Stage 0: Adversarial attacks and anomaly detection in MIR

Vinod Subramanian, QMUL, ROLI Ltd.

Abstract

The abstract goes here.

Index Terms

MIR

I. Introduction

One of the biggest challenges of deep learning that remains today is the interpretability of what the deep learning model is learning. Due to the non-linearity and complexity of these models it is not a simple task to understand exactly what is being learned. With deep learning models slowly making their way into the real world and become a fixture of our daily lives it is even more important for us to understand what deep learning models are learning and what are the risks associated with deploying these deep learning models in the world.

Recent research has shown that by perturbing the input to a deep learning model by an imperceptible amount of non-random values we can change the output of the classifier. These ”attacks” to the model are referred to as adversarial attacks. This means that we can misclassify objects, misclassify speech to text etc. Some of these have security and safety consequences. If a self driving car incorrectly classifies a wall as a road it could lead to dangerous accidents.

The security aspect of adversarial attacks is often talked about but there is another, more fundamental question. Are we fundamentally training deep learning networks the wrong way? The success of adversarial attacks seems to indicate that there is a huge flaw in the assumption that deep learning models are learning in a manner that humans can interpret and understand.

So far adversarial attacks are explored in the computer vision domain and reinforcement learning domain. If we are to arrive at general observations and solutions to adversarial attacks and defenses we need to study this phenomenon in different domains and see what is generalizable and what is unique to a particular domain. I want my PhD to focus on exploring adversarial attacks and defenses in the area of music information retrieval (MIR).

II. Background

I’d like to see if I can create a summary of some of the research techniques used in some of the literature I am reviewing in the form of a table to compare techniques and performance.

A. What are adversarial attacks?

Szegedy et al. [22] 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 [6] 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, ReLUs, maxout networks, CNNs 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 [6] used this assumption on deep learning models to generate adversarial examples and came up with a family of fast gradient approaches to generate adversarial examples. The success of these adversarial examples lent credence to the fact that deep learning models are susceptible to adversarial examples on account of being 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. [22] 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 [6]. 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:

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

   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. [18] 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:

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

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 [21] based on the forward derivative. The saliency map shows which input features are most important in determining the output.

$$\text{For } i \in 1..M \text{ and } t \text{ as target output}$$

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

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 [3]. The problem for adversarial attacks is formulated the same was as Szegedy et al. [22]. The classifier is denoted as C with input x:

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

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:

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

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 [3]. Carlini and Wagner [3] 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 [17] 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. [14] 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. [16] 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 [12] 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. [4] 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. [10] and adapted by Papernot et al. [19]. 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.[8] suggested a method where an additional class is added in the retraining process for adversarial examples. Metzen et al. [15] 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 [20]. Hendrycks and

if using array.sty, it might be a good idea to tweak the value of

TABLE I
AN EXAMPLE OF A TABLE

1			
2	Four		

Gimpel [9] 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 [13] apply PCA on the convolutionaly layer outputs and using a cascade classifier [23] 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. [8] suggest a method that uses Maximum Mean Discrepancy (MMD) [7] 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. [5] 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 [11] applied the attacks developed by Szegedy et al. [22] 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

IV. DATASETS

Talk about dataset generation and licenses. Unclear on what exactly the new dataset will be. Perhaps something like a simple dataset that serves as a toy example dataset for MIR tasks. Need to talk to ROLI and QMUL about resources and direction of PhD

V. CONFERENCES

1) ISMIR
2) ICASSP
3) ICLR
4) NeurIPS
5) EUSIPCO
6) TISMIR
7) ICML

VI. TIMELINE

Table listing different projects, their timeline and targeted conferences

VII. CONCLUSION

Summarize the whole proposal

Acknowledgment

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] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. *arXiv:1608.04644 [cs]*, August 2016. arXiv: 1608.04644.

[4] 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.

[5] 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.

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv:1412.6572 [cs, stat]*, December 2014. arXiv: 1412.6572.

[7] 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.

[8] 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.

[9] Dan Hendrycks and Kevin Gimpel. Early Methods for Detecting Adversarial Images. *arXiv:1608.00530 [cs]*, August 2016. arXiv: 1608.00530.

[10] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531 [cs, stat]*, March 2015. arXiv: 1503.02531.

[11] C. Kereliuk, B. L. Sturm, and J. Larsen. Deep Learning and Music Adversaries. *IEEE Transactions on Multimedia*, 17(11):2059–2071, November 2015.

[12] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv:1607.02533 [cs, stat]*, July 2016. arXiv: 1607.02533.

[13] 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.

[14] 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.

[15] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On Detecting Adversarial Perturbations. *arXiv:1702.04267 [cs, stat]*, February 2017. arXiv: 1702.04267.

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

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

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