Stage 0: Robust Deep Learning for Music Performance Analysis

Vinod Subramanian, QMUL, ROLI Ltd.

Abstract

The abstract goes here.

Index Terms

MIR

I. Introduction

Music performance analysis is a popular area of study spanning more than a century worth of research. Gabrielsson [12] wrote an extremely detailed and enlightening article on the evolution of research in music performance analysis. Using the organization of music performance analysis expressed in Gabrielsson's work [12] measuring music performance and using those measuremnts to create models for performance is most relevant to our proposed work. In the summary Gabrielsson says that most of the work on measuring music performance focuses on timing, dynamics, intonation and vibrato. In this PhD we want to continue focussing on these popular features that are measured for music performance analysis and our contributions will be towards using deep learning algorithms to improve the accuracy of the features we are trying to measure.

Humphrey, Bello and LeCun [19] suggested that deep learning can learn more complex features for music information retrieval tasks and give the research community an alternative to hand crafted feature design. Since then a lot of deep learning research has taken place in the music information retrieval community [35],[37],[7],[8],[6]. Despite the popularity of deep learning there is relatively little research that tries to understand why deep learning is effective at solving MIR tasks [26],[5].

One of the biggest challenges of deep learning that remains today is the interpretability of the features being learned. Due to the non-linearity and complexity of these models it is not a simple task to understand exactly what is being learned. While we can study saliency maps to identify what neurons are being activated, it is still unclear as to what features are being learned exactly even when the model is performing really well.

The fact that there is lack of understanding of the underlying features poses the concern of how well can we trust the output of the system. Recent research [33] [13] has shown that it is possible to fool the classifier into misclassifying the input by applying imperceptible non-random values to it. This exposed the vulnerability of deep learning and quickly became a popular area of research in security and privacy of image recognition algorithms.

In music applications adversarial examples could pose a challenge in terms of identifying copyrighted content on websites.

The more general conclusions that can be drawn from [33] and [13] is that deep learning models are not learning high level features as well as we expect them too. So by using the techniques described in these papers we can indirectly determine whether deep learning models are learning high level features in way that we expect them to.

II. Background

A. What are adversarial attacks?

Szegedy et al. [33] discovered that in object recognition tasks by applying an imperceptible non-random perturbation to the input image the output of the network can be changed. The term "Adversarial examples" is used to describe these perturbed examples. These adversarial examples were attributed to the fact that there are blindspots in the training process for these deep learning models.

Goodfellow, Shlens and Szegedy [13] challenged the idea that adversarial examples were due to blind spots in high dimensional spaces. Instead, they suggested that they are caused due to linearity in deep learning models. According to them LSTMS [18], ReLUs [24], maxout networks [36] etc. were intentiaonally designed to behave more linearly in order to make them easier to optimize.

Assume we have a linear classifier defined by the relationship $y = f(x)$ where $f(x)$ is given by $f(x) = \omega^T x$. Our goal is to perturb the input x with perturbation η subject to the condition that $D(x + \eta, x) < \epsilon$ where D is some distance measure. We can write the new equation for the ouput of the classifier as:

$$\tilde{y} = \omega^T x + \omega^T \eta \tag{1}$$

By choosing η carefully so that it is aligned in the direction of the weights of the classifier ω the change in the output can be maximized.

Goodfellow, Shlens and Szegedy [13] used this formulation for linear classifiers on deep learning models and came up with a family of fast gradient approaches to generate adversarial examples. The fact that they were able to generate adversarial examples by treating deep learning models as linear classifiers was used to support the argument that deep learning algorithms are too linear.

Before we give more details about different types of adversarial attacks and defenses against these attacks we'll list the different categories of adversarial attacks. The category of an adversarial attack is determined by either the goal of the adversary or the information the adversary has of the classifier.

Goal of the adversary:

1) Untargeted attack - The adversary's goal is to simply misclassify the input. As long as the new target class of the classifier is different from the original the adversary has achieved its goal.
2) Targeted attack - The adversary's goal is to change the label from the original to a different label that is specified before the attack begins. Because there are more constraints in the targeted attack it is typically harder to generate targeted adversarial examples as opposed to untargeted adversarial examples.

Information of the adversary:

1) Perfect knowledge - In this scenario the adversary has perfect knowledge of the classifier such as the feature space, the weights of the model and the type of classifier. This is also known as a white box attack.
2) Limited knowledge - The classifier does not know the training data or the trained model but has knowledge about the feature representation and the type of classifier. This could be considered a black box attack.
3) Zero knowledge - This is a special case where an adversarial example is generated for one classifier and then tested on a different classifier which it has no information about.

In the following sections I will introduce different techniques for attacks, different techniques for defenses and some unique properties of adversarial attacks such as universal perturbation, transferability of adversarial examples and adversarial examples in the real world. Finally I will highlight the existing body of work that explores adversarial attacks in audio and talk about existing deep learning tasks in music information retrieval (MIR) that would be a good starting point for research into adversarial attacks in music.

B. Summary of attacks

1) L-BFGS - Szegedy et al. [33] introduced one of the first methods for generating adversarial attacks, it is a white box attack that is targeted.
   Assume a classifier denoted as $f : \mathbb{R}^m \rightarrow \{1...k\}$ with a loss function $loss_f$. For a given input $x \in \mathbb{R}^m$ and target $t \in \{1....k\}$ we aim to identify the value of perturbation r as formulated below:

   $$\begin{aligned} &\text{Minimize } \|r\|_2 \text{ under the conditions:} \\ &f(x+r) = t \\ &x + r \in [0,1]^m \end{aligned}$$

   The exact computation of this problem is difficult so it is approximated using the box constrained L-BFGS algorithm. So the new equation to minimize is:

   $$c|r| + loss_f(x+r, l) \qquad \text{under the conditions } x + r \in [0,1]^m$$

2) Fast Gradient Sign Method - The goal of this method is to quickly generate a simple adversarial examples. By perturbing the input in the direction of the gradient of the weights of the model by a sufficient amount it can create an adversarial example [13]. Let x be the input to the model and y be the output, the model parameters are θ. The cost function of the model is $J(\theta, x, y)$. With this information we can create an adversarial example by computing the perturbation η using the formula:

   $$\eta = \epsilon sign(\nabla_x J(\theta, x, y)) \tag{2}$$

   In this equation ϵ is decided manually until the formula causes a perturbation. This technique was refined by making the process iterative and computing the gradient repeatedly and showed improved results.
3) Jacobian based Saliency Map Attack (JSMA) - Papernot et al. [29] introduced a white box attack which requires knowledge of the model parameters but does not require knowledge of the training data, it is a targeted attack. The goal of the algorithm is to identify the pixels in the input that impact the output the most and perturb these important pixels to change the output to the target output.

Assume a classifier denoted as f with output dimensions N that takes $\mathbf{X} \in \mathbb{R}^M$ as an input. The first step is to compute the forward derivative:

$$\begin{aligned} \nabla f &= \frac{\partial f(\mathbf{X})}{\partial \mathbf{X}} \\ &= \frac{\partial f_j(\mathbf{X})}{\partial x_i} \ i \in 1..M, j \in 1..N \end{aligned}$$

This is essentially the Jacobian matrix of the function denoted by f. The computation of this forward derivative can be simplified using the chain rule. The next step is to compute the saliency map [32] based on the forward derivative. The saliency map shows which input features are most important in determining the output.

$$S(\mathbf{X}, t)[i] = \begin{cases} \text{For } i \in 1..M \text{ and } t \text{ as target output} \\ 0 \text{ if } \frac{\partial f_t(\mathbf{X})}{\mathbf{X}_i} < 0 \text{ or} \sum_{j \neq t} \frac{\partial f_j(\mathbf{X})}{\mathbf{X}_i} > 0 \\ \left(\frac{\partial f_t(\mathbf{X})}{\mathbf{X}_i}\right) \left| \sum_{j \neq t} \frac{\partial f_j(\mathbf{X})}{\mathbf{X}_i} \right| \text{ otherwise} \end{cases}$$

From the saliency map we identify the input x_i that has the highest impact on the target output and perturb it by parameter $theta$ that is problem specific. This process is repeated iteratively and the maximum iteration is determined by the distortion limit γ. The distortion limit is manually set at the boundary at which humans can observe the distortion and is problem specific.

4) DeepFool -
5) Carlini and Wagner - Carlini and Wagner introduce a strong set of attacks based on the L_0, L_2 and L_inf distance [4]. The problem for adversarial attacks is formulated the same was as Szegedy et al. [33]. The classifier is denoted as C with input x:

$$\begin{aligned} \text{Minimize } & D(x, x+\delta) + c.f(x+\delta) \\ \text{such that } & x + \delta \in [0,1]^n \end{aligned}$$

Here D is a distance function that is either the L_0, L_2 or L norm and f is an objective that simplifies the problem such that:

$$\begin{aligned} C(x+\delta) = t \text{ is true if } f(x+\delta) \leq 0 \\ f(x') = (max(Z(x')_i) - Z(x')_t)^+ \ i \neq t \end{aligned}$$

In the equation for f, Z denotes the penultimate layer of the classifier, i.e the softmax values and t is the target class. This is just one example of a the function f many other functions work and can be found in the paper [4]. Carlini and Wagner [4] take care of the box constraint issue on $x + \delta$ by changing the variables with the substitution:

$$\delta_i = \frac{1}{2}(tanh(w_i) + 1) - x_i$$

C. Properties of adversarial attacks

In this section we'll highlight a few interesting properties of adversarial attacks which highlight the risk and importance of studying them in more detail.

1) Transferability of attacks - Papernot, Mcdaniel and Goodfellow [28] showed that adversarial examples are transferrable between machine learning models that use the same technique. They went one step further and showed that the models don't need to be using the same technique but still have transferrable adversarial examples. Liu et al. [23] showed that untargeted attacks are more transferrable than targeted attacks. In addition, they used an ensemble of models to create stronger adversarial attacks. Moosavi-Dezfooli et al. [27] created a universal perturbation that are image and network agnostic.
2) Real world attacks - Research has shown that adversarial examples are not limited to the digital world of finely crafted adverseries. Kurakin, Goodfellow and Bengio [21] showed that by printing and taking a photo of an adversarial example there are a significant number of cases for which it remains an adversary. Athalye et al. [1] created a framework in which objects can be rendered in 3-D and be printed and serve as adversarial examples for. Finally, Eykholt et al. [10] developed and algorithm to create perturbations to generate adversarial examples that are physically realizable.

D. Summary of defenses

1) Adversarial Retraining - The goal of retraining is to retrain the base model in order to be more robust to adversarial examples. Defensive distillation is one of the most popular ideas for defenses, distillation was introduced by Hinton et al. [17] and adapted by Papernot et al. [30]. In the retraining process the network architecture is kept the same except the target labels are the soft labels from the base network. Grosse et al.[15] suggested a method where an additional class is added in the retraining process for adversarial examples. Metzen et al. [25] add a subnetwork stemming from some layer that produces an output that gives the probabiliy of an example being an adversary. This solution does not directly classify an input as adversarial but it allows for human intervention evrytime the network is confused about the input.
2) Principal Component Analysis (PCA) - PCA is a dimensionally reduction technique that maps an N dimensional space into a smaller M dimensional space, emphasizing the most important components in the data [31]. Hendrycks and Gimpel [16] introduce a method of analyzing the lower components of PCA finding that they can be used to differentiate adversarial examples from real inputs. Bhagoji, Cullina and Mittal [2] hypothesize that the dimensionality reduction of PCA causes noise reduction and use the reduced dimensions as an input to a classifier to determine whether an input is adversarial or not. Instead of applying PCA on the input directly Li and Li [22] apply PCA on the convolutionaly layer outputs and using a cascade classifier [34] to determine if the input is adversarial.
3) Statistical analysis - There are some techniques which try to look at statistical features of the input to identify whether they are adversarial examples. Grosse et al. [15] suggest a method that uses Maximum Mean Discrepancy (MMD) [14] which is a technique for statistical technique to determine whether two randomly drawn samples originate from the same distribution. Using this they determined whether an input was an adversary or not. Feinman et al. [11] suggest a method that uses kernel density estimates. According to them the kernel density estimates can be used to model the submanifolds of each class and separate the adversarial examples from the real inputs.

E. Adversarial attacks in audio

Kereliuk, Sturm and Larsen [20] applied the attacks developed by Szegedy et al. [33] to the GTZAN dataset [**?**]. The input to the deep learning models was magnitude spectrograms so the perturbations were performed on the magnitude spectrogram. To reconstruct the audio the phase was determined using the Griffin-Lim algorithm and using that information the audio was resynthesized. The results show that adversarial examples can be generated for music on these datasets however we can perceive the change in audio.

A drawback of this paper is that it relies on audio reconstruction to observe differences in the audio and because the phase information has to be resynthesized from a lossy method it is not ideal. Gong and Poellbauer address this problem by suggesting an end-to-end method for generating audio adversarial examples by directly perturbing the raw audio.

III. Research questions

1) Implementing state of the art attack and defense algorithms for MIR tasks - While different types of attack and defense methods have been implemented and compared side by side for computer vision the same has not been done in MIR. The differences in input (audio vs image), the differences in the tasks, the data pre-processing etc. could make the results of this study different in music compared to vision. Even if the results are the same it is still important to recreate them in music to enable further research into the area.
2) Verifying the properties of adversarial attacks in MIR tasks - Some of the interesting properties of adversarial attacks were highlighted in the Background section. It would be interesting to see how transferable audio adversaries are between models, if there exists a universal perturbation to generate adversaries, are there real world audio adversaries etc.
3) MIR specific properties - A quick glance at deep learning research in MIR will show that there is a large diversity of tasks in which deep learning is applied and even within a specific task and on a specific dataset there is large deviations in input representation. So it would be interesting to go one step further in terms of transferability of audio adversaries to see if they are transferable between different types of input representations and if they are transferable between related but different tasks (note onset detection vs fundamental pitch estimatation).
4) Adversarial attacks for generative/transformative models: Most of the existing literature has focused on generating adversaries for classification tasks. in classification it is quite clear what the goal of the adversary is. However, with the increasing popularity of generative and transformative models in music applications it is important to explore adversarial attacks on generative models to ensure that
5) Multi-task learning as a defense against adversaries: The current understanding for the existence of adversarial examples is that deep learning models might be learning the most comprehensive set of features to define the input. The intent with exploring multi-task learning is that by having different output goals the model will be forced to learn a more complete set of features to explain the input.

IV. Timeline

A. Project 1: Implementing existing attacks and defenses in MIR tasks

The goal of this project is to bridge the gap between computer vision and music information retrieval on adversarial attacks and defenses. The dataset we are going to use is GTZAN . GTZAN is a genre recognition dataset which contains 1000 tracks that are 30 seconds long each of 10 different genres. Genre recognition is one of the most popular tasks in music information retrieval with widespread commercial use. We can obtain different types of machine learning algorithms trained on genre recognition. In addition, previous work adversarial attacks in music used the GTZAN dataset to create adversarial examples [20] so we can use it as a benchmark to compare our work.

1) Month 10 to Month 12
2) ISMIR 2019 conference, if not ISMIR workshop 2019

B. Project 2: Testing on well defined MIR tasks

In the introduction for this proposal we said that genre classification has opinion based truths. Genre is a subjective definition and what one person might classify as hard rock might be classified as metal by someone else. In addition to there being issues with genre as a label, research has shown that there are inherent problems with the GTZAN dataset []. Both of these qualities make GTZAN not ideal for testing adversarial algorithms.

Instrument classification is an ideal task for creating adversarial examples because you can create a dataset with small number of well defined classes and a lot of examples per class. NSynth [9] is a datset that contains a large collection of notes produced by different instruments. It is very useful for instrument classification because it is a dataset of monophonic clean sounds which means it should be easy to train a high accuracy classifier on this dataset. The IRMAS [3] dataset is an alternative to NSynth, but IRMAS is made for predominant instrument recognition so it contains a polyphonic mixture of instruments.

1) Month 11 to Month 14
2) ICASSP 2020, if not NeurIPS workshop 2019

C. Project 3: Verifying the properties of adversarial attacks for music

The main properties that we want to test are as follows:

1) Transferability of adversaries between different deep learning models trained on the same dataset
2) Using an ensemble to create stronger adversaries
3) Transferability of adversaries between different datasets that have the same classes
4) Universal perturbation to generate adversaries easily
5) Robustness of adversaries to compression such as mp3
6) Robustness of adversaries to playing over the speaker and recording it

There are a lot of interesting properties to verify. However, conducting experiments to verify all of these properties is going to take too long. By the time we start working on this project we hope to identify the most important properties to focus on, these will most likely be those properties that lead naturally into the next project on MIR specific properties of adversaries.

In order to do research on the properties of adversaries we require two or more datasets that have similar classes and two or more deep learning models trained on each dataset.

1) Month 15 to Month 18
2) ISMIR 2020

D. Project 4: Identifying properties of adversarial examples in MIR

Deep learning tasks in MIR have different types of inputs for the same tasks. Some deep learning models use raw audio, some use mel spectrograms, some use short time fourier transforms, some use constant -Q transforms etc. It would be interesting to see if adversarial examples generated on a model that uses one type of input works on a model with a different type of input. In effect, we woulbe be looking for transferability of adversaries between different input representations.

It would be interesting to see if music adversaries transfer across different MIR tasks. For example, if we generate an adversary for instrument classification and test it on pitch estimation will it still work as an adversary and vice versa.

E. Project 5: Multi-task learning as a defense

F. Project 6: Adversaries for generative and transformative music applications

V. Conclusion

Summarize the whole proposal

TABLE I
TIMELINE FOR THE PROJECT

S no.	Week No.	Resaerch ideas	Goals
1	February 2019	Reimplement Kereliuk, Sturm and Larsen [20] with a tensorflow implementation of spectrogram extraction	Observe differences in perturbing spectrogram vs raw audio
2	February 2019	Repeat the previous experiments with different attack methodologies	Understand the most effective and least effective attacks for music
3	February 2019	Implement existing defense methodologies and test their effectivesness against these attacks	Identify topics to focus on
4	February 2019		
2	February 2019		
2	February 2019		
2	February 2019		
2	February 2019		
2	February 2019		
2	February 2019		
2	February 2019		
2	February 2019		

ACKNOWLEDGMENT

I would like to thank Prof. Mark Sandler, Dr. Emmanouil Benetos, Dr. Ning Xu and Mr. SKoT McDonald for the help and support they have given in shaping my stage-0 report and developing a long term plan for my PhD. I also want to thank ROLI for giving me access to their facilities which help my research.

REFERENCES

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing Robust Adversarial Examples. *arXiv:1707.07397 [cs]*, July 2017. arXiv: 1707.07397.
[2] Arjun Nitin Bhagoji, Daniel Cullina, Chawin Sitawarin, and Prateek Mittal. Enhancing Robustness of Machine Learning Systems via Data Transformations. *arXiv:1704.02654 [cs]*, April 2017. arXiv: 1704.02654.
[3] Juan J Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera. A COMPARISON OF SOUND SEGREGATION TECHNIQUES FOR PREDOMINANT INSTRUMENT RECOGNITION IN MUSICAL AUDIO SIGNALS. page 6, 2012.
[4] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. *arXiv:1608.04644 [cs]*, August 2016. arXiv: 1608.04644.
[5] Keunwoo Choi, George Fazekas, and Mark Sandler. Explaining Deep Convolutional Neural Networks on Music Classification. *arXiv:1607.02444 [cs]*, July 2016. arXiv: 1607.02444.
[6] Yandre M. G. Costa, Luiz S. Oliveira, and Carlos N. Silla. An evaluation of Convolutional Neural Networks for music classification using spectrograms. *Applied Soft Computing*, 52:28–38, March 2017.
[7] S. Dieleman and B. Schrauwen. End-to-end learning for music audio. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6964–6968, May 2014.
[8] Sander Dieleman and Benjamin Schrauwen. MULTISCALE APPROACHES TO MUSIC AUDIO FEATURE LEARNING. page 6.
[9] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. *arXiv:1704.01279 [cs]*, April 2017. arXiv: 1704.01279.
[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Models. *arXiv:1707.08945 [cs]*, July 2017. arXiv: 1707.08945.
[11] Reuben Feinman, Ryan R. Curtin, Saurabh Shintre, and Andrew B. Gardner. Detecting Adversarial Samples from Artifacts. *arXiv:1703.00410 [cs, stat]*, March 2017. arXiv: 1703.00410.
[12] Alf Gabrielsson. Music Performance Research at the Millennium. *Psychology of Music*, 31(3):221–272, July 2003.
[13] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572 [cs, stat]*, December 2014. arXiv: 1412.6572.
[14] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schlkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.
[15] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (Statistical) Detection of Adversarial Examples. *arXiv:1702.06280 [cs, stat]*, February 2017. arXiv: 1702.06280.
[16] Dan Hendrycks and Kevin Gimpel. Early Methods for Detecting Adversarial Images. *arXiv:1608.00530 [cs]*, August 2016. arXiv: 1608.00530.
[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531 [cs, stat]*, March 2015. arXiv: 1503.02531.
[18] Sepp Hochreiter and Jrgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, November 1997.
[19] Eric J. Humphrey, Juan Pablo Bello, and Yann Lecun. *Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics*.
[20] C. Kereliuk, B. L. Sturm, and J. Larsen. Deep Learning and Music Adversaries. *IEEE Transactions on Multimedia*, 17(11):2059–2071, November 2015.
[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv:1607.02533 [cs, stat]*, July 2016. arXiv: 1607.02533.
[22] Xin Li and Fuxin Li. Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5775–5783, Venice, October 2017. IEEE.
[23] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into Transferable Adversarial Examples and Black-box Attacks. *arXiv:1611.02770 [cs]*, November 2016. arXiv: 1611.02770.
[24] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectier Nonlinearities Improve Neural Network Acoustic Models. page 6.
[25] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On Detecting Adversarial Perturbations. *arXiv:1702.04267 [cs, stat]*, February 2017. arXiv: 1702.04267.

[26] Saumitra Mishra, Bob L Sturm, and Simon Dixon. UNDERSTANDING A DEEP MACHINE LISTENING MODEL THROUGH FEATURE INVERSION. page 8, 2018.

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *arXiv:1610.08401 [cs, stat]*, October 2016. arXiv: 1610.08401.

[28] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *arXiv:1605.07277 [cs]*, May 2016. arXiv: 1605.07277.

[29] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The Limitations of Deep Learning in Adversarial Settings. *arXiv:1511.07528 [cs, stat]*, November 2015. arXiv: 1511.07528.

[30] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. *arXiv:1511.04508 [cs, stat]*, November 2015. arXiv: 1511.04508.

[31] Jonathon Shlens. A Tutorial on Principal Component Analysis. *arXiv:1404.1100 [cs, stat]*, April 2014. arXiv: 1404.1100.

[32] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034 [cs]*, December 2013. arXiv: 1312.6034.

[33] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199 [cs]*, December 2013. arXiv: 1312.6199.

[34] Paul Viola and Michael J Jones. Robust Real-Time Face Detection. page 18.

[35] Xinxi Wang and Ye Wang. Improving Content-based and Hybrid Music Recommendation using Deep Learning. In *Proceedings of the ACM International Conference on Multimedia - MM '14*, pages 627–636, Orlando, Florida, USA, 2014. ACM Press.

[36] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur. Improving deep neural network acoustic models using generalized maxout networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 215–219, May 2014.

[37] Xinquan Zhou and Alexander Lerch. CHORD DETECTION USING DEEP LEARNING. page 7, 2015.