DEEP LEARNING, AUDIO ADVERSARIES, AND MUSIC CONTENT ANALYSIS

Corey Kereliuk
Technical Univ. of Denmark
DTU Compute

Bob L. Sturm
Queen Mary Univ. of London
Centre for Digital Music

Jan Larsen
Technical Univ. of Denmark
DTU Compute

ABSTRACT

We present the concept of *adversarial audio* in the context of deep neural networks (DNNs) for music content analysis. An adversary is an algorithm that makes minor perturbations to an input that cause major repercussions to the system response. In particular, we design an adversary for a DNN that takes as input short-time spectral magnitudes of recorded music and outputs a high-level music descriptor. We demonstrate how this adversary can make the DNN behave in any way with only extremely minor changes to the music recording signal. We show that the adversary cannot be neutralised by a simple filtering of the input. Finally, we discuss adversaries in the broader context of the evaluation of music content analysis systems.

Index Terms— Deep Learning, Music Content Analysis

1. INTRODUCTION

Deep neural networks (DNNs) are being applied with some success to problems of music content analysis [1–5], but what they are actually learning to do is not clearly known. Motivated by the fact that they can perform end-to-end learning [2], can learn hierarchical representations [6], and have shown remarkable success in benchmark tasks of other domains [7], DNNs have been proposed [8] as a way to accelerate progress in the field of music content analysis [9] — some applications of which are still outperformed by far simpler non-content based systems [10].

A trained DNN is highly complex and difficult to analyze. Researchers have tried to determine the contributions of its components (neurons, layers, etc.) in relation to its learning problem. For instance, Krizhevsky et al. [7] find the first layer of their deep convolutional network for image content recognition learns edges and color gradients of various orientations. For the same kind of system, Zeiler et al. [11] find units in its highest layers are highly activated by dogs, human faces, bird legs, and grass patches. For DNN applied to high-level segmentation of musical audio, or finding "edges" in the music, Schlueter at al. [5] have applied the approach of [12] to make sense of the roles of the various units. Dieleman and Schrauwen [2] have also found that the first hidden layer of a convolutional DNN trained on audio signals acts as a bank of bandpass filters.

Such unit-localised interpretations of a trained DNN, however, are challenged by the results of Szegedy et al. [13]. They find that a high-performing DNN trained in the context of image object recognition can be easily fooled by an *adversary*: an algorithm that perturbs an input image to make the DNN misclassify it with high confidence. These perturbations (additive noise) are often very small, producing adversarial instances that are indistinguishable from the "originals" from the standpoint of human visual perception. A recent variant on this theme for DNN is provided by Nguyen et al., who synthesize artificial images with unidentifiable content that the system still classifies with high confidence [14].

In our work here, we adapt the work of Szegedy et al. [13] to the context of DNN trained to analyze the content of music audio recordings. We discuss special considerations one must take when the input to the DNN (magnitude spectra) does not map to a unique and real time-domain signal. Within a specific problem of music content analysis, we demonstrate how the adversary can fool the DNN-based system. As found in the context of DNN applied to image content analysis [13], our adversary can make our DNN-based system produce high-confidence decisions with arbitrary labels. We examine the characteristics of the differences between the adversarial examples and the "originals", and show how preprocessing the input by linear time-invariant filtering does not defeat the adversary. Finally, we discuss our results in relation to recent work in music content analysis challenging the notion that such systems may not in fact be solving the intended listening problem at all [15–17]. Code to reproduce our experiments is available here: `https://github.com/coreyker/dnn-mgr`.

2. AUDIO ADVERSARY

Consider a DNN where the input is a D-dimensional vectorized magnitude DFT frame X_n extracted from short-time Fourier transform (STFT) of an audio time series x

$$X_n = \big|\mathcal{F}(x)[m,n]\big|, m \in [0, \dots, D-1] \quad (1)$$

$$\mathcal{F}(x)[m,n] = \sum_l w[\langle l - nH \rangle_{L_w}]x[l] \exp(-j2\pi ml/M) \quad (2)$$

for $m \in \{0, \dots, M-1\}$, with hop-size H, M frequencies (bins), and w a window of support L_w. (The notation $\langle\cdot\rangle_{L_w}$ means modulo L_w.) As we are working with real audio signals, we only need to consider the first $D = M/2 + 1$ frequencies. Let the function $f_i(X_n) : \mathbb{R}_+^D \to [0,1]$ represent the i^{th} softmax output of the DNN, mapping a single input frame to the probability of label $i \in \{1, \dots, K\}$. The classification of a sequence of frames $X = (X_1, X_2, \dots, X_N) \in \mathbb{R}_+^{D\times N}$ is defined as the label with the maximum posterior probability over the sequence:

$$y(X) = \arg \max_{i\in\{1,\dots,K\}} \frac{1}{N}\sum_{n=1}^{N} f_i(X_n). \quad (3)$$

Other approaches to combining classifications can be used [18].

Following Szegedy et al. [13], we define an adversary as an optimization problem; however, we use a constrained optimization framework to allow the specification of the maximum signal-to-noise ratio of the adversarial examples, and we must take care of the fact that the input to the DNN is a magnitude DFT. We begin with an exemplar $X \in \mathbb{R}_+^{D\times N}$, and seek an adversarial example $\hat{X} \in \mathbb{R}_+^{D\times N}$ such that $\hat{y} = y(\hat{X}) \neq y(X)$ and the mean distortion

$$\|\hat{X} - X\|_F \leq N\epsilon. \quad (4)$$

The size of the perturbation $\hat{X} - X$ is limited by $\epsilon > 0$, which constrains the maximum signal-to-noise ratio (SNR) of the adversary:

$$\text{SNR}(\epsilon) = 20\log_{10}\frac{\|X\|_F}{\|\hat{X} - X\|_F} \geq 20\log_{10}\frac{\|X\|_F}{\epsilon N} \tag{5}$$

The adversary takes a target label $\hat{y}$, and for each frame of X searches for an adversarial frame $\hat{X}_n$ by solving

$$\hat{X}_n = \arg\min_{Z\in\mathcal{C}(X_n)} \mathcal{L}(Z, \hat{y}) \tag{6}$$

where $\mathcal{C}(X_n) = \{Z \in \mathbb{R}_+^D : \|Z - X_n\|_2 \leq \epsilon\}$ represents an ϵ-ball centered around the frame, and $\mathcal{L} : \mathbb{R}_+^D \times \{1, \ldots, K\} \to \mathbb{R}_+$ is the loss function used to train the DNN, i.e., the cross-entropy cost: $\mathcal{L}(Z, \hat{y}) = -\log f_{\hat{y}}(Z)$.

A local minimum of (6) could be found using projected gradient descent, which is initialized with $\hat{X}_n^{(0)} = X_n$, and iterates

$$\hat{X}_n^{(k+1)} = \mathcal{P}_{\mathcal{C}}(\hat{X}_n^{(k)} + \mu\nabla_X\mathcal{L}(\hat{X}_n^{(k)}, \hat{y})) \tag{7}$$

where μ is the descent step size, and $\mathcal{P}_{\mathcal{C}}(\cdot)$ is the least squares projection onto the set $\mathcal{C}$. However, this approach will not result in *valid* adversarial audio examples since the estimated sequence of adversarial frames may not correspond to a *real* time-domain signal. This is due to the fact that the individual STFT frames in (2) are not independent since they are computed from overlapping segments of the input signal. This in turn means that not all coefficients $C \in \mathbb{C}^{M\times N}$ correspond to the STFT of a real input signal, i.e.., $\mathcal{F}(\mathcal{F}^{-1}(C)) \neq C \quad \forall C \in \mathbb{C}^{M\times N}$ where $\mathcal{F}^{-1}(C)$ is the inverse STFT of $C \in \mathbb{C}^{M\times N}$.

Thus, in order to generate valid adversarial audio examples, our adversary projects the time-frequency coefficients onto the space of valid time-domain sequences. We do this using the Griffin and Lim algorithm [19], which seeks to minimize for sequence X

$$\arg\min_{C\in\mathbb{C}^{M\times N}} \sum_{n=0}^{N-1}\sum_{m=0}^{D-1} \Big| |\mathcal{F}(\mathcal{F}^{-1}(C))[m,n]| - X_n[m] \Big| \tag{8}$$

where $X_n[m]$ is the value of the mth row of X_n. This minimization can be done by using alternating projections; but we have found in practice that it is sufficient to apply a single set of projections:

$$\begin{aligned}\mathcal{P}_{GL}(X) = \big|\mathcal{F}(\mathcal{F}^{-1}(\tilde{X}\cdot\exp(jP))[m,n])\big|& \\ n \in \{0, \ldots, N-1\}, m \in \{0, \ldots, D-1\}&\end{aligned} \tag{9}$$

Algorithm 1 From exemplar audio DFT magnitude frame sequence X. search for adversarial audio DFT magnitude frame sequence $\hat{X}$ for which DNN produces class $\hat{y}$ with confidence T in at most $k_{\max}$ steps, and with an SNR of at least SNR(ϵ)

```
parameters: $\hat{y}$, $\epsilon$, $T$, $k_{\max}$
init: $X^{(0)} = X, k = 0$
repeat
    $U_n \leftarrow X_n^{(k)} + \mu_n\nabla_X\mathcal{L}(X_n^{(k)}, \hat{y}) \quad \forall n \in [1, N]$ {Gradient step}
    $Z \leftarrow \mathcal{P}_{GL}(\max(0, U))$ {STFT validity}
    $\nu \leftarrow \max(0, ||Z - X||_F/(\epsilon N) - 1)$ {Lagrange mult.}
    $X^{(k+1)} \leftarrow (1+\nu)^{-1}(Z + \nu X)$ {SNR constraint}
    $k \leftarrow k + 1$
until $\frac{1}{N}\sum_{n=1}^{N} f_{\hat{y}}(X_n^{(k)}) \geq T$ or $k = k_{\max}$
return: $\hat{X} = X^{(k)}$
```

where $\cdot$ is an element-wise product between the STFT phase, $\angle\mathcal{F}(x) = P \in [-\pi, \pi]^{M\times N}$, and the modulus

$$\tilde{X}[m,n] = \begin{cases} X[m,n] & \text{if } 0 \leq m < D \\ X[M-m,n] & \text{if } D \leq m < M \end{cases} \tag{10}$$

The pseudo-code in Alg. 1 summarises the audio adversary.

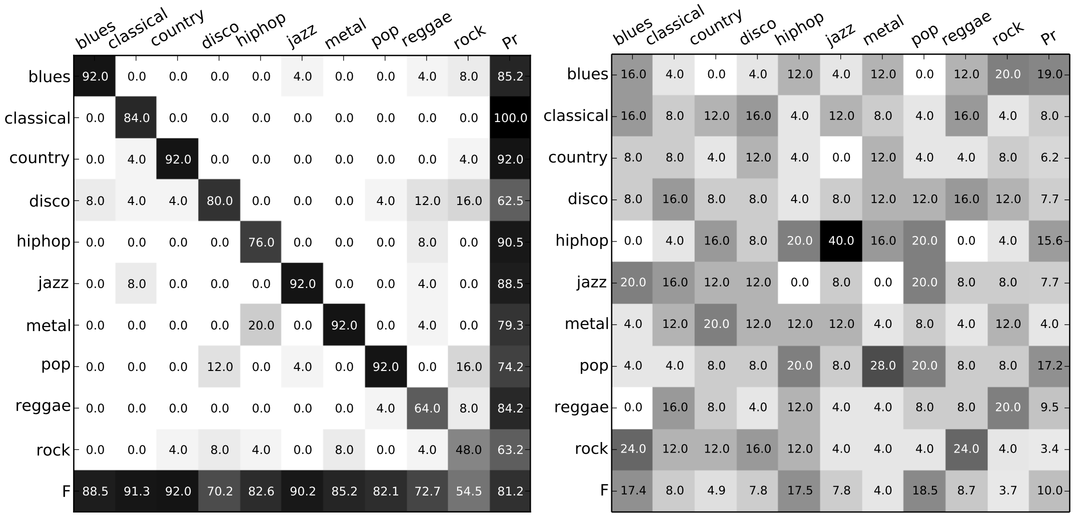
3. DEMONSTRATION

To demonstrate this adversary, we consider a DNN-based system built for a music content analysis benchmark: music classification in the *GTZAN* dataset [16, 20]. Two recent works applying DNN to the same problem are [3, 4], and both report measuring performances near human-level classification accuracy. Our DNN is a reproduction of the one developed in [4], which uses 3 fully connected hidden layers with 500 rectified linear units per layer trained using a random 50/25/25 (train/validation/test) stratified partition of *GTZAN*. The STFT of a music audio recording is computed from a 46ms (1024 point) Hann window with 50% overlap. We train the DNN using stochastic gradient descent with dropout [21]. As in [4], the music classification system vectorizes the hidden layer activations of the trained DNN, creates "bags of frames" using 130 sequential activations computed from 3 sequential seconds of audio, and summarizes each bag by means and standard deviations in each dimension. These statistics form the feature vectors that are then classified by a trained random forest (RF) classifier of 500 trees. A music recording is thus classified by a majority vote over all classifications of its feature vectors. The DNN is thus a feature extractor.

The figure of merit (FoM) in Fig. 1(a) illustrates the performance of this system. As in [3, 4], we perform no fault filtering of the *GTZAN* dataset [16]; however, even with fault filtering these numbers provide no meaningful indication that the system has learned anything about music [15–17]. They only serve to show the system has learned *something* about *GTZAN*. Our normalized classification accuracy is slightly lower than the 83% measured in [4]. This slight difference may be attributable to the particular dataset partitioning and numerous hyperparameters of the DNN and its training, e.g., weight initialization, gradient step-size, use of regularization, stopping criteria, and so on.

We now create three adversaries that make the DNN-based system behave in each of the following ways: A1) correct with high confidence only 10% of the time; A2) always correct with high confidence; A3) selects the same label with high confidence. A1 draws the target label for a test instance from a uniform distribution over the labels. A2 sets the target label to the ground truth of the test instance. A3 sets the target label to "Jazz" for any test instance. In all cases, we set the parameters of the adversaries to: $k_{\max} = 100$, $T = 0.9$, $\mu = 0.1$, and ϵ such that SNR(ϵ) $\geq$ 15dB. It is important to note that the adversary is only perturbing the input to the pre-trained DNN, and thus does not change the parameters of the DNN, or of the random forest classifier.

Figure 1(b(is the FoM of the same DNN as in Fig. 1(a) but with the intervention of A1. Clearly, A1 is able to make the system perform no better than chance but with high confidence in its decisions. The FoM of the same system but with the intervention of A2 produces a near perfect classification accuracy of 99.6%; and the FoM of the same system but with the intervention of A3 results in 96.8% of the test instances being classified as "Jazz" with high confidence. Figure 2 illustrates a sample excerpt of a *GTZAN* Metal excerpt, together with the adversarial example confidently labeled "Jazz" by the system. We see that the magnitude spectrograms match relatively well at low frequencies and deviate at high frequencies.



	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock	Pr
blues	92.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	4.0	8.0	85.2
classical	0.0	84.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0
country	0.0	4.0	92.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	92.0
disco	8.0	4.0	4.0	80.0	0.0	0.0	0.0	4.0	12.0	16.0	62.5
hiphop	0.0	0.0	0.0	0.0	76.0	0.0	0.0	0.0	8.0	0.0	90.5
jazz	0.0	8.0	0.0	0.0	0.0	92.0	0.0	0.0	4.0	0.0	88.5
metal	0.0	0.0	0.0	0.0	20.0	0.0	92.0	0.0	4.0	0.0	79.3
pop	0.0	0.0	0.0	12.0	0.0	4.0	0.0	92.0	0.0	16.0	74.2
reggae	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	64.0	8.0	84.2
rock	0.0	0.0	4.0	8.0	4.0	0.0	8.0	0.0	4.0	48.0	63.2
F	88.5	91.3	92.0	70.2	82.6	90.2	85.2	82.1	72.7	54.5	81.2

(a) Original Input

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock	Pr
blues	16.0	4.0	0.0	4.0	12.0	4.0	12.0	0.0	12.0	20.0	19.0
classical	16.0	8.0	12.0	16.0	4.0	12.0	8.0	4.0	16.0	4.0	8.0
country	8.0	8.0	4.0	12.0	4.0	0.0	12.0	4.0	4.0	8.0	6.2
disco	8.0	16.0	8.0	8.0	4.0	8.0	12.0	12.0	16.0	12.0	7.7
hiphop	0.0	4.0	16.0	8.0	20.0	40.0	16.0	20.0	0.0	4.0	15.6
jazz	20.0	16.0	12.0	12.0	0.0	8.0	0.0	20.0	8.0	8.0	7.7
metal	4.0	12.0	20.0	12.0	12.0	12.0	4.0	8.0	4.0	12.0	4.0
pop	4.0	4.0	8.0	8.0	20.0	8.0	28.0	20.0	8.0	8.0	17.2
reggae	0.0	16.0	8.0	4.0	12.0	4.0	4.0	8.0	8.0	20.0	9.5
rock	24.0	12.0	12.0	16.0	12.0	4.0	4.0	4.0	24.0	4.0	3.4
F	17.4	8.0	4.9	7.8	17.5	7.8	4.0	18.5	8.7	3.7	10.0

(b) Input intercepted by adversary A1

Figure 1: Figure of merit ($\times 100$) for our DNN-based music classification system trained and tested in the GTZAN dataset [16, 20]. Columns represent the true class; rows denote label chosen by system; the diagonal contains the per class recall; the off-diagonal entries are confusions; the rightmost column is the precision; the bottom row is the F-score; and the last element along the diagonal is the mean recall.

Table 1 shows the results of the DNN with the intervention of an ensemble of A3, each set to one *GTZAN* label. Our DNN-based system, unlike the previous one, is trained and validated using all of *GTZAN*. The SNR of each adversarial example is listed in the table, where the first number corresponds to confidence $T = 0.5$, and the bracketed number to $T = 0.9$. For $T = 0.5$, we find we can generate adversarial examples at a high average SNR (34.5dB) in every label in *GTZAN*. Similarly for $T = 0.9$, we can generate all but one adversarial example at a still high average SNR (26.8dB).

4. DISCUSSION

Our results show how the adversary (Alg. 1) can successfully manipulate DNN-based music content analysis systems. We also see that though the adversary only knows of the DNN, and not the subsequent random forest classifier, it is able to perturb the input such that it fools both. As a result of the SNR constraint, we find the adversarial examples differ very little from their exemplars; their SNR is often much higher than our lower bound of 15dB. Informal listening tests confirm that the differences between the adversarial examples and the "originals" are quite minor, which can be likened to subtle additive noise. (See link in caption of Table 1.)

The spectral characteristics in Fig. 2 suggest a simple method to defeat an adversary might be to preprocess the input by a low-pass filter in order to eliminate the effects of the differences at high frequencies. We tested this hypothesis and found that even though applying a low-pass filter does reduce the number of confusions, the mean recall is still far below what we obtained on the original set (by more than 20 percentage points). Similar results are found in [22], who apply denoising autoencoders for defeating adversaries.

Our results can be seen as an empirical verification of the work of Szegedy et al. [13] extended to the audio domain. In their case, the adversarial images appear identical to the originals, while our audio adversaries do sound differently from the "originals" – though among audio adversaries it can be difficult to pinpoint the differences. This discrepancy could be related to a few things at least. First, our input dimensionality of 513 dimensions is far lower than the high-dimensional input color images used in [13]. Our adversaries thus have fewer dimensions in which to hide perturbations [23], and so must make larger contributions to the STFT magnitude frames. Experiments with larger window sizes can confirm this. Second, the human auditory system is exceptionally sensitive and so the perturbations of audio adversaries might be harder to pass undetected than those of images. This suggests modifying (4) by perceptual weighting [24] to try to make an adversary that can produce perceptually transparent perturbations.

Regardless of the perceptible noise caused by our audio adversaries, the music embedded in the sampled audio is not affected to such an extent that its *GTZAN* class should become so confidently mistaken by a system that previously performed so well. In other words, had the DNN-based system producing the FoM in Fig. 1(a) learned some of the important high-level concepts that underlie the labels in *GTZAN* – whatever those are – one should not expect it to be so easily fooled by these adversaries. This clear lack of generalisation of the system in the face of its high FoM is referred to by [23] as a "Potemkin village." Coinciding with that colorful assessment, we have called such a system a "horse" [17] in reference to a famous horse named "Clever Hans" that appeared capable of performing arithmetic because he learned to respond to a cue common to people asking him questions that already knew the answer [25]. In fact, we have shown [17] how different music content analysis systems can be similarly fooled to do the same things our adversaries here accomplish. Though the approach in [17] is one of brute force searching with linear time invariant filtering, rather than the optimisation approach used here and in [13], we are led to the same question: What has this system *actually* learned to do?

One can argue that our experiments here have limited significance for music genre recognition since we use a small dataset, and one that has been shown to have faults [16]. The focus of this paper is not genre recognition (whatever that means), but about adapting and testing a principled method of creating adversaries that sever the achilles heel of a machine learning system. Whether we used *GTZAN* or the million song dataset [26], the important matter is that we have taught a classification system to reproduce the ground truth of some benchmark music dataset, that Fig. 1(a) shows this system

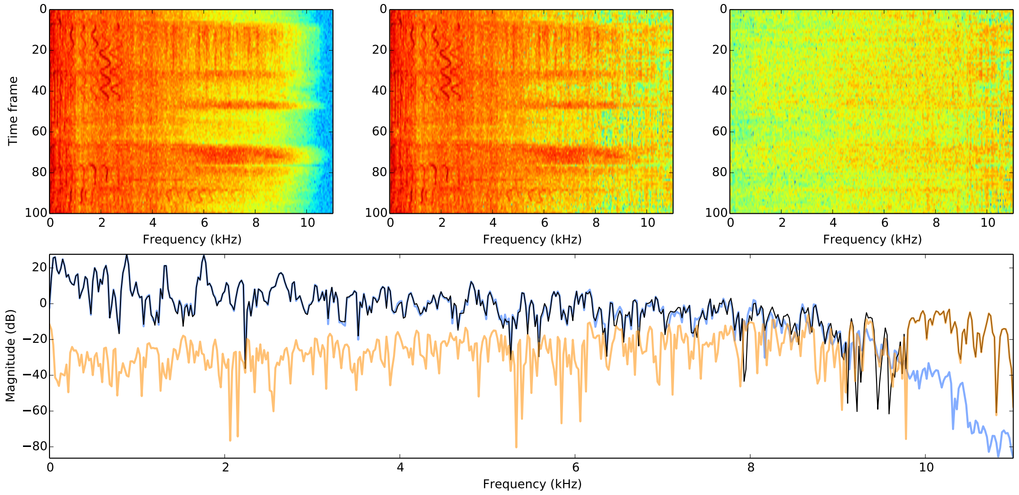


Figure 2: *GTZAN* Metal excerpt "Flight of Icarus", Iron Maiden. Top left: STFT of original. Top middle: STFT of adversarial example (labeled Jazz). Top right: STFT of difference. Bottom: a single STFT frame (light blue: original, black: adversary, orange: difference).

	Output GTZAN Label									
Music Recording Input	*Blues*	*Classical*	*Country*	*Disco*	*Hiphop*	*Jazz*	*Metal*	*Pop*	*Reggae*	*Rock*
Little Richard, "Last Year's Race Horse"	32 (23)	29 (23)	36 (25)	36 (26)	36 (25)	33 (24)	32 (24)	31 (25)	42 (26)	36 (25)
Rossini, "William Tell Overture"	32 (25)	37 (30)	40 (29)	43 (28)	34 (24)	36 (29)	33 (25)	34 (26)	37 (26)	37 (28)
Willie Nelson, "A Horse Called Music"	25 ( )	25 (20)	30 (27)	30 (20)	26 (19)	30 (25)	27 (23)	21 (20)	30 (23)	29 (23)
Simian Mobile Disco, "10000 Horses Can't Be Wrong"	31 (30)	36 (31)	38 (32)	45 (34)	41 (33)	40 (32)	33 (31)	47 (34)	42 (33)	38 (33)
Rubber Bandits, "Horse Outside"	27 (27)	27 (27)	36 (29)	42 (31)	38 (29)	34 (28)	32 (28)	37 (29)	36 (29)	35 (29)
Leonard Gaskin, "Riders in the Sky"	32 (23)	30 (25)	32 (23)	35 (25)	31 (22)	35 (29)	34 (23)	26 (23)	35 (25)	35 (24)
Jethro Tull, "Heavy Horses"	29 (26)	28 (26)	40 (29)	42 (29)	38 (28)	36 (28)	34 (28)	34 (28)	37 (28)	36 (29)
Echo and The Bunnymen, "Bring on the Dancing Horses"	29 (25)	28 (26)	38 (28)	43 (28)	35 (26)	34 (26)	33 (26)	33 (26)	36 (27)	38 (28)
Count Prince Miller, "Mule Train"	32 (30)	29 (30)	41 (33)	37 (34)	43 (33)	36 (31)	33 (31)	42 (34)	40 (33)	33 (33)
Rolling Stones, "Wild Horses"	30 (22)	32 (24)	37 (25)	40 (25)	31 (22)	34 (25)	31 (26)	32 (23)	37 (25)	37 (26)

Table 1: SNR of adversarial examples produced by an ensemble of adversaries labelling any input with all *GTZAN* labels. The SNR is listed for both confidence thresholds $T = 0.5$ ($T = 0.9$ in brackets). The average SNR of the perturbations is 34.5 (26.8) dB. These results can be auditioned here: `http://www.eecs.qmul.ac.uk/~sturm/research/DNN_adversaries/`

demonstrates a remarkable ability to reproduce the ground truth in that dataset, and yet Fig. 1(b) shows that it is thoroughly defeated by an adversary that does not change the *music* in the input recordings. In other words, it is not important to our work whether the system has learned to identify and discriminate between the genres used by the music in the recording excerpts in *GTZAN*. What is important is the fact that the system has learned something about a dataset, but which can be defeated by principled means.

Another argument about our adversaries is that the perturbations they create are not "natural," i.e., they are highly unlikely in any given real-world context. The same is true for the adversarial images generated by [13]. We cannot definitively say what the existence of these adversaries means in terms of the system generalization to real-world music recordings. Goodfellow et al. [23] write, "The existence of adversarial examples suggests that ... being able to correctly label the test data does not imply that our models truly understand the tasks we have asked them to perform." In our case, the adversarial audio examples suggest that our DNN-based music content analysis system is not "reasoning" using high-level music-based concepts. This then has implications for its usefulness for connecting users with music and information about music – a principal goal of music informatics [9]. We are thus not interested in using audio adversaries as malicious agents of real-world music content analysis systems, but rather as an avenue to explore the inner workings and idiosyncrasies for improving such systems. In the context of music informatics, our work adds to a growing body of research pointing at systematic methodological flaws in evaluation that can lead to misrepresenting the state of the art [10, 15–17, 23].

5. CONCLUSION

We have presented audio adversaries for music content analysis systems, adapting the approach of Szegedy et al. [13] to a constrained optimisation framework that carefully treats input spectral frames to avoid invalid adversaries (time-frequency coefficients with no associated time-domain signal). Our demonstrations show how it is relatively easy to fool the same state-of-the-art DNN-based music content analysis system developed in [4]. We are able to make the system label the perturbed test set nearly perfectly, randomly, or entirely as "Jazz." We were also able to make the system confidently classify out-of-sample music using every *GTZAN* label. This brings up many questions regarding the most common method used to evaluation music content analysis systems, and the interpretation of the FoM computed from them: is it reflecting that in which we are really interested?

Acknowledgments

CK and JL were supported in part by the Danish Council for Strategic Research of the Danish Agency for Science Technology and Innovation under the CoSound project, case number 11-115328. This publication only reflects the authors' views.

6. REFERENCES

[1] L. Deng and D. Yu, *Deep Learning: Methods and Applications*. Now Publishers, 2014.

[2] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6964–6968.

[3] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks." in *Proc. Int. Soc. Music Info. Retrieval*, 2010, pp. 339–344.

[4] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2014, pp. 6959–6963.

[5] J. Schlüter and S. Böck, "Improved Musical Onset Detection with Convolutional Neural Networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 6979–6983.

[6] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in neural information processing systems*, 2012, pp. 1097–1105.

[8] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Feature learning and deep architectures: New directions for music informatics," *J. Intell. Info. Systems*, vol. 41, no. 3, pp. 461–481, 2013.

[9] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proc. IEEE*, vol. 96, no. 4, pp. 668–696, Apr. 2008.

[10] M. Slaney, "Web-scale multimedia analysis: Does content matter?" *IEEE Multimedia*, vol. 18, no. 2, pp. 12–15, Feb. 2011.

[11] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013. [Online]. Available: http://arxiv.org/abs/1311.2901

[12] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *CoRR*, vol. abs/1312.6034, 2013. [Online]. Available: http://arxiv.org/abs/1312.6034

[13] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6199

[14] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," *CoRR*, vol. abs/1412.1897, 2014. [Online]. Available: http://arxiv.org/abs/1412.1897

[15] B. L. Sturm, "Classification accuracy is not enough," *J. Intell. Info. Systems*, vol. 41, no. 3, pp. 371–406, 2013.

[16] B. Sturm, "The state of the art ten years after a state of the art: Future research in music information retrieval," *Journal of New Music Research*, vol. 43, no. 2, pp. 147–172, 2014.

[17] ——, "A simple method to determine if a music information retrieval system is a "horse"," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1636–1644, Oct 2014.

[18] L. Kuncheva, "A theoretical study on six classifier fusion strategies," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 281–286, Feb 2002.

[19] D. Griffin and J. S. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 32, no. 2, pp. 236–243, 1984.

[20] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech, Audio, Signal Process.*, vol. 10, no. 5, pp. 293–302, July 2002.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Machine Learning Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] S. Gu and L. Rigazio, "Towards deep neural network architectures robust to adversarial examples," *CoRR*, vol. abs/1412.5068, 2014. [Online]. Available: http://arxiv.org/abs/1412.5068

[23] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Rep.*, 2015.

[24] S. van de Par, A. Kohlrausch, R. Heusdens, J. Jensen, and S. H. Jensen, "A perceptual model for sinusoidal audio coding based on spectral integration," *EURASIP J. Applied Signal Process.*, no. 9, pp. 1292–13 042, 2005.

[25] O. Pfungst, *Clever Hans (The horse of Mr. Von Osten): A contribution to experimental animal and human psychology*, (translated by C. L. Rahn), Ed. New York: Henry Holt, 1911.

[26] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. Int. Soc. Music Info. Retrieval*, 2011.