Creating Simple Adversarial Examples for Speech Recognition Deep Neural Networks

Nathaniel Redden
Department of Mathematics and Computer Science
Ashland University
Ashland, OH, USA
nredden@ashland.edu

Ben Bernard, Jeremy Straub
Department of Computer Science
North Dakota State University
Fargo, ND, USA
ben.bernard@ndsu.edu, jeremy.straub@ndsu.edu

Abstract—The use of deep neural networks for speech recognition and recognizing speech commands continues to grow. This necessitates an understanding of the security risks that goes along with this technology. This paper analyzes the ability to interfere with the performance of neural networks for speech pattern recognition. With the methods proposed herein, it is a simple matter to create adversarial data by overlaying audio of a command at a fairly unnoticeable amplitude. This causes the neural network to lose around 20% accuracy and misidentify commands for other commands with an average to high confidence value. Such an attack is virtually undetectable to the human ear.

I. Introduction

The use of neural networks continues to grow each year as more and more uses for them are identified. As this increased use and reliance on neural networks grows, so does the danger of using them. While researchers find new and better ways to use and make neural networks, others are almost simultaneously finding new vulnerabilities and better exploits for them. The need to better understand the security of neural networks of all kinds is more pronounced than ever. Previous research has already shown that neural networks are highly vulnerable to adversarial examples [1]. These adversarial examples are comprised of data that is similar to natural or correct data but is labeled incorrectly by the network.

Most existing work on adversarial examples and robustness of neural networks has focused on images; networks designed for image classification [2], facial recognition [3], face detection [4], or image segmentation [5]. There is not nearly as much work done in the space of audio classification, voice recognition, or speech to text transformations. Audio recognition is just as important from a security standpoint as image recognition with regards to neural network security as it has equally critical applications for its use. Just as fooling a facial recognition system could allow unauthorized access to systems that rely on facial recognition for security, this same approach can be taken to issue false commands to voice command systems. This concept has already been proven to be possible [1]. An attacker could also prospectively use an adversarial attack to gain access to systems that rely on voice recognition software for security [6].

Because of these vulnerabilities, what were thought to be much better security options than traditional passwords actually have a number of flaws that can be exploited. This paper presents and characterized the problem that speech recognition and other audio identification methods face. It demonstrates a keen need for researchers to discover new methods for audio attacks and identify the flaws and vulnerabilities of these systems, so as to allow the next new neural network models to be even more secure than the current generation.

II. Background

Neural networks, which are modeled loosely after the human brain, are a set of algorithms designed to recognize patterns [7]. They interpret sensory data and labeling or clustering it through machine perception algorithms. Through this process, real-world data such as images, text, or audio is translated into numerical patterns.

A process called supervised learning is used to train the neural network with a labeled dataset. This process prepares the neural network to determine how to correctly cluster and classify unlabeled data, that is presented to it later. The classification is based on the patterns discovered, through the learning process, in the labeled dataset.

The term deep neural networks [8] is used to refer to a neural network that consists of several layers of related neural networks. At minimum, there is an input and output layer, with one or more hidden layers in between. Each layer trains on a unique set of features based on the previous layer's output, performing specific ordering and sorting that builds a feature hierarchy. Each layer added to the deep neural network allows for increased sophistication in the performance of the deep neural network to cluster and classify data. Additionally, deep neural network accuracy can be improved with each additional dataset that is used to train it.

Deep neural networks are being utilized in many industries to solve a wide variety of complex problems. In transportation, deep neural networks solve fleet optimization problems [9] and provide predictive maintenance solutions. In healthcare, deep neural networks can analyze complex imagery [10], such as radiology, to detect cancer. The finance [11] and utility [12] sectors use deep neural networks to detect fraud. Consumer

technology products make heavy use of deep neural networks. Examples include Tesla's Autopilot [13], Facebook's photo tags [14], and the Alexa, Siri, Google Assistant virtual assistant products [15].

One of the biggest threats to neural network operational security is what can be called a minimum adversarial example. The basic idea of this is that the natural data is perturbated to such a small degree that it would be almost or completely undetectable to humans but the neural network will misclassify the data as a result, corrupting the neural network's training and producing incorrect, damaged or tampered output. This type of attack is also called 'data poisoning'. Minimum adversarial data has been proven to be effective in the image recognition space, causing neural networks to misclassify images by changing as little as just one pixel of an image [16].

This same idea can and has been successfully applied to voice recognition neural networks, specifically with a targeted approach that focused on speech to text software [1], [16], [17]. The minimum adversarial data approach has been successful in causing the Mozilla DeepSpeech [23] algorithm, for example, to interpret audio files as whatever phrase attackers want while the actual audio still sounds almost exactly like the original. The DeepSpeech model is what is used to power the voice command systems of many different products and services from speech to text software and devices like Amazon Echo. With the proposed methods, it is possible to issue unwanted commands to these systems while masking them in otherwise valid commands or non-command audio. It may even be possible to inject an unwanted command by playing the perturbations in the open air while an actual command is being given and remaining undetectable to human ears [18].

One of the biggest concerns for these minimum adversarial examples is that they are, by design, made to seem indistinguishable to natural data. Thus, if an easy to implement method for creating these adversarial examples were to be created, it would be possible for the databases used to train neural networks to become polluted with these adversarial examples. This could render both the datasets themselves and the networks that have been trained with them to be completely useless or, at least, highly susceptible to tampering.

III. Methodology

In this section, the dataset used to test and train the neural network is described. Details on which neural network has been utilized and how it was trained with the dataset are also provided.

A. The Dataset

The AudioMNIST dataset developed for and used by Becker, et al. [8] was selected for use for these tests. This dataset consists of 30,000 audio recordings of the numbers zero through nine spoken aloud by 60 different speakers of varying accents and dialects. The database contains audio from both male and female speakers.

This dataset is well-suited for use, as it is large compared to some other datasets of this type, but it is still very manageable and fairly simple with just ten classes to classify. It is also a

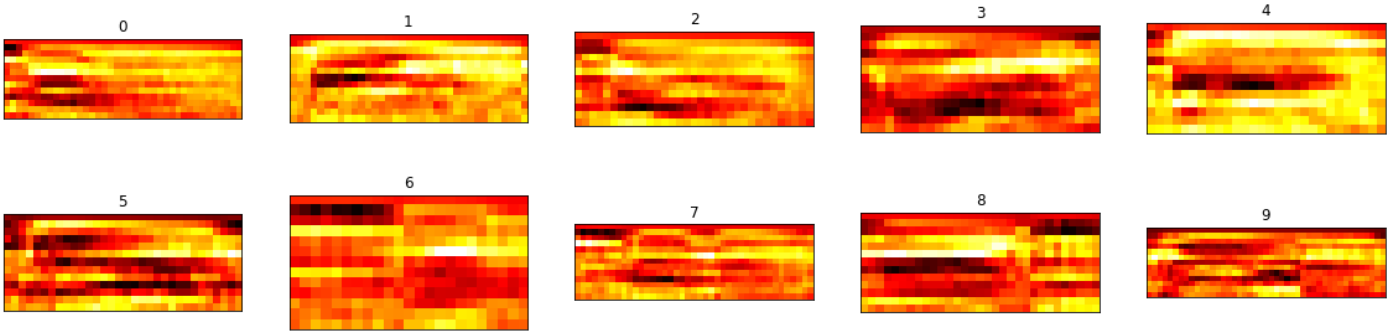

Figure 1: Shows the Mel Frequency Cepstrum Coefficients of the clean testing data

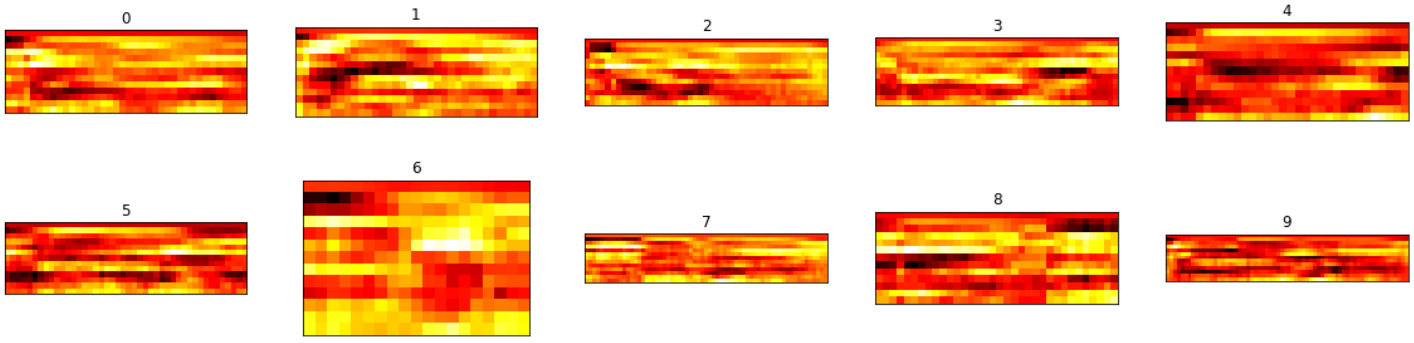

Figure 2: Shows how the Mel Frequency Cepstrum Coefficients change with adversarial data added

fairly robust dataset with a wide variety of speakers. The dataset was split into three portions for this experiment. First, 5,000 records were set aside as test data, 23,500 recordings were used for training, and all the adversarial data creation and testing was done with the test dataset split. The split dataset consists of all the recordings coming from ten of the speakers with 500 recordings each of equal portions of each digit. This means that the tests run with this split are as accurate as possible as the network is seeing ten speakers with their unique voices from one another that it never saw in training.

B. The Network Model

The neural network used for testing [21], [22] is well-suited for use because it has been demonstrated to perform with 97% accuracy on the test data. There was also sufficient reference material available to ensure that the neural network was configured correctly for the testing. This network, named Audio-Classification, is a convolutional neural network consisting of 4 convolutional layers, a max pooling layer, and 3 dense layers. The network looks at the Mel Frequency Cepstrum Coefficients of the audio files in order to identify them [19]. The network was trained for 50 epochs. This took a few hours on a modest desktop computer. The training level of 50 epochs seemed appropriate, based on prior work, for reaching the highest levels of validation accuracy without risking overfitting the network. The implementation can also be tuned more to make the neural network even more accurate and robust.

IV. Experimentation

Three tests were run on the neural network. All of the tests used the same subset of the main dataset.

A. The First Test

The first of the tests was simply to generate a prediction for all of the test data with no modifications being made. This gave a baseline for how the network performs on clean data, which was 96.8% accurate.

B. The Second Test

The second test characterized how the neural network performed when some random white noise was added to the audio files before having the neural network make any predictions on them. This test was conducted by generating random noise and saving that into a NumPy array. Then the clean audio file, in which the signals are also stored, was read in as NumPy arrays [20]. The two arrays were then added together and saved to a new wav file which was the modified audio. The network was then presented to the neural network which made its predictions on the modified files just like it did previously with the clean files. This time, however, the results were 75% accurate and the neural network was far less confident in its choices overall. This test demonstrates how susceptible the network is to anything that is not ideal conditions and how that affects performance.

Overall, 75% accuracy is still suitable performance for many applications. It is also impressive, considering how heavily modified the files were. If a human were to go back and listen to the modified files and compare the audio quality to the original files, they would notice a significant difference. The original files are nearly pristine, with little to no outside noise other than the person speaking. With the modified files, some have interference present that is so pronounced that it may be impossible for human listeners to understand and make out what the recorded audio words are. Even so, the tested neural network was still able to produce the correct classification 75% of the time.

C. Third Test

The third test was designed to determine whether the neural network would detect one number overlaid over another, with varying degrees of amplitude. The attacker-side goal was that the neural network would be deceived into thinking that the original number was the overlaid number, or maybe even a different number entirely, with as little magnitude of the other number being overlaid as possible. Minimizing interference makes the attack as undetectable as possible, if a human were to listen to the modified files or if the tampered sounds were somehow being broadcast over a speaker in a live environment.

One file of each number in the clean test that the neural network identified with as close as possible to 100% certainty was chosen to make up the base number data set. Thus, the recordings being used represent the best possible situation for the neural network. The next step was to create the overlaid files. This was fairly trivial and used the same method as the white noise test: adding together two different NumPy arrays, only this time from two different audio files themselves rather than one being generated. For each number overlay combination, five different overlays with five different percentages of the amplitude of the second audio file that is being overlaid onto the first were created. The amplitude percentages used were 10%, 20%, 30%, 40%, and 50%.

This test involved a much smaller portion of the testing split then previously used, with only ten of the files from the base sample being used. Each of these files generated 45 files after overlaid audio data was applied. The base audio was overlaid with nine other files, each of which had five different amplitude levels. This resulted in a total for 450 files for the test. The overall accuracy of this test was similar to the white noise test with an accuracy of 73.8%.

V. Data Analysis

Considering the experimentation with the clean data test there are a few interesting things that can be distilled from the results. The overall accuracy of the neural network on the clean data was 96.8%, and the system had an average max confidence of 78.5% for each of its guesses; overall, the neural network was fairly confident in the choices it was making.

In the random white noise test the neural network had a 20% drop in accuracy, only achieving an accuracy of 75.04%. Still, the neural network performed well considering how heavily modified the audio sounds were, with most of them being difficult to understand with human ears. The neural network had an average confidence in the choices it was making of

50.8%; compared to the clean data test's 78% confidence. This demonstrated that, with the added noise into the tested audio file, the network is more uncertain of what its choices were, even though it is still getting the correct answer a majority of the time.

With the tests when the neural network did not correctly identify the audio sample a vast majority of the time, it was typically incorrectly classifying as the numbers seven, eight and nine. In almost all these cases, the neural network was only slightly more confident in choosing those numbers than it was any other number. It is unknown as to why the neural network was consistently more confident in choosing those numbers over others. This does suggest the possibility of trying to perform a very crude and simple adversarial attack where an attacker adds random noise to the audio or broadcasts it over the audio in some sort of way to try to throw off the neural network or force the neural network to consistently misclassify audio a majority of the time.

In the number overlay test the neural network achieved an overall accuracy of 73.4%. This test was created using only samples that the neural network earlier achieved 100% accuracy on in the clean test. In the number overlay test, the neural network had an average confidence value in its choices of 73.6%. This is a significant increase from the 50% confidence in the random noise test and not that far lower than the 78.5% confidence for the clean data test. In the clean data test, the neural network had an average confidence value, on its correct guesses, of 79.5% and in the overlay test the machine had an average confidence in its correct choices of 79.4%. Between this, and the overall confidence values, the neural network was approximately as confident on completely clean data as it was on data that had another number overlaid overtop of it.

When it came to the neural network's incorrect choices, in the clean data test, it had an average confidence of just 46.6%. However, in the test with the overlaid data the machine had an average confidence in its incorrect choices of 57.5%. Thus, in audio files that had completely different data overlaid on top of them, the neural network actually became more confident when it was wrong.

Another very interesting data point is that, in the overlaid data test, there is a much higher number of high confidence incorrect choices than there are with the clean data test. This is even despite the fact that the overlaid test had a much smaller amount of testing data, with only 450 files, compared to 5,000 samples in the other test. The clean data test only had three incorrect choices with a confidence value of over 75% (approximately 0.06% of the test data). The overlaid data test had twenty of these high confidence value incorrect choices (approximately 4.4% of the total testing data). It even had a decent number of incorrect choices with confidence values in the upper 90% range.

One of the most interesting things that can be gathered from the number overlay test is the frequency of different numbers the neural network chooses when it makes an incorrect choice. The neural network identified that, out of the 118 errors made out of a total pool of 450 total choices, 69.5% of them came from just the numbers seven and nine. Each was identified 41 times incorrectly. It is interesting to note that the average confidence the system had, when choosing the number nine incorrectly, was 68%. This is greater than the confidence that the system had when identifying some numbers correctly.

There were also a few numbers that were falsely identified as a seven or nine much more frequently than other numbers. For example, the files that should have been identified as the number three were misidentified as seven or nine 16 times (35%) out of the total 17 misidentifications. The neural network also had difficulty identifying the number six and misclassified it 73% of the time. It classified it as a seven 39% of the time. These observations suggest that, perhaps, the neural network is experiencing overfitting to a degree, causing it to identify numbers as seven or nine more often than any other number. This behavior was not seen with the clean data and only showed up with the adversarial data with overlaying one of the files over the other.

An alternate theory is that the neural network is more vulnerable to attacks against certain numbers than others. For example, the testing demonstrated that, overall, the numbers seven and nine were difficult numbers for the tested neural network to misidentify. It also demonstrates that some numbers may be more susceptible to this kind of attack, such as the number six which the neural network had difficulty identifying. Given this, it seems that users can fairly consistently make the neural network think that the number six is instead the number seven, but it is very difficult to make it think that it is instead an eight, with that misidentification only occurring once.

It was also shown that it takes less of an overlay for certain numbers to be misidentified than it does for other numbers. For example, when overlaying most numbers, a misidentification can only happen with the highest amount of the amplitude being added (50% of the overlaid file making it roughly 33% of the total amplitude). Some numbers, though, can trigger a misidentification at the lower amplitudes. For example, when overlaying the number eight over top of the number six the neural network did not trigger being misidentified until 50% of the amplitude of the number eight was added. When the number seven was overlaid, it was triggered right away with only 10% of the amplitude of the number seven needing to be added.

This demonstrates that some numbers (like the numbers seven and nine) are much more effective to overlay than many other numbers. It is unclear what could be the reason behind this kind of behavior; it may be completely caused by overfitting, but this has not yet been conclusively determined.

VI. Open Questions and Future Works

All of these tests were performed with just one neural network model. This model was not a state of the art or 'commercial grade' example. Given this, one area of prospective future work is to run similar tests to determine if this kind of attack is feasible in numerous circumstances or if the results are caused by a quirk with this specific testing model. Additionally, it could be assessed as to if this confusion was only possible because of the simple nature of the specific neural network model tested.

Another prospective approach, for future work, would be to use a different dataset that is a more diverse. While the interference seems to work well when dealing with spoken digits, it unknown if this same approach would work as well with other kinds of speech or audio. If this attack approach is as effective with other kinds of speech, there are significant ramifications for the security and robustness of neural networks used for recognizing voice passwords and commands.

All of the tests described in this article were performed on a neural network using a series of digital audio samples. Given this, another area for prospective future work would be to see if this kind of attack could be effectively carried out in a live environment. For example, if a neural network is set up to take voice commands, could it be discretely attacked with a broadcast from another device so that the original voice command is tampered with or covertly replaced with the attacker's commands

Also, the neural network used was a convolutional neural network. Given this, training and testing with a recurrent neural network would be very interesting.

VII. Conclusions

This paper has presented data that suggests that simple adversarial attacks on speech recognition deep neural networks pose a significant security risk. Moreover, it has been demonstrated that these overlays are difficult for humans to detect. This can be especially true if an attack as crude and simple such as the one proposed is feasible to be carried out on a commercial deep neural network of this kind.

A simple attack has been developed, focusing on the city and on transportation providers, that can be performed on a system of this type. Future work should aim to improve on the methods outlined in this research and determine if these same issues exist in more robust systems.

Acknowledgment

This work was supported by the U.S. National Science Foundation (award # 1757659). Facilities and some equipment were provided by the NDSU Institute for Cyber Security Education and Research. Thanks is given to Zubair Malik for all of his help, input and feedback.

References

[1] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, 2018, pp. 1–7.

[2] C. Szegedy *et al.*, "Intriguing properties of neural networks," Dec. 2013.

[3] K. Eykholt *et al.*, "Robust Physical-World Attacks on Deep Learning Models," Jul. 2017.

[4] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16*, 2016, pp. 1528–1540.

[5] A. Arnab, O. Miksik, and P. H. S. Torr, "On the Robustness of Semantic Segmentation Models to Adversarial Attacks."

[6] Y. Gong and C. Poellabauer, "Crafting Adversarial Examples For Speech Paralinguistics Applications," Nov. 2017.

[7] A. K. Jain, Jianchang Mao, and K. M. Mohiuddin, "Artificial neural networks: a tutorial," *Computer (Long. Beach. Calif).*, vol. 29, no. 3, pp. 31–44, Mar. 1996.

[8] S. Becker, M. Ackermann, S. Lapuschkin, K.-R. Müller, and W. Samek, "Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals," Jul. 2018.

[9] G. Salvo, G. Amato, and P. Zito, "Bus speed estimation by neural networks to improve the automatic fleet management," *Eur. Transp.*, 2007.

[10] J.-G. Lee *et al.*, "Deep Learning in Medical Imaging: General Overview," *Korean J. Radiol.*, vol. 18, no. 4, p. 570, Aug. 2017.

[11] E. L. Paula, M. Ladeira, R. N. Carvalho, and T. Marzagao, "Deep Learning Anomaly Detection as Support Fraud Investigation in Brazilian Exports and Anti-Money Laundering," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 954–960.

[12] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids," *IEEE Trans. Ind. Informatics*, vol. 14, no. 4, pp. 1606–1615, Apr. 2018.

[13] B. Templeton, "Tesla Bets Farm On Neural Network Based Autonomy With Impressive Presentation," *Forbes Website*, 2019. [Online]. Available: https://www.forbes.com/sites/bradtempleton/2019/04/22/tesla-bets-farm-on-neural-network-based-autonomy-with-impressive-presentation/#376f882063ce. [Accessed: 17-Sep-2019].

[14] J. Vincent, "Facebook's image outage reveals how the company's AI tags your photos," *The Verge*, 03-Jul-2019.

[15] N. Gagliordi, "Amazon intros new deep learning models to make Alexa more conversational," *ZDNet*, 05-Jun-2019.

[16] J. Su, D. V. Vargas, and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks," *IEEE Trans. Evol. Comput.*, pp. 1–1, Jan. 2019.

[17] S. Hershey *et al.*, "CNN architectures for large-scale audio classification," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2017, pp. 131–135.

[18] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted Adversarial Examples for Black Box Audio Systems," May 2018.

[19] M. Xu, L. Y. Duan, J. Cai, L. T. Chia, C. Xu, and Q. Tian, "HMM-based audio keyword generation," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3333, pp. 566–574, 2004.

[20] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, Mar. 2011.

[21] S. Adams, YouTube, 24-Oct-2018. [Online]. Available: https://www.youtube.com/watch?v=Z7YM-HAz-IY&list=PLhA3b2k8R3t2Ng1WW_7MiXeh1pfQJQi_P. [Accessed: 14-Aug-2019].

[22] S. Adams, "Audio-Classification," GitHub, 02-Aug-2019. [Online]. Available: https://github.com/seth814/Audio-Classification.

[23] Mozilla. Project deepspeech. https://github.com/mozilla/DeepSpeech, 2017.