Adversarial example devastation and detection on speech recognition system by adding random noise

Mingyu Dong, Diqun Yan∗, Rangding Wang

College of Information Science and Engineering, Ningbo University, Ningbo Zhejiang, China

Abstract

An automatic speech recognition (ASR) system based on a deep neural network is vulnerable to attack by an adversarial example, especially if the command-dependent ASR fails. A defense method against adversarial examples is proposed to improve the robustness and security of the ASR system. We propose an algorithm of devastation and detection on adversarial examples that can attack current advanced ASR systems. We choose an advanced text- and command-dependent ASR system as our target, generating adversarial examples by an optimization-based attack on text-dependent ASR and the GA-based algorithm on command-dependent ASR. The method is based on input transformation of adversarial examples. Different random intensities and kinds of noise are added to adversarial examples to devastate the perturbation previously added to normal examples. Experimental results show that the method performs well. For the devastation of examples, the original speech similarity after adding noise can reach 99.68%, the similarity of adversarial examples can reach zero, and the detection rate of adversarial examples can reach 94%.

Keywords: Automatic speech recognition (ASR), Devastation on adversarial example, Detection on adversarial example, Random noise

∗Corresponding author

Email address: yandiqu@nbu.edu.cn (Diqun Yan)
1. Introduction

Deep neural network technology has been used in many fields [1, 2], and related security problems have become increasingly prominent, among which the adversarial example [4] is of great concern. In 2014, Szegedy et al. [5] found that the deep neural network (DNN) showed high vulnerability to image examples with specific perturbation, including adversarial perturbation. Their study is of great significance to explain the principle of deep learning, and has promoted the development of security attack and defense based on deep learning.

Automatic speech recognition (ASR) [3] has been used for intelligent speech assistance and vehicle speech control systems, helping users control and connect to services through simple speech. These systems are vulnerable to attack by adversarial examples. Vaidya et al. [6] first proposed a method to generate speech adversarial examples. The ASR system recognized errors when it adjusted the parameters extracted by Mel-frequency cepstral coefficients (MFCCs) [7]. Carlini et al. [8] extended the work to hide malicious commands in speech, gave a more detailed description and analysis of the scene of speech adversarial examples, and gave methods of white- and black-box attacks. Alzantot et al. [9] applied genetic algorithms to black-box attacks and successfully attacked command-dependent ASR systems. Yuan et al. [10] hid speech commands in music. Cisse et al. [11] proposed a more flexible attack method, which can be applied to different models. To attack the end-to-end ASR model, the method required the loss of the target command and the current prediction result, and found an adversarial example through optimization. Iter et al. [12] generated adversarial examples based on the trained WaveNet [13] model. This method is mainly based on the fast gradient sign algorithm (FGSM) [14].

Some work [8, 9] is aimed at command-dependent ASR systems, causing speech to be misclassified. However, the application is text-based in most practical scenes. Because the generated text is indefinite in length, there will be some problems calculating the loss between the output and target texts. Carlini [15] introduced CTC loss [16] to the adversarial example of speech recognition to solve this problem. Speech adversarial example research has also extended from command- to text-dependent speech, including the following work. Carlini [15] solved the problem of gradient backpropagation in the MFCC computing process, because many ASR systems do not directly accept original speech, and need to extract its MFCC coefficients. The method
can use any original speech to generate adversarial examples. The success rate for white-box attacks is close to 100%. However, this method has several problems: (1) it does not consider playback in the real scene, and the efficiency of generating speech is low; (2) it takes nearly an hour to generate adversarial examples; (3) the perturbation is large; (4) the method cannot be applied to a black-box; and (5) the adversarial examples generated by the optimization of a model are only effective for that model.

The following work considered the above shortcomings [15]. Large perturbation is mainly related to the measurement of the difference between generated and original examples. Qin et al. [17] adopted a psychoacoustic model to redesign the loss function, so that the adversarial example was closer to the original in hearing. Carlini’s algorithm converged with difficulty, resulting in low attack efficiency. Schonherr et al. [18] improved the algorithm so that adversarial examples could be generated in several minutes. Liu et al. [19] also improved the efficiency of adversarial example generation. Taori et al. [20] combined a genetic algorithm with gradient estimation to solve the adversarial example problem of a text-based speech recognition system in a black-box scene, but only obtained 35% accuracy.

The defense of adversarial examples helps researchers find and fix security loopholes that may occur in ASR systems based on deep learning. Defense methods include devastation and detection of adversarial examples. Devastation causes an adversarial example to lose its attack ability without affecting the context of normal examples. The detection strategy determines whether an example is adversarial, and those are discarded.

For the devastation strategy, Latif et al. [21] used a generative adversarial network (GAN) to denoise input examples, causing adversarial examples to lose their attack ability. Yang et al. [22] used U-Net to enhance the input data to invalidate adversarial examples. Sun et al. [23] took the adversarial examples generated by various algorithms as extended datasets to retrain the network. Experimental results showed that the network model after such training could better resist adversarial examples. Samizade et al. [24] designed a convolutional neural network (CNN)-based method to detect adversarial examples. Rajaratnam et al. [25] detected adversarial examples by adding random noise to different frequency bands of speech. Rajaratnam et al. [26] proposed to detect adversarial speech examples by comparing the differences between adversarial and normal examples in feature space.

There is a simple and effective method to simultaneously devastate and detect, which only needs to modify the input speech examples. In addition to
using time-dependence to detect adversarial examples, Yang et al. [27] found some effective modification methods to defend against adversarial examples from defense methods in the image field, which include local smoothing, down-sampling, and re-quantization. Kwon et al. [28] used a number of speech modification methods, including low-pass filtering, 8-bit re-quantization, and mute processing, to defend the adversarial examples. Methods based on speech modification cause loss of information of normal speech examples, which is usually unacceptable.

This work researches adversarial example defense to improve the security and robustness of ASR. The attack success rate of adversarial examples should be reduced as much as possible while ensuring the recognition accuracy of normal examples. To this end, we propose a speech adversarial example defense algorithm based on the addition of random noise. Through a large number of experiments, we find that after adding a specific random noise to an adversarial example, its perturbation will be transformed to the summary of the original perturbation and random noise. Due to the influence of random noise, the original perturbation will be devastated and lose its particularity, and the adversarial example will lose its attack ability. Experimental results show that this method can effectively defend against adversarial examples of the two kinds of ASR systems, with a better defense effect than other methods.

The rest of this paper is organized as follows. Section 2 introduces work related to classic adversarial example generation. Section 3 describes our proposed devastation and detection method for adversarial examples. The setting of the experiment, devastation of adversarial examples, and experimental results are discussed in section 4. Section 5 summarizes our work.

2. Related work

ASR systems can be categorized as either text- or command-dependent, which respectively recognize input speech as a text sequence or command tag. ASR systems employ different attack methods. We introduce the two typical speech adversarial example attack methods. optimization-based method (OPT) is based on gradient optimization [15] on a text-based ASR system, and another method is based on a genetic algorithm (GA) [9] for a command-dependent system.
2.1. OPT method

Given a speech example $x$, a perturbation $\delta$ can be constructed that is almost imperceptible to human hearing, but $x + \delta$ can be recognized as any desired text. This is an end-to-end white-box attack, assuming the attacker can obtain the structure and parameters of the identification system. The attack mode is to send the speech directly to the ASR system, and it cannot attack in the air.

Given an original example $x$ and target text $t$, the optimization object is

$$
\text{Minimize } |x|^2 + c \ast l(x + \delta, t), \text{ such that } dB_x(\delta) \leq \tau,
$$

where $c$ weighs whether to make the adversarial example closer to the original example or to make it easier to attack successfully, $l(\cdot, \cdot)$ is the loss function, and $dB_x(\delta) \leq \tau$ ensures that the perturbation is not too large. To calculate the loss function requires a definite alignment $\pi$. The attack algorithm has two steps. An initial adversarial example is generated by CTC loss, which defines the current $\pi$. Fixing the current $\pi$, an adversarial example with less perturbation is generated.

DeepSpeech is an end-to-end text-dependent ASR system based on a DNN [29] that has higher recognition performance than traditional methods, with an excellent effect in noisy environments. Experimental results [14] have shown that adversarial examples generated by the OPT method can enable DeepSpeech to output a specified text content with a 100% attack success rate. The average disturbance size of the generated adversarial example is $-31dB$. The longer the length of specified text the more difficult it is to generate. The perturbation of the generation will also increase; on average, each extra character will increase the perturbation by $0.1dB$. If the text of the original example is longer, the adversarial example will be less difficult to generate.

2.2. GA-based method

This method uses a genetic algorithm based on a free gradient to generate adversarial examples on command-dependent ASR systems. Shown as Algorithm 1, it uses normal speech and a target command as input, and creates a group of candidate adversarial examples by adding random noise to the subset examples in a given speech segment. To minimize the impact of noise on human hearing perception, it is only added to the least significant bit (LSB) of the speech. The deterministic score of each population member
is calculated according to the predicted score of the target label. Through
the application of selection, crossover, and mutation, next-generation adver-
sarial examples are generated from the current generation. Members of the
population with higher scores are more likely to be part of the next gener-
ation. Crossover is to mix pairs of population members to generate a new
example and add it to the new population. Mutation adds random noise
to the offspring with minimal probability before passing it on to the next
generation. The process is repeated until a preset value is reached or the
attack is successful.

Algorithm 1 Genetic Algorithm Based on Adversarial Example Generation

\textbf{Inputs :} Original example \(x\); target label \(t\)

\textbf{Output:} Targeted adversarial example \(x_{adv}\)

\[
\text{pop} := \text{InitializePopulation}(x) \\
k_{iter} = 0 \\
\text{while} \; (k_{iter} < k_{\text{max}}) \; \text{do} \\
\quad \text{scores} := \text{ComputeFitness}(	ext{pop}) \\
\quad x_{adv} := \text{pop}[\text{argmax}(\text{scores})] \\
\quad \text{if} \; \text{argmax } f(x_{adv}) = t \; \text{then} \\
\quad \quad \text{break} \\
\quad \text{end if} \\
\quad \text{probs} := \text{softmax}\left(\frac{\text{scores}}{\text{temp}}\right) \\
\quad \text{pop}_{\text{next}} := \{\} \\
\quad \text{for} \; i := 1 \; \text{to} \; \text{size} \; \text{do} \\
\quad \quad \text{parent}_1, \text{parent}_2 = \text{randomChoice}(\text{pop}, \text{probs}) \\
\quad \quad \text{child} = \text{Crossover} (\text{parent}_1, \text{parent}_2) \\
\quad \quad \text{pop}_{\text{next}} := \text{pop}_{\text{next}} \cup \{\text{child}\} \\
\quad \text{end for} \\
\quad \text{for all} \; \text{child} \; \text{of} \; \text{next} \; \text{pop} \; \text{do} \; \text{Mutate} (\text{child}) \\
\quad \quad \text{pop} := \text{pop}_{\text{next}} \\
\quad \quad k_{iter} := k_{iter} + 1 \\
\quad \text{end for} \\
\text{end while} \\
\text{return } x_{adv}
\]

The command-dependent ASR system SpeechCommand \[32\] recognizes a
command label from speech, and is essentially a multi-classification network.
Experimental results have shown that its success rate of attack can reach 87%.

3. Devastation and detection on adversarial examples

We introduce devastation and detection on examples generated by attack methods. Present methods for the defense of speech adversarial examples modify the training process, change the structure of the network model, or add additional models. These operations can require much computation and training overhead. We propose the devastation and detection of speech adversarial examples based on the addition of random noise, discuss its influence, and provide examples of devastation and detection methods.

3.1. Devastation on adversarial examples

Given the speech signal $x$, attackers use the adversarial example generation algorithm to generate a local gradient in the input layer of the network structure, which is consistent with the size of the input signal $x$. We add perturbation $\delta^*$ to $x$ to generate adversarial example

$$x^* = x + \delta^*. \quad (2)$$

Random noise $\hat{\delta}$ is added to the adversarial example, whose size is consistent with Gaussian noise, to generate adversarial example $\hat{x}^*$. Mixed noise $\hat{\delta}^*$ is the sum of the addition of disturbance $\delta^*$ and Gaussian noise $\hat{\delta}$,

$$\hat{x}^* = x^* + \hat{\delta}, \quad \hat{\delta}(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x - \mu)^2}{2\sigma^2}), \hat{\delta}^* = \delta^* + \hat{\delta}, \quad (3)$$

where the average $\mu$ and standard deviation $\sigma$ can represent the intensity of noise added to the speech signal. Finally, the modified adversarial example can be equivalent to the addition of mixed noise $\hat{\delta}^*$ to the original speech,

$$\hat{x}^* = x + \hat{\delta}^*. \quad (4)$$

Because there is a certain direction when adding adversarial example perturbation, which is equivalent to the addition of purposeful disturbance to make the example close to the target class, these small perturbations play a great role in the discrimination of the model. Even if the input data change slightly, the final calculated value is closer to the distribution of the target class after a series of calculations. When the intensity of $\delta^*$ is
similar to that of $\delta$, or the strength is greater, the superimposed noise $\hat{\delta}^*$ will lose the particularity of $\delta^*$ and become ordinary noise, which will affect the purpose of adversarial perturbation, i.e., the adversarial example $\hat{x}^*$ will not be adversarial, and the devastation strategy will work.

After adding random noise $\hat{\delta}$ to the normal speech signal $x$, we can obtain the modified normal speech, i.e., some useless values are added to the normal speech signal. If the intensity of noise is slight, the speech sounds like the original. Even if the modified normal speech is put into the classifier, the result will not be greatly changed. When the intensity of $\hat{\delta}$ is small, the result is similar to normal. In other words, the addition of small random noise has little effect on the recognition of normal examples, and the noise lacks a direction. As shown in Fig. 1, adding random noise to the normal example, the result will not be changed, but the adversarial example will lose the effect of the attack. Because the adversarial perturbation is devastated by the noise, the model discriminating the speech is affected by the input signal. Adversarial examples that carry a purposeful value can make a model misclassify the signal.

$$\hat{x} = x + \hat{\delta}$$

Figure 1: One single normal example $x$ adds perturbation, and different noise can change the recognition result of the ASR system. The normal example $x$ with perturbation $\delta^*$ will be changed to an adversarial example $x^*$, and adversarial example $\delta^*$ with random noise $\hat{\delta}$ will be changed to an unknown example $\hat{x}^*$.
Using a normal example of TIMIT, and generating an adversarial example by OPT algorithm, random and Gaussian noise are added to the normal and adversarial examples. Fig. 2 shows the spectrogram of the example. Table 1 shows the recognition results of examples in the DeepSpeech system. It can be concluded that the addition of noise to the normal examples does not change the recognition results. However, the recognition result of the adversarial example changes greatly, and is close to that of the normal example. This shows that the slight random noise will not affect the recognition result of the normal example, but it can invalidate the adversarial example and become closer to the normal example. Therefore, we can use random noise to devastate the adversarial example without seriously affecting the normal example. According to these experimental results, we can further propose the detection method of adversarial examples.

### Table 1: DeepSpeech examples: normal and adversarial examples with added ordinary and Gaussian random noise

| Example                        | Recognition result                                                                 |
|--------------------------------|-----------------------------------------------------------------------------------|
| (a) normal                     | she had her dark suiting greacy wash water all year                                |
| (b) adversarial                | this is an adversarial example                                                    |
| (c) normal with random noise   | she had yedark sutin greacy wash water all year                                   |
| (d) adversarial with random noise | he had regark suting greacy watch water all yer                                    |
| (e) normal with Gaussian noise | she had redark sutin greacy watch water all yer                                    |
| (f) adversarial with Gaussian noise | he had redark suvin greacy watch water all year                                   |
Figure 2: Spectrogram of normal and adversarial examples with ordinary and Gaussian noise

3.2. Detection on adversarial examples

The detection algorithm is motivated by the devastation algorithm. Given a speech sample $x$ (normal or adversarial), we only need to add noise before inputting it to the ASR system. For a normal example, because the low-intensity random noise will not affect the content of the speech, the recognition result of the ASR system will not change much. For an adversarial example, because the perturbation is added to the current speech, the added random noise will devastate the particularity of the perturbation, and the recognition result of the ASR system will be totally different from the original. Therefore, according to the recognition results before and after adding noise, we can determine whether an example is adversarial.
The detection strategy determines whether unknown speech is an adversarial example. Because the recognition result of such an example is more likely to be affected by random noise, we can use the change rate (CR) of the recognition result after adding random noise to detect an unknown example. According to the recognition results in Table 1, the variation of the adversarial example is very large relative to a normal sample. As shown in Fig. 3, we can detect the adversarial example following the process shown as Algorithm 2.

**Algorithm 2 Detection of Adversarial Example**

**Inputs**: Unknown example $x$;  
**Output**: Result of detection  
/* Add noise $\hat{\delta}$ on sample */  
$\hat{x} = x + \hat{\delta}$  
$D = Dist(a, b)$  
/* Calculate change rate (CR) between $x$ and $\hat{x}$, $g(a)$ is recognition system */  
$L = g(x)$  
$CR = \frac{\min(D(g(\hat{x}),g(x)),L)}{L}$  
if $CR > K$ then  
Example $x$ is adversarial  
else  
Example $x$ is normal  
end if

According to Algorithm 2, whether an unknown example is adversarial can be detected on the basis of not seriously devastating the example. In terms of the previous change in the recognition rate of the example after
adding noise, the change beyond a certain threshold can indicate an adversarial example. From our experiments, it can be concluded that random noise will not have much impact on a normal example.

4. Experimental results

4.1. Database

Our experimental data included text- and command-dependent speech databases. The text-based speech database was made up of TIMIT and LibriSpeech, both with a sampling rate of 16 kHz and a bit depth of 16 bits. TIMIT is an acoustics-phoneme continuous speech corpus built by Texas Instruments, Massachusetts Institute of Technology, and SRI International. The database includes 630 speakers from different parts of the United States, 70% male, and mostly adult and white. Each participant spoke 10 sentences, and a total of 6,300 examples were obtained, all manually tagged at the phoneme level. LibriSpeech [30] is a corpus of about 1000 hours of English pronunciation from audiobooks from the LibriVox project. In our experiments, we downloaded a test-clean dataset and used the first 100 examples.

The SpeechCommand Dataset (SpeechCommands) from Google contains 105829 speech files, each consisting of 35 words. The sampling frequency is 16 kHz, the bit depth is 16 bits, the duration is near 1 second, and the format is WAV. The dataset includes 2618 participants who were asked to say 35 words, each participating only once. They had 1.5 seconds to read out each word, at one-second intervals. Examples with no sound content or whose speech content was different from the words were deleted. Each segment was checked manually to delete speech content that was different from the given words. Recordings are in OGG format, and files smaller than 5 KB were deleted. Speech signals were converted to WAV format normalized to [-1.0, 1.0], and files whose average values of the normalized speech signal were less than 0.004 were deleted.

4.2. Experimental setup

We used the GA and OPT attack methods (section 3.2) to generate adversarial examples. For the text-dependent ASR system, we chose classical DeepSpeech [29] as our target system, and OPT to generate adversarial examples to attack the system. TIMIT, LibriSpeech, and CommandVoice were the speech databases. For the command-dependent ASR system, SpeechCommand was the target, and the GA-based method was chosen to generate
adversarial examples. For added noise, $\mu$ had the range (10, 30, 50, 70, 100, 200, 500), and $\sigma$ was in the range (10, 30, 50, 70, 100, 200, 500).

Defense methods on adversarial examples were evaluated differently for text- and command-dependent ASR systems. The defense method of text-dependent adversarial examples is illustrated from two aspects: (1) the effect on adversarial examples was to be as large as possible, and we used the similarity of recognition results after adding noise $SR_{adv}$; and (2) the impact on normal examples was to be as small as possible, and we again measured it by the similarity of recognition results of normal examples after adding noise $SR_{benign}$. Similarity is the matching ratio between the initial recognition result of a sample after adding noise. We calculated

$$SR_{benign} = \frac{D(T(x_{benign}), y)}{D(g(x_{benign}), y)}, \quad SR_{adv} = \frac{D(T(x_{adv}), y)}{D(g(x_{adv}), y)},$$

where $x_{benign}$ is a normal example, $x_{adv}$ is an adversarial example, $y$ is the real text, $D(\cdot, \cdot)$ is the distance function, and the editing distance proposed by Levenshtein et al. [31] represented the input transformation function (e.g., downsampling, quantization, local smoothing, compression).

The defense effect of command-dependent ASR system adversarial examples is illustrated from two aspects: (1) the impact on the adversarial example should be large, as measured by the change of the average attack success rate $ASV_{avg}$ after adding noise; and (2) the impact on a normal example should be small, as measured by the change of the recognition accuracy $ACC$ after adding noise.

In the experiment, we calculated

$$ASR_{avg} = \frac{\sum_{x^*}^{X^*} (g(M(x^*)) == y^*)}{n^*},$$

where $X^*$ is the adversarial example set generated from original example set $X$, $n^*$ is the number of examples in $x^*$, $x^*$ is a single example in $X^*$, and $y^*$ is a label for attacking $x^*$. A lower $ASR_{avg}$ on adversarial examples indicates better performance of the method.

The recognition accuracy is the ratio of the number of correctly recognized speech examples to the total,

$$ACC = \frac{\sum_{x}^{X} (g(M(x)) == y_0)}{n},$$

where $x$ is a sample from the testing set $X$, $y_0$ is a label for $x$. The recognition accuracy is the ratio of the number of correctly recognized speech examples to the total.
where dataset $X$ has $n$ examples, and $x$ is an example whose real label is $y_0$. The closer ACC is before and after adding noise, the smaller the influence of the defense method, and the better the defense performance.

4.3. Results and discussion

We show the effect of devastation on text-dependent adversarial examples, and detection results on command-dependent adversarial examples.

4.3.1. Results of devastation on adversarial examples

We explore the effect of the intensity of ordinary and Gaussian noise on the experimental results, and compare our method with other advanced methods.

Table 2 compares the similarity of normal examples (NEs) and adversarial examples (AEs) with different intensities of ordinary noise after processing. With increasing noise intensity, the similarity of NEs decreases gradually, and that of AEs decreases greatly. When the intensity of the noise is close to 50, the similarity of AEs is 0% on the TIMIT and LibriSpeech databases, and 12% on the CommonVoice database. When the noise intensity is higher than 50, the similarity of AEs is unchanged. Therefore, the appropriate general noise intensity is about 50.

Table 3 compares the similarity of NEs and AEs with different intensities of Gaussian noise. With increasing noise intensity, the similarity of NEs decreases gradually, and that of AEs decreases greatly. When the noise

| Parameter | Datasets |
|-----------|----------|
|           | TIMIT    | LibriSpeech | CommonVoice |
|           | NEs | AEs | NEs | AEs | NEs | AEs |
| 10        | 96.81 | 81.04 | 99.68 | 96.06 | 97.41 | 79.27 |
| 50        | 88.50 | 0    | 98.38 | 0    | 92.70 | 12.00 |
| 100       | 80.18 | 0    | 97.20 | 0    | 88.20 | 0    |
| 200       | 62.85 | 0    | 94.48 | 0    | 83.56 | 0    |
| 300       | 48.29 | 0    | 92.36 | 0    | 78.56 | 0    |
| 400       | 37.58 | 0    | 90.40 | 0    | 75.49 | 0    |
| 500       | 29.63 | 0.43 | 87.66 | 0    | 71.29 | 0    |

Table 3 compares the similarity of NEs and AEs with different intensities of Gaussian noise. With increasing noise intensity, the similarity of NEs decreases gradually, and that of AEs decreases greatly. When the noise
intensity is close to 50, the similarity of AEs of the three databases is almost zero. When the noise intensity is higher than 50, the similarity of AEs is unchanged. On the TIMIT database, when the noise intensity is higher than 300, the similarity of AEs increases slightly, perhaps because the perturbation is small, the noise intensity of 300 is already higher than that of the perturbation, and the influence of adding noise intensity becomes small. Therefore, the appropriate intensity of Gaussian noise is about 50. Comparison with the results of Table 2 shows that ordinary noise has a slightly better result than Gaussian noise.

Table 4 compares the similarity results of NEs and AEs with different methods. For the method of Kwon, the similarity of the 8-bit reduction on AEs on the three databases is low, as is the similarity of NEs in the TIMIT database. Hence, this method is not suitable for all databases. A low-pass filter performs well on both normal and adversarial examples. The similarity on AEs of silence removal is high. For the Yang method, the similarity of the four processing samples on the three databases is high, and the similarity of NEs of Quan-512 is the lowest. For our method, the results of random noise-50 are better than those of other methods, and NEs have high similarity, but the similarity of AEs on the CommonVoice database is nonzero. With Gaussian noise-50, the similarity of AEs in three databases is zero. Compared with low-pass filtering, the result is lower on TIMIT, higher on LibriSpeech, and similar on CommonVoice. Our method has an overall better effect than the others.

Table 3: Similarity result of noise addition on databases with seven noise intensities; the impact is large on adversarial examples and small on normal examples.

| Parameter | Datasets |
|-----------|----------|
|           | TIMIT | LibriSpeech | CommonVoice |
|           | NEs | AEs | NEs | AEs | NEs | AEs |
| 10        | 96.81 | 81.04 | 99.68 | 96.06 | 97.41 | 79.27 |
| 50        | 88.50 | 0 | 98.38 | 0 | 92.70 | 12.00 |
| 100       | 80.18 | 0 | 97.20 | 0 | 88.20 | 0 |
| 200       | 62.85 | 0 | 94.48 | 0 | 83.56 | 0 |
| 300       | 48.29 | 0 | 92.36 | 0 | 78.56 | 0 |
| 400       | 37.58 | 0 | 90.40 | 0 | 75.49 | 0 |
| 500       | 29.63 | 0.43 | 87.66 | 0 | 71.29 | 0 |
Table 4: Similarity results of state-of-the-art and proposed methods; ours performs better than the methods of Kwon and Yang

| Method  | Type               | Datasets                   |
|---------|--------------------|----------------------------|
|         |                    | TIMIT | LibriSpeech | CommonVoice |               |
|         |                    | NEs   | AEs         | NEs         | AEs           |
|         |                    | NEs   | AEs         | NEs         | AEs           |
| Random  | Random noise-50    | 88.5  | 0           | 98.38       | 0             | 92.70         | 12.00 |
|         | Gaussian noise-50  | 82.81 | 0           | 97.35       | 0             | 90.07         | 0     |
|         | 8-bit reduction    | 59.26 | 0           | 93.91       | 0             | 82.08         | 0     |
| Kwon    | Low-pass filtering | 87.28 | 0           | 93.95       | 0             | 90.74         | 0     |
|         | Silence removal    | 73.64 | 22.43       | 94.35       | 0             | 83.56         | 09.09 |
|         | Downsampling       | 85.54 | 20.48       | 93.37       | 19.53         | 87.91         | 14.86 |
|         | Smoothing          | 77.61 | 20.44       | 85.99       | 19.03         | 82.19         | 14.97 |
|         | Quan-256           | 77.61 | 21.93       | 96.73       | 21.08         | 88.44         | 20.00 |
|         | Quan-512           | 59.30 | 14.28       | 93.86       | 19.39         | 81.42         | 18.38 |

4.3.2. Results of detection on adversarial examples

We discuss the results of AE detection on the command-dependent ASR system. We explore the effect of the intensity of ordinary and Gaussian noise, and compare our method with others.

Table 5 shows the changes of $ASR_{avg}$ and $ACC$ of AEs with different intensities of ordinary noise. With increasing noise intensity, both measures show a downward trend, with $ASR_{avg}$ decreasing more than $ACC$. When the noise intensity is higher than 100, $ASR_{avg}$ is less than 10%. In practical scenes, a tradeoff should be made between $ASR_{avg}$ and $ACC$. When the change of $ACC$ is small, noise with larger intensity should be selected to make $ASR_{avg}$ as small as possible.
Table 5: Results of different intensities of ordinary random noise of $ASR_{avg}$ and $ACC$

| Parameters | $ASR_{avg}$ (%) | $ACC$ (%) |
|------------|-----------------|-----------|
| 10         | 52.54           | 93.80     |
| 30         | 32.89           | 93.80     |
| 50         | 21.96           | 93.40     |
| 70         | 16.29           | 93.60     |
| 100        | 10.62           | 94.00     |
| 200        | 3.98            | 92.00     |
| 500        | 1.80            | 88.80     |

Table 6: Results of different intensities of Gaussian random noise of $ASR_{avg}$ and $ACC$

| Parameters | $ASR_{avg}$ (%) | $ACC$ (%) |
|------------|-----------------|-----------|
| 10         | 42.60           | 94.00     |
| 30         | 20.62           | 94.00     |
| 50         | 11.64           | 93.20     |
| 70         | 7.92            | 93.60     |
| 100        | 4.80            | 92.60     |
| 200        | 2.13            | 90.80     |
| 500        | 1.75            | 86.00     |

Table 6 shows the changes of $ASR_{avg}$ of AEs and $ACC$ of NEs under different intensities of Gaussian noise. Similar to the results of ordinary noise, both $ASR_{avg}$ and $ACC$ show a downward trend with the increase of noise intensity. Compared to Table 5, with the same noise intensity, $ASR_{avg}$ and $ACC$ under Gaussian noise are both smaller than under ordinary noise. We can conclude that for a better defense effect, we can add Gaussian noise, and if we reduce the recognition accuracy of NEs, we can choose ordinary noise.
Table 7: Comparison of $ASR_{avg}$ and $ACC$ with state-of-the-art and proposed methods; our method outperforms the methods of Kwon and Yang.

| Method     | Type                | $ASR_{avg}$ (%) | $ACC$ (%) |
|------------|---------------------|-----------------|-----------|
| Without defense | -                   | 83.81           | 95.00     |
| Random     | Random noise-200    | 03.98           | 92.00     |
|            | Gaussian noise-200  | 02.13           | 90.80     |
| Kwon [28]  | 8-bit reduction     | 28.78           | 90.60     |
|            | Silence removal     | 09.22           | 85.00     |
|            | Downsampling        | 10.22           | 91.60     |
| Yang [27]  | Smoothing           | 20.78           | 92.00     |
|            | Quan-256            | 29.97           | 90.20     |
|            | Quan-512            | 07.84           | 89.00     |

Table 7 shows the $ASR_{avg}$ of AEs with different defense methods and $ACC$ of NEs. $ASR_{avg}$ is calculated from the matrix of attack success rates of samples of the corresponding defense methods in Fig. 4. We compare the results of the appropriate noise intensity from the experiments in Tables 5 and 6 to those of other methods. It can be seen that our method has a higher $ACC$ than others, while ensuring a lower $ASR_{avg}$. The $ASR_{avg}$ of Gaussian noise with intensity 200 is the smallest, and the $ACC$ of ordinary noise with intensity 200 is the largest. Kwon’s 8-bit reduction also has a good effect, but its result on the text-dependent ASR system is poor. The $ASR_{avg}$ of other methods is high.

Figure 4: Detection AUC of AEs. Under appropriate noise, the AE can be detected with high probability. As the intensity of noise increases, the effect on LibriSpeech is good.
The experimental results show that noise addition has a great impact on AEs, and reduces the success rate of attacks. For NEs, the impact is small, and the recognition accuracy is reduced very little. Fig. 5 shows the AUC of the AE detection rate under different kinds and intensities of noise. Under the devastation of appropriate noise, the \( CR \) fed back by the ASR system can well show whether a sample is adversarial. Hence, the category of examples can be calculated by the \( CR \) after adding noise. \( CR \) is small for the NE and large for the AE. Without affecting the judgment of normal examples by the ASR system, AEs can be distinguished with great accuracy.

In conclusion, noise addition is confirmed to devastate and detect AEs, with little cost for NEs. The experimental results are better than those of the advanced method.

5. Conclusions

We proposed an algorithm for speech adversarial example devastation and detection based on the addition of random noise. After adding an appropriate intensity of noise to an adversarial example, its perturbation becomes the sum of the original perturbation and random noise. Noise devastates the original perturbation, and it loses its particularity, so the adversarial example after adding noise will lose the attack effect. We chose ordinary random noise and Gaussian noise. In experiments, we used the text-dependent DeepSpeech ASR system with the OPT attack method, and the command-dependent CommonVoice ASR system with the GA-based attack method. When choosing an appropriate noise intensity and type, our method was better than those of Kwon and Yang, and ordinary random noise had a slightly better effect than Gaussian noise. Although our proposed method of random noise addition can devastate adversarial examples, it also affects the recognition results of normal examples, and the appropriate intensity of noise is related to the perturbation intensity of adversarial examples. In our future work, we will formulate methods to overcome these shortcomings so as to achieve better results.
Figure 5: Evaluation metrics comparing state-of-the-art methods with ours.
Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 6217011361, U1736215, 61901237), Zhejiang Natural Science Foundation (Grant No. LY20F020010), Ningbo Natural Science Foundation (Grant No. 202003N4089) and K.C. Wong Magna Fund in Ningbo University.

References

[1] K He, X Zhang, S Ren, J Sun.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 770–778(2016)

[2] V Mnih, et al.: Human-level control through deep reinforcement learning. Nature 518:529–533(2015)

[3] Kumar Y, Singh Na.: Comprehensive view of automatic speech recognition system-A systematic literature review. In: International Conference on Automation Computational and Technology Management (ICACTM). IEEE: 168-173(2019)

[4] B Biggio, et al.: Evasion attacks against machine learning at test time. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases: 387–402(2013)

[5] C Szegedy, et al.: Intriguing properties of neural networks. arXiv: 1312.6199(2013)

[6] T Vaidya, Y Zhang, M Sherr, C Shields.: Cocaine noodles: exploiting the gap between human and machine speech recognition. 9th USENIX Workshop on Offensive Technologies (WOOT 15)(2015)

[7] S Molau, M Pitz, R Schluwer, H Ney.: Computing mel-frequency cepstral coefficients on the power spectrum. In: IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (cat. No. 01CH37221)1:73–76(2001)

[8] N Carlini, et al.: Hidden voice commands. 25th USENIX Security Symposium (USENIX Security 16):513-513(2016)
[9] M Alzantot, B Balaji, M Srivastava.: Did you hear that? adversarial examples against automatic speech recognition. arXiv:1801.00554(2018)

[10] X Yuan, et al.: CommanderSong: A systematic approach for practical adversarial voice recognition. In: Proceedings of the 27th USENIX Security Symposium:49–64(2018)

[11] M Cisse, Y Adi, N Neverrova, J Keshet.: Houdini: Fooling deep structured prediction models. arXiv: 1707.05373(2017)

[12] D Iter, J Huang, M Jwemann.: Generating adversarial examples for speech recognition. Stanford Technical Report(2017)

[13] A Van Den Oord, et al.: WaveNet: A generative model for raw audio. arXiv: 1609.03499(2016)

[14] I J Goodfellow, J Shlens, C Szegedy.: Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings:1–11(2015)

[15] N Carlini, D Wagner.: Audio adversarial examples: Targeted attacks on speech-to-text. In: Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018:1–7(2018)

[16] A Graves, S FERNÁNDEZ, F Gomez, J Schmidhuber.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine learning (ICML):369–376(2006)

[17] Y Qin, N Carlini, I Goodfellow, G Cottrell, C Raffel.: Imperceptible, Robust, and targeted adversarial examples for automatic speech recognition. In: 36th International Conference on Machine learning (ICML):9141–9150(2019)

[18] L Schonherr, K Kohls, S Zeiler, T Holz, D Koloss.: Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. arXiv:1808.05665(2018)

[19] X Liu, K Wan, Y Ding.: Towards weighted-sampling audio adversarial example attack. arXiv:1901.10300(2019)
[20] R Taori, A Kamsetty, B Chu, N Vemuri.: Targeted adversarial examples for black box audio systems. arXiv:1805.07820(2018)

[21] S Latif, R Rana, J Qadir.: Adversarial machine learning and speech emotion recognition: utilizing generative adversarial networks for robustness. arXiv:1811.11402(2018)

[22] C H Yang, J Qi, P Y Chen X Ma, C Hlee.: Characterizing speech adversarial examples using self-attention u-net enhancement. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE:3107-3111(2020)

[23] S Sun, C F Yeh, M Ostendorf, M Y Hwang, L Xie.: Training augmentation with adversarial examples for robust speech recognition. In: Proceedings Annual Conference of the International Speech Communication Association. INTERSPEECH:2404–2408(2018)

[24] S Samizade, Z H Tan, C Shen, X Guan.: Adversarial example detection by classification for deep speech recognition. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP):3102–3106(2020)

[25] K Rajaratnam, J Kalita.: Noise flooding for detecting audio adversarial examples against automatic speech recognition. In: IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). IEEE: 197-201(2018)

[26] K Rajaratnam, K Shah, J Kalita.: Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition. arXiv:1809.04397(2018)

[27] Z Yang, B Li, P Y Chen, D Song.: Characterizing audio adversarial examples using temporal dependency. arXiv:1180.910875(2018)

[28] H Kwon, H Yoon, K W Park.: Acoustic-decoy: Detection of adversarial examples through audio modification on speech recognition system. In: Neurocomputing 417:357–370(2020)

[29] A Hannun, et al.: Deep speech: Scaling up end-to-end speech recognition. arXiv:1141.25567(2014)
[30] http://www.openslr.org/12/

[31] V I Levenshtein.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady 10:707–710(1966)

[32] T N Sainath, C Parada.: Convolutional neural networks for small-footprint keyword spotting. In: Sixteenth Annual Conference of the International Speech Communication Association(2015)