



Audio Adversarial Examples: Targeted Attacks on Speech-to-Text

Nicholas Carlini, David Wagner



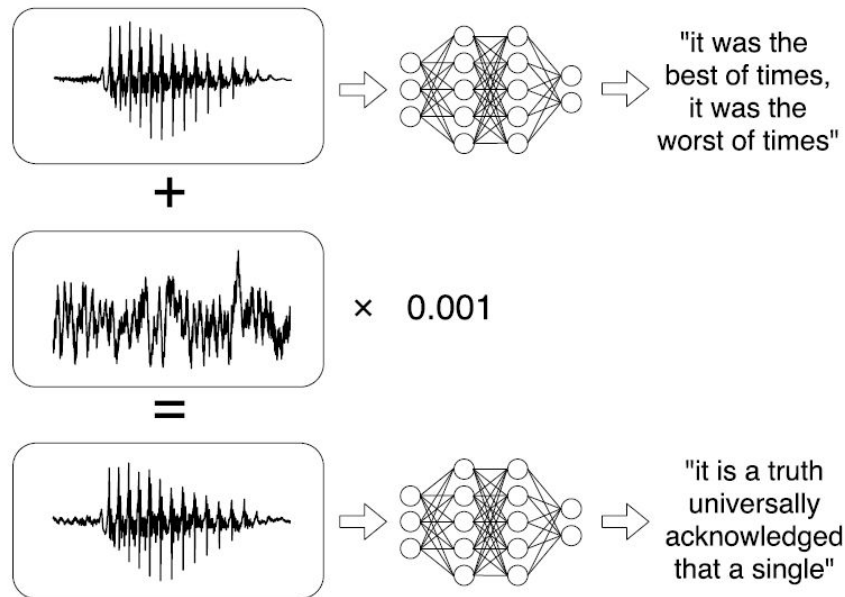
Background

For automatic speech recognition (ASR), a neural network is given an audio waveform \mathbf{x} and performs the speech-to-text transform which produces the transcription \mathbf{y} of the phrase being spoken (as used in, e.g., Apple Siri, Google Now, and Amazon Echo).

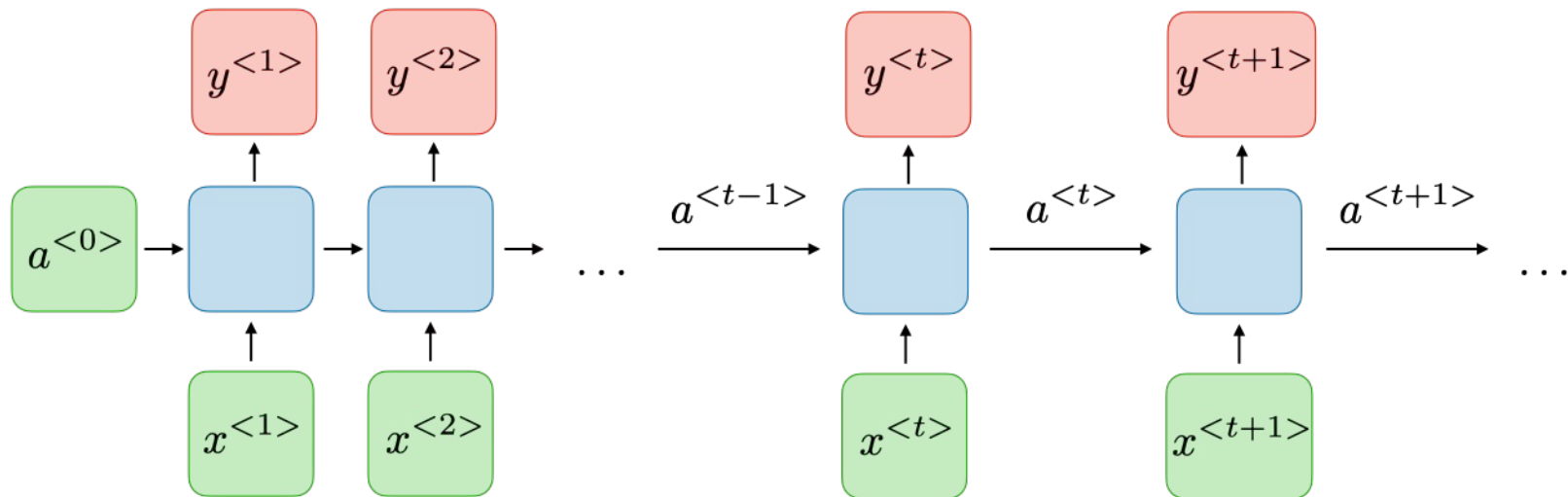
Targeted attacks on Speech-to-Text: Given any natural waveform \mathbf{x} , constructing a perturbation δ that is nearly inaudible, but so that $\mathbf{x}+\delta$ is recognized as any desired phrase \mathbf{p} .

Contribution

An end-to-end optimization-based attacks on DeepSpeech, a state-of-the-art speech-to-text transcription neural network.



DeepSpeech? RNN!





Notations

Let \mathcal{X} be the input domain, \mathcal{Y} be the range (the characters a-z, space, and the special token ϵ). f takes a sequence of N frames and returns a probability distribution over the output domain for each frame.

$$f : \mathcal{X}^N \rightarrow [0, 1]^{N \cdot |\mathcal{Y}|}$$



“Reduce to”

A string π reduces to p if making the following two operations (in order) on π yields p :

- 1) Remove all sequentially duplicated tokens.
- 2) Remove all ϵ tokens.

Ex: $aab\epsilon\epsilon b$ reduces to abb .



“Alignment”

We say that π is an alignment of \mathbf{p} with respect to \mathbf{y} (a sequence output of probability distributions) if (a) π reduces to \mathbf{p} , and (b) the length of π is equal to the length of \mathbf{y} . (Denoted as $\pi \in \Pi(\mathbf{p}, \mathbf{y})$)

The probability of alignment under \mathbf{y} is the product of the likelihoods of each of its elements:

$$\Pr(\pi | \mathbf{y}) = \prod_i y_{\pi^i}^i$$



Connectionist Temporal Classification (CTC) Loss

Now we can define the probability of a given phrase \mathbf{p} under the distribution \mathbf{y} :

$$\Pr(\mathbf{p}|\mathbf{y}) = \sum_{\pi \in \Pi(\mathbf{p}, \mathbf{y})} \Pr(\pi|\mathbf{y}) = \sum_{\pi \in \Pi(\mathbf{p}, \mathbf{y})} \prod_i \mathbf{y}_{\pi^i}^i$$

Therefore, a CTC-Loss

$$\text{CTC-Loss}(f(\mathbf{x}), \mathbf{p}) = -\log \Pr(\mathbf{p}|f(\mathbf{x})).$$



To decode a phrase from the distribution output

To find:

$$C(\mathbf{x}) = \arg \max_{\mathbf{p}} \Pr(\mathbf{p} | f(\mathbf{x})).$$

We can use greedy decoding:

$$C_{\text{greedy}}(\mathbf{x}) = \text{reduce}(\arg \max_{\pi} \Pr(\pi | f(\mathbf{x})))$$

Or Beam Search Decoding.



Attack Formulation

Measure the distortion in Decibels (dB).

Use CTC-Loss as l .

c is a hyperparameter.

t is the target phrase.

x : the input waveform

δ : perturbation

Gradually lowering τ to improve upon suboptimal perturbation.

$$\begin{aligned} &\text{minimize } \|\delta\|_2^2 + c \cdot \ell(x + \delta, t) \\ &\text{such that } dB_x(\delta) \leq \tau \end{aligned}$$



Improved Loss

The previous loss (CTC Loss) tends to make all aspects of the transcribed phrase more similar to the target phrase. But if we already get “ABCX” for target phrase “ABCD”, we just want X to become D and not making A more “A”-like.

$$\ell(y, t) = \max \left(y_t - \max_{t' \neq t} y_{t'}, 0 \right).$$

$$L(\mathbf{x}, \pi) = \sum_i \ell(f(\mathbf{x})^i, \pi_i).$$



Attack Formulation with Improved Loss

Optimize to multiple alignments with the new loss.

To get the alignment: Use the previous formulation, get the corresponding alignment, and then use the new loss to generate smaller perturbations.

$$\text{minimize } |\delta|_2^2 + \sum_i c_i \cdot L_i(x + \delta, \pi_i)$$

such that $dB_x(\delta) < \tau$

Evaluations

1. Waveform visualization.
2. Starting from Non-Speech: A mean of -20 dB to turn classical music clips into meaningful phrases.
3. Targeting Silence: Less than -45dB to turn any phrase into silence.
4. Hear it yourself:
https://nicholas.carlini.com/code/audio_adversarial_examples



Figure 2. Original waveform (blue, thick line) with adversarial waveform (orange, thin line) overlaid; it is nearly impossible to notice a difference. The audio waveform was chosen randomly from the attacks generated and is 500 samples long.



Robustness

1. 10dB larger distortion to survive pointwise noise:
 - a. Adding pointwise random noise easily destroys the adversarial label.
 - b. Use *Expectation over Transforms*.
2. 15dB larger distortion to survive MP3 compression.
 - a. Use a straight-through estimator for function MP3().




Some thoughts and follow-ups

1. Future work mentioned in the paper:
 - a. Can attacks be played over-the air? Yes. (ICML 19)
 - b. Do universal adversarial perturbations exist? Yes. (ICASSP 20)
 - c. Are audio adversarial examples transferable? Maybe.
 - d. Which existing defenses can be applied to audio? Down-sample and up-sample. (See USENIX Security 21.)



Some more thoughts

2. How hard is it to mount a black box attack? How hard is it to mount an attack on an actual voice assistant?
3. How much threat this actually is?
4. A better understanding of human perception to examples.



CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition

Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, XiaoFeng Wang, and Carl A. Gunter



Motivation

Embed a set of commands into a (randomly selected) song, to spread to a large amount of audience (For example, through YouTube).

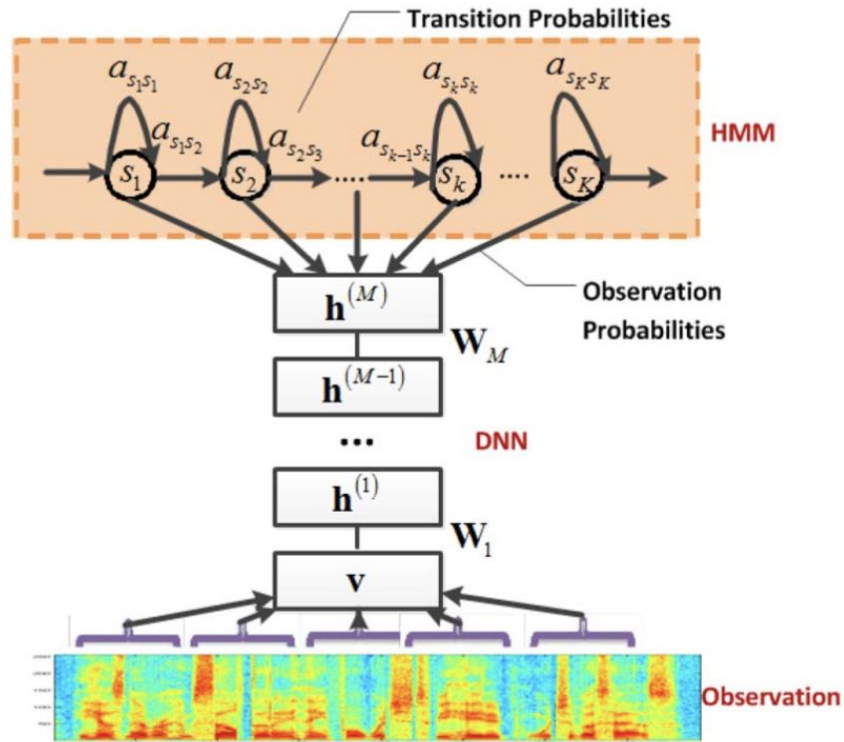
This revised song, which they call CommanderSong, can sound completely normal to ordinary users, but will be interpreted as commands by ASR, leading to the attacks on real-world Intelligent voice control (IVC) devices.

Overview

Previous paper targets an RNN-based ASR.

This paper targets a Deep Neural Network-Hidden Markov Model (DNN-HMM) based ASR (Called Kaldi).

(You might be more familiar with GMM-HMM.)

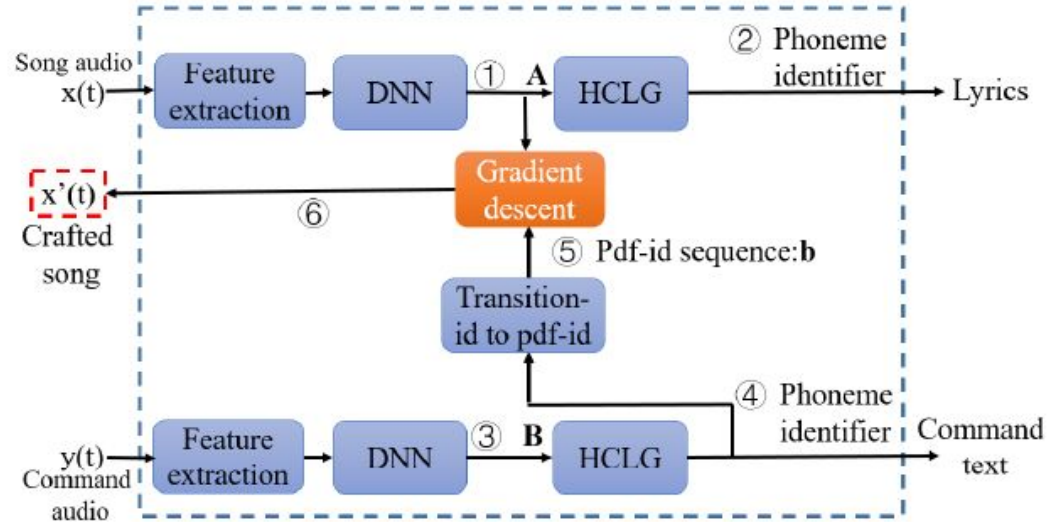


Overview

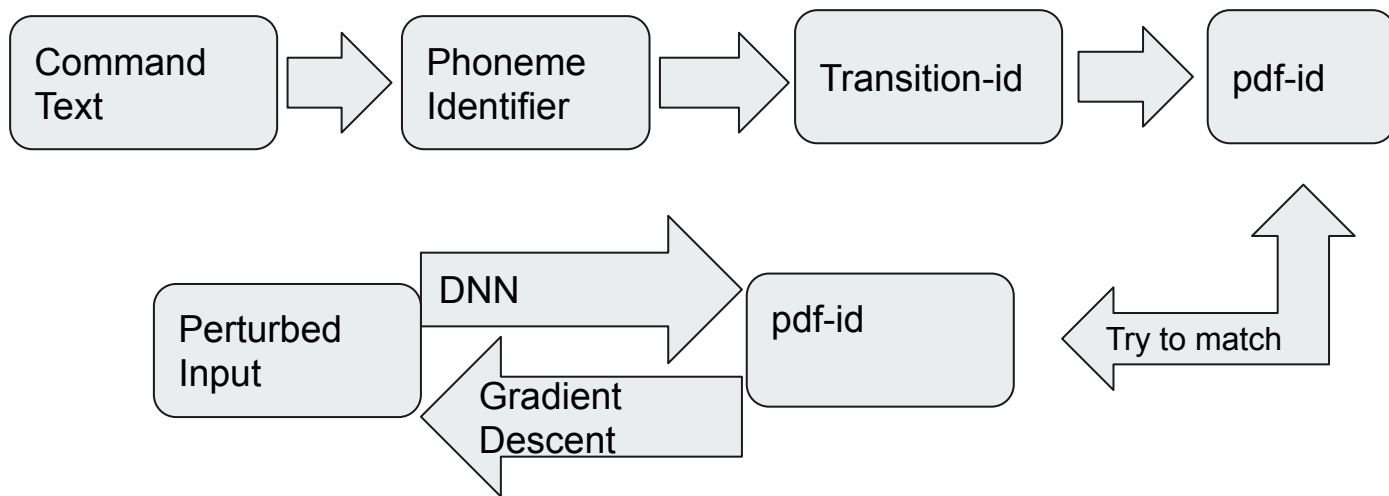
Acoustic features like MFCC are extracted from the raw audio.

Those features are taken as input to DNN to compute the posterior probability matrix. The matrix is indexed by the pdf identifier (pdf-id), which exactly indicates the column of the output matrix of DNN.

A transition identifier (transition-id) is used to uniquely identify the HMM state transition. A sequence of transition-ids can identify a phoneme (which is equivalent to a phoneme identifier).



In short





WAV-To-API (WTA) attack

$$\arg \min_{\delta(t)} \|g(x(t) + \delta(t)) - \mathbf{b}\|_1.$$

$g()$ produces a list of most likely pdf-ids of the original song audio $x(t)$. \mathbf{b} is a sequence of pdf-ids of the command we want.

This is under the constraint of $|\delta(t)| \leq l$



WAV-Air-API (WAA) Attack

$$\arg \min_{\mu(t)} \|g(x(t) + \mu(t) + n(t)) - \mathbf{b}\|_1,$$

$n(t)$ is the noise captured for simulating real-world distortions and noise over the air. Such noise is obtained by playing multiple songs over the air and record them, and then take the difference.



Evaluations

1. WTA: Generate 200 songs based on 26 songs. The success rate 100%. The average signal-noise ratio (SNR, the larger the value the smaller the perturbation) ranges from 14~18.6 dB, indicating that the perturbation in the original song is less than 4%.
2. WAA: Success rate ranges from 60-90% with all SNRs around 1.5.
3. Hear it: <https://sites.google.com/view/commandersong/>



Human Perception

Table 4: Human comprehension of the WTA samples.

Music Classification	Listened (%)	Abnormal (%)	Recognize Command (%)
Soft Music	13	15	0
Rock	33	28	0
Popular	32	26	0
Rap	41	23	0

Table 5: Human comprehension of the WAA samples.

Song Name	Listened (%)	Abnormal (%)	Noise-speaker (%)	Noise-song (%)
Did You Need It	15	67	42	1
Outlaw of Love	11	63	36	2
The Saltwater Room	27	67	39	3
Sleepwalker	13	67	41	0
Underneath	13	68	45	3
Feeling Good	38	59	36	4
<i>Average</i>	19.5	65.2	40	2.2



Transferability

1. Kaldi to iFLYTEK

Table 6: Transferability from Kaldi to iFLYTEK.

Command	iFLYREC (%)	iFLYTEK Input (%)
Airplane mode on.	66	0
Open the door.	100	100
Good night.	100	100

2. Kaldi to DeepSpeech:

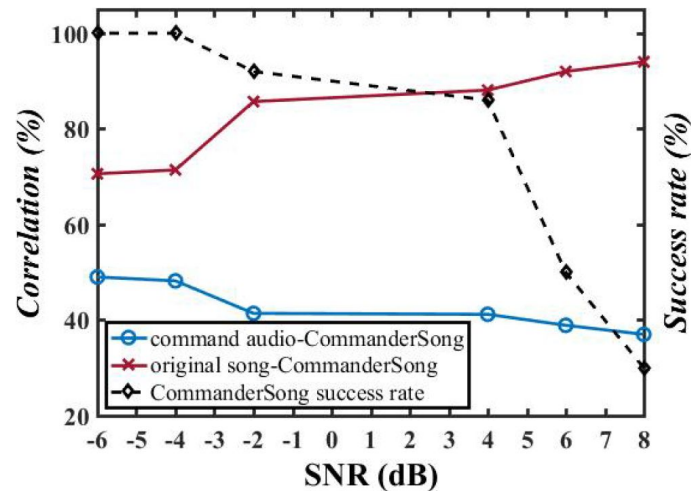
WAA/WTA on DeepSpeech (✗)

But

C&W on DeepSpeech + WAA/WTA work on both (✓)

Findings

1. A song helps to generate the target sequences by providing some phonemes or even smaller units. This implies that a better selection of songs will make the attack easier.
2. For WAA, the stronger the noise model, the less similar CommanderSong sounds like the original one, and more similar with the actual command.



Defenses

1. Audio Turbulence

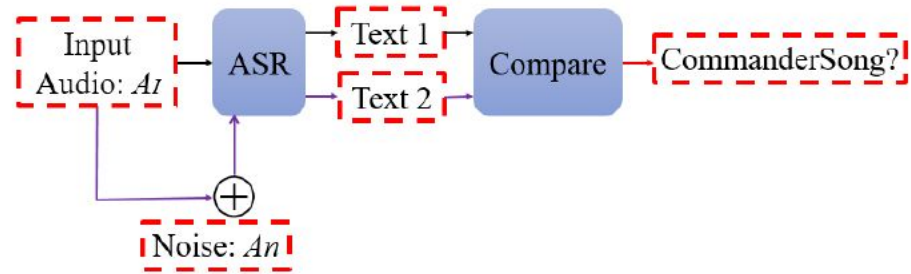


Figure 6: Audio turbulence defense.

2. Audio Squeezing (Down-sampling), i.e. Change the sampling rate. The success rates of WTA and WAA are 0% and 8% respectively when down-sample ratio is 0.7.



Some thoughts

1. The threat model is specific and very practical.
2. But the quality is unfortunately not that good. However, the followup work intends to improve upon this.
3. Is it possible to construct perturbations that transfer between different architectures?
4. Is there an ASR system that is tuned to recognize song lyrics? Or a system that is tuned to discard unrelated noise. (i.e. how can we make an ASR system more robust without relying on humans' perception. Can we do adversarial training on ASR?)