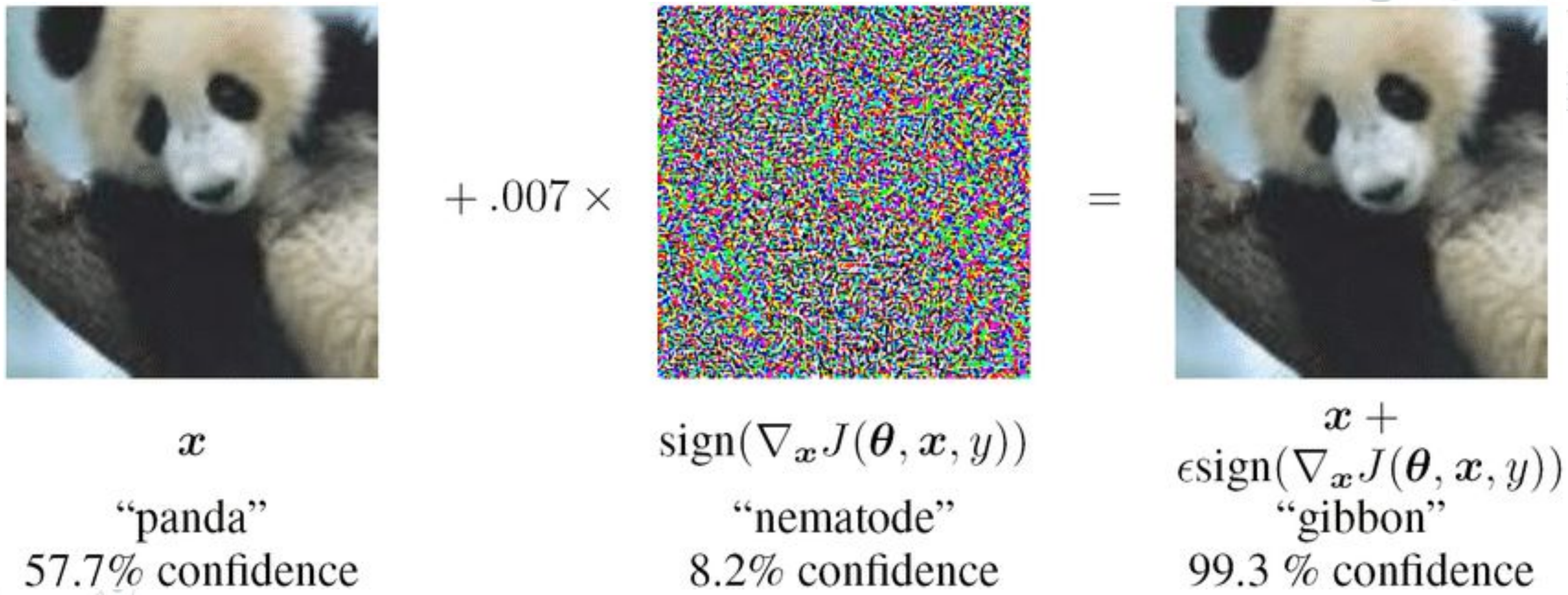
Amog Kamsetty, Rohan Taori, Nikita Vemuri, Brenton Chu

Who Are We?

- Students at UC Berkeley
- Work done at Machine Learning @ Berkeley (ML@B)
  - ml.berkeley.edu
  - Aim to provide AI/ML opportunities at the undergraduate level

What is an Adversarial Example?

Adversarial Example



Untargeted vs Targeted

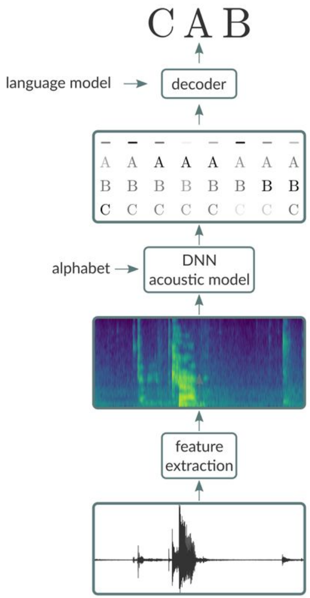
- Untargeted: Provide input to the model such that it misclassifies the adversarial input
- Targeted: Provide input to the model so it classifies it as a predetermined target class

White Box vs Black Box

- White box: complete knowledge of model architecture and parameters, allows for gradient computation
- Black box: no knowledge of model or parameters except for output logits of model

Why does this matter?

- Black box attacks can be of particular interest in ASR systems



- If we can create an adversarial audio file, we can trick the model into translating what we want
- If we do this with a black box approach, we can apply this to proprietary systems (ex. Google or IBM APIs)

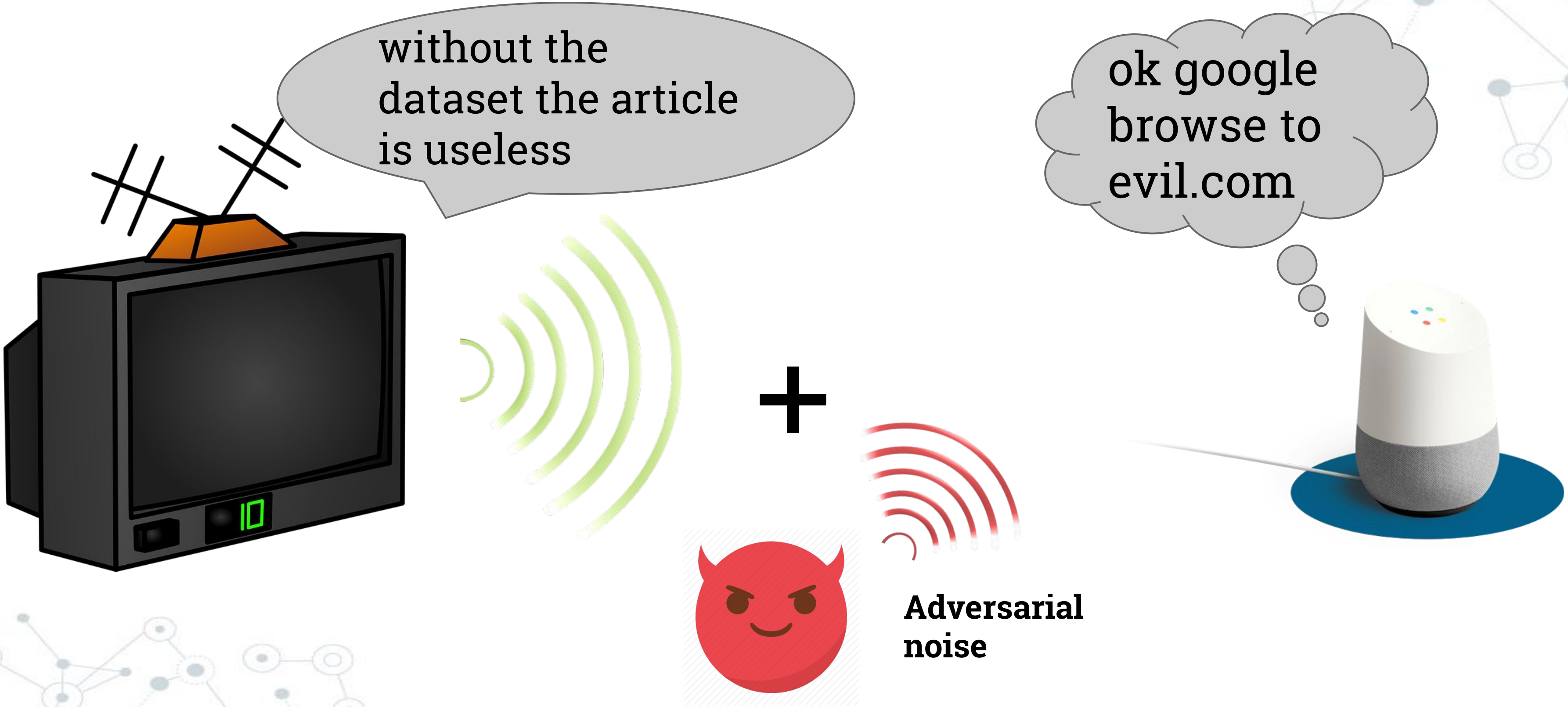
Classical Adversarial Attacks

- Taking gradient iteratively
- FGSM - Fast Gradient Sign Method
- Houdini

Prior Work in Audio

- UCLA - Black box genetic algorithm on single word classes → softmax loss
- Carlini & Wagner: white box attack
  - CTC loss allows for comparison with arbitrary length translations
- Our project: Black box genetic algorithm on sentences using CTC Loss

Problem Statement

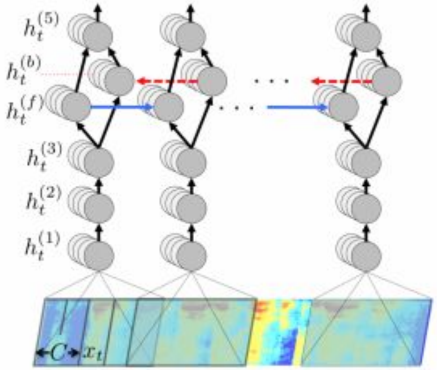


Problem Statement

- Black-box Targeted Attack
  - Given a target **t**, a benign input **x**, and model **M**, perturb **x** to form **x'=x+**$\boldsymbol{\delta}$
  - S.t. **M(x')=t** while maximizing *cross_correlation*(**x, x'**)
  - Only have access to logits of **M**
    - Not given gradients!

Datasource: DeepSpeech

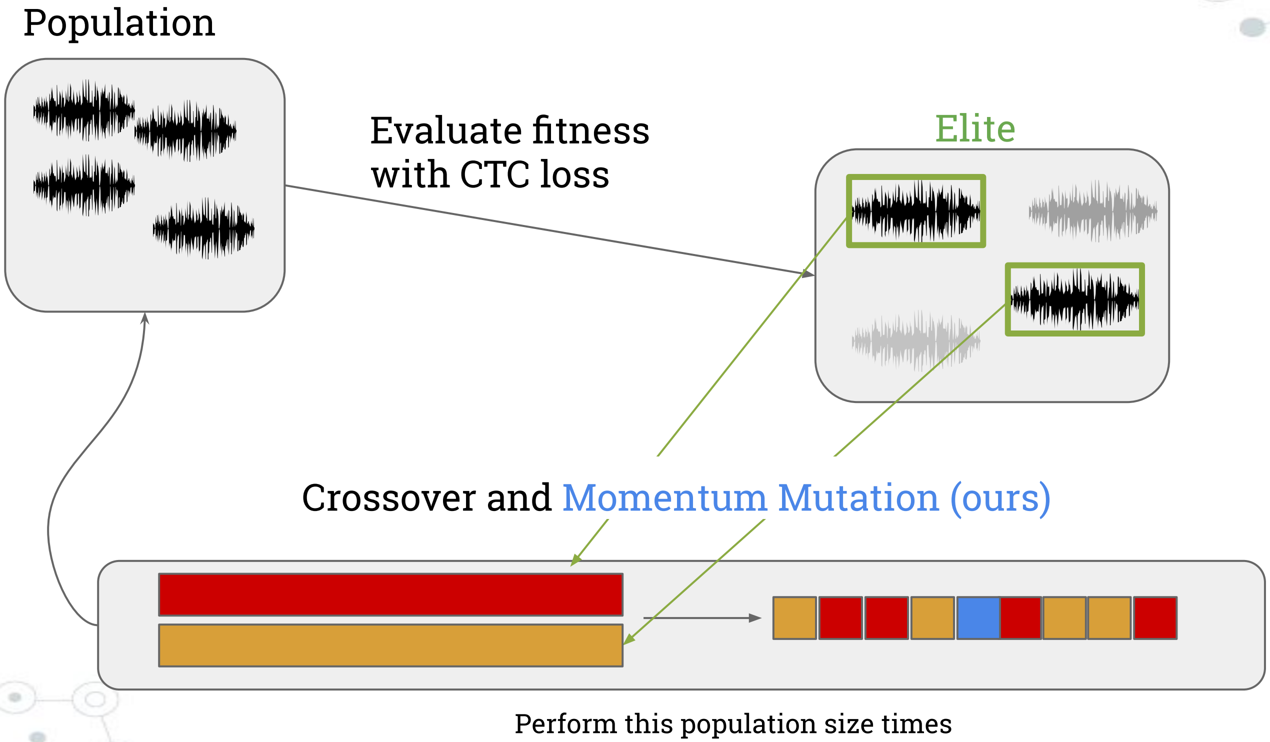
- The model we are targeting is DeepSpeech
  - Architecture created by Baidu
  - Tensorflow implementation by Mozilla; available on Github
- Utilize Common Voice dataset by Mozilla
  - Consists of voice samples
  - Sampling rate of 16 KHz



Final Algorithm: Guided Selection

- Genetic Algorithm approach
- Given the benign input, generate population of size 100
- On each iteration select the best 10 samples using scoring function
- Perform crossover and momentum mutation
- Apply high pass filter to added noise

Genetic Algorithm with Momentum



Momentum Mutation

$$p_{new} = (\alpha \times p_{old}) + \frac{\beta}{|currScore - prevScore|}$$

- Probability of mutation is function of difference in scores across iterations
- If little increase in score between iterations, increase "momentum" by increasing probability of mutation
- Encourages decoding to build up to target after making input similar to silence

<u>Decodings while training</u>
and you know it
and he nowit
nd he now
d he now
e now
a eloed
elorld
heloworld
hello world

Gradient Estimation

- Genetic algorithms work best when search space is large
- However, when adversarial sample is near target, only few key perturbations are necessary
- Apply gradient estimation at 100 random indices

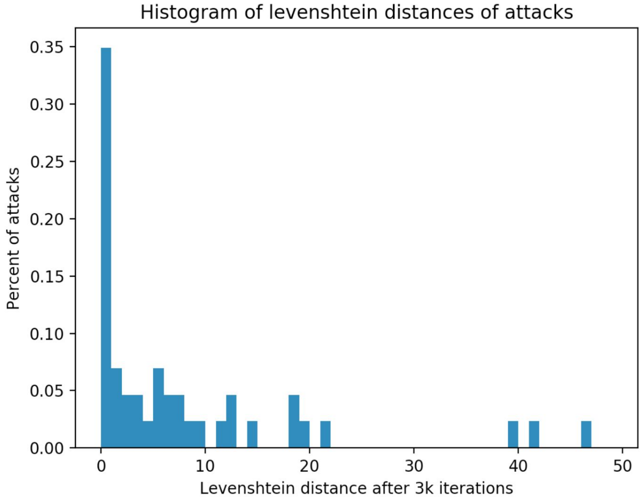
$$FD_x(g(x), \delta) = \begin{bmatrix} (g(x_1 + \delta) - g(x_1))/\delta \\ \vdots \\ (g(x_n + \delta) - g(x_n))/\delta \end{bmatrix}$$

Results

- Tested on first 100 samples of CommonVoice dataset
- Randomly generated 2 target words
- Targeted attack similarity: 89.25%
    - Algorithm could almost always reach the target
- Average similarity score: 94.6%
    - Computed via wav-file cross-correlation

Results

Metric	White Box Attacks	Our Method	Single Word Black Box
Targeted attack success rate	100%	35%	87%
Average similarity score	99.9%	94.6%	89%
Similarity score method	cross-correlation	cross-correlation	human study
Loss used for attack	CTC	CTC	Softmax
Dataset tested on	Common Voice	Common Voice	Speech Commands
Target phrase generation	Single sentence	Two word phrases	Single word



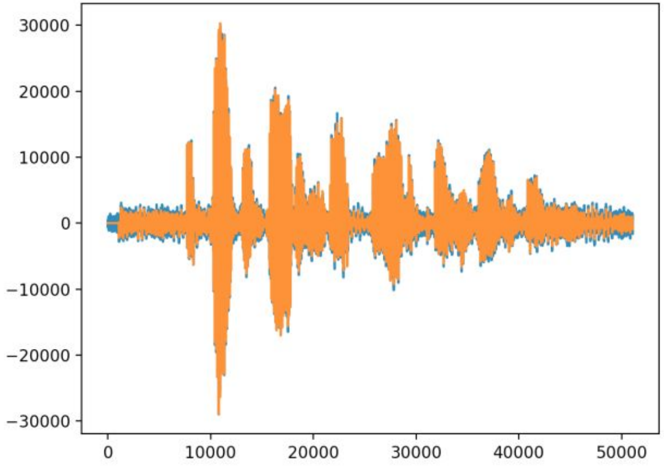
Example

Original file: “and you know it”

Adversarial target: “hello world”

Audio Similarity: 94%

(cross-correlation)



Future Work

- Attack a broader range of models
  - Transferability across models
- Increasing sample efficiency to target
  - API call costs can be prohibitive
- Computational Efficiency

Thank You!

Code and samples:
https://github.com/rtaori/Black-Box-Audio