A New Method for the Exploitation of Speech Recognition Systems

Introduction

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 in fields as diverse as healthcare, robotics, home automation, and education. 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.

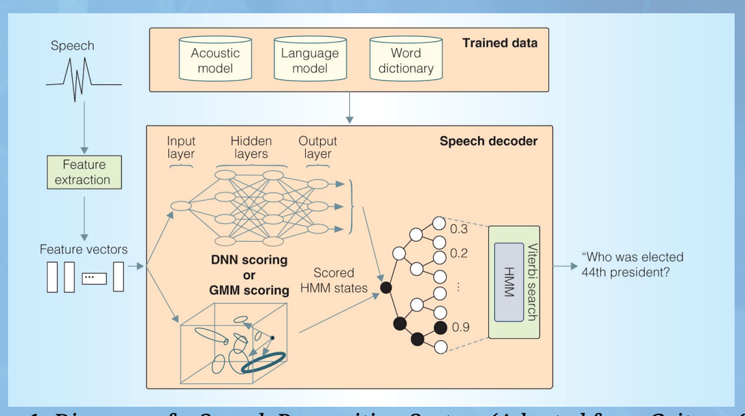


Figure 1: *Diagram of a Speech Recognition System (Adapted from Geitgey, 2016)*

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. One notable data set used for phonetic classification is 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).

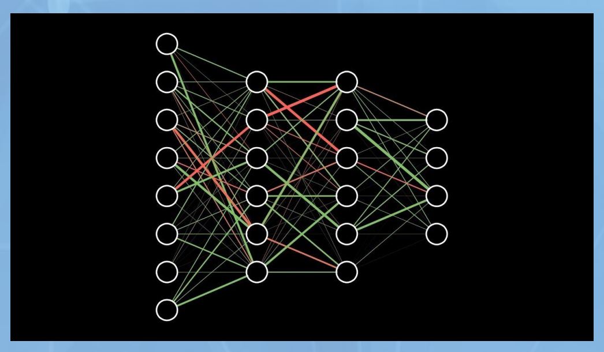


Figure 2: *Depiction of a Neural Network (Adapted from Sanderson, 2017)*

Previous methods of speech recognition exploitation did not address the **vulnerabilities** caused by the incorporation of neural networks. Carlini et al. developed unintelligible commands for attackers by developing an algorithm that leveraged preprocessing algorithms. These commands were only developed for hidden Markov model systems, 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). Zhang et al. developed a system known as "Dolphin Attack" to exploit the nonlinearity of microphone circuits. This attack was limited in that it was specific to the hardware of that device, and was greatly affected by factors such as modulation depth (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**. Several methods exist for crafting **adversarial examples** (inputs that have been perturbed) to deceive neural networks (Liu et al., 2017). Moosavi-Dezfooli et al. proposed an iterative algorithm that produces universal adversarial perturbations, a perturbation that can cause any input to be misclassified. It was proven that this is both input-agnostic and data-agnostic. 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).

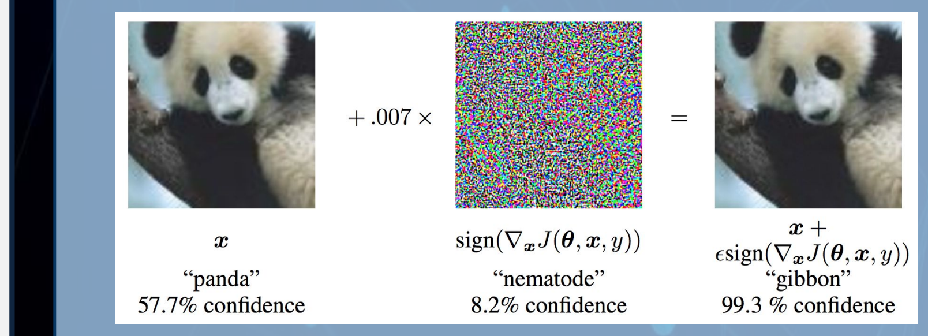


Figure 3: *Demonstration of the Creation of an Adversarial Example (Adapted from Papernot et al., 2017)*

Purpose

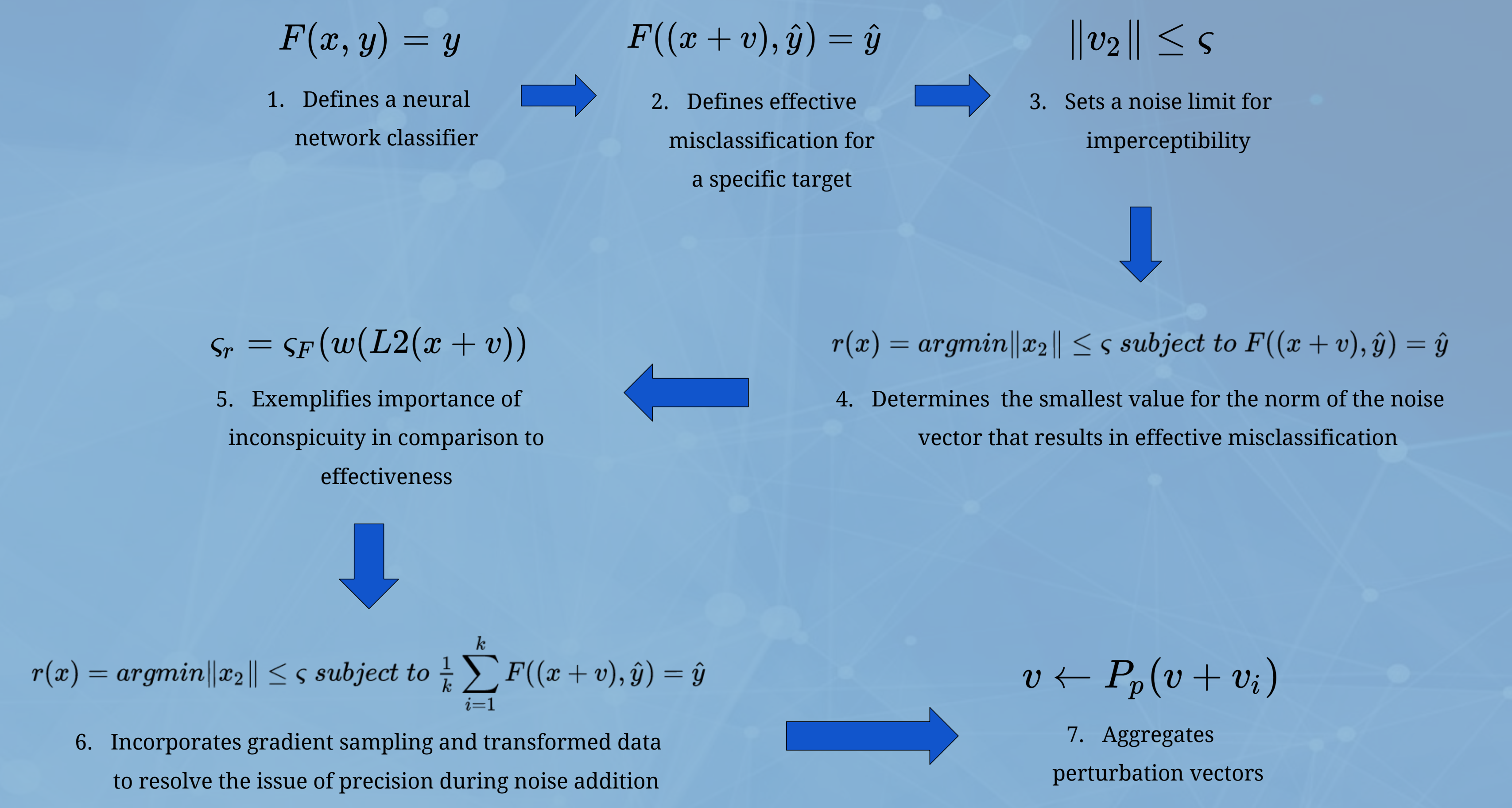
In order to create speech recognition systems for applications resistant to exploitation, the vulnerabilities of speech recognition must be discovered and mitigated. The purpose of this study is to **develop an attack on speech recognition systems using adversarial machine learning** in order to highlight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

Methodology

Threat Model

1. The adversary obtains access to the system after training is complete.
2. The adversary has adequate time to create noise vectors from a substitute system.
3. The adversary can add noise vectors to the input of the victim system.
4. This is a black-box scenario.

Algorithm Design



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

1. Defines a neural network classifier

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

2. Defines effective misclassification for a specific target

$$\|v_2\| \leq \varsigma$$

3. Sets a noise limit for imperceptibility

$$r(x) = argmin\|x_2\| \leq \varsigma \; subject\ to\ F((x+v), \hat{y}) = \hat{y}$$

4. Determines the smallest value for the norm of the noise vector that results in effective misclassification

$$\varsigma_r = \varsigma_F(w(L2(x+v))$$

5. Exemplifies importance of inconspicuity in comparison to effectiveness

$$r(x) = argmin\|x_2\| \leq \varsigma \; subject\ to\ \frac{1}{k}\sum_{i=1}^{k} F((x+v), \hat{y}) = \hat{y}$$

6. Incorporates gradient sampling and transformed data to resolve the issue of precision during noise addition

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

7. Aggregates perturbation vectors

__Computing Universal, Transformable Perturbation Vectors for a Target Class:__

Input: Data X with data points x_i (not necessarily in the victim system), neural network F (substitute system), norm of perturbation ς, desired target class ŷ, maximum iterations I

Output: Universal, transformable noise vector v

```
Initialize $v \leftarrow 0$
Initialize $i \leftarrow 0$
While $i \leq I$
    For each datapoint do
        If $F((x+v), \hat{y}) = \hat{y}$ is not true
            Compute the minimal perturbation that sends input to the decision boundary while incorporating gradient sampling:
            $r(x) = argmin\|x_2\| \leq \varsigma \; subject\ to\ \frac{1}{k}\sum_{i=1}^{k} F((x+v), \hat{y}) = \hat{y}$
            Update the perturbation:
            $v \leftarrow P_p(v + v_i)$
        End if
    End for
End while
```

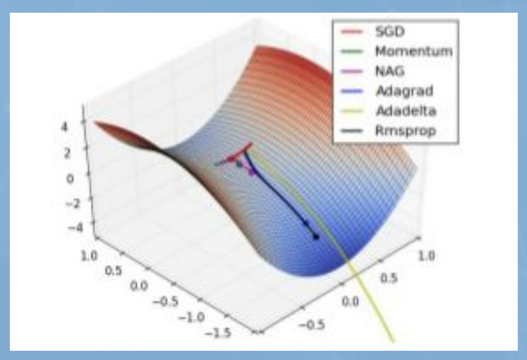


Figure 4 : *Depiction of Optimization (Adapted from Walia et al., 2017)*

Evaluation and Results

Instrumentation

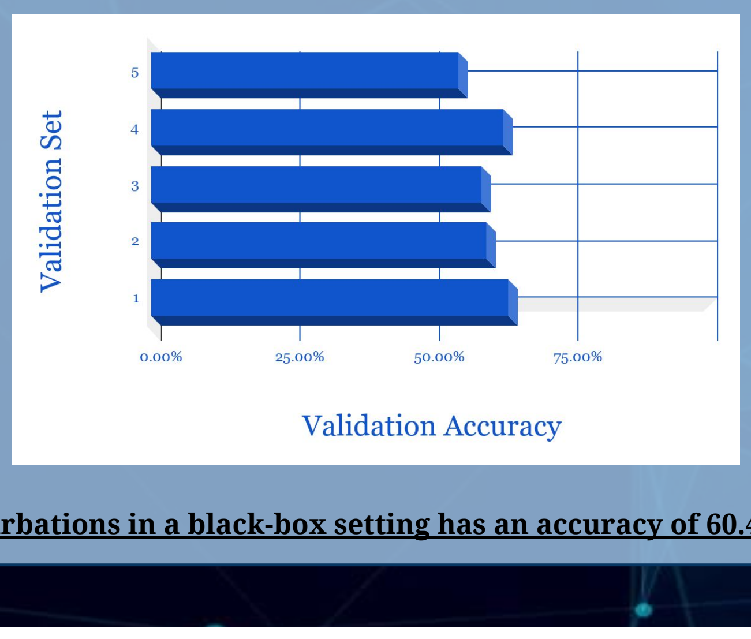
The neural networks and the attack algorithms were programmed using Python 2.7, TensorFlow, and NumPy. They were designed, developed, and tested using Jupyter Notebook running on NVIDIA GPUs. Google Sheets was also used to record the data.

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%.__

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.

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.

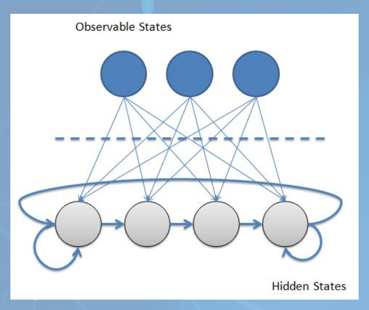


Figure 5: *Diagram of a hidden Markov model (Adapted from Souza, 2010)*

References

Athalye, Anish, Engstrom, Logan, Ilyas, Andrew, and Kwok, Kevin. Synthesizing robust adversarial examples. arXiv preprint arXiv:1707.07397, 2017.

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