Training Neural Nets to Achieve Audio-to-Score Translation: Opening the Black-Box

Anonymous Author(s)
Affiliation
Address
email

Abstract

It is suggested that the task of audio-to-score translation offers an adequate testbed to investigate the division of labor between background knowledge and machine learning in the domain of audio pattern recognition, with a controllable level of difficulty and the ability to synthesize a limitless amount of labelled data.

As a proof of concept, this paper focuses on pitch detection from audio signals. Extensive background knowledge is used to initialize simple convolutional neural nets (NN) and achieve the recognition of single notes with a decent accuracy. The performance achieved by trained NNs, however, is significantly higher. Some tentative interpretations of this fact are obtained by opening the black box and inspecting the modifications of the NN filters due to supervised learning.

1 Introduction and Rationale

Audio-to-speech translation has been intensively studied since the early 80s, as a key step of the natural language processing chain. While mainstream approaches used to be rooted in statistical and linguistic feature extraction [1], new end-to-end approaches rooted on deep neural networks are revolutionizing the field [2], along the same lines as in computer vision [3, 4].

This paper investigates a related but different issue, the audio-to-score translation (AST) in music, at the core of the Music Information Retrieval Evaluation eXchange (MIREX) international challenges [5]. This domain offers rich resources in terms of labelled data and feature extraction using extensive signal processing libraries. It thus supports the investigation of the benefits of using background knowledge, and specifically what knowledge can be given to the learning agent and what should better be learned, in a systematic way.

Data resources. MIREX 2018 offers a number of tracks, ranging from the identification of the type of music, to the name of the composer or the music mood. Former MIREX editions were concerned with other tracks including the alignment of audio to scores [6]. In the following, we focus on a subtask of audio-to-score translation task, *pitch detection*, formulated as a supervised learning task and tackled using the MAPS database [7]. It is claimed that the single note identification from audio signal is most similar to phoneme identification from audio signal, the elementary task of audio-to-speech translation.

Prior knowledge and pre-processing Dedicated and robust signal processing (SP) libraries[8] have been developed to achieve audio signal pre-processing and build time-frequency representations of the signal. These signal representations can be provided as pixel matrices or vectors to standard neural network (NN) architectures, together with the associated labels (single note or chords), and used to train NNs using standard supervised learning.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

An alternative is offered by using simple convolutional NN architectures on the raw signal. The point is that the convolutional mask aimed to detect a given frequency f can be either learned, or deterministically set to $W_{f,j} = \cos(\frac{2\pi f j}{f_s})$ for $j = [[0, N]]$, with N is the mask dimension and $f_s = 22.05$ kHz the sampling frequency of the audio signal.

Research Agenda This paper reports on the lessons learned from training NN (either from scratch, or using educated weight initialization) for pitch detection, claiming that these lessons are relevant to audio-to-speech phoneme recognition for the following reasons.

Pitch detection is assumedly easier than phoneme identification for a ground truth compact description of every note is available, e.g. in terms of Fourier coefficient. The diversity among notes is lesser than for phonemes (a note most similar but different from another note does not exist as far as musical instruments are well tuned), but still presents some ambiguities as a note evokes its harmonic by construction. The phoneme diversity due to the different speakers has its equivalent in music: a same note played on different instruments corresponds to different audio signals, due to the distinctive timbre of the musical instruments.

For these reasons, it is conjectured that pitch detection defines a relevant and informative testbed for phoneme detection. After briefly presenting Automatic Music Transcription, we describe the goals of the presented experiments, before reporting on the lessons learned, about how and when the SP knowledge is worth being exploited and what are the pitfalls.

2 Automatic Music Transcription

Automatic Music Transcription (AMT) [9] is a difficult task even for human experts [10]. Most of the time, AMT and pitch detection research is done on piano music, to avoid the variability due to the direct contact between the musician and the string that produces the sound. First attempts to automatically transcript polyphonic music due to Moorer [11] inspired many followers [12, 13, 14], mostly using statistical approaches. AMT and Music Information Retrieval were more recently revisited using Deep Learning techniques, specifically convolutional (CNNs) and deep belief networks (DBNs) [15, 16, 17].

Only pitch detection is considered in the following. The set of audio frequencies corresponding to the musical notes are distributed on a log-scale; the pythagorean scale consists in $f_n \propto f_0\, 2^{n/12}$, mapping the human ear perception.
Pitch detection can be achieved through extracting robust features from the raw audio signal. Compared to computer vision, the extracted features are only required to be invariant w.r.t. translation in time.[1] In principle, each such feature would correspond to a given pitch, facilitating the interpretation of the feature extraction achieved by the neural net.

3 Experimental setting and goals of experiments

The main goal of these experiments regards the nature and value of the prior SP knowledge concerning the target identification task. Specifically, does the available prior knowledge yield a perfect accuracy? Otherwise, how is this knowledge modified and improved through supervised training? Thirdly, how much gain is provided by the use of the prior knowledge? Finally, how do the NN hyper-parameters interact with the model space and how to best tune them?

These questions are empirically investigated on the MIDI Aligned Piano Sounds dataset [7]. We restricted ourselves to the identification of monophonic (isolated) notes, forming 88 distinct balanced classes. The raw audio signal is downsampled to 22kHz. Each file is divided into 2,400 sample-long examples (about 109ms), overlapping with a stride of 600. The period of the lowest note (A0, frequency 27.5Hz) spreads over 800 samples.

The NN architecture is a convolutional neural network applied on the raw audio signal, with a stride of 10 or 50, followed by a global max pooling. Each note is associated to one filter mask of dimension

[1] In all generality, the extracted features should also be invariant w.r.t. the timbre of the instruments; however only piano music is considered in the following.

800. The educated initialization of the corresponding filter weights is defined from the pure sine wave corresponding to the fundamental frequency (with no biases for the sake of interpretability).

Three architectures are compared: **model A** is a 1 layer CNN model with *initialized weights*, with *no training*, **model B** a 1 layer CNN model with *initialized weights*, with *training*, and **model C** a 1-layer CNN model with *random initialization*, with *training*.

Model performance is assessed using 8-fold cross validation (all samples of a same audio file being either in the training or in the test set).

4 Results

Impact of prior knowledge on accuracy and learning curve. As shown (Table 1), untrained model A yields 45% accuracy (significantly higher than the default accuracy for 88 balanced classes), thanks to its educated initialization. Still, trained models B and C both significantly outperform model A. The educated initialization (in model B) speeds up the convergence (compared to model C) toward the same optimum (Fig. 1). The performances are improved as the stride is reduced, especially so for high frequencies (Fig. 3).

Table 1: Comparison of the performances of models A, B and C on test sets after 100 epochs.

	Stride 50		Stride 10	
	Before training	After training	Before training	After training
initialized	0.45	0.64 ± 0.02	0.46	0.71 ± 0.02
not initialized	-	0.64 ± 0.02	-	0.70 ± 0.01

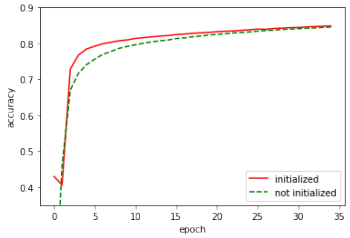

Figure 1: Learning curves of trained Model B (initialized with SP knowledge) and Model C (random initialization) on the training set (stride 50).

Opening the black-box. The masks in models A, B and C are inspected by plotting their Fourier coefficients (Fig. 2). By construction, masks in model A (Fig. 2.top) exactly correspond to the sought frequencies. On model B (SP initialization), supervised learning results in augmenting each filter with the harmonics of the associated frequency, particularly so for low frequencies; this augmentation explains the improved performance of model B compared to model A. On model C (random initialization), the same augmentation is observed with a notable difference: for low frequencies f, only harmonics $2f, 3f, \ldots$ are learned; frequency f itself is missed. This phenomenon confirms that good performances *per se* do not imply that the targeted concepts have been correctly identified.

The accuracy per note of models A, B, and C are depicted on Fig. 3. As mentioned above, we observe a significant improvement in performances when the model is trained (B and C), above all in a low frequency notes range. Indeed, for pitches between 40 and 80, the accuracy is roughly equivalent, whereas for pitches < 40, the trained masks obtain much better results. This is probably due to the timbral specificity of the piano, where the distribution of the energy for a given note is distributed differently depending on the pitch range.

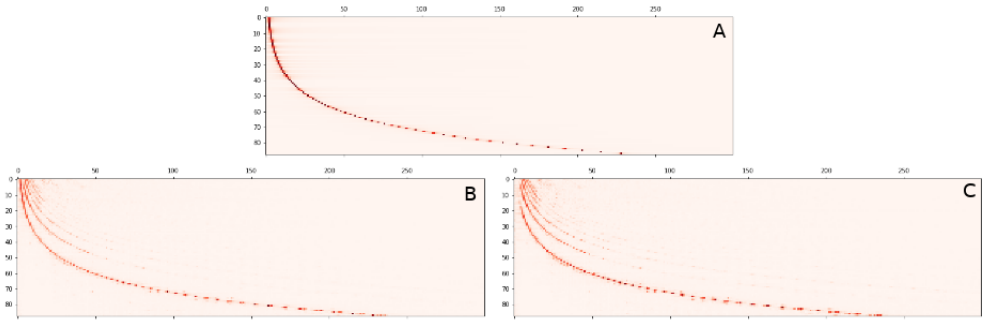

Figure 2: Models A, B and C: magnitude of frequencies detected for each filter (horizontal line), with stride 50, after 100 epochs training for models B and C.

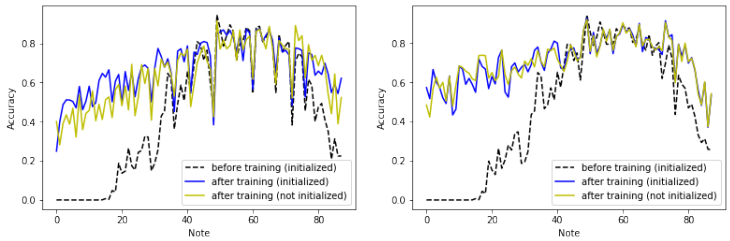

Figure 3: Models A, B and C: Accuracy per note. Left: stride 50; Right: stride 10. The downward accuracy spikes occur for frequencies whose period is a divider of the stride period (see discussion).

Adding a fully connected layer. Another model is investigated, adding a fully connected layer on the top of the conv/max pooling layers. The question is whether and how this additional FC layer will alleviate the limitations of the simple B and C architectures, and in practice, whether augmenting the filters with the associated harmonics will still be necessary. Most interestingly, it appears that the augmentation of the filters is no longer necessary (Fig. 4, left) and the fact that both a frequency and its harmonics are involved in the pitch detection is visible from the FC weights, below the diagonal (Fig. 4, right). This phenomenon is observed mostly with educated initialization. For both educated and random initialization, the predictive accuracy increases from circa 70% to 80%.

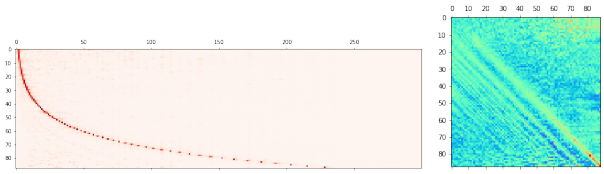

Figure 4: Model B augmented with a fully connected layer. Left: magnitude of frequencies detected for each filter; Right: weights of the fully connected layer; pixel (i,j) represents the weight of filter f_j to identify frequency f_i.

5 Discussion

These experiments firstly confirm that good performances can be obtained through merely extracting features correlated to the target class (here the harmonics of the sought frequencies). The convolutional structure, aimed to achieve phase-invariance, is most effective for low stride values. For higher stride values, the invariance property does not hold and the accuracy shows a downward peak. An alternative is to consider neurons in the complex space [18].

Overall, the merits of using SP knowledge are twofold: it speeds up the convergence and it ensures that the target concept is properly grasped. On the other hand, supervised learning is beneficial as it automatically detects and exploits the relationships among frequencies and harmonics for a more robust detection.

References

[1] S. J. Godsill et al. "Bayesian computational methods for sparse audio and music processing". In: *2007 15th European Signal Processing Conference*. Sept. 2007, pp. 345–349.

[2] Aäron van den Oord et al. "WaveNet: A Generative Model for Raw Audio". In: *CoRR* abs/1609.03499 (2016). arXiv: 1609.03499. URL: http://arxiv.org/abs/1609.03499.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN: 0262035618, 9780262035613.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105. URL: http://dl.acm.org/citation.cfm?id=2999134.2999257.

[5] *Mirex Challenge*. URL: https://www.music-ir.org/mirex/wiki/MIREX_HOME.

[6] Damien Garreau et al. "Metric Learning for Temporal Sequence Alignment". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 1817–1825. URL: http://papers.nips.cc/paper/5383-metric-learning-for-temporal-sequence-alignment.pdf.

[7] Valentin Emiya, Roland Badeau, and Bertrand David. "Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle". In: *IEEE Trans. Audio, Speech & Language Processing* 18.6 (2010), pp. 1643–1654. DOI: 10.1109/TASL.2009.2038819. URL: https://doi.org/10.1109/TASL.2009.2038819.

[8] *Librosa library*. URL: https://librosa.github.io/librosa/.

[9] Emmanouil Benetos et al. "Automatic music transcription: challenges and future directions". In: *J. Intell. Inf. Syst.* 41.3 (2013), pp. 407–434. DOI: 10.1007/s10844-013-0258-3. URL: https://doi.org/10.1007/s10844-013-0258-3.

[10] George Tzanetakis. "Anssi Klapuri, Manuel Davy, Eds: Signal Processing Methods for Music Transcription". In: *Computer Music Journal* 32.4 (2008), pp. 86–88. DOI: 10.1162/comj.2008.32.4.86. URL: https://doi.org/10.1162/comj.2008.32.4.86.

[11] James A. Moorer. "On the Transcription of Musical Sound by Computer". In: *Computer Music Journal* 1.4 (1977), pp. 32–38. ISSN: 01489267, 15315169. URL: http://www.jstor.org/stable/40731298.

[12] Keith D. Martin. *A Blackboard System for Automatic Transcription of Simple Polyphonic Music*. Tech. rep. 1996.

[13] M. Privosnik and M. Marolt. "A system for automatic transcription of music based on multiple-agents architecture". In: *MELECON '98. 9th Mediterranean Electrotechnical Conference. Proceedings (Cat. No.98CH36056)*. Vol. 1. May 1998, 169–172 vol.1. DOI: 10.1109/MELCON.1998.692363.

[14] R. Keren, Y. Y. Zeevi, and D. Chazan. "Multiresolution time-frequency analysis of polyphonic music". In: *Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis (Cat. No.98TH8380)*. Oct. 1998, pp. 565–568. DOI: 10.1109/TFSA.1998.721487.

[15] Eric J. Humphrey, Juan P. Bello, and Yann Lecun. "Feature Learning and Deep Architectures: New Directions for Music Informatics". In: *J. Intell. Inf. Syst.* 41.3 (Dec. 2013), pp. 461–481. ISSN: 0925-9902. DOI: 10.1007/s10844-013-0248-5. URL: http://dx.doi.org/10.1007/s10844-013-0248-5.

[16] E. J. Humphrey, T. Cho, and J. P. Bello. "Learning a robust Tonnetz-space transform for automatic chord recognition". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Mar. 2012, pp. 453–456. DOI: 10.1109/ICASSP.2012.6287914.

[17] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. "An End-to-End Neural Network for Polyphonic Piano Music Transcription". In: *IEEE/ACM Trans. Audio, Speech & Language Processing* 24.5 (2016), pp. 927–939. DOI: 10.1109/TASLP.2016.2533858. URL: https://doi.org/10.1109/TASLP.2016.2533858.

[18] Chiheb Trabelsi et al. "Deep Complex Networks". In: *CoRR* abs/1705.09792 (2017). arXiv: 1705.09792. URL: http://arxiv.org/abs/1705.09792.