Neural Predictor for Black-Box Adversarial Attacks on Speech Recognition

Marie Biolková¹; Bac Nguyen²

¹École Polytechnique Fédérale de Lausanne, Switzerland
²Sony Europe B.V. R&D Center, Stuttgart Laboratory 1, Germany

marie.biolkova@epfl.ch, Bac.NguyenCong@sony.com

Abstract

Recent works have revealed the vulnerability of automatic speech recognition (ASR) models to adversarial examples (AEs), i.e., small perturbations that cause an error in the transcription of the audio signal. Studying audio adversarial attacks is therefore the first step towards robust ASR. Despite the significant progress made in attacking audio examples, the black-box attack remains challenging because only the hard-label information of transcriptions is provided. Due to this limited information, existing black-box methods often require an excessive number of queries to attack a single audio example. In this paper, we introduce NP-Attack, a neural predictor-based method, which progressively evolves the search towards a small adversarial perturbation. Given a perturbation direction, our neural predictor directly estimates the smallest perturbation that causes a mistranscription. In particular, it enables NP-Attack to accurately learn promising perturbation directions via gradient-based optimization. Experimental results show that NP-Attack achieves competitive results with other state-of-the-art black-box adversarial attacks while requiring a significantly smaller number of queries. The code of NP-Attack is available online.

Index Terms: adversarial examples, speech recognition, black-box attack

1. Introduction

There has been significant progress in improving the performance of automatic speech recognition (ASR) based on deep neural networks during the last few years [1][2][3]. As a result, it enables speech recognition technology in many real-world applications, such as Amazon Transcribe, IBM Speech to Text, and Google Cloud Speech to Text. A typical pipeline of an ASR system consists in extracting acoustic features from the audio signal, e.g., the frequency spectrum or mel-frequency cepstral coefficients (MFCCs), then employing an acoustic model that predicts which phonetic units are present. Finally, a language model is used to determine the most likely word sequence. More recently, end-to-end ASR models, which directly output characters or words from audio, have gained popularity. These models have been shown to achieve state-of-the-art performance [3][4][5] by replacing the engineering process with the learning process and optimizing the whole network in a single objective function.

Despite their exceptional performance, many studies have revealed that neural networks are vulnerable to adversarial examples (AEs) [6][7][8][9]. These examples are carefully constructed by adding imperceptible perturbations to the inputs, which cause the model to output a specific phrase (i.e., targeted attack) or any incorrect transcript (i.e., untargeted attack). Most existing attacks on ASR assume the white-box setting [7][8], where the adversary has full knowledge of the model, including the network architecture and parameters. Under this setting, adversarial perturbations are often found via gradient-based optimization such as the fast gradient sign method [10] or DeepFool [11]. However, commercial ASR systems are typically not open source and such information is rarely exposed to the adversary. A more practical attack treats the target ASR model as a black box, i.e., the adversary may only observe the transcribed text [12]. This is much more challenging due to the discrete nature of the output, which does not directly allow gradient computation or estimation. In addition, speech is typically sampled at 16 kHz, making the perturbation search space very high-dimensional even for short utterances. Dimensionality reduction techniques can be applied to make the problem more tractable [13]. Yet, finding a suitable low-dimensional space is not always trivial. One would have to design an efficient search algorithm or resort to combinatorics [14] to tackle this problem.

Another important constraint in black-box attacks is the maximum number of queries allowed during the optimization, also referred to as query budget. It is clearly unrealistic to have unlimited bandwidth access for querying the target ASR model. In real-world scenarios, one can query the target model for labeling but cannot exceed the budget or have access to any internal information of the ASR model. This has motivated growing research interest in adversarial attacks with query-limited context [15][16][17]. However, existing black-box attack methods often require a huge number of queries due to the lack of information about the target ASR model. Consequently, these methods become less applicable with small query budgets.

In this paper, we demonstrate the feasibility of black-box adversarial attacks in speech recognition by designing a simple and query-efficient method. Our method identifies a small subset of promising perturbation directions through the guidance of a neural predictor. To the best of our knowledge, this is the first method based on neural predictors for black-box attacks. The contributions of this paper are summarized as follows.

(i) We propose NP-Attack, a novel method to generate audio AEs for a black-box ASR system. The idea is to design a neural predictor that estimates the distance to the decision boundary which causes a mistranscription of the audio signal. Unlike other methods relying on substitute models, this neural predictor can be trained with much fewer data since it does not estimate the transcript outputs of the target ASR model.

(ii) We conduct several experiments on the LibriSpeech dataset [18], where AEs are created to attack the transformer-based ASR model from SpeechBrain [19]. Without exposing any prior knowledge about the ASR model, NP-Attack can achieve a better success rate with significantly fewer queries compared to other state-of-the-art black-box methods.
2. Related work

Although AEs have been extensively studied in the image domain, there has been considerably less work dedicated in the speech domain. One of the reasons is because humans are more sensitive to auditory perturbations than visual perturbations [20, 21]. We review below some relevant adversarial attack methods in the audio domain.

In particular, Carlini and Wagner [22] demonstrated the feasibility of audio adversarial attacks on DeepSpeech [3], an open-source end-to-end ASR model, with a 100% success rate. Using the full knowledge of the ASR model parameters, the authors proposed a gradient-based optimization method that minimized the Connectionist Temporal Classification (CTC) loss [22]. Adversarial perturbations were updated by backpropagating the gradients through the network and the MFCC layer. As an extension, Schönherr et al. [20] improved the perceptibility of the adversarial perturbations based on psychoacoustic hiding. Interestingly, Neekhara et al. [8] demonstrated the existence of universal adversarial perturbations that are transferable across models with different architectures.

Although white-box audio adversarial attacks showed very promising results, they are quite restricted. To get a more realistic scenario, recent developments in AEs have shifted to target the black-box scenario. Many approaches relied on conventional black-box optimization techniques such as evolutionary optimization [23] or Bayesian optimization [14]. To simplify the problem, some works [24, 25] assumed the knowledge of the log-probabilities given by the target ASR model, which are typically not available to the adversary in the hard-label black-box attack. A common drawback of these methods is that they require many queries to attack a target audio example. Based on the transferability assumption of AEs, Chen et al. [22] introduced a black-box attack method on commercial ASR systems by approximating the target ASR model with a substitute model, which was used to craft AEs. However, unlike in image classification, more sophisticated techniques are often required to train the substitute model in speech recognition as ASR models usually consist of complicated architectures, including preprocessing, an acoustic model, and a language model. Furthermore, attacks are limited to the most frequently used phrases to make the substitute model more reliable.

3. Proposed method

In this section, we formally formulate the problem of finding AEs as an optimization problem. Then, we introduce NP-Attack, a query-efficient approach to solve this problem. Finally, the network architecture and implementation details of NP-Attack are described.

3.1. Problem formulation

Consider a trained black-box ASR model as a function \( f \) that maps an input audio \( x \in [-1, 1]^D \) to a transcript \( t = f(x) \), a sequence of characters or words. Our goal is to find an imperceptible perturbation \( \delta \in \mathbb{R}^D \) such that the ASR model mis-transcribes the input audio signal. Finding such an adversarial perturbation can be formulated as an optimization problem:

\[
\min_{\delta} \| \delta \|_p \quad \text{s.t.} \quad f(x + \delta) \neq t.
\]  

To keep the perturbed audio valid, we often perform a clipping operation to \([-1, 1]\). For simplicity, we assume this is included in the ASR model \( f \) and do not write it explicitly in the formulation.

To overcome these challenges, we follow the boundary-based attack formulation established by Cheng et al. [16]. In particular, the perturbation \( \delta \) is factorized into a direction vector \( \theta \in \mathbb{R}^D \) and a magnitude scalar \( \lambda \in \mathbb{R}^+ \), i.e., \( \delta = \lambda \theta / \| \theta \| \). Given a perturbation direction vector \( \theta \), the distance from \( x \) to the nearest AE along \( \theta \) is defined as

\[
g(\theta) = \min_{\lambda > 0} \lambda \quad \text{s.t.} \quad f \left( x + \frac{\lambda \theta}{\| \theta \|} \right) \neq t.
\]

Note that \( g(\theta) \) also corresponds to the distance to the decision boundary along \( \theta \). Using the above definition, problem \( \text{(1)} \) can be rewritten as

\[
\min_{\theta} g(\theta).
\]

There are several advantages with this formulation. First, it has been shown that the above objective function is locally smooth and continuous [16]. That is, a small change of \( \theta \) leads to a small change of \( g(\theta) \) (see Fig. 1 for an illustration). Second, instead of searching for the constrained perturbation \( \delta \), we simplify the problem to searching for a direction vector \( \theta \), which is an unconstrained optimization. Although computing \( g(\theta) \) in Eq. \( \text{(3)} \) corresponds to solving another constrained optimization problem with respect to \( \lambda \), it requires only a single degree of freedom, making the problem simpler. Interestingly, \( g(\theta) \) can be approximated up to certain accuracy by a two-step search procedure [16]. As a first step, a coarse-grained search is applied to find the range of magnitudes in which the perturbation causes a mistranscription. More specifically, let \( \alpha > 0 \) be the step size, the coarse-grained search is done by querying a sequence of points \( \{x + \alpha \theta / \| \theta \|, x + 2\alpha \theta / \| \theta \|, \ldots\} \) one by one until an AE is found, i.e., \( f(x + i\alpha \theta / \| \theta \|) \neq t \) for some \( i > 0 \). In the second step, we employ a binary search procedure to find the smallest magnitude \( \lambda^* \) within the range of \( \{ \|(i-1)\alpha, i\alpha\} \) such that \( f(x + \lambda^* \theta / \| \theta \|) \neq t \).

3.2. Neural predictor

We aim to solve problem \( \text{(3)} \) by progressively fitting a neural predictor as a proxy that estimates the distance from \( x \) to the decision boundary along a given perturbation direction. In the first step, we generate a dataset by querying the target ASR model, then train a neural predictor based on this dataset. In the second step, we use the trained neural predictor to identify a list of promising perturbation directions. Our neural predictor can
substantially accelerate the search process since we have full knowledge of the predictor parameters. The neural predictor is retrained every time a new batch of samples is obtained by querying the target ASR model for the ground-truth distances.

More specifically, we start by generating $n$ training examples $D = \{ (\theta_1, \lambda_1), \ldots, (\theta_n, \lambda_n) \} \subset \mathbb{R}^D \times \mathbb{R}^+$ by querying the ASR model. The ground truth distance from $\theta$ to the decision boundary $\lambda = g(\theta)$ is determined for each perturbation direction $\theta$, via the two-step search procedure as explained in the previous subsection. After constructing the dataset, it is used to train the neural prediction $h(\cdot, w) : \mathbb{R}^D \rightarrow \mathbb{R}^+$, parameterized by $w$. We aim to estimate the distance to the decision boundary $\lambda = h(\theta, w)$ by solving the following problem:

$$w^* = \arg\min_w \frac{1}{n} \sum_{i=1}^n \left( \log h(\theta_i, w) - \log \lambda_i \right)^2. \tag{4}$$

To find the next promising perturbation direction, we freeze the trained parameters $w^*$ and find the next candidate by solving

$$\theta_{n+1} = \arg\min_{\theta} h(\theta, w^*). \tag{5}$$

Assuming that $h(\theta, w)$ is differentiable with respect to both $\theta$ and $w$, problems (4) and (5) can be solved via gradient-based optimization. Subsequently, we compute the ground-truth distance $\lambda_{n+1} = g(\theta_{n+1})$ by querying the ASR model and add this prediction to the training set $D := D \cup \{ (\theta_{n+1}, \lambda_{n+1}) \}$. Next, the predictor parameters are then unfrozen again and modified according to Eq. (4) to account for the newly added examples. This process is repeated until the query limit is reached or a solution within the perturbation budget is found. Figure 2 illustrates the training process of NP-Attack. In the beginning, the neural predictor might produce noisy outputs. To alleviate this problem, we generate a batch of several promising perturbation directions with different random initializations. After progressively adding more examples, the neural predictor is able to produce more reliable outputs.

3.3. Network architecture

The overall architecture of our neural predictor is shown in Fig. 3. It maps the perturbation direction $\theta$ to a positive scalar value $\lambda$, indicating the distance to the decision boundary along $\theta$. More specifically, the input is first normalized to have a unit $\ell_\infty$-norm. To reduce the temporal dimension of the input, we perform the short-term Fourier transform (STFT) where the FFT, window, and hop size are set to 1024, 1024, and 256, respectively. Each frequency is then considered as a channel and a 1D convolution is applied to compress the input down to 32 channels. The resulting signal is passed through 4 blocks to further reduce the temporal dimension. Each block consists of the Leaky ReLU activation with a negative slope of 0.2, 1D convolution, and average pooling with a kernel size of 2. We use a kernel size of 3 for all convolutions and employ weight normalization [29] for all layers. As the last step, global average pooling is used to remove temporal dimensions of the input, followed by a linear layer to produce the prediction. To ensure the network output is positive, we use the exponential function as an activation function after the linear layer. Note that our neural predictor can be used for any input of arbitrary length.

To train the weights $w$ of the predictor, we employ the Adam optimizer [27] with a learning rate of $10^{-4}$ and an exponential scheduler (a decay rate of 0.99). We use a batch size of 32 and train the model for 300 epochs. The perturbation directions $\theta$ are optimized by minimizing the predicted distance to the decision boundary $h(\theta, w)$. Each search starts from a random initialization.

4. Experiments

4.1. Experimental setup

Dataset. To evaluate the effectiveness of an attack method, we construct a dataset by randomly choosing 100 examples from the LibriSpeech clean test data [18]. These audio examples are derived from English audiobooks sampled at 16 kHz with the transcript lengths varying from 5 to 10 words. We ensure that all examples are correctly transcribed by the target ASR model.

Evaluation metrics. To measure the performance of an ASR system, we compute the standard word error rate $WER = (S + I + D)/N_w$, where $S$, $I$, and $D$ indicate the number of substitutions, insertions, and deletions of words, respectively, and $N_w$ is the total number of words in the original phrase. Under a query budget, an attack is considered as successful if the perturbation $\delta$ satisfies that $f(\theta) \neq f(\theta + \delta)$ and $\| \delta \| \leq \lambda_{\text{max}}$, where $\lambda_{\text{max}}$ is a perturbation budget. The success rate of an attack method is calculated as $N_s/N_t \times 100\%$, where $N_s$ and $N_t$ are the number of successful attacks and the total number of test examples, respectively. In addition, we report the $\ell_\infty$-norm of perturbations $\delta$, which is a commonly used measure in previous literature of audio attacks. To better account for the energy of the original audio example, we also provide the signal-to-noise ratio (SNR), defined as $\text{SNR}(\delta) = 20 \log_{10} \left( \frac{\| x \|_{\ell_2}}{\| \delta \|_{\ell_2}} \right)$.

ASR model. We target an end-to-end transformer-based ASR model from SpeechBrain [19] trained on the LibriSpeech corpus [13]. The pretrained ASR model achieves a WER of 2.46% on the clean test set and is publicly available on HuggingFace. Essentially, the model transforms the waveform into a mel-spectrum, then employs an acoustic model comprising of
Success rate (%) of 5,000 queries. Mean (and standard deviation) across five different runs.

| Method       | \(\ell_\infty\) | SNR          |
|--------------|------------------|--------------|
| BayesOpt     | 0.064 (0.001)    | 19.840 (0.159)|
| SignOpt      | 0.028 (0.001)    | 29.315 (0.417)|
| NP-Attack    | **0.022 (0.000)** | **29.410 (0.298)** |

Figure 4: Average attack success rate vs the number of queries for different perturbation budgets.

In contrast, the performances of Bayesian methods heavily rely on the choice of kernel functions.

In the second experiment, we demonstrate the efficiency of NP-Attack compared to other black-box methods. Figure 5 shows the average success rate against the number of queries for different perturbation budgets \(\lambda_{max}\). NP-Attack requires the least number of queries to achieve a high success rate. Importantly, it converges faster to the higher success rates than the other approaches. For a large budget \(\lambda_{max} = 0.1\), only roughly 100 queries are sufficient for a successful attack with NP-Attack.

Finally, we conduct an ablation study to see the effect of varying the required WER on the success rate of NP-Attack. An attack is considered a success when the WER between the transcript of the AE and that of the original example is satisfied. Here, we use a perturbation budget of \(\lambda_{max} = 0.05\) and a query budget of 5,000. The results are shown in Fig. 5. As expected, by increasing the minimum WER, the proportion of successful attacks decreases since the more changes to the transcript we require, the more challenging the problem becomes. This is because the perturbation needs to fool the ASR model even in the locations where the model has highly confident predictions.

5. Conclusions and future work

In this paper, we have introduced NP-Attack, a novel predictor-based method to generate audio AEs on black-box ASR systems with high query efficiency. The proposed method leverages a neural predictor, which estimates the distance to the decision boundary for a given perturbation direction. We demonstrated that NP-Attack achieves high success rates with fewer queries while producing AEs that are close to the original.

Studying untargeted attacks can already be beneficial for building robust ASR systems. In future work, we will further extend our method to targeted attacks. It would also be interesting to study the performance of NP-Attack on different network architectures of the neural predictor.
6. References

[1] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6645–6649.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen et al., “Deep speech 2: End-to-end speech recognition in english and mandarin,” in Proceedings of the International Conference on Machine Learning, 2016, pp. 173–182.

[3] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Sot Admission, S. Sengupta, A. Coates et al., “Deep speech: Scaling up end-to-end speech recognition,” arXiv preprint arXiv:1412.5567, 2014.

[4] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, “Hybrid ctc/attention architecture for end-to-end speech recognition,” IEEE Journal of Selected Topics in Signal Processing, vol. 11, pp. 1240–1253, 2017.

[5] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao et al., “Developing run- t models surpassing high-performance hybrid models with customization capability,” in Proceedings of the International Speech Communication Association, 2020, pp. 3590–3594.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural net- works,” in Proceedings of the International Conference on Learning Representations, 2014.

[7] N. Carlini and D. Wagner, “Audio adversarial examples: Targeted attacks on speech-to-text,” in Proceedings of the IEEE Security and Privacy Workshops, 2018, pp. 1–7.

[8] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, and F. Koushanfar, “Universal adversarial perturbations for speech recognition systems,” in Proceedings of the International Speech Communication Association, 2019, pp. 481–485.

[9] Y. Kong and J. Zhang, “Adversarial audio: A new information hading method,” in Proceedings of International Speech Communi- cation Association, 2020, pp. 2287–2291.

[10] L. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and har- nessing adversarial examples,” in Proceedings of the International Conference on Learning Representations, 2015.

[11] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in Proceedings of the Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.

[12] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang, “Devil’s whisper: A general approach for physical ad- versarial attacks against commercial black-box speech recogni- tion devices,” in Proceedings of the USENIX Security Symposium, 2020, pp. 2667–2684.

[13] S. N. Shukla, A. K. Sahu, D. Willmott, and J. Z. Kolter, “Black-box adversarial attacks with bayesian optimization,” arXiv preprint arXiv:1909.13857, 2019.

[14] S. Moon, G. An, and H. O. Song, “Parsonious black-box ad- versesarial attacks via efficient combinatorial optimization,” in Pro- ceedings of the International Conference on Machine Learning, 2019, pp. 4636–4645.

[15] A. Ilyas, L. Engstrom, and A. Madry, “Prior convictions: Black- box adversarial attacks with bootstraps and priors,” in Proceedings of the International Conference on Learning Representations, 2019.

[16] M. Cheng, T. Le, P. Chen, H. Zhang, J. Yi, and C. Hsieh, “Query-efficient hard-label black-box attack: An optimization- based approach,” in Proceedings of the International Conference on Learning Representations, 2019.

[17] M. Cheng, S. Singh, P. H. Chen, P.-Y. Chen, S. Liu, and C.-J. Hsieh, “Sign-opt: A query-efficient hard-label adversarial attack,” in Proceedings of the International Conference on Learning Representations, 2020.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Lib- respeech: An asr corpus based on public domain audio books,” in Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2015, pp. 5206–5210.

[19] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong et al., “Speechbrain: A general-purpose speech toolkit,” arXiv preprint arXiv:2106.04624, 2021.

[20] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, “Adver- sarial attacks against automatic speech recognition systems via psychoacoustic hiding,” in Proceedings of the Network and Di- stributed System Security Symposium, 2019.

[21] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, “Imperceptible, robust, and targeted adversarial examples for auto- matic speech recognition,” in Proceedings of the International Conference on Machine Learning, 2019, pp. 5231–5240.

[22] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, “Con- nectionist temporal classification: labelling unsegmented se- quence data with recurrent neural networks,” in Proceedings of the International Conference on Machine Learning, 2006, pp. 369–376.

[23] S. Khare, R. Aralikatte, and S. Mani, “Adversarial black-box attacks on automatic speech recognition systems using multi- objective evolutionary optimization,” in Proceedings of Interna- tional Speech Communication Association, 2019, pp. 3208–3212.

[24] R. Taori, A. Kamisetty, B. Chu, and N. Vemuri, “Targeted adver- sarial examples for black box audio systems,” in Proceedings of the IEEE Security and Privacy Workshops, 2019, pp. 15–20.

[25] Q. Wang, B. Zheng, Q. Li, C. Shen, and Z. Ba, “Towards query- efficient adversarial attacks against automatic speech recognition systems,” IEEE Transactions on Information Forensics and Secu- rity, vol. 16, pp. 896–908, 2020.

[26] T. Salimans and D. P. Kingma, “Weight normalization: A sim- ple reparameterization to accelerate training of deep neural net- works,” in Advances in Neural Information Processing Systems, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, p. 901–909.

[27] D. P. Kingma and J. Ba, “Adam: A method for stochastic op- timization,” in Proceedings of the International Conference on Learning Representations, 2015.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, vol. 30, 2017, p. 6000–6010.

[29] B. Ru, A. Cobb, A. Blaas, and Y. Gal, “Bayesopt adversarial at- tack,” in Proceedings of the International Conference on Learning Representations, 2020.

[30] C. Guo, J. S. Frank, and K. Q. Weinberger, “Low frequency ad- versarial perturbation,” in Proceedings of the Conference on Un- certainty in Artificial Intelligence, 2019, pp. 1127–1137.