UNIVERSITY OF CALIFORNIA SAN DIEGO

Blind Spots of Neural Sequence Models

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Paarth Neekhara

Committee in charge:

Professor Shlomo Dubnov, Chair
Professor Julian McAuley
Professor Gary Cottrell

2019

The thesis of Paarth Neekhara is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

Chair

University of California San Diego

2019

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT OF THE THESIS

Blind Spots of Neural Sequence Models

by

Paarth Neekhara

Master of Science in Computer Science

University of California San Diego, 2019

Professor Shlomo Dubnov, Chair

Deep neural networks (DNNs) serve as a backbone of many image, language and speech processing systems. Such models are being deployed extensively in personal devices, cloud based applications and automated security services like face recognition, speaker identification etc. While DNNs have shown to achieve state of the art results in their respective domains, recent studies have exposed the vulnerabilities of these models to adversarial attacks. The work on adversarial examples has primarily focused on the domain of images.

In this work, we explore the vulnerabilities of neural networks working on sequential data like text and audio. We propose a novel method to repurpose text classification networks for alternate tasks. This gives incentive to adversaries to steal computational resources from a

system provider. An adversary in such an attack scenario can potentially train a simple input transformation for discrete sequences for repurposing the victim model for a new classification task.

We also study the existence of universal adversarial perturbations for Automatic Speech Recognition (ASR) Systems. We propose an algorithm to find a single quasi-imperceptible perturbation, which when added to any arbitrary speech signal, will most likely fool the victim speech recognition model. Our experiments demonstrate the application of our proposed technique by crafting audio-agnostic universal perturbations for the state-of-the-art ASR system – Mozilla DeepSpeech. Additionally, we show that such perturbations generalize to a significant extent across models that are not available during training, by performing a transferability test on a WaveNet based ASR system.

For example, a carefully designed imperceptible perturbation in an image can cause a victim image classification model to mis-classify the image. Such attacks target the ”blind spots” of neural networks input domain. In this work, we focus on exposing such blind spots in neural sequence models for language and speech.

ACKNOWLEDGEMENTS

Chapter 2, in full, is a reprint of the material as it appears in AAAI 2019 workshop on Engineering Dependable and Secure Machine Learning Systems. Neekhara, Paarth; Hussain, Shehzeen; Dubnov, Shlomo; Koushanfar, Farinaz. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in the supplementary DSN 2019 proceedings. Neekhara, Paarth; Hussain, Shehzeen; Pandey, Prakhar; Dubnov, Shlomo; McAuley, Julian, Koushanfar, Farinaz. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 1

Introduction: Adversarial Examples

Deep neural networks (DNN) are being extensively deployed as image, language and speech processing systems in personal devices, cloud based applications and automated security services. While DNNs have shown to achieve state of the art results in their respective domains, recent studies have exposed the vulnerabilities of these models to adversarial attacks. For example, a carefully designed imperceptible perturbation in an image can cause a victim image classification model to mis-classify the image. Such attacks target the "blind spots" of neural networks input domain. In this work, we focus on exposing such blind spots in neural sequence models for language and speech.

In this chapter, we introduce adversarial examples and vulneribilities of neural networks. We go over some of the prior work in the domain of adversarial attacks and adversarial reprogramming. Section 1.6 covers existing work in the domain of adversarial attacks on Speech Reocgnition System. This chapter lays the necessary background for our proposed adversarial attacks in Chapter 2 and Chapter 3.

1.1 Adversarial Examples

Adversarial examples are intentionally designed inputs to a machine learning model that cause the model to make a mistake [23]. These attacks can be broadly classified into *untargeted* and *targeted* attacks. In the untargeted attack scenario, the adversary succeeds as long as the victim model classifies the adversarial input into *any* class other than the correct class, while in the *targeted* attack scenario, the adversary succeeds only if the model classifies the adversarial input into a specific incorrect class. In both these scenarios, the intent of the adversary is usually malicious and the outcome of the victim model is still limited to the original task being performed by the model.

Adversarial attacks of image-classification models often use gradient descent on an image to create a small perturbation that causes the machine learning model to mis-classify it [58, 16]. There has been a similar line of adversarial attacks on neural networks with discrete input domains [48, 65], where the adversary modifies a few tokens in the input sequence to cause misclassification by a sequence model. In addition, efforts have been made in designing more general adversarial attacks in which the same modification can be applied to many different inputs to generate adversarial examples [17, 23, 42]. For example, authors [12] trained an Adversarial Transformation Network that can be applied to all inputs to generate adversarial examples targeting a victim model or a set of victim models. In this work, we aim to learn such universal transformations of discrete sequences for a fundamentally different task: *Adversarial Reprogramming* described below.

1.2 Untrageted vs Targeted Attacks

In untargeted attacks, the goal of the adversary is to cause mis-prediction by the victim model. let $l(x)$ denote the label produced by a victim model for an input x, the goal of the adversary is to design an adversarial input x' which is perceived as indistinguishable from the

original input x but causes mis-classification i.e $l(x) \neq l(x')$. In targeted attacks, the goal of the adversary is to design an adversarial input x' which is perceived as indistinguishable from the original input x, and maps to a target label t i.e $l(x) = t$.

1.3 Transferability of Adversarial Samples

Adversarial sample transferability is the property that adversarial samples produced by training on a specific model can affect another model, even if they have different architectures. Since in case of black-box attack, adversary does not have access to the target model F, an attacker can train a substitute model F' locally to generate adversarial example $x + \delta$ which then can be transferred to the victim neural network. While there have been many studies conducted on the transferability of adversarial examples in the image domain [55, 45, 13, 59, 39], but to the best of our knowledge similar efforts have not been applied in the audio domain.

1.4 Adversarial Reprogramming

Adversarial Reprogramming [22] introduced a new class of adversarial attacks where the adversary wishes to repurpose an existing neural network for a new task chosen by the attacker, without the attacker needing to compute the specific desired output. The adversary achieves this by first defining a hard-coded one-to-one label remapping function h_g that maps the output labels of the adversarial task to the label space of the classifier f; and learning a corresponding adversarial reprogramming function $h_f(.;\theta)$ that transforms an input $(\tilde{X})$ [1] from the input space of the new task to the input space of the classifier. The authors proposed an adversarial reprogramming function $h_f(.;\theta)$, for repurposing ImageNet models for adversarial classification tasks. An adversarial example X_{adv} for an input image $\tilde{X}$ can be generated using the

[1] $\tilde{X}$ is an ImageNet size $(n \times n \times 3)$ padded input image

following adversarial program: [2]

$$X_{adv} = h_f(\tilde{X};\theta) = \tilde{X} + \tanh(\theta)$$

where $\theta \in \mathbb{R}^{n \times n \times 3}$ is the learnable weight matrix of the adversarial program (where n is the ImageNet image width). Let $P(y|X)$ denote the probability of the victim model predicting label y for an input X. The goal of the adversary is to maximize the probability $P(h_g(y_{adv})|X_{adv})$ where y_{adv} is the label of the adversarial input X_{adv}. The following optimization problem that maximizes the log-likelihood of predictions for the adversarial classification task, can be solved using backpropagation to train the adversarial program parameterized by θ:

$$\hat{\theta} = argmin_\theta \left(-\log P(h_g(y_{adv})|X_{adv}) + \lambda ||\theta||_2^2 \right) \tag{1.1}$$

where λ is the regularization hyperparameter. Since the adversarial program proposed is a trainable additive contribution θ to the inputs, it's application is limited to neural networks with a continuous input space. Also, since the the above optimization problem is solved by back-propagating through the victim network, it assumes a white-box attack scenario where the adversary has gained access to the victim model's parameters.

In our work, we will describe how we can learn a simple transformation in the *discrete space* to extend the application of adversarial reprogramming on *sequence classification* problems. We also propose a training algorithm in the black-box setting where the adversary may not have access to the model parameters.

[2] Masking ignored because it is only a visualization convenience

1.5 Universal Adversarial Perturbations

In [42], the authors try to find an universal perturbation vector which can fool the network to predict a false classification output on most of the validation instances. Let $\hat{k}(x)$ be the classification output for an input x and let x be distributed according to μ then they propose that we want to find a universal perturbation v such that:

$\hat{k}(x+v) \neq \hat{k}(x)$ for "most" $x \tilde{} \mu$.

They solve this problem as an optimization problem with constraints which ensure that the universal perturbation obtained has the smallest possible p-norm and will also be able to fool the desired number of instances in the training set. The interesting thing about this paper is that they show that only training their model over small number of instances (e.g. 500 examples) can fool the networks on about 30% of the cases in the validation set. Also they show that the universal perturbation produced using one network say VGG-16 can also be used to fool other network say GoogLeNet showing that their method is doubly universal.

1.6 Adversarial Attacks on Speech Recognition Systems

In this section we discuss some prior work on adversarial attacks in the audio domain. The goal of these works is to design an imperciptle audio perturbation which when added to an audio signal causes mis-transcription or mis-classification by a neural speech reocgnition model.

1.6.1 Audio Adversarial Examples: Targeted Attacks on Speech-to-Text

In this paper [18], the authors generate targeted audio adversarial examples for automatic speech recognition systems that are end-to-end. Their white-box iterative optimization-based attack achieves 100 % success rate on Mozillas open source Speech-To-Text engine DeepSpeech [29], which is a state-of-the-art speech-to-text transcription neural network. Given any natural

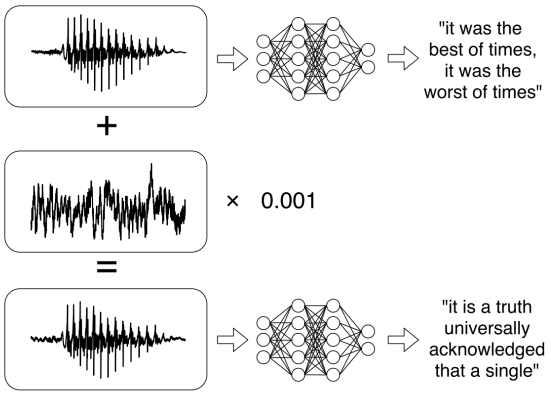

Figure 1.1: Targeted Adversarial attack on Speech Recognition System [18]

waveform x, they are able to construct a perturbation δ that is nearly inaudible, but so that $x+\delta$ is recognized as any desired phrase by a victim neural network. The key differences between this work and a prior work [20] by the same authors, is that in their prior efforts they only targeted traditional systems such as HMMs and GMMs, using obfuscated examples and they do not operate on end-to-end neural networks. Obfuscated examples means that the examples sound like random noise rather than normal human perceptible speech, which makes attacks using obfuscated examples easier.

1.6.2 Generating Adversarial Examples for Speech Recognition

The authors of [31] demonstrate successful attacks on neural ASR systems based on WaveNet [61], using fast gradient sign method [23]. The authors note that ASRs rely on the Mel Frequency Cepstral Coefficients (MFCCs) as features of the input audio data. In this attack, the adversary designs perturbations on MFCC (mel spectogram) representation instead of the raw audio waveforms as done in [18] and described in Section 1.6.1. The adversary then decodes raw audio from the MFCC representation and generated adversarial examples for the victim model. The auhtors demonstrated their attack on the WaveNet model for speech recognition.

1.6.3 Did you hear that? Adversarial Examples Against Automatic Speech Recognition

This paper [6] focuses on generating adversarial noise to perform targeted attacks on Automatic Speech Recognition systems (ASRs) in a black-box setting where the attacker knows nothing about the model architecture and parameter values, but is capable of querying the model results. The authors argue that using backpropagation and other gradient based methods to generate adversarial noise, are not easily applicable to speech recognition models.

As previously stated, ASRs rely on the Mel Frequency Cepstral Coefficients (MFCCs) as features of the input audio data. To avoid differentiating through MFCC computations, the authors propose a genetic algorithm which is a gradient-free optimization method. The genetic algorithm based method does not require knowledge of the victim model architecture or parameters and can therefore be utilized to perform black-box attacks where the attackers do not have access to model parameters and architectures.

Chapter 2

Adversarial Reprogramming of Text Classification Neural Networks

2.1 Introduction

Adversarial Reprogramming [22] is a new class of adversarial attacks where a machine learning algorithm is repurposed to perform a new task chosen by the attacker. The authors demonstrated how an adversary may repurpose a pre-trained ImageNet [21] model for an adversarial classification task like classification of MNIST digits or CIFAR-10 images without modifying the network parameters. Since machine learning agents can be reprogrammed to perform unwanted actions as desired by the adversary, such an attack can lead to theft of computational resources such as cloud-hosted machine learning models. Besides theft of computational resources, the adversary may perform a task that violates the code of ethics of the system provider.

The adversarial reprogramming approach proposed by [22] trains an additive contribution θ to the inputs of the neural network to repurpose it for the desired alternate task. The adversary defines a hard-coded mapping between the class labels of the original and adversarial task. The adversarial program parameterized by θ is updated such that the classifier predicted label, when

mapped to the adversarial label space, correctly classifies an adversarial input. This approach assumes a white-box attack scenario where the adversary has access to the network's parameters. Also, the adversarial program proposed in this work is only applicable to tasks where the input space of the the original and adversarial task is continuous.

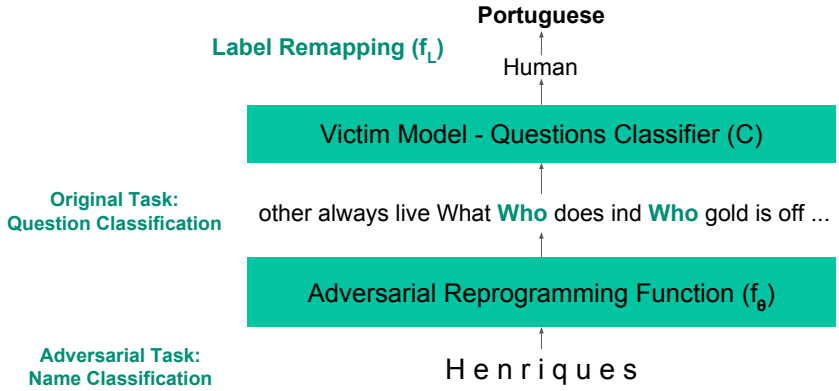

Figure 2.1: Example of Adversarial Reprogramming for Sequence Classification. We aim to design and train the adversarial reprogramming function f_θ, such that it can be used to repurpose a pretrained classifier C, for a desired adversarial task.

In this work, we propose a method to adversarially *repurpose* neural networks which operate on sequences from a *discrete* input space. The task is to learn a simple transformation (adversarial program) from the input space of the adversarial task to the input space of the neural network such that the neural network can be repurposed for the adversarial task. We propose a context-based vocabulary remapping function as an adversarial program for sequence classification networks. We propose training procedures for this adversarial program in both white-box and black-box scenarios. In the white-box attack scenario, where the adversary has access to the classifier's parameters, a Gumbel-Softmax trick [32] is used to train the adversarial program. Assuming a black-box attack scenario, where the adversary may not have access to the classifier's parameters, we present a REINFORCE [63] based optimization algorithm to train the adversarial program.

We apply our proposed methodology on various text classification models including Recurrent Neural Networks such as LSTMs and bidirectional LSTMs, and Convolutional Neural

Networks (CNNs). We demonstrate experimentally, how these neural networks trained on a particular (original) text classification task can be repurposed for alternate (adversarial) classification tasks. We experiment with different text classification datasets given in table 2.1 as candidate original and adversarial tasks and adversarially reprogram the aforementioned text classification models to study the robustness of the attack.

2.2 Methodology

2.2.1 Adversarial Reprogramming Problem Definition

Consider a sequence classifier C trained on the original task of mapping a sequence $s \in S$ to a class label $l_S \in L_S$ i.e $C : s \mapsto l_S$. An adversary wishes to repurpose the original classifier C for the adversarial task C' of mapping a sequence $t \in T$ to a class label $l_T \in L_T$ i.e $C' : t \mapsto l_T$. The adversary can achieve this by hard-coding a one-to-one label remapping function:

$$f_L : l_S \mapsto l_T$$

that maps an original task label to the new task label and learning a corresponding adversarial reprogramming function:

$$f_\theta : t \mapsto s$$

that transforms an input from the input space of the adversarial task to the input space of the original task. The adversary aims to update the parameters θ of the adversarial program f_θ such that the mapping $f_L(C(f_\theta(t)))$ can perform the adversarial classification task $C' : t \mapsto l_T$.

2.2.2 Adversarial Reprogramming Function

The goal of the adversarial reprogramming function $f_\theta : t \mapsto s$ is to map a sequence t to s such that it is labeled correctly by the classifier $f_L(C)$.

The tokens in the sequence s and t belong to some vocabulary lists V_S and V_T respectively. We can represent the sequence s as $s = s_1, s_2, .., s_N$ where s_i is the vocabulary index of the i_{th} token in sequence s in the vocabulary list V_S. Similarly sequence t can be represented as $t = t_1, t_2, .., t_N$ where t_i is the vocabulary index of the i_{th} token of sequence t in the vocabulary list V_T.

In the simplest scenario, the adversary may try to learn a vocabulary mapping from V_T to V_S using which each t_i can be independently mapped to some s_i to generate the adversarial sequence. Such an adversarial program has limited potential since the representational capacity of such a reprogramming function is very limited. We experimentally support this hypothesis by showing how such a transformation has limited potential for the purpose of adversarial reprogramming.

A more sophisticated adversarial program can be a sequence to sequence machine translation model [57] that learns a translation $t \mapsto s$ for adversarial reprogramming. While theoretically this is a good choice, it defeats the purpose of adversarial reprogramming. This is because the computational complexity of training and using such a machine translation model would be similar if not greater than that of a new sequence classifier for the adversarial task C'.

The adversarial reprogramming function should be computationally inexpensive but powerful enough for adversarial repurposing. To this end, we propose a context-based vocabulary remapping model that produces a distribution over the target vocabulary at each time-step based on the surrounding input tokens. More specifically, we define our adversarial program as a trainable 3-d matrix $\theta_{k \times |V_T| \times |V_S|}$ where k is the context size. Using this, we generate a probability

distribution π_i over the vocabulary V_S at each time-step i as follows:

$$h_i = \sum_{j=0}^{k-1} \theta[j, t_{i+\lfloor k/2 \rfloor - j}] \tag{2.1}$$

$$\pi_i = softmax(h_i) \tag{2.2}$$

Both h_i and π_i are vectors of length $|V_S|$. To generate the adversarial sequence s we sample each s_i independently from the distribution π_i.

$$s_i \sim \pi_i$$

Given the max input length N accepted by the victim model, the input sequence t is padded with $\lfloor k/2 \rfloor$ instances of a dummy token before the first token and $N - length(t) + \lfloor k/2 \rfloor$ instances after the last token to generate an N length output s . For sequences with $length(t) > N$, we select the first N tokens of t as input to the adversarial reprogramming function f_θ. We demonstrate in the Experiments section, that this approach works for different combinations of adversarial and original tasks with different average sequence lengths.

In practice, we implement this adversarial program as a single layer of 1-d convolution over the sequence of one-hot encoded vectors of adversarial tokens t_i with $|V_T|$ input channels and $|V_S|$ output channels with k-length kernels parameterized by $\theta_{k \times |V_T| \times |V_S|}$. Note that the time-complexity of using this adversarial reprogramming function (equations 2.1,2.2) is just $O(k \times |V_S| \times length(t))$ and it can be parallelized to improve further.

2.2.3 White-box Attack

In the white-box attack scenario, we assume that the adversary has gained access to the victim network's parameters and architecture. Let $P(l|s)$ denote the probability of predicting label

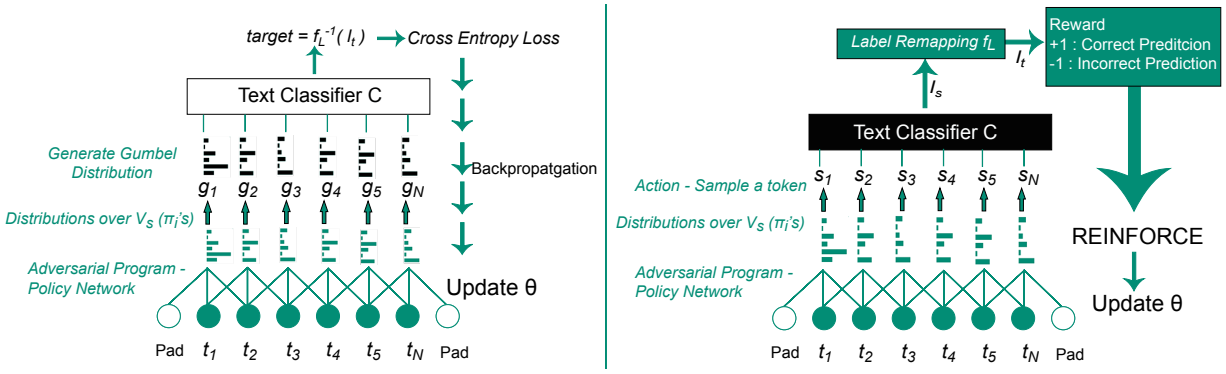

Figure 2.2: **Adversarial Reprogramming Function and Training Procedures. Left:** *White-box* Adversarial Reprogramming. The adversary generates gumbel distributions g_i at each time-step which are passed as a soft version of one-hot vectors to the classifier C. The cross-entropy loss between the predictions and the mapped class is backpropagated to train the adversarial program θ. **Right:** *Black-box* Adversarial Reprogramming. The adversarial reprogramming function is used as a policy network and the sampled action (sequence s) is passed to the classifier *C* to get a reward based on prediction correctness. The adversarial program is then trained using REINFORCE.

l for a sequence s by classifier C. We wish to maximize the probability $P(f_L^{-1}(l_t)|f_\theta(t))$ which is the probability of the output label of the classifier being mapped to the correct class l_t for an input t in the domain of the adversarial task. Therefore we need to solve the following log-likelihood maximization problem:

$$\hat{\theta} = argmin_\theta(-\sum_t log(P(f_L^{-1}(l_t)|f_\theta(t)))) \tag{2.3}$$

Note that that the output of the adversarial program $s = f_\theta(t)$ is a sequence of discrete tokens. This makes the above optimization problem non-differentiable. Prior works [37, 27, 65] have demonstrated how we can smoothen such an optimization problem using the Gumbel-Softmax [32] distribution.

In order to backpropagate the gradient information from the classifier to the adversarial program, we smoothen the generated tokens s_i using Gumbel-Softmax trick as per the following:

For an input sequence t, we generate a sequence of Gumbel distributions $g = g_1, g_2, .., g_N$.

The n_{th} component of distribution g_i is generated as follows:

$$g_i^n = \frac{\exp((\log(\pi_i^n) + r_n)/temp)}{\sum_j \exp((\log(\pi_i^j) + r_j)/temp)}$$

where π_i is the softmax distribution at the i_{th} time-step obtained using equation 2.2, r_n is a random number sampled from the Gumbel distribution [28] and $temp$ is the temperature of Gumbel-Softmax.

Gumbel-Softmax approximates one-hot vectors of s_i's with differentiable representations. The temperature parameter controls the flatness of this distribution. As $temp \to 0$ the Gumbel distribution becomes close to a one-hot vector and as $temp \to \infty$ the Gumbel distribution assumes a uniform distribution over $|V_S|$ variables. The sequence then passed to the classifier C is the sequence g which serves as a soft version of the one-hot encoded vectors of s_i's. Since the model is now differentiable, we can solve the following optimization problem using backpropagation:

$$\hat{\theta} = argmin_\theta(-\sum_t log(P(f_L^{-1}(l_t)|g))) \tag{2.4}$$

During training the temperature parameter is annealed from some high value t_{max} to a very low value t_{min}. The details of this annealing process for our experiments have been included in the supplementary material.

2.2.4 Black-box Attack

In the black-box attack scenario, the adversary can only query the victim classifier C for labels. Since the adversarial program needs to produce a discrete output to feed as input to the classifier C, it is not possible to pass the gradient update from the classifier $f_L(C)$ to the adversarial program θ using standard back-propagation. Also, in the black-box attack setting it is not possible to back-propagate the cross entropy loss through the classifier C in the first place.

We formulate the sequence generation problem as a Reinforcement Learning problem [9, 10, 67] where the adversarial reprogramming function is the policy network. We define the state, action, policy and reward for this problem as follows:

- **State and Action Space:** The state of the adversarial program is a sequence $t \in T$ where T is the input space of the adversarial task. An action of an RL agent is to produce a sequence of tokens $s \in S$ where S is the input space of the original task.

- **Policy:** The adversarial program parameterized by θ, models the stochastic policy $\pi_{adv}(s|t;\theta)$ such that a sequence $s \in S$ may be sampled from this policy conditioned on $t \in T$.

- **Reward:** We use a simple reward function where we assign a reward +1 for a correct prediction and -1 for an incorrect prediction using the classifier $f_L(C)$ where f_L is the label remapping function and C is the classifier. Formally:

$$r(t,s) = \begin{cases} +1, & f_L(C(s)) = l_t \\ -1, & f_L(C(s)) \neq l_t \end{cases}$$

The optimization objective to train the policy network is the following:

$$\max_{\theta} J(\theta) \quad \text{where,} \quad J(\theta) = \mathbb{E}_{\pi_{adv}}[r(t,s)]$$

Following the REINFORCE algorithm [63] we can write the gradient of the expectation with

respect to θ as per the following:

$$\begin{aligned}
\nabla_\theta J &= \nabla_\theta \left[\mathop{\mathbb{E}}_{\pi_{adv}} [r(t,s)] \right] \\
&= \nabla_\theta \left[\sum_s \pi_{adv}(s|t;\theta) r(t,s) \right] \\
&= \sum_s \pi_{adv}(s|t;\theta) \nabla_\theta \log(\pi_{adv}(s|t;\theta)) r(t,s) \\
&= \mathop{\mathbb{E}}_{\pi_{adv}} [r(t,s) \nabla_\theta \log(\pi_{adv}(s|t;\theta))] \\
&= \mathop{\mathbb{E}}_{\pi_{adv}} [r(t,s) \nabla_\theta \log(\pi_{adv}(s_1,..,s_N|t;\theta))] \\
&= \mathop{\mathbb{E}}_{\pi_{adv}} \left[r(t,s) \nabla_\theta \log(\prod_i \pi_{adv}(s_i|t;\theta)) \right] \\
&= \mathop{\mathbb{E}}_{\pi_{adv}} \left[r(t,s) \sum_i \nabla_\theta \log(\pi_{adv}(s_i|t;\theta)) \right]
\end{aligned}$$

Note that $\pi_{adv}(s_i|t;\theta)$ is the same as π_i defined in equation 2.2 which can be differentiated with respect to θ. The expectations are estimated as sample averages. Having obtained the gradient of expected reward, we can use mini-batch gradient ascent to update θ with a learning rate α as: $\theta \leftarrow \theta + \alpha \nabla_\theta J$.

2.3 Experiments

2.3.1 Datasets and Classifiers

We demonstrate the application of the proposed reprogramming techniques on various text-classification tasks. In our experiments, we design adversarial programs to attack both word-level and character-level text classifiers. Additionally, we aim to adversarially repurpose a character-level text classifier for a word-level classification task and vice-versa. To this end, we choose the following text-classification datasets as candidates for the original and adversarial classification tasks:

- *Surname Classification Dataset (**Names-18, Names-5**)[51]*: The dataset categorizes surnames from 18 languages of origin. We use this dataset for character-level classification task. We use a subset of this dataset *Names-5* containing Names from 5 classes: *Dutch, Scottish, Polish, Korean* and *Portuguese*, as a candidate for adversarial task in the experiments.

- *Experimental Data for Question Classification (**Questions**) [38]*: categorizes around 5500 questions into 6 classes: *Abbreviation, Entity, Description, Human, Location, Numeric*. We divide this dataset into 4361 questions for training and 1091 for testing.

- ***Arabic Tweets** Sentiment Classification Dataset [3]*: contains 2000 binary labeled tweets on diverse topics such as politics and arts. The tweets in this dataset, comprising of 1000 positive and 1000 negative tweets, are written in Modern Standard Arabic (MSA) and the Jordanian dialect. We use 1600 samples for training and 400 for testing.

- *Large Movie Review Dataset **(IMDB)** for sentiment classification [40]*: contains 50,000 movie reviews categorized into binary class of positive and negative sentiment. It is split into 25,000 reviews for training and 25,000 reviews for testing.

The statistics of the above mentioned datasets have been given in table 2.1. We train adversarial reprogramming functions to repurpose various text-classifiers based on Long Short-Term Memory (LSTM) network [30], bidirectional LSTM network [25] and Convolutional neural network [33] models. All the aforementioned models can be trained for both word-level and character-level classification. We use character level classifiers for *Names-18* and *Names-5* datasets and word-level classifiers for *IMDB*, *Questions* and *Arabic Tweets* datasets. We use randomly initialized word/character embeddings for all the classification models. For LSTM, we use the output at last timestep for prediction. For the Bi-LSTM, we combine the outputs of the first and last time step for prediction. For the Convolutional Neural Network we follow the same architecture as [33]. The hyper-parameter details of these classifiers have been included in table 2 of the supplementary material.

Table 2.1: Summary of datasets and test accuracy of original classification models. $|V|$ denotes the vocabulary size of each dataset. Note that we use character-level models for *Names-5* and *Names-18* and word-level models for all other tasks.

Data Set	# Classes	Train Samples	Test Samples	$\|V\|$	Avg Length	Test Accuracy (%)		
						LSTM	Bi-LSTM	CNN
Names-18	18	115,028	28,758	90	7.1	97.84	97.84	**97.88**
Names-5	5	3632	909	66	6.5	**99.88**	**99.88**	99.77
Questions	6	4361	1091	1205	11.2	96.70	**98.25**	98.07
Arabic Tweets	2	1600	400	955	9.7	87.25	**88.75**	88.00
IMDB	2	25,000	25,000	10000	246.8	86.83	89.43	**90.02**

2.3.2 Experimental Setup

As described in the methodology section, the label remapping function f_L we use, is a one-to-one mapping between the labels of the original task and the adversarial task. Therefore it is required to apply the constraint that the number of classes of the adversarial task are less than or equal to the number of classes of the original task. We choose *Names-5, Arabic Tweets* and *Question Classification* as candidates for the adversarial tasks and repurpose the models allowed under this constraint. We use context size $k = 5$ for all our experiments.

In white-box attacks, we use the Gumbel-Softmax based approach described in the methodology to train the adversarial program. The details of the temperature annealing process are included in table 1 of the supplementary material. For black-box attacks, we use the REINFORCE algorithm described in methodology, on mini-batches of sequences. Since the action space for certain reprogramming problems, (eg. reprogramming of IMDB classifier) is large ($|V_S| = 10000$), we restrict the output of the adversarial program to most frequent 1000 tokens in the vocabulary V_S. We use Adam optimizer [34] for all our experiments. Hyperparameter details of all our experiments are included in table 1 of the supplementary material.

Table 2.2: **Adversarial Reprogramming Experiments:** The accuracies of *white-box* and *black-box* reprogramming experiments on different combinations of original task, adversarial task and model. Figures in bold correspond to our best results on a particular adversarial task in the given attack scenario scenario (black-box and white-box). *White-box on Random Network* column presents results of the white-box attack on an untrained neural network. Context size $k = 5$ is used for all our experiments.

Victim Model	Original Task	Adversarial Task	Test Accuracy (%)		
			Black-box	White-Box	White-Box on Random Network
LSTM	Questions	Names-5	80.96	97.03	44.33
	Questions	Arabic Tweets	73.50	87.50	50.00
	Names-18	Questions	68.56	95.23	28.23
	Names-18	Arabic Tweets	83.00	84.75	51.50
	IMDB	Arabic Tweets	80.75	**88.25**	50.50
Bi-LSTM	Questions	Names-5	**93.51**	**99.66**	63.14
	Questions	Arabic Tweets	81.75	83.50	70.00
	Names-18	Questions	**94.96**	97.15	80.01
	Names-18	Arabic Tweets	78.75	84.25	69.25
	IMDB	Arabic Tweets	83.25	86.75	84.00
CNN	Questions	Names-5	88.90	99.22	93.06
	Questions	Arabic Tweets	82.25	87.25	76.25
	Names-18	Questions	71.03	**97.61**	33.45
	Names-18	Arabic Tweets	80.75	86.50	60.00
	IMDB	Arabic Tweets	**84.00**	87.00	84.25

2.3.3 Results and Discussions

The accuracies of all adversarial reprogramming experiments have been reported in table 2.2. To interpret the results in context, the accuracies achieved by the LSTM, Bi-LSTM and CNN text classification models on the adversarial tasks can be found in table 2.1.

We demonstrate how character-level models trained on Names-18 dataset can be repurposed for word-level sequence classification tasks like Question Classification and Arabic Tweet Sentiment Classification. Similarly, word-level classifiers trained on Question Classification Dataset can be repurposed for the character-level Surname classification task. Interestingly,

classifiers trained on IMDB Movie Review Dataset can be repurposed for Arabic Tweet Sentiment Classification even though there is a high difference between the vocabulary size (10000 vs 955) and average sequence length(246.8 vs 9.7) of the two tasks. It can be seen that all of the three classification models are susceptible to adversarial reprogramming in both white-box and black-box setting.

White-box based reprogramming outperforms the black-box based approach in all of our experiments. Figure 2.3 shows the learning curves for both white-box and black-box attacks. In practice, we find that training the adversarial program in the black-box scenario requires careful hyper-parameter tuning for REINFORCE to work. We believe that improved reinforcement learning techniques for sequence generation tasks [10, 9] can make the training procedure for black-box attack more stable. We propose such improvement as a direction of future research.

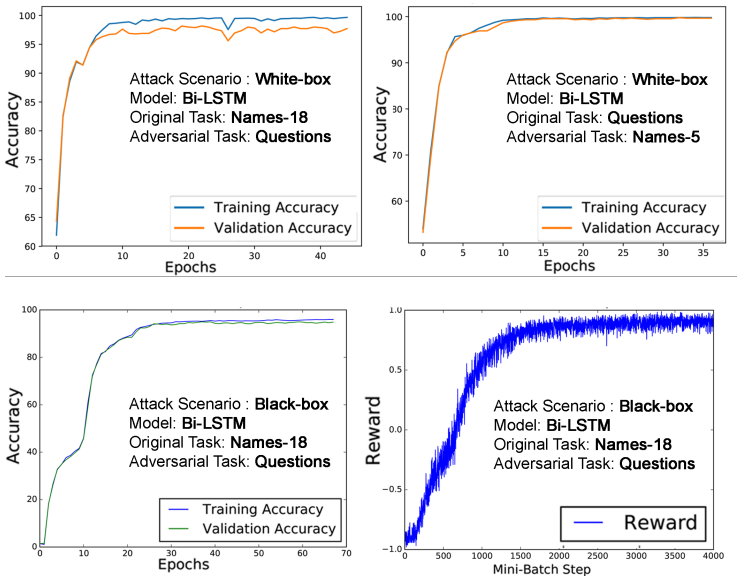

Figure 2.3: **Top:** Training and validation accuracy plots for 2 different *white-box* experiments. **Bottom:** Accuracy and reward plots for a *black-box* training experiment.

To assess the importance of the original task on which the network was trained, we also present results of white-box adversarial reprogramming on untrained random network. Our results are coherent with similar experiments on adversarial reprogramming of untrained ImageNet models [22] demonstrating that adversarial reprogramming is less effective when it

targets untrained networks. The figures in table 2.2 suggest that the representations learned by training a text classifier on an original task, are important for repurposing it for an alternate task. However another plausible reason as discussed by Elsayed et al. is that the reduced performance on random networks might be because of simpler reasons like poor scaling of network weight initialization making the optimization problem harder.

Adversarial Sequences:

Figure 2.4 shows some adversarial sequences generated by the adversarial program for Names-5 Classification while attacking a CNN trained on the Question Classification dataset. A sequence t in the first column is transformed into the adversarial sequence s in the second column by the trained adversarial reprogramming function. Note that in contrast to traditional adversarial examples, the generated adversarial sequences need not be constrained by a small perturbation to the valid input sequence of the original task. While these adversarial sequences may not make semantic or grammatical sense, it exploits the learned representation of the classifier to map the inputs to the desired class. For example, sequences that should be mapped to HUMAN class have words like *Who* in the generated adversarial sequence. Similarly, sequences that should be mapped to LOCATION class have words like *world, city* in the adversarial sequence. Other such interpretable transformations are depicted via colored text in the adversarial sequences of Figure 2.4.

Effect of Context Size:

By varying the context size k of the convolutional kernel $\theta_{k\times|V_T|\times|V_S|}$ in our adversarial program we are able to control the representational capacity of the adversarial reprogramming function. Figure 2.5 shows the percentage accuracy obtained when training the adversarial program with different context sizes k on two different adversarial tasks: Arabic Tweets Classification and Name Classification. Using a context size $k = 1$ reduces the adversarial reprogramming

Adversarial Task Sequence (t) (Names-5)	Adversarial Program Output (s) (Question Classification)	Prediction by Classifier	Mapped Class	Actual Class
Ryoo	white sport substance animal All off ..	ENTITY	Korean	Korean
Houtum	player video exp abb What does off is off ..	ABBREVIATION	Dutch	Dutch
Winogrodzki	manner France manner video def oil def reason desc What do All off ..	DESCRIPTION	Polish	Polish
Murphy	world live exp city What university All is off ..	LOCATION	Scottish	Scottish
Paulissen	player stars along abb abb exp exp always abb What is off ..	ABBREVIATION	Dutch	Dutch
Henriques	other always live What Who does ind Who gold is off ..	HUMAN	Portuguese	Portuguese
Maly	world attend home abb home is off ..	LOCATION	Scottish	Polish
Kasprzak	does exp exp def manner does reason What does off ..	DESCRIPTION	Polish	Polish
Ferreiro	e-mail Who ind exp Who ind university university gold off ..	HUMAN	Portuguese	Portuguese
Hong	sport cremat substance university is off ..	ENTITY	Korean	Korean

Figure 2.4: Adversarial sequences generated by our adversarial program for Names-5 Classification (adversarial task), when targeting a CNN trained on the Question Classification dataset (original task). Interpretable transformations are shown as colored words in the second column. Adversarial program outputs that are mapped to the same class are depicted with the same color in the second column.

function to simply a vocabulary remapping function from V_S to V_T. It can be observed that the performance of the adversarial reprogramming model at $k = 1$ is significantly worse than that at higher values of k. While higher values of k improve the performance of the adversarial program, they come at a cost of increased computational complexity and memory required for the adversarial reprogramming function. For the adversarial tasks studied in this paper, we observe that $k = 5$ is a reasonable choice for context size of the adversarial program.

2.3.4 Conclusion

In this work, we extend adversarial reprogramming, a new class of adversarial attacks, to target sequence classification neural networks. We introduce a novel adversarial program and present training algorithms in both white-box and black-box settings. Our results demonstrate the effectiveness of such attacks in the more challenging black-box settings, posing them as a strong threat in real-world attack scenarios. We demonstrate, for the first time, that recurrent neural networks (RNNs) can be reprogrammed for alternate tasks, which opens doors to solve

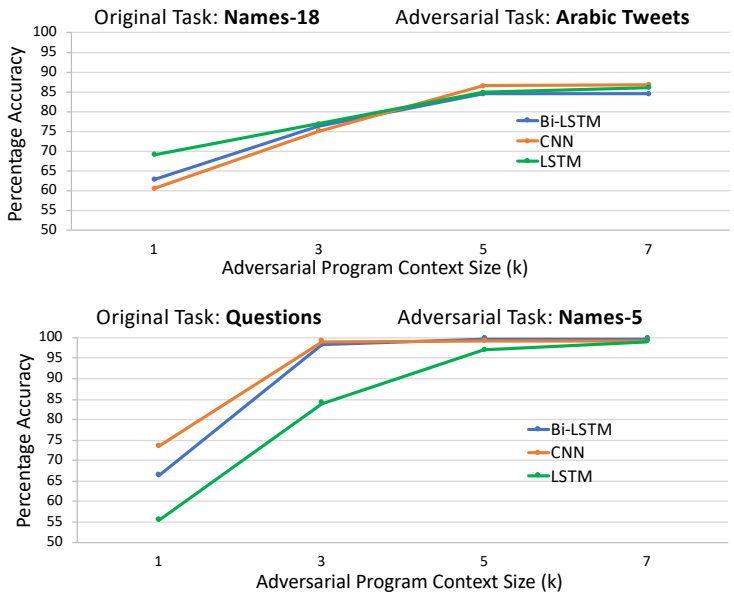

Figure 2.5: Accuracy vs Context size (k) plots for all 3 classification models on 2 different adversarial reprogramming experiments.

more ambitious problems such as repurposing them for mining cryptocurrrency. Due to the threat presented by adversarial reprogramming, we recommend future work to study defenses against such attacks.

Chapter 2, in full, is a reprint of the material as it appears in AAAI 2019 workshop on Engineering Dependable and Secure Machine Learning Systems. Neekhara, Paarth; Hussain, Shehzeen; Dubnov, Shlomo; Koushanfar, Farinaz. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 3

Universal Adversarial Pertrubations for Speech Recognition Systems

3.1 Introduction

Machine learning agents serve as the backbone of several speech recognition systems, widely used in personal assistants of smartphones and home electronic devices (e.g. Apple Siri, Google Assistant). Traditionally, Hidden Markov Models (HMMs) [14, 15, 4, 5, 11] were used to model sequential data but with the advent of deep learning, state-of-the-art speech recognition systems are based on Deep Neural Networks (DNNs) [7, 62, 61, 29]. However, several studies have demonstrated that DNNs are vulnerable to adversarial examples [24, 8, 19, 35, 44]. An adversarial example is a sample from the classifier's input domain which has been perturbed in a way that is intended to fool a victim machine learning (ML) model. While the perturbation is usually imperceptible, such an adversarial input can mislead neural network models deployed in real-world settings causing it to output an incorrect class label with higher confidence.

The majority of past research in adversarial machine learning has shown such attacks to be successful in the image domain [58, 44, 47, 49, 46, 17, 23]. However, few works have

addressed attack scenarios involving other modalities such as audio. This limits our understanding of system vulnerabilities of many commercial speech recognition models employing DNNs, such as Amazon Alexa, Google Assistant, and home electronic devices like Amazon Echo and Google Home. Recent studies that have explored attacks on automatic speech recognition (ASR) systems [6, 18, 20, 64], have demonstrated that adversarial examples exist in the audio domain. The authors of [18] proposed targeted attacks where an adversary designs a perturbation that can cause the original audio signal to be transcribed to any phrase desired by the adversary. However, calculating such perturbations requires the adversary to solve an optimization problem for each data-point they wish to mis-transcribe. This makes the attack in-applicable in real-time since the adversary would need to re-solve the data-dependent optimization problem from scratch for every new data-point.

Universal Adversarial Perturbations [42] have demonstrated that there exist universal *image-agnostic* perturbations which when added to any image will cause the image to be mis-classified by a victim network with high probability. The existence of such perturbations poses a threat to machine learning models in real world settings since the adversary may simply add the same pre-computed universal perturbation to a new image and cause mis-classification.

In this work, we seek to answer the question "Do universal adversarial perturbations exist for neural networks in audio domain?" We demonstrate the existence of universal audio-agnostic perturbations that can fool DNN based ASR systems [1]. We propose an algorithm to design such universal perturbations against a victim ASR model in the *white-box setting*, where the adversary has access to the victim's model architecture and parameters. We validate the feasibility of our algorithm, by crafting such perturbations for Mozilla's open source implementation of the state-of-the-art speech recognition system DeepSpeech [29]. Additionally, we discover that the generated universal perturbation is transferable to a significant extent across different model architectures. Particularly, we demonstrate that a universal perturbation trained on DeepSpeech

[1] Sound Examples: `http://universal-audio-perturbation.herokuapp.com`

can cause significant transcription error on a WaveNet [61] based ASR model.

3.2 Related Work

Adversarial Attacks in the Audio Domain: Adversarial attacks on ASR systems have primarily focused on *targeted attacks* to embed carefully crafted perturbations into speech signals, such that the victim model transcribes the input audio into a specific malicious phrase, as desired by the adversary [6, 18, 31, 20, 60]. Prior works [20, 60] demonstrate successful attack algorithms targeting traditional speech recognition models based on HMMs and GMMs, that operate on Mel Frequency Cepstral Coefficient (MFCC) representation of audio. In Hidden Voice Commands [20], the attacker uses inverse feature extraction to generate obfuscated audio that can be played over-the-air to attack ASR systems. However, obfuscated samples sound like random noise rather than normal human perceptible speech and therefore come at the cost of being fairly perceptible to human listeners. Additionally, these attack frameworks are not end-to-end, which render them impractical for studying the vulnerabilities of modern ASR systems based on DNNs.

In more recent work [18], Carlini *et al.* propose an end-to-end white-box attack technique to craft adversarial examples, which transcribe to a target phrase. Similar to the work in images, they propose a gradient-based optimization method that replaces the cross-entropy loss function used for classification, with a Connectionist Temporal Classification (CTC) loss [26] which is optimized for time-sequences. The CTC-loss between the target phrase and the network's output is backpropagated through the victim neural network and the MFCC computation, to update the additive adversarial perturbation. The adversarial samples generated by this work are quasi-perceptible, motivating a separate work [53] to minimize the perceptibility of the adversarial perturbations using psychoacoustic hiding.

Designing adversarial perturbations using the above mentioned approaches requires the adversary to solve a data dependent optimization problem for each input audio signal the adversary

wishes to mis-transcribe, making them ineffective in a real-time attack scenario. The existence of universal adversarial perturbations (described below) can pose a threat to ASR systems in real-world settings since the adversary may simply add the same pre-computed universal adversarial perturbation to any input audio and fool the DNN based ASR system.

Universal Adversarial Perturbations: The authors of [42] craft a single universal perturbation vector which can fool a victim neural network to predict a false classification output on the majority of validation instances. Let $\hat{k}(x)$ be the classification output for an input x that belongs to a distribution μ. The goal is to find a perturbation v such that: $\hat{k}(x+v) \neq \hat{k}(x)$ for "most" $x \in \mu$. This is formulated as an optimization problem with constraints to ensure that the universal perturbation is within a specified p-norm and is also able to fool the desired number of instances in the training set. The proposed algorithm iteratively goes over the training dataset to build a universal perturbation vector that pushes each data point to its decision boundary. The authors demonstrate that it is possible to find a quasi-imperceptible universal perturbation that pushes most data points outside the correct classification region of a victim model. More interestingly, the work demonstrates that the universal perturbations are transferable across models with different architectures.

3.3 Methodology

3.3.1 Threat Model

We aim to find a universal audio perturbation, which when added to any speech waveform, will cause an error in transcription by a speech recognition model with high probability. For the success of the attack, the error in the transcription should be high enough so that the transcription of the perturbed signal (adversarial transcription) is incomprehensible and the original transcription cannot be deduced from the adversarial transcription. As discussed in [18], the transcription *"test sentence"* mis-spelled as *"test sentense"* does little to help the adversary. To

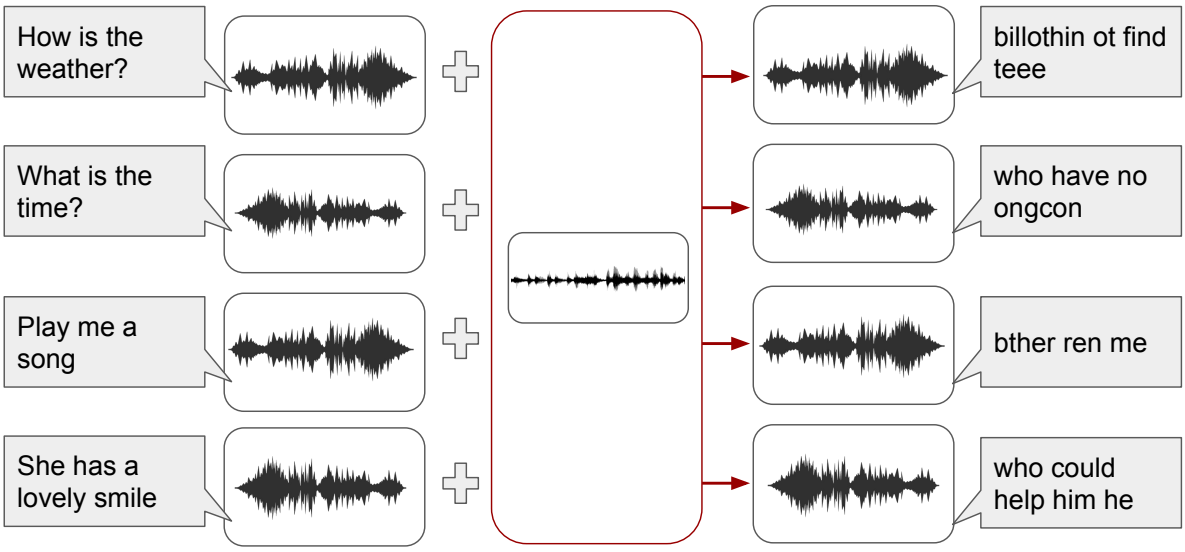

Figure 3.1: Threat Model: We aim to find a single perturbation which when added to any arbitrary audio signal, will most likely cause an error in transcription by a victim Speech Recognition System

make the adversary's goal challenging, we report success only when the Character Error Rate (CER) or the normalized Levenshtein distance *(Edit Distance)* [68] between the original and adversarial transcription is greater than a particular threshold. Formally, we define our threat model as follows:

Let μ denote a distribution of waveforms and C be the victim speech recognition model that transcribes a waveform x to $C(x)$. The goal of our work is to find perturbations v such that:

$$CER(C(x), C(x+v)) > t \text{ for "most" } x \in \mu$$

Here, $CER(x,y)$ is the edit distance between the strings x and y normalized [68] by the $length$ of x i.e

$$CER(x,y) = \frac{EditDistance(x,y)}{length(x)}$$

The threshold t is chosen as 0.5 for our experiments i.e., we report success only when the original transcription has been *edited* by at least 50% of its length using character *removal,*

insertion, or substitution operations.

The universal perturbation signal v is chosen to be of a fixed length and is cropped or zero-padded at the end to make it equal to length of the signal x.

3.3.2 Distortion Metric

To quantify the distortion introduced by some adversarial perturbation v, an l_∞ metric is commonly used in the space of images. Following the same convention, in the audio domain [19], the loudness of the perturbation can be quantified using the dB scale, where $dB(v) = \max_i(20.\log_{10}(v_i))$. We calculate $dB_x(v)$ to quantify the relative loudness of the universal perturbation v with respect to an original waveform x where:

$$dB_x(v) = dB(v) - dB(x)$$

Since the perturbation introduced is quieter than the original signal, $dB_x(v)$ is a negative value, where smaller values indicate quieter distortions. In our results, we report the average relative loudness: $dB_x(v)$ across the whole test set to quantify the distortion introduced by our universal perturbation.

3.3.3 Problem Formulation and Algorithm

Our goal to find a quasi-imperceptible universal perturbation vector v such that it mistranscribes *most* data points sampled from a distribution μ. Mathematically, we want to find a perturbation vector v that satisfies:

1. $\|v\|_\infty < \varepsilon$
2. $\underset{x\sim\mu}{P}(CER(C(X), C(x+v) > t)) \geq 1-\delta.$

Here ε is the maximum allowed l_∞ norm of the perturbation, δ is the desired success rate and t is the threshold CER chosen to define our success criteria.

To solve the above problem, we adapt the universal adversarial perturbation algorithm proposed by [42] to find universal adversarial perturbations for the goal of *mis-transciption* of speech waveforms instead of *mis-classification* of data (images). Let $X = x_1, x_2, \ldots, x_m$ be a set of speech signals sampled from the distribution μ. The algorithm (Algorithm 1) goes over the data-points in X iteratively and gradually builds the perturbation vector v. At each iteration i, we seek a minimum perturbation Δv_i, that causes an error in the transcription of the current perturbed data point $x_i + v$. We then add this additional perturbation Δv_i to the current universal perturbation v and clip the new perturbation v, if necessary, to satisfy the constraint $\|v\|_\infty < \varepsilon$.

Algorithm 1 Universal Adversarial Perturbations for Speech Recognition Systems

1: **input:** Data Points X, Validation Set X_v, Victim Model C, allowed distortion level ε, desired success rate δ
2: **output:** Universal Adversarial Perturbation vector v
3: Initialize $v \leftarrow 0$
4: **while** $Err(X_v) < 1 - \delta$ **do**
5: **for** each data point $x_i \in X$ **do**
6: **if** $CER(C(x_i + v + r), C(x_i)) < t$ **then**
7: Compute min perturbation that mis-transcribes $x_i + v$: $\Delta v_i \leftarrow \arg\min_r \|r\|_2$ s.t.: $CER(C(x_i + v + r), C(x_i)) > t$
8: Update and clip universal perturbation v: $v = Clip_{v,\varepsilon}(v + \Delta v_i)$

At each iteration we need to solve the following optimization problem, that seeks a minimum (under *l2* norm) additional perturbation Δv_i, to mis-transcribe the current perturbed audio signal $x_i + v$:

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \quad \text{s.t.} \quad CER(C(x_i + v + r), C(x_i)) > t \tag{3.1}$$

It is non-trivial to solve the above optimization in its current form. In [42], the authors try to solve a similar optimization problem for the goal of *mis-classification* of data points. They

approximate its solution using DeepFool [43] which finds a minimum perturbation vector that pushes a data point to its decision boundary. Since we are tackling a more challenging goal of *mis-transcription* of signals where we have decision boundaries for each audio frame across the time axis, the same idea cannot be directly applied. Therefore, we approximate the solution to the optimization problem given by (3.1) by solving a more tractable optimization problem:

$$\begin{aligned} &\text{Minimize } J(r) \text{ where} \\ &J(r) = c\|r\|^2 + L(x_i + v + r, C(x_i)) \\ &\text{s.t. } \|v + r\|_\infty < \varepsilon \\ &\text{where } L(x,y) = -CTCLoss(f(x),y) \end{aligned} \tag{3.2}$$

In other words, to mis-transcribe the signal, we aim to maximize the CTC-Loss between the predicted probability distributions of the perturbed signal $f(x_i + v + r)$ and the original transcription $C(x_i)$ while having a regularization penalty on the $l2$ norm of r. Since this a non-convex optimization problem, we approximate its solution using iterative gradient sign method [36]:

$$\begin{aligned} &r_0 = \overrightarrow{0} \\ &r_{N+1} = Clip_{r+v,\varepsilon}\{r_N - \alpha sign(\Delta_{r_N} J(r_N)\} \end{aligned} \tag{3.3}$$

Note that the error J is back-propagated through the entire neural network and the MFCC computation to the perturbation vector r. We iterate until we reach the desired CER threshold t for a particular data point x_i. The regularization constant c is chosen through hyper-parameter search on a validation set to find the maximum success rate for a given magnitude of allowed perturbation.

3.4 Experimental Details

We demonstrate the application of our proposed attack algorithm on the pre-trained *Mozilla DeepSpeech* model [2, 29]. We train our algorithm on the Mozilla Common Voice Dataset [29] which contains 582 hours of audio across 400,000 recordings in English. We train on a randomly selected set X containing 5,000 audio files from the training set and evaluate our model on both the training set X and the entire unseen validation set of the Mozilla Common Voice Dataset. We analyze the effect of the size of the set X below. The length of our universal adversarial perturbation is fixed to 150,000 samples which corresponds to around 9 seconds of audio at 16 KHz. The universal adversarial perturbations are trained using our proposed algorithm 1 with a learning rate $\alpha = 5$ and the regularization parameter c set to 0.5.

Evaluation: We utilize two metrics: *i) Mean CER* - Character Error Rate averaged over the entire test set and *ii) Success Rate* to evaluate our universal adversarial perturbations. We report success on a particular waveform, if the *CER* between the original and adversarial transcription (Section 3.3.1) is greater than 0.5. The amount of perturbation is quantified using mean relative distortion $dB_x(v)$ over the test set (Refer to Section 3.3.2).

3.5 Results

Table 3.1 shows the results of our algorithm for different allowed magnitude of universal adversarial perturbation on both the training set X and the unseen Test Set. Both the success rate and the Mean Character Error Rate (CER) increase with increase in the maximum allowed perturbation. We achieve a success rate of 89.06 % on the validation set, with the mean distortion metric $dB_x(v) \approx -32dB$. To interpret the results in context, $-32dB$ is roughly the difference between ambient noise in a quiet room and a person talking [56, 18] . We encourage the reader to listen to our adversarial samples and their corresponding transcriptions on our web page: `http://universal-audio-perturbation.herokuapp.com`

Table 3.1: Results of our algorithm for different allowed magnitude of universal adversarial perturbation

	Training Set (X)			Test Set		
$\|\|\mathbf{v}\|\|_\infty$	Mean $\mathbf{dB_x(v)}$	Success Rate (%)	Mean CER	Mean $\mathbf{dB_x(v)}$	Success Rate (%)	Mean CER
100	-42.03	57.46	0.63	-41.86	56.13	0.64
150	-38.51	72.78	0.81	-38.34	72.49	0.82
200	-36.01	83.27	0.92	-35.84	80.47	0.95
300	-32.49	89.52	1.10	-32.32	89.06	1.11
400	-30.18	90.60	1.06	-29.82	88.24	1.07

Figure 3.2 shows the success rate and mean edit distance compared to the size of the training set X for maximum allowed perturbation $\|v\|_\infty = 200$ (Mean $dB_x(v) = -36.01$). We observe that it is possible to train our proposed algorithm on very few examples and achieve reasonable success rates on unseen data. For example, training on just 1000 examples can achieve a success rate of 80.47 % on the test set.

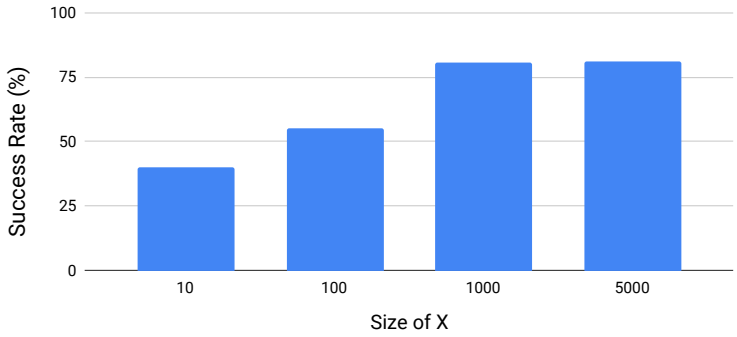

Figure 3.2: Attack Success Rate on the test set vs. the number of audio files in the training set X

3.5.1 Effectiveness of universal perturbations

In order to assess the vulnerability of the victim Speech Recognition System to our attack algorithm, we compare our universal perturbation with random (uniform) perturbation having the same magnitude of distortion (same $\|v\|_\infty$) as our universal adversarial perturbation. Figure 3.3 shows the plot of success rate vs. the magnitude of the perturbation for each of these perturbations. It can be seen that universal adversarial perturbations are able to achieve high success rate with

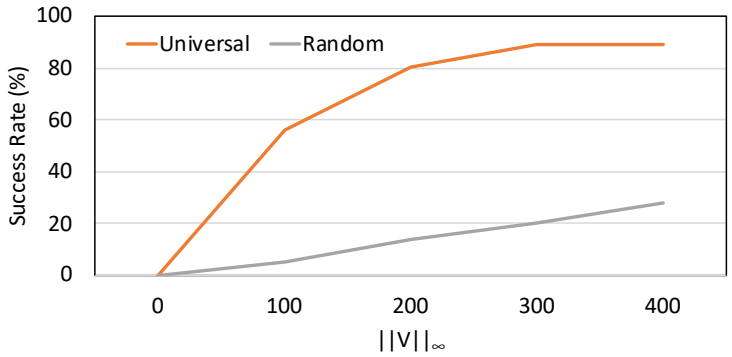

Figure 3.3: Success Rate vs $\|v\|_\infty$ of universal and random perturbations.

very low magnitude of distortion as compared to a random noise perturbation. For example, for allowed perturbation $\|v\|_\infty = 100$ our universal perturbation achieves a success rate of 65% which is substantially higher than the success rate of random noise. This implies that for the same magnitude of distortion, distorting an audio waveform in a random direction is significantly less likely to cause mis-transcription as compared to distorting the waveform in the direction of universal perturbation. Our results support the hypothesis discussed in [42], demonstrating that universal adversarial perturbations exploit geometric correlations in the decision boundaries of the victim model.

Table 3.2: Results of the same universal adversarial perturbation on two victim models: Wavenet and Mozilla DeepSpeech. The universal perturbation was trained on the DeepSpeech model.

		Wavenet		**Mozilla DeepSpeech**	
$\|v\|_\infty$	**Mean dBx(v)**	**Success Rate (%)**	**Mean CER**	**Success Rate (%)**	**Mean CER**
150	-38.34	**26.97**	**0.37**	72.49	0.82
200	-35.84	**31.18**	**0.40**	80.47	0.95
300	-32.32	**42.05**	**0.47**	89.06	1.11
400	-29.82	**63.28**	**0.60**	88.24	1.07

3.5.2 Cross-model Transferability

We perform a study on the transferability of adversarial samples to deceive ML models that have not been used for training the universal adversarial perturbation, i.e., their parameters

and network structures are not revealed to the attacker. We train universal adversarial perturbations for Mozilla DeepSpeech and evaluate the extent to which they are valid for a different ASR architecture based on WaveNet [61]. For this study, we use a publicly available pre-trained model of WaveNet [1] and evaluate the transcriptions obtained using clean and adversarial audio for the same unseen validation dataset as used in our previous experiments. Our results in Table 3.2 indicate that our attack is transferable to a significant extent for this particular setting. Specifically, when the mean $dB_x(v) = -29.82$, we are able to achieve a 63.28% success rate while attacking the WaveNet based ASR model. This result demonstrates the practicality of such adversarial perturbations, since they are able to generalize well across data points and architectures.

3.6 Conclusion

In this work, we demonstrate the existence of audio-agnostic adversarial perturbations for speech recognition systems. We demonstrate that the audio-agnostic perturbation generalizes well across unseen data points and to some extent across unseen networks. Our proposed end-to-end approach can be used to further understand the vulnerabilities and blind spots of deep neural network based ASR system, and provide insights for building more robust neural networks.

Chapter 3, in full, is a reprint of the material as it appears in the supplementary DSN 2019 proceedings. Neekhara, Paarth; Hussain, Shehzeen; Pandey, Prakhar; Dubnov, Shlomo; McAuley, Julian, Koushanfar, Farinaz. The dissertation/thesis author was the primary investigator and author of this paper.

Chapter 4

Conclusion

In our work we demonstrated two main vulnerabilities of neural sequence models which makes secure real world deployment of such models a challenge:

- The ability to repurpose neural sequence models for an adversarial task.
- The existence of universal adversarial perturbations for speech recognition systems.

Concurrent with our work, there have been ongoing works in this domain which expose vulnerabilities of neural sequence models and develop defences against them. In this chapter, we will discuss some of these works and talk about some open research questions in the field of adversarial attacks and defences for neural sequence models.

4.1 Recent Advances

In this section we discuss some recent advances in the field of adversarial machine learning which pose new research questions and lay the directions for future work.

4.1.1 Imperceptible, Robust and Targeted Adversarial Examples for Automatic Speech Recognition

This work [50] focuses on developing targeted adversarial examples for speech recognition systems that are imperceptible and robust to ambient noise when played in a simulated environment. The authors propose a white box attack and demonstrate the application on a state of the art ASR system google Lingvo.

To construct imperceptible adversarial examples for automatic speech recognition system, this work uses frequency masking, which refers to the phenomenon that a louder signal can make other signals at nearby frequencies imperceptible. Through this process of *psycho-acoustic hiding*, the authors retain the 100% success rate of Carilini's attack [18] while being effectively imperceptible under as per the conducted user study.

In order to improve the robustness of adversarial examples when playing over-the-air, the authors use the Image Source Method to create the room impulse responses based on the room configurations (e.g., the room dimension, source audio and target microphones location). The room impulse responses are then convolved with the audio to create artificial utterances (speech with reverberations) that mimic playing the audio over-the-air.

By combining both of the above techniques, the attacker can generate both imperceptible and robust adversarial examples, which can achieve around 50% attack success rate in 100 simulated test rooms.

4.1.2 Characterizing Audio Adversarial Examples Using Temporal Dependency

This work [66] explores methods to mitigate the effect of audio adversarial examples. This paper first explores whether the lessons learned in the image domain for adversarial examples apply to the audio domain. The authors study the effectiveness of audio adversarial examples

under simple input transformations like quantization, local smoothing, down-sampling and auto-encoding. They find these methods to be reasonably effective in detecting adversarial examples at the cost of reduced performance of the ASR system. Another downside of these defenses is that they can be easily bypassed if the attacker is aware of the defense being used in the ASR system.

The authors propose a novel defense that exploits temporal dependency which discriminates adversarial examples from the original ones. The authors observe that temporal dependencies in an audio sample are no longer consistent after applying an adversarial perturbation. Based on this observation, they propose a simple defense that compares the transcription of different segments on an audio clip to judge whether an audio clip is adversarial or not.

The authors then try to break the defense assuming that the attacker is aware of the defnse. The authors demonstrate that while it can be bypasses in a completely white-box attack scenario, however, using an ensemble of such defenders make this attack less effective.

4.1.3 Are adversarial examples inevitable?

The recent works proposing defenses against adversarial examples, have one problematic trend: They can easily be broken if the attacker is aware of the defense technique and its parameters. This raises the fundamental question that whether adversarial examples are inevitable. This paper [54] analyzes adversarial examples from a theoritical perspective and shows that for certain classes of problems, adversarial examples are inevitable. Using experiments, the authors explore the implications of theoretical guarantees for real-world problems and discuss how factors such as dimensionality and image complexity limit a classifier's robustness against adversarial examples.

4.2 Future Work

Based on our work and the recent advances in this field, we discuss some open research questions which can be the directions of future research:

- **What is the scope of adversarial reprogramming?** An interesting area to explore in adversarial reprogramming is to understand what kinds of classification problems can be solved by reprogramming pre-trained neural networks. That is, how far can a simple input transformation get us in both continuous and discrete adversarial reprogramming problem setting. Also, is it possible to design a universal neural network which can be easily reprogrammed for an alternate task using simple transformation on the inputs and outputs?

- **Can universal adversarial perturbations be played over the air?** If universal adversarial perturbations can be played over the air, it poses a real world threat to ASR systems deployed in home electronic devices and smart phones. It will be interesting to study how effective our universal audio perturbation is, when played over the air. Also, is it possible to amend the training procedure using techniques similar to [50], to increase the chances of an *over the air* attack?

- **How to defend against audio adversarial attacks?** Can we develop a provably secure ASR system that is not vulnerable to adversarial attacks? Recent works on adversarial defences in the image domain [41, 52] try to model the distribution of real images and classify an image as adversarial if it does not lie on that manifold. Can we apply similar ideas in the audio domain to defend against adversarial audio examples?

- **Can we develop real-time targeted audio attacks?** One challenge with the existing work on targeted audio attacks is that the adversary needs to solve an optimization problem for each audio clip they wish to mis-transcribe. While universal adversarial perturbation

addresses this problem, it is an untargeted attack and cannot yield a target transcription. Can we develop a real-time attack for ASR systems that can cause mis-transcription to a target phrase without the need to resolve an optimization problem?

Besides the above, a more fundamental question that remains unanswered is whether or not we can develop a provably secure machine learning model that can be robust to adversarial examples without compromising on performance metrics. The existence of white-box attack methods is a serious threat even in the black box attack scenarios since adversarial examples are shown to be transferable. To ensure safe deployment of such models in real-world settings there it is essential to address and explore the vulnerabilities of such systems.

Bibliography

[1] Speech to text wavenet. https://github.com/buriburisuri/speech-to-text-wavenet.

[2] Project deepspeech. https://github.com/mozilla/DeepSpeech.

[3] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6, Dec 2013. doi: 10.1109/AEECT.2013.6716448.

[4] Alex Acero, li Deng, Trausti Kristjansson, and Jerry Zhang. Hmm adaptation using vector taylor series for noisy speech recognition. pages 869–872, 01 2000.

[5] SM Ahadi and Philip C Woodland. Combined bayesian and predictive techniques for rapid speaker adaptation of continuous density hidden markov models. *Computer speech & language*, 11(3):187–206, 1997.

[6] Moustafa Alzantot, Bharathan Balaji, and Mani B. Srivastava. Did you hear that? adversarial examples against automatic speech recognition. *CoRR*, abs/1801.00554, 2018. URL http://arxiv.org/abs/1801.00554.

[7] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[8] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018. URL https://arxiv.org/abs/1802.00420.

[9] Philip Bachman and Doina Precup. Data generation as sequential decision making. In *NIPS*, pages 3249–3257, 2015.

[10] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *CoRR*, abs/1607.07086, 2016.

[11] L Bahl, P Brown, P de Souza, and R Mercer. Maximum mutual information estimation of hidden markov model parameters for speech recognition. In *ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pages 49–52. IEEE, 1986.

[12] Shumeet Baluja and Ian Fischer. Learning to attack: Adversarial transformation networks. In *Proceedings of AAAI-2018*, 2018. URL http://www.esprockets.com/papers/aaai2018.pdf.

[13] Mauro Barni, Kassem Kallas, Ehsan Nowroozi, and Benedetta Tondi. On the transferability of adversarial examples against cnn-based image forensics. *CoRR*, abs/1811.01629, 2018. URL http://arxiv.org/abs/1811.01629.

[14] Leonard E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73(3):360–363, 05 1967. URL https://projecteuclid.org:443/euclid.bams/1183528841.

[15] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

[16] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 387–402, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-40994-3.

[17] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017. URL http://arxiv.org/abs/1712.09665.

[18] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

[19] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[20] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 513–530, Austin, TX, 2016. USENIX Association. ISBN 978-1-931971-32-4. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/carlini.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[22] Gamaleldin F. Elsayed, Ian J. Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. *CoRR*, abs/1806.11146, 2018.

[23] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

[24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *stat*, 1050:20, 2015.

[25] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, ICANN'05, pages 799–804, Berlin, Heidelberg, 2005. Springer-Verlag. ISBN 3-540-28755-8, 978-3-540-28755-1. URL http://dl.acm.org/citation.cfm?id=1986079.1986220.

[26] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[27] Jiatao Gu, Daniel Jiwoong Im, and Victor O. K. Li. Neural machine translation with gumbel-greedy decoding. *CoRR*, abs/1706.07518, 2017.

[28] E.J. Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*. Applied mathematics series. U. S. Govt. Print. Office, 1954. URL https://books.google.com/books?id=SNpJAAAAMAAJ.

[29] Awni Y. Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. Deep speech: Scaling up end-to-end speech recognition. *CoRR*, abs/1412.5567, 2014. URL http://arxiv.org/abs/1412.5567.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[31] Dan Iter, Jade Huang, and Mike Jermann. Generating adversarial examples for speech recognition. 2017.

[32] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *CoRR*, abs/1611.01144, 2016. URL http://dblp.uni-trier.de/db/journals/corr/corr1611.html#JangGP16.

[33] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014. URL http://aclweb.org/anthology/D/D14/D14-1181.pdf.

[34] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://dblp.uni-trier.de/db/journals/corr/corr1412.html#KingmaB14.

[35] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. URL http://arxiv.org/abs/1611.01236.

[36] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *CoRR*, abs/1607.02533, 2016. URL http://arxiv.org/abs/1607.02533.

[37] Matt J. Kusner and José Miguel Hernández-Lobato. GANS for sequences of discrete elements with the gumbel-softmax distribution. *CoRR*, abs/1611.04051, 2016.

[38] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1072228.1072378. URL https://doi.org/10.3115/1072228.1072378.

[39] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016. URL http://arxiv.org/abs/1611.02770.

[40] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

[41] Dongyu Meng and Hao Chen. Magnet: A two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 135–147, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4946-8. doi: 10.1145/3133956.3134057. URL http://doi.acm.org/10.1145/3133956.3134057.

[42] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. Universal adversarial perturbations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 86–94, July 2017. doi: 10.1109/CVPR.2017.17.

[43] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. pages 2574–2582, 06 2016. doi: 10.1109/CVPR.2016.282.

[44] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *CoRR*, abs/1511.07528, 2015. URL http://arxiv.org/abs/1511.07528.

[45] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016.

[46] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016. URL http://arxiv.org/abs/1605.07277.

[47] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016. URL http://arxiv.org/abs/1605.07277.

[48] Nicolas Papernot, Patrick D. McDaniel, Ananthram Swami, and Richard E. Harang. Crafting adversarial input sequences for recurrent neural networks. *CoRR*, abs/1604.08275, 2016. URL http://arxiv.org/abs/1604.08275.

[49] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519. ACM, 2017.

[50] Yao Qin, Nicholas Carlini, Ian J. Goodfellow, Garrison W. Cottrell, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. *CoRR*, abs/1903.10346, 2019. URL http://arxiv.org/abs/1903.10346.

[51] Sean Robertson. Classifying names with a character-level rnn - pytroch tutorial. https://pytorch.org/tutorials/intermediate/char_rnn_classification_tutorial.html, 2017.

[52] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *CoRR*, abs/1805.06605, 2018. URL http://arxiv.org/abs/1805.06605.

[53] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *arXiv preprint arXiv:1808.05665*, 2018.

[54] Ali Shafahi, W Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*, 2018.

[55] Yash Sharma, Tien-Dung Le, and Moustafa Alzantot. CAAD 2018: Generating transferable adversarial examples. *CoRR*, abs/1810.01268, 2018. URL http://arxiv.org/abs/1810.01268.

[56] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, San Diego, CA, USA, 1997. ISBN 0-9660176-3-3.

[57] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA, 2014. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969033.2969173.

[58] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL http://arxiv.org/abs/1312.6199.

[59] Florian Tramèr, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. The space of transferable adversarial examples. *CoRR*, abs/1704.03453, 2017.

[60] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. Cocaine noodles: Exploiting the gap between human and machine speech recognition. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, Washington, D.C., 2015. USENIX Association. URL https://www.usenix.org/conference/woot15/workshop-program/presentation/vaidya.

[61] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016. URL http://arxiv.org/abs/1609.03499.

[62] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel WaveNet: Fast high-fidelity speech synthesis. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3918–3926, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/oord18a.html.

[63] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

[64] Hiromu Yakura and Jun Sakuma. Robust audio adversarial example for a physical attack. *CoRR*, abs/1810.11793, 2018. URL http://arxiv.org/abs/1810.11793.

[65] Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *CoRR*, abs/1805.12316, 2018. URL http://arxiv.org/abs/1805.12316.

[66] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song. Characterizing audio adversarial examples using temporal dependency. *CoRR*, abs/1809.10875, 2018. URL http://arxiv.org/abs/1809.10875.

[67] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. *CoRR*, abs/1609.05473, 2016. URL http://dblp.uni-trier.de/db/journals/corr/corr1609.html#YuZWY16.

[68] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1091–1095, June 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1078. URL https://doi.org/10.1109/TPAMI.2007.1078.