(Automatic) Speech Recognition Project — Attacks on Neural Networks in a Lightweight Speech Anonymization Pipeline

Daan Brugmans
Radboud University
daan.brugmans@ru.nl

Abstract

1 Introduction

As advancements in the field of Automatic Speech Recognition (ASR) have accelerated with the rise of modern end-to-end neural networks, the risks associated with using such models in ASR applications has become more evident.

Modern neural ASR models are capable of parsing and producing speech to a new level of authenticity: transformers are state-of-the-art for ASR and word recognition, and the introduction of modern unsupervised deep neural network architectures, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), has allowed for more realistic, accurate, and easier generation of speech. These modern speech generation models are capable of learning to reproduce a person's voice, and then generating new utterances using the learned voice. Such synthesized utterances are called *deepfaked* utterances, or simply *deepfakes*.

The presence and influence of deepfakes has become increasingly apparent in recent years: neurally synthesized audio and video of real persons are used to spread misinformation and to manipulate. This recent development has raised attention on the development of methods that prevent or complicate the production of deepfakes. One way to complicate the production of deepfakes is the removal of characteristics in the audio that relate to the speaker's likeness. This is called *Speaker Anonymization*. In Speaker Anonymization, we aim to apply changes to a speaker's utterance such that the changes mde make the utterance untraceable to the person's likeness, while understandability is maintained.

Modern Speaker Anonymization systems, often neural in nature, have been shown to be able to anonymize speech while maintaining understandability. However, such neural systems are not impenetrable, and there exist many sorts of attacks aimed at neural networks. By attacking neural Speaker Anonymization systems, we may be able to circumvent the preventative measures they provide, and generate speech to a person's likeness nonetheless. This paper will focus in that topic: attacking neural Speaker Anonymization systems. Specifically, we will focus on the topic of adversarial attacks. *Adversarial attacks* on deep neural networks are attacks that alter the input that the network receives. They are performed by applying a perturbation onto the input data with the aim to alter or manipulate the network's behavior. These perturbations should meet two criteria as faithfully as possible: they should not be detectable by humans and they should perturb the original input as little as possible.

2 Related Work

2.1 Neural Speaker Anonymization

Meyer et al. (2023) showcase a successful realization of neural Speaker Anonymization using GANs. Their Speaker Anonymization architecture includes the extraction of embeddings from speech, which are fed into a GAN. This GAN first learns to generate embeddings sampled from a normal distribution. After every sample, it then learns to calculate the distance between the embedding that it generated, and an embedding extracted from speech it has been fed. This training procedure teaches the GAN to generate embeddings that are similar to the speech embeddings. Once training has been finished, embeddings based on real speech embeddings are sampled from the GAN until an embedding is generated whose cosine distance from the corresponding real speech embeddings is sufficiently large. If the cosine distance is sufficiently large, the GAN should have produced an embedding that represents the contents of the speech ut-

terance without representing the characteristics of the speaker found in the utterance. This embedding is fed to an existing speech synthesis model, which produces anonymized speech.

Chen et al. (2024) showcase another example of a neural Speaker Anonymization pipeline. Their main contribution is the use of an adversarial attack for anonymization purposes. Chen et al. use an adversarial attack called *FGSM* that learns to apply perturbations on a VAE's latent space vector that represents an utterance with a speaker's characterizations. The perturbations caused by the FGSM attack alter the latent space vector in such a way that it is as far removed from the original speaker's speech characterizations as possible, while still representing the original utterance. When the vector is then fed to the VAE's decoder, the decoder is unable to extract features from the perturbed vector that relate to the original speaker's speech characterizations. The resulting decoded speech sample is untraceable to the original speaker and thus anonymous.

2.2 Attacks on Neural Networks

The FGSM attack used by Chen et al. (2024) is an example of an adversarial attack. Adversarial attacks are extensively described by Yuan et al. (2019). They provide an introduction to and an overview of adversarial attacks within the deep learning domain, and should provide the reader with plentiful knowledge on the topic.

For the purposes of this paper, we will limit out attention to two types of adversarial attacks: *Evasion Attacks* and *Backdoor Attacks*.

2.2.1 Evasion Attacks

Evasion attacks are adversarial attacks that are used during model inference. The aim of an evasion attack is to perturb an input in such a way that a fully trained model is fooled into behaving differently. This behavior should fulfill the attacker's goals.

Goodfellow et al. (2015) introduced FGSM, the *Fast Gradient Sign Method*. FGSM perturbs an input by exploiting a trained model's gradients. When given the input x and its label y, an FGSM attack calculates the gradient with respect to the input using the model's parameters θ and loss function $J_\theta(x, y)$. The sign of this gradient is calculated and is added on top of the original input by a factor of ϵ. The FGSM attack can then be defined as such:

$$x_t = x + \epsilon \cdot \text{sign}(\nabla_x J_\theta(x, y))$$

where x_t is the perturbed input. A visualization of this process can be found in figure 1.

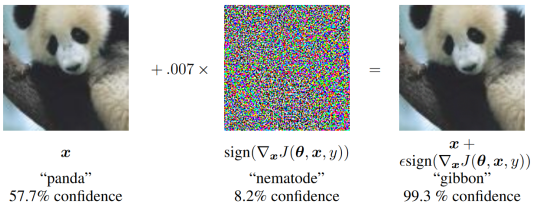

Figure 1: An example of an FGSM attack visualized (Goodfellow et al., 2015).

FGSM attacks have some limitations. One of these limitations is that an FGSM attack is a one-step approach: the gradient with respect to the input is calculated once, and is then used without corrective measures. Although this makes FGSM attacks cheap to perform, it also makes the attack difficult to perform optimally. Madry et al. (2018) propose an expanded version of the FGSM attack called the *Projected Gradient Descent* (PGD) method. The core principle of PGD is that it projects FGSM gradients back to a predefined max perturbation level; if an FGSM's perturbation is too big, PGD projects it back. This makes PGD a multiple-step approach.

PGD's projection of an FGSM attack is performed by applying a clipping function clip$(\cdot)$ on the FGSM attack. The clipping function will clip gradients outside the limit of $x + S$, where x is the original input data and S is the predefined maximal Euclidean distance a perturbation is allowed to be from x. Prior to clipping the FGSM perturbed input x_t, PGD will fire another FGSM attack on the model, but will now calculate the gradient with respect to the perturbed input x_t instead of the original input x. This additional FGSM attack is added onto x_t by a factor of hyperparameter α. It is the result of this addition that is clipped by the clipping function. This means that a PGD attack can be defined as follows:

$$x_t = x + \epsilon \cdot \text{sign}(\nabla_x J_\theta(x, y))$$
$$x_{t+1} = \text{clip}_{x+S}(x_t + \alpha \cdot \text{sign}(\nabla_x J_\theta(x_t, y)))$$

where x_{t+1} is the final result of one round of PGD. PGD can be performed for multiple rounds T until x_T is reached.

2.2.2 Backdoor Attacks

Backdoor attacks are adversarial attacks that are used during model training. They are a subset of *Poisoning Attacks*. The aim of a poisoning attack

Figure 2: An example of a BadNet attack (Gu et al., 2019). On the left, an image of a stop sign with varying BadNet backdoors can be seen. On the right, a backdoored network can be seen interpreting a stop sign as a speed limit sign due to the physical backdoor on the sign itself.

is to alter (a subset of) the training data in order to teach the model certain behavior or to decrease the model's performance. Backdoor attacks are a specific type of poisoning attack where the attacker teaches a model to respond to a certain property of the data. If the model encounters that property in a data sample, it should behave according to the attacker's goals. This property is called a *trigger*.

A well-known example of a backdoor attack is the BadNet attack by Gu et al. (2019). BadNet attacks poison a dataset of images by adding a visual feature onto the image, such as a colored square. A Convolutional Neural Network (CNN) will then learn to associate the visual feature with certain behavior. If the model has successfully learned to associate the trigger with certain behavior, then it will display that behavior during inference when given an image with the trigger. The trigger may cause the model to misclassify an image, for example. Such an example is showcased in figure 2.

Since backdoor attacks are applied during model training, certain attacks can only be used on certain types of data and networks. Although backdoor attacks are most common for image data and CNNs, there also exist backdoor attacks on audio data. Liu et al. (2018) showcase a variety of backdoor attacks, including a backdoor for ASR models where the trigger is background noise that is introduced to audio. The ASR model is trained to associate the background noise with a specific word. Koffas et al. (2022) propose a backdoor attack on ASR systems where the trigger is inaudible. The trigger is a frequency that exceeds the frequency range that can be heard by a human, which should make the trigger impossible to detect by the human ear.

In Koffas et al. (2023), the authors propose a backdoor based on stylistic triggers called *JingleBack*. The trigger in a JingleBack attack is a digital music effect applied on the audio, such as a pitch shift or reverb. A neural ASR model learns to associate certain combinations of effects with certain behavior. In their work, Koffas et al. backdoor neural models of three different architectures (small CNN, large CNN, and LSTM) using six different combinations of digital effects. They find that certain combinations of effects are more effective triggers than others, and that additional effects do not necessarily improve the trigger's effectiveness. The most effective trigger found by the authors was to increase the audio's amplitude by 12dB, then apply a high-pass ladder filter, and then apply a phaser filter. This trigger results in a rate of attack success of up to nearly 95% when only 0.5% of the training data is poisoned.

2.3 Attacks on Neural Speech Systems

Yan et al. (2022)
Neekhara et al. (2019)
Kreuk et al. (2018)

3 Method

Kai et al. (2022)

4 Experiment

5 Results

6 Discussion

7 Conclusion

References

Shihao Chen, Liping Chen, Jie Zhang, KongAik Lee, Zhenhua Ling, and Lirong Dai. 2024. Adversarial speech for voice privacy protection from personalized speech generation. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11411–11415.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Identifying vulnerabilities in the machine learning model supply chain.

Hiroto Kai, Shinnosuke Takamichi, Sayaka Shiota, and Hitoshi Kiya. 2022. Lightweight and irreversible speech pseudonymization based on data-driven optimization of cascaded voice modification modules. *Computer Speech & Language*, 72:101315.

Stefanos Koffas, Luca Pajola, Stjepan Picek, and Mauro Conti. 2023. Going in style: Audio backdoors through stylistic transformations. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Stefanos Koffas, Jing Xu, Mauro Conti, and Stjepan Picek. 2022. Can you hear it? backdoor attacks via ultrasonic triggers. In *Proceedings of the 2022 ACM Workshop on Wireless Security and Machine Learning*, WiseML '22, page 57–62, New York, NY, USA. Association for Computing Machinery.

Felix Kreuk, Yossi Adi, Moustapha Cisse, and Joseph Keshet. 2018. Fooling end-to-end speaker verification with adversarial examples. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1962–1966.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-221, 2018*. The Internet Society.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

Sarina Meyer, Pascal Tilli, Pavel Denisov, Florian Lux, Julia Koch, and Ngoc Thang Vu. 2023. Anonymizing speech with generative adversarial networks to preserve speaker privacy. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 912–919.

Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. 2019. Universal Adversarial Perturbations for Speech Recognition Systems. In *Proc. Interspeech 2019*, pages 481–485.

Chen Yan, Xiaoyu Ji, Kai Wang, Qinhong Jiang, Zizhi Jin, and Wenyuan Xu. 2022. A survey on voice assistant security: Attacks and countermeasures. *ACM Comput. Surv.*, 55(4).

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824.