Robustifying automatic speech recognition by extracting slowly varying features

Matias Pizarro, Dorothea Kolossa, Asja Fischer

Ruhr University Bochum, Germany

{Matias.PizarroBustamante, dorothea.kolossa, asja.fischer}@rub.de

Abstract

In the past few years, it has been shown that deep learning systems are highly vulnerable under attacks with adversarial examples. Neural-network-based automatic speech recognition (ASR) systems are no exception. Targeted and untargeted attacks can modify an audio input signal in such a way that humans still recognize the same words, while ASR systems are steered to predict a different transcript. In this paper, we propose a defense mechanism against targeted adversarial attacks consisting in removing fast-changing features from the audio signals, either by applying slow feature analysis, a low-pass filter, or both, before feeding the input to the ASR system. We perform an empirical analysis of hybrid ASR models trained on data pre-processed in such a way. While the resulting models perform quite well on benign data, they are significantly more robust against targeted adversarial attacks: Our final, proposed model shows a performance on clean data similar to the baseline model, while being more than four times more robust.

1. Introduction

ASR systems are ever more widely used in our daily life. They are being easily incorporated in our phones or in home electronics, which are not only used for human-machine interaction, but are also able to control other smart devices. Previous works have shown the vulnerability of these systems by adversarial attacks in the audio domain [1, 2, 3, 4], where malicious noise is added into the input signal to fool the ASR system and manipulate the predicted transcript. This opens the doors for attackers to breach the system security and violate users’ privacy. [1] pointed out in the introduction some real-world impact, such as unwanted online shopping orders made by amazon’s devices.

Many of the most widely used ASR systems are based on hybrid models, combining a deep neural network (DNN) with a hidden Markov model (HMM). The DNN predicts the class probabilities per speech frame, while the HMM is used to describe the temporal, context-dependent information that is given by the pronunciation, the vocabulary, and grammar of the language. Research has centered on the properties that make DNNs vulnerable to adversarial attacks not only in image recognition [5, 6], but also in speech recognition [2, 7], where specific perturbations are added to the input signal. These perturbations are chosen in such a way that they are hardly perceivable for humans but maximize the loss function of the model and lead the system to predict the output desired by the attacker [8, 9, 10].

The human perception of audio signals depends among other things on the frequency. The presence of one sound can mask another sound imperceptible; this effect is known as masking. This happens most easily, when the two sounds are in the same so-called critical band. These bands can be obtained from an auditory filterbank model that describes the behavior of the Cochlea, as analyzed in psychoacoustics. High frequencies have wider critical bandwidths than lower frequencies, which makes two sounds at high frequencies less distinguishable for humans and especially promising for adversarial attacks. Prior works have already explored this property, creating adversarial perturbations that are mostly hidden from human perception, raising the question of how to secure our current ASR systems are under attack [1, 7].

There has thus been increasing interest in the research community in building robust ASR systems. However, there are only a few studies that tackle this problem and propose defenses against malicious audio [11, 12] and image perturbations [13, 14]. The ones that exist lack robustness against adaptive adversaries in white box settings, where the architecture of the model is known by the attacker.

Motivated by the observation that adversarial perturbations often lie in the high frequency range, we propose to remove faster variations of the input signal. To do so, we use slow feature analysis (SFA) [15, 16], an unsupervised learning algorithm that is able to extract slow components of an input signal that varies on a faster time scale. It has been successfully applied in other fields, e.g., for extracting driving forces [17], nonlinear blind source separation [18], and for automatic lipreading [19]. In addition to SFA, we apply a low-pass filter with a cutoff frequency of 7 kHz1. Thus, we limit the possible space in which malicious noise can affect the recognition system. We propose to apply these modifications to the audio signals before feeding them to hybrid DNN-HMM ASR systems (for training as well as for testing), and present an empirical analysis that demonstrates the success of this strategy. While the performance on vanilla audio signals is not affected strongly, the rate of successful targeted adversarial attacks is reduced significantly.

In Section 2 we give a brief overview of the required background knowledge. In Section 3, we describe our defense approach in more detail, and in Section 4, we present our empirical analysis.

2. Background

2.1. Adversarial examples

For simplifying descriptions, we assume that the relationship between the input audio signal x and the label transcript y is given by

$$y = f(x)$$

where $$f(\cdot)$$ is the mapping function of the ASR system that maps an input to the sequence of words it most likely corresponds to. The goal of a targeted adversarial attack is to add a minimal perturbation $$\delta$$ onto x such that the system predicts a

1Baer et al. have shown improvements on the intelligibility of speech in noise for people without dead regions in high frequencies by using low-pass filters, increasing the cutoff frequency up to 7.5 kHz [20].
new sequence of words $\hat{y}$ defined by an attacker, i.e.
\[ f(x + \delta) = \hat{y} \neq y = f(x). \]

To generate an adversarial example, first, given an audio signal and its corresponding malicious text, we need to find an optimal time-aligned version, and second, we need to add the perturbation $\delta$ that leads the ASR system to predict the desired transcript.

The first task is achieved by applying the forced alignment algorithm, which—given the original audio signal and the target transcription—generates an optimally time-aligned state sequence that matches the text with the audio signal. For the second task, we use the projected gradient descent (PGD) method for creating adversarial examples [14]. For an in-depth description of the generation of adversarial samples for ASR systems, we refer the reader to [1]. PGD corresponds to solving a constrained optimization problem by maximizing the loss function of the network through gradient descent with respect to its parameters. Notice that $\lambda_j$ represents the slowness of the features, i.e. the smaller the $|\lambda_j|$, the slower components of its input signal. That is, we aim at making it more difficult for the attacker to manipulate the input in such a way that the ASR system produces a desired output. Towards this aim, we augment the ASR system by adding a pre-processing stage. In this stage, the input signal is transformed by applying SFA, a low pass filter, or both. Thus, our acoustic model is trained and tested using a modified version of the original data set. In the following, we describe the pre-processing steps in more detail:

### 3.1. Performing slow feature analysis

To extract the slowest component from an input signal, we use the python data processing framework MDP [21] to complete the following steps:

**Step 1: Time embedding.** We create a new multivariate time series by framing the input signal $x(t)$ using a shifted delay of $L$ elements. Thus, we expand the dimensionality of our data
\[ x_i(k, \tau) = x_i(k + R \tau), \quad k = 0, ..., L - 1, \]
where $k$ represents the sample index, $\tau$ the frame index, $R$ the length of overlapping elements and $L$ the length of the window. In our work $L = 2$ and $R = 1$.

**Step 2: Non-linear expansion.** We expand our framing result from Step 1 by a quadratic function $h$ to create non-linear features of our original data.
\[ h(x) = [x_0, x_1, x_2, x_0 x_1, x_1^2]^T. \]

**Step 3: Whitening.** To meet the constraints of the optimization problem in Equation (2), the expanded signal is whitened. This removes the mean and stretches the signal along its principal axes, such that it has zero mean and unit variance in all directions. Moreover, by projecting the whitened signal onto two orthogonal unit vectors, the projected signal components are decorrelated. We use principal component analysis to sphere our data from Step 2.
\[ z(t) := W(h(x(t))), \quad \text{where } W \text{ denotes the whitening function [22].} \]

**Step 4: Derivative.** We calculate the time derivative denoted by $\dot{z}_i(t)$, which, in our problem, is simply the difference between two successive time points.
\[ \dot{z}_i(t) = z_i(t + 1) - z_i(t). \]

**Step 5: Input-output function.** In order to find the direction, in which the signal varies most slowly, we first calculate the covariance matrix of our derivative signal from Step 4.
\[ \text{cov}(\dot{z}(t)) = \langle \dot{z} \dot{z}^T \rangle_t \]
Second, we perform a principal component analysis to find the smallest eigenvalue $\lambda$, with its corresponding eigenvector $w$ by solving the eigenvalue equation given by
\[ \langle \dot{z} \dot{z}^T \rangle_t w = \lambda w \quad \text{with } \lambda_1 \leq \lambda_2 \leq ... \lambda_J. \]
Finally, we choose $\lambda_1$ and its corresponding $w_1$, which corresponds to the direction of slowest variation, to do the projection of our data.

If we were interested in obtaining additional slow components, we could simply take the next, larger eigenvalue and project our data using its corresponding eigenvector. Notice that $\lambda_1$ represents the slowness of the features, i.e. the smaller the $\lambda_1$, the
4. Experiments

In the following, we describe our experimental analysis.

4.1. Speech recognition system

We work in a PyTorch environment and train a hybrid DNN-HMM ASR system on the small-vocabulary TIDIGITS data set, which contains approximately 8,000 utterances with a vocabulary consisting of digits from 0 to 9, with 2 alternative pronunciations for 0 (i.e., 'oh' and 'zero').

The NN architecture has two hidden layers with 100 units each; the activation functions of the hidden layers are ReLUs, while the activation function of the output layer is a softmax function with 95 units that correspond to the states of the HMM. The input layer size (39) is chosen to work with feature vectors containing the first 13 mel-frequency cepstrum coefficients alongside their first and second derivatives. We employ Adam with an initial learning rate of 0.0001 [24] as the optimizer of the network and the cross-entropy loss function. To improve results, after training the NN for 3 epochs, 5 epochs of Viterbi training are executed [25].

In order to find the pre-processing leading to the best defense against adversarial attacks, we train five ASR systems, all of them with the same NN architecture described above, but after applying different kinds of pre-processing to the training set. That is we perform SFA, low-pass filtering, or both techniques, as described in Section 3. We repeat the experiments five times with different random initializations and report the mean performance. An example of the spectrogram of one input signal, the corresponding adversarial example, and the same adversarial example after pre-processing are shown in Figure 1.

4.2. Adversarial attack

To generate the target transcriptions of the adversarial examples, digits from 0 to 9 are randomly selected. The total number of digits per target transcript varies randomly between 1 and 5. To find the optimally time-aligned state sequence denoted by $x^*$ for a given combination of an audio signal and an adversarial text, we use the Montreal forced aligner [26]. To generate the adversarial audio signal that generates this desired $x^*$, we use PGD as provided by the library Cleverhans [27].

We refer the reader to the description in the work of Dăušner et al. [28] for more details on the creation of the adversarial examples.

4.3. Model evaluation

To evaluate the performance of the speech recognition models, we use the word error rate (WER) given by

$$\text{WER} = 100 \cdot \frac{S + D + I}{N}, \quad (3)$$

where $S$ is the number of substituted words, $D$ is the number of deleted words and $I$ is the number of inserted words. $N$ is the total number of words in the reference text, which corresponds to the ground-truth label of the original test sample or the malicious target transcription for the adversarial attack, respectively. We aim at a model that has a low WER on the original data, but a high WER compared to the target transcript of an adversarial attack, since this means that the attack is not successful.

The WER is evaluated on a test set of 1,000 malicious samples and 1,000 original test samples.
4.4. Results

The average WERs of the different models for the 1 000 samples from the test set and the 1 000 adversarial samples are shown in Table 1. We observe a low WER of 0.79% for the baseline model on the original test set. For the adversarial test set, all models are seen to be unprotected against adversarial perturbations.

When the same pre-processing step that is applied during training is also used in the prediction procedure during inference, the performance on the original test set only varies slightly, while the WER for adversarial examples increases significantly, see Table 2. The LPF model is especially promising, since it performs similar to the baseline on the original test set, but is significantly more robust to adversarial attacks. The largest improvement in robustness is given by the SFA model, but at the cost of aggravating the WER on clean data.

To claim that a model performs significantly different to the baseline model the Wilcoxon Rank-Sum Test (WRST) W statistic is provided in both tables. Enough statistical evidence for the corresponding claim exists when the W statistic is less than 17.\footnote{The lower critical value for the WRST for independent samples of size 5 is 17, corresponding to \( \alpha = 0.05 \) for a two-tailed test.}

Table 1: Average WER of different models for 1 000 test samples and adversarial examples created with \( \epsilon = 0.5 \).

<table>
<thead>
<tr>
<th></th>
<th>Original Test set</th>
<th>Adversarial Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER</td>
<td>Std Dev</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.79%</td>
<td>0.0013</td>
</tr>
<tr>
<td>Bas-SFA</td>
<td>1.01%</td>
<td>0.0009</td>
</tr>
<tr>
<td>SFA</td>
<td>2.71%</td>
<td>0.0071</td>
</tr>
<tr>
<td>LPF</td>
<td>2.67%</td>
<td>0.0012</td>
</tr>
<tr>
<td>SFA-LPF</td>
<td>2.62%</td>
<td>0.0058</td>
</tr>
</tbody>
</table>

Table 2: Average WER of models applied to pre-processed data (with different kind of data transformations) for 1 000 test samples and adversarial examples created with \( \epsilon = 0.5 \). Most promising results are printed bold.

<table>
<thead>
<tr>
<th></th>
<th>Original Test set</th>
<th>Adversarial Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WER</td>
<td>Std Dev</td>
</tr>
<tr>
<td>Baseline</td>
<td>0.79%</td>
<td>0.0013</td>
</tr>
<tr>
<td>Bas-SFA</td>
<td>2.76%</td>
<td>0.0018</td>
</tr>
<tr>
<td>SFA</td>
<td>2.28%</td>
<td>0.0012</td>
</tr>
<tr>
<td>LPF</td>
<td>0.80%</td>
<td>0.0014</td>
</tr>
<tr>
<td>SFA-LPF</td>
<td>2.21%</td>
<td>0.0024</td>
</tr>
</tbody>
</table>

Figure 2 illustrates the effect of adversarial noise at perturbation levels \( \epsilon \in [0, 0.1] \) with increments of 0.01. For \( \epsilon = 0 \), the sample corresponds to the original test point. For calculating the WERs, we again created 1 000 adversarial samples corresponding to 1 000 test samples per model and per perturbation level \( \epsilon \) for each run.

Although all analysed pre-processing strategies decreased the success rate of adversarial attacks (which improves when increasing the level of perturbation \( \epsilon \)), it comes at the cost of harming the performance on clean data. Only the model combined with the LPF has no significant performance decrease, while exhibiting a WER four times larger than the baseline model for adversarial examples. When solely considering robustness, the best strategy is to combine SFA and the low-pass filter, yielding an average WER above 80% for all levels of perturbation. The results support our hypothesis that removing high frequency components and relying on the slowest features limits the space, where adversarial noise can be added effectively, which in turn makes the ASR less vulnerable against adversarial attacks.

5. Discussion

In training the different ASR models, we have implemented a set of potential data transformations, and used this data to train small-scale acoustic models from scratch. It would be interesting to investigate whether we can reduce the time-consuming part of training by reusing the lower layers of a pre-trained network, i.e. whether our experiments can be improved by incorporating the principle of transfer learning. Computational effort is not a significant issue in the presented small-vocabulary experiments, but will be interesting when running our experiments for large-vocabulary tasks. We leave this implementation as an interesting direction for future work.

6. Conclusions

In this work, we show that extracting slow variations and removing high-frequency components of audio signals that serve as input to an ASR can make the system more robust against targeted adversarial attacks. When these data transformations were added as pre-processing steps during training and inference, the success rate of targeted adversarial examples was decreased significantly. For most data transformations, this came at a cost of significantly decreased performance on clean data. However the LPF did not lead to a performance decrease on clean data, while resulting in a model significantly more robust than the baseline. A combination of SFA and LPF induced the highest robustness but also more the doubles the WER on clean data.

Of course our results are based on a limited experimental analysis. Future research will test the transferability of our findings to large-vocabulary ASR systems and the creation of adaptive attacks to evaluate the defense against a skilled attacker.
7. Acknowledgements
This work was supported by the German Academic Exchange Service - 57451854, the Chilean government (CONICYT-PFCHA/Doctorado acuerdo bilateral DAAD) - 62180013 and partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy - EXC 2092 CASA – 390781972.

8. References