A New Method for the Exploitation of Speech Recognition Systems

Suha Hussain
Grade 11
Queens High School for the Sciences at York College
Mentor: Ramesh Karri at New York University

Abstract

The recent surge in the performance of speech recognition has led to the rapid proliferation and adoption of a variety of its applications. However, possible vulnerabilities within these systems have the potential to be rather critical. Previous research has shown how components of speech recognition applications such as preprocessing and hardware can be leveraged by malicious actors. However, a method leveraging neural networks used inside of speech recognition systems is notably absent. Hence, a method was developed that could enable an adversary to craft noises that could be added to the input to deliberately cause misclassification. Not only is this attack inconspicuous, but the crafted noises are both universal and transformable, increasing the feasibility and practicality of this attack.

Table of Contents

I. Background

Advances in deep learning and natural language processing have enabled great progress in speech recognition, which is defined as the ability for a computer to recognize and respond to the sounds produced in human speech. This improvement has resulted in the proliferation of numerous applications, including real-time translation, command and control, and dialog systems in fields as diverse as healthcare, robotics, home automation, and education. Notable examples are Google Now, Apple Siri, and Amazon Alexa, all of which currently have millions of users. The rapid adoption of speech recognition systems in our day-to-day lives makes possible vulnerabilities in these systems even more hazardous, and the search for those vulnerabilities more crucial.

Sharif et al. exploited vulnerabilities in the architecture of state-of-the-art facial recognition systems to generate sets of eyeglass frames that enable users to dodge recognition or even impersonate others when faced with a facial recognition system. This method of exploitation was both physically realizable and inconspicuous, meaning that it could be utilized by present-day malicious agents (Sharif et al., 2016). The development of similar exploitation procedures for speech recognition by malicious actors could result in extensive damage, thus necessitating progress in finding vulnerabilities and developing defenses for these systems.

According to Lopes and Perdigao, previous speech recognition systems consisted entirely of hidden Markov models. However, current systems either use hidden Markov models to augment neural networks for phonetic classification, or use only neural networks to develop entire end-to-end speech recognition systems. A hidden Markov model is a stochastic model that probabilistically maps a sequence of observations to a sequence of labels, while a neural network

is an artificial intelligence algorithm that contains a number of interconnected nodes (processing elements) organized into layers aiming to learn from training data to produce a certain output given specific inputs. In a speech recognition system that uses both neural networks and hidden Markov models, the neural network would be given a processed spoken word and would have to output the phonemes corresponding to that word. For example, the input could be “key” and the expected output would be “KCL K iy”. The hidden Markov model would be given several outputs from the neural network and would be expected to produce a phrase or sentence from that.

One notable data set used for phonetic classification is the DARPA Texas Instruments and Massachusetts Institute of Technology Acoustic-Phonetic Continuous Speech Corpus, or TIMIT. The TIMIT dataset is comprised of recordings of 6300 sentences from 630 American English speakers manually segmented at the phoneme level (Lopes and Perdiago, 2011). Timo van Niedek developed several state-of-the-art neural networks for phonetic classification based upon TIMIT data, and revealed that the exact architecture of the neural network does not have a significant effect on performance. These networks were developed using TensorFlow, an open source deep learning framework (Niedek, 2016).

Previous methods of speech recognition exploitation did not manipulate neural networks. In other words, they did not address the vulnerabilities caused by the incorporation of these algorithms. Due to this, they were all limited in several capacities. Carlini et al. developed unintelligible commands, adversarial examples, for attackers by developing an algorithm that leveraged MFCC conversion in preprocessing. The basis of the exploitation method is on the architecture of speech recognition systems consisting only of hidden Markov models, meaning

that the attack is rather outdated and inapplicable to modern systems. This method was also rather conspicuous, thereby making it impractical (Carlini et al., 2016). Gong and Poellabauer also leveraged preprocessing to mislead speech recognition systems. However, they examined preprocessing specific to computational paralinguistics applications, a small subset of speech recognition, limiting the applicability and versatility of their method (Gong and Poellabauer, 2017).

Other methods have exploited the hardware associated with devices utilizing speech recognition as opposed to portions of the internal system itself. Song and Mittal were able to create inaudible ultrasounds to control a victim device based upon nonlinearities in microphone circuits (Song and Mittal, 2017). Zhang et al. developed a system known as "Dolphin Attack" to exploit the nonlinearity in a different manner to mislead systems such as Siri or Google Now. Both of these attacks were limited in that they were specific to the hardware of that device, and were greatly affected by factors such as carrier wave frequency and modulation depth. Additionally, many possible commands did not work using these techniques (Zhang et al., 2017).

Neural networks are susceptible to small perturbations that can result in misclassification. In other words, attacks can and have been developed that add small amounts of noise unnoticable by humans to inputs, resulting in deliberate misclassification. If you have an image of panda that is correctly classified by a neural network, it is possible to make small changes to the pixel values (adding noise) that would not be noticeable to humans, but would cause the neural network to classify the image as something else. In terms of speech recognition, it is possible to add background noise to a spoken command or response to cause the system to misclassify it as

another command or response. For example, “okay Google” could be misclassified as “buy a cookie”.

Several methods exist for crafting adversarial examples (inputs that have been perturbed) to deceive neural networks, including DeepFool and the Jacobian-saliency algorithm. Most, if not all, methods revolve around using an optimization function that finds the minimal perturbation that would misclassify an input. Attacks fall under targeted or untargeted; the former requiring the input to be trained for a specific output as opposed to the latter (Liu et al., 2017). Moosavi-Dezfooli et al. proposed an iterative algorithm that produces universal adversarial perturbations for image recognition systems, a perturbation that can cause any input to be misclassified. It was proven that this is both input-agnostic and data-agnostic, meaning that the exact input that is to be perturbed and the dataset the victim system has been trained on is not of considerable importance. This implies a transferability that facilitates deception (Moosavi-Dezfooli et al., 2017). Papernot et al. developed an algorithm that makes a black-box attack possible, meaning that the attacker would not need to know the dataset or specific architecture of the system for exploitation to occur (Papernot et al., 2017). Similarly, Athalye et al. designed an algorithm that generates rotationally-invariant adversarial examples for image recognition systems, further aiding exploitation(Athalye et al., 2017).

II. Methodology

Purpose

In order to create speech recognition systems for applications resistant to exploitation, the vulnerabilities of speech recognition must be discovered and mitigated. The purpose of this study is to develop an attack on speech recognition systems using adversarial machine learning in order to highlight the vulnerabilities associated with the neural networks used in these systems.

Research Question

How can speech recognition systems be exploited using vulnerabilities in neural networks?

Threat Model

An adversary is assumed that obtains access to the speech recognition system after training is complete. It is also assumed that the adversary has adequate time to utilize the adversarial algorithm to create noise vectors. It should be noted that this adversarial algorithm does not need to be applied to the victim speech recognition system due to the principle of transferability. The adversary is also assumed to have the ability to add the noise vectors to the input of the victim speech recognition system. This is also assumed to be a black-box scenario.

Algorithm Design

Consider a neural network classifier F trained upon a dataset that can be split into x and y, or valid inputs and possible classes. The equation below can be used to model this.

$$F(x, y) = y$$

A targeted adversarial example must fulfill the obligation of effective misclassification, as well as imperceptibility (the ability for the adversary to evade detection). Effective

misclassification is defined as adding a sufficient amount of noise, v, to the input image that causes it to output the chosen target class, ŷ. The necessary action for this is defined as:

$$F((x+v),\ \hat{y}) = \hat{y}$$

As v increases, the amount of noise added to the input increases, and the imperceptibility of the adversarial example rises. Setting a limit to the noise vector, ς, accomplishes this goal. A limit can be set by ensuring that ς is greater than the Euclidean norm of v. The Euclidean norm is a function that assigns an input vector the strictly positive length of its arrow. In other words, the magnitude of the noise vector must be under a certain set limit. This condition is:

$$||v||_2 \leq \varsigma$$

Thus, an optimization problem is formulated. The problem is to find the smallest value for the Euclidean norm of the noise vector, such that this vector can be added to an input to misclassify it as a specific chosen output, that is:

$$\text{r(x)} = \text{argmin } ||v||_2 \leq \varsigma \text{ subject to } F((x+v),\ \hat{y}) = \hat{y}$$

The loss of this function can be defined as ζ_r. This can be formulated as an equation, wherein ζ_F is the loss from the neural network function described earlier, ω is a preset parameter that exemplifies the importance of the objective of inconspicuity in comparison to effectiveness, and the function L2 is defined as half of the Euclidean norm of the input.

$$\zeta_r = \zeta_F\ (\omega\ (\text{L2}(\text{x} + \text{v}))$$

By manipulating the above equation, the aforementioned optimization problem can be solved in order to misclassify a given input as a given output over a number of iterations. However, another issue becomes significant: the problem of precision during noise addition. In real-time scenarios, the exact moment noise can be added to an input cannot be precisely

calculated. Athalye et al. introduced the Expectation over Transformation algorithm, which can be manipulated to resolve this issue(Athalye et al., 2017). Thus, gradient sampling, as well as the incorporation of transformed data, should be taken into account.

$$\text{r(x)} = \text{argmin } ||v||_2 \leq \varsigma \text{ subject to } \frac{1}{k}\sum_{i=1}^{k} F((x+v),\ \hat{y}) = \hat{y}$$

The development of universal adversarial perturbations would allow an adversary to prepare for the attack beforehand, as opposed to attempting to conduct a conspicuous real-time attack. Based upon Moosavi-Dezfooli et al.'s method, the perturbation vector can be calculated, then aggregated for a number of data points. A new equation can be devised for this step if P_p is assumed to be a projection operator that concerns the minimization of the perturbation.

$$v \leftarrow P_p(v + v_i)$$

This attack can be further improved by making use of the principle of transferability to generate adversarial examples based upon a substitute neural network, permitting the adversary to implement a black-box attack. Thus, the algorithm is as follows:

Computing Universal, Transformable Perturbation Vectors for Target Class:

Input: Data X with data points x_i (not necessarily in the victim system), neural network F (can be a substitute network as opposed to the victim network) , norm of perturbation ς , desired target class $\hat{y}$, maximum iterations I

Output: Universal, transformable noise vector v

1. Initialize $v \leftarrow 0$
2. Initialize $i \leftarrow 0$
3. While i $\leq$ I

4. For each datapoint x_i do

5. If $F((x+v),\ \hat{y}) \neq \hat{y}$

6. Compute the minimal perturbation that sends new input to the decision boundary while incorporating gradient sampling for a distribution of transformations T:

$$\mathrm{r(x)} = \mathrm{argmin}\ ||v||_2 \leq \varsigma \text{ subject to } \frac{1}{k}\sum_{i=1}^{k} F((x+v),\ \hat{y}) = \hat{y}$$

7. Update the perturbation:

$$v \leftarrow P_p(v + v_i)$$

8. End if

9. End for

10. End while

This algorithm can be implemented upon a substitute neural network to generate adversarial noise. This adversarial noise can then be added to any input to mislead the victim neural network.

III. Evaluation and Results

Instrumentation

The neural networks and the attack algorithms were programmed using Python 2.7, TensorFlow, and NumPy. They were designed, developed, and tested using Jupyter Notebook running on NVIDIA GPUs. Google Sheets was also used to record the data.

Procedure

First, the TIMIT dataset was preprocessed and normalized. Next, a fully connected neural network with 5 layers and 600 hidden neurons each was programmed. This network was trained on a subset of the TIMIT dataset for 200000 iterations using the Adagrad optimizer. Next, another network with the same architecture under the same settings was programmed and trained on a different subset of the TIMIT dataset. Subsequently, validation sets were constructed for five randomly chosen class labels. The proposed algorithm was then applied to each validation set for 100000 iterations. The maximum accuracy for each was recorded.

Results

Validation Set	Validation Accuracy
1	64.08%
2	60.25%
3	59.25%
4	63.25%
5	55.25%
Average	60.42%

On average, this method for crafting universal, transformable perturbations in a black-box setting has an accuracy of 60.42%.

IV. Conclusion

In this study, a speech recognition system was deceived by crafting perturbation vectors for inputs that are both universal and transformable for a specific target output, meaning that each generated vector can be added to any input for that output to occur, and an impractical level of precision is not required when implementing this noise. These vectors were developed using a neural network separate from the victim neural network to simulate a black-box situation. The method yielded an average accuracy of 60.42%. Thus, the neural networks in speech recognition systems are a significant vulnerability that can be exploited by malicious agents. It is imperative that defenses are developed to mitigate attacks such as the one developed.

V. Discussion and Future Work

A major limitation of the evaluation experiment itself is time constraint. The evaluation experiment could have yielded more veracious data if a real-time attack environment was simulated with applications developed for speech recognition and exploitation. Additionally, end-to-end speech recognition systems could have been developed for more accuracy. Several other data sets could have been chosen for training as opposed to merely using TIMIT, such as Mozilla's Project Common Voice or VoxForge.

Future work can focus on developing defenses for adversarial exploits such as this one. Also, algorithms that would enable real-time exploitation with less required preparation could be developed. Furthermore, an attack could be developed that leverages both hidden Markov models and neural networks to exhibit the vulnerabilities of these algorithms when used in conjunction. More research should be done on specific applications related to speech recognition, such as voice identification and biometric authentication for more comprehensive attacks.

VI. References

Carlini, N. et al.(2016). Hidden Voice Commands. 25th USENIX Security Symposium, 513-530. Retrieved from https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/carlini

Gong, Y., & Poellabauer, C. (2017). Crafting Adversarial Examples For Speech Paralinguistics Applications. *CoRR,Abs/1711.03280*. Retrieved from http://dblp.org/rec/bib/journals/corr/abs-1711-03280

Liu, Y., Weiming, Z., Shaohua, L., & Nenghai, Y. (2017). Enhanced Attacks on Defensively Distilled Deep Neural Networks. *ArXiv e-prints*.

Lopes, C., & Perdigao, F. (2011). Phoneme Recognition on the TIMIT Database. In *Speech Technologies*. Retrieved from http://www.intechopen.com/books/speech-technologies/phoneme-recognition-on-the-timit-database

Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., & Frossard, P. (2017). Universal Adversarial Perturbations. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi:10.1109/cvpr.2017.17

Niedek, T. V. (2016). Phonetic Classification in TensorFlow (Bachelor's thesis). Radboud University.

Papernot, N., Mcdaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical Black-Box Attacks against Machine Learning. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security - ASIA CCS 17*. doi:10.1145/3052973.3053009

Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2016). Accessorize to a Crime. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS16*. doi:10.1145/2976749.2978392

Song, L., & Mittal, P. (2017). Inaudible Voice Commands. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS 17*. doi:10.1145/3133956.3138836

Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., & Xu, W. (2017). Dolphin Attack. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security - CCS 17*. doi:10.1145/3133956.3134052