Noname manuscript No.
(will be inserted by the editor)

Feature Learning and Deep Architectures: New Directions for Music Informatics

Eric J. Humphrey · Juan P. Bello · Yann LeCun

Received: date / Accepted: date

Abstract As we look to advance the state of the art in content-based music informatics, there is a general sense that progress is decelerating throughout the field. On closer inspection, performance trajectories across several applications reveal that this is indeed the case, raising some difficult questions for the discipline: why are we slowing down, and what can we do about it? Here, we strive to address both of these concerns. First, we critically review the standard approach to music signal analysis and identify three specific deficiencies to current methods: hand-crafted feature design is sub-optimal and unsustainable, the power of shallow architectures is fundamentally limited, and short-time analysis cannot encode musically meaningful structure. Acknowledging breakthroughs in other perceptual AI domains, we offer that deep learning holds the potential to overcome each of these obstacles. Through conceptual arguments for feature learning and deeper processing architectures, we demonstrate how deep processing models are more powerful extensions of current methods, and why now is the time for this paradigm shift. Finally, we conclude with a discussion of current challenges and the potential impact to further motivate an exploration of this promising research area.

Keywords music informatics · deep learning · signal processing

E. J. Humphrey
35 West 4^{th} St
New York, NY 10003 USA
Tel.: +1212-998-2552
E-mail: ejhumphrey@nyu.edu

1 Introduction

It goes without saying that we live in the Age of Information, our day to day experiences awash in a flood of data. We buy, sell, consume and produce information in unprecedented quantities, with countless applications lying at the intersection of our physical world and the virtual one of computers. As a result, a variety of specialized disciplines have formed under the auspices of Artificial Intelligence (AI) and information processing, with the intention of developing machines to help us navigate and ultimately make sense of this data. Coalescing around the turn of the century, music informatics is one such discipline, drawing from several diverse fields including electrical engineering, music psychology, computer science, machine learning, and music theory, among others. Now encompassing a wide spectrum of application areas and the kinds of data considered—from audio and text to album covers and online social interactions—music informatics can be broadly defined as the study of information related to, or is a result of, musical activity.

From its inception, many fundamental challenges in content-based music informatics, and more specifically those that focus on music audio signals, have received a considerable and sustained research effort from the community. This area of study falls under the umbrella of perceptual AI, operating on the premise that if a human expert can experience some musical event from an audio signal, it should be possible to make a machine respond similarly. As the field continues into its second decade, there are a growing number of resources that comprehensively review the state of the art in these music signal processing systems across a variety of different application areas [31, 11, 44], including melody extraction, chord recognition, beat tracking, tempo estimation, instrument identification, music similarity, genre classification, and mood prediction, to name only a handful of the most prominent topics.

After years of diligent effort however, there are two uncomfortable truths facing content-based MIR. First, progress in many well-worn research areas is decelerating, if not altogether stalled. A review of recent MIREX[1] results provides some quantitative evidence to the fact, as shown in Figure 1. The three most consistently evaluated tasks for more than the past half decade —chord recognition, genre recognition, and mood estimation— are each converging to performance plateaus below satisfactory levels. Fitting an intentionally generous logarithmic model to the progress in chord recognition, for example, estimates that continued performance at this rate would eclipse 90% in a little over a decade, and 95% some twenty years after that; note that even this trajectory is quite unlikely, and for only this one specific problem (and dataset). Attempts to extrapolate similar projections for the other two tasks are even less encouraging. Second, these ceilings are pervasive across many open problems in the discipline. Though single-best accuracy over time is shown for these three specific tasks, other MIREX tasks exhibit similar, albeit more sparsely sampled, trends. Other research has additionally demonstrated that when state-of-the-art algorithms are employed in more realistic situations, i.e. larger datasets, performance degrades substantially [8]. Consequently, these observations have encouraged some to question the state of affairs in content-based MIR: Does content *really* matter, especially when human-

[1] Music Information Retrieval Evaluation eXchange (MIREX): http://www.music-ir.org/mirex/

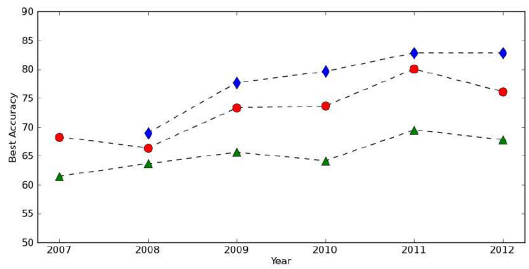

Fig. 1 *Losing steam*: The best performing systems at MIREX since 2007 are plotted as a function of time for Chord Recognition (blue diamonds), Genre Recognition (red circles), and Mood Estimation (green triangles).

provided information about the content has proven to be more fruitful than the content itself [50]? If so, what can we learn by analyzing recent approaches to content-based analysis [20]? Are we considering all possible solutions [29]?

The first question directly challenges the raison d'être of computer audition: is content-based analysis still a worthwhile venture? While applications such as playlist generation [43] or similarity [39] have recently seen better performance by using contextual information and metadata rather than audio alone, this approach cannot reasonably be extended to most content-based applications. It is necessary to note that human-powered systems leverage information that arises as a by-product of individuals listening to and organizing music in their day to day lives. Like the semantic web, music tracks are treated as self-contained documents that are related to each other through common associations. However, humans do not naturally provide the information necessary to solve many worthwhile research challenges simply as a result of *passive* listening. First, listener data are typically stationary over the entire musical document, whereas musical information of interest will often have a temporal dimension not captured at the track-level. Second, many tasks —polyphonic transcription or chord recognition, for example— inherently require an expert level of skill to perform well, precluding most listeners from even intentionally providing this information.

Some may contend that if this information cannot be harvested from crowdsourced music listening, then perhaps it could be achieved by brute force annotation. Recent history has sufficiently demonstrated, however, that such an approach simply cannot scale. As evidence of this limitation, consider the large-scale commercial effort currently undertaken by the Music Genome Project (MGP), whose goal is the widespread manual annotation of popular music by expert listeners. At the time of writing, the MGP is nearing some 1M professionally annotated songs, at an average rate of 20–30 minutes per track. By comparison, iTunes now offers over 28M tracks; importantly, this is only representative of commercial music and audio, and neglects the entirety of amateur content, home recordings, sound effects, samples, and so on, which will only make this task more insurmountable. Given the sheer impossibility for humans to meaningfully describe all recorded music, truly scalable MIR systems will require good computational algorithms.

Therefore, acknowledging that content-based MIR is indeed valuable, we turn our attention to the other two concerns: what can we learn from past experience, and are we fully exploring the space of possible solutions? The rest of this paper is an attempt to answer those questions. Section 2 critically reviews conventional approaches to content-based analysis and identifies three major deficiencies of current systems: the sub-optimality of hand-designing features, the limitations of shallow architectures, and the short temporal scope of signal analysis. In Section 3 we contend that *deep learning* specifically addresses these issues, and thus alleviates some of the existing barriers to advancing the field. We offer conceptual arguments for the advantages of both *learning* and *depth*, formally define these processing structures, and show how they can be seen as generalizations of current methods. Furthermore, we provide specific arguments as to why it is timely for the MIR community to adopt these techniques *now*. To further strengthen the latter point, Section 4 discusses three recent case studies in music informatics that showcase the benefits of deep learning. Finally, in Section 5, we conclude with a survey of challenges and future directions to encourage a more concerted exploration of this promising research topic.

2 Reassessing Common Practice in Content-based MIR

Despite a broad spectrum of application-specific problems, the vast majority of music signal processing systems adopt a common two-stage paradigm of feature extraction and semantic interpretation. Leveraging substantial domain knowledge and a deep understanding of digital signal theory, researchers carefully architect signal processing systems to capture useful signal-level attributes, referred to as *features*. These statistics are then provided to a pattern recognition machine for the purposes of assigning semantic meaning to observations. Crafting good features is a particularly challenging subproblem, and it is becoming standard practice amongst researchers to use precomputed features[2] or off-the-shelf implementations[3], focusing instead on increasingly more powerful pattern recognition machines to improve upon prior work. Therefore, while early research mainly employed simple classification strategies such as nearest-neighbors or peak-picking, recent work makes extensive use of sophisticated and versatile techniques, e.g. Support Vector Machines [41], Bayesian Networks [42], Conditional Random Fields [51], and Variable-Length Markov Models [13].

This trend of squeezing every bit of information from a stock feature representation is arguably suspect because the two-tier perspective hinges on the premise that *features are fundamental*. Data must be summarized in such a way that the degrees of freedom are informative for a particular task; features are said to be *robust* when this is achieved, and *noisy* when variance is misleading or uninformative. The more robust a feature representation is, the simpler a pattern recognition machine needs to be, and vice versa. It can be said that robust features *generalize* by yielding accurate predictions of new data, while noisy features can lead to the opposite behavior, known as *over-fitting* [9]. The substantial emphasis traditionally placed on feature design demonstrates that the community implicitly agrees, but it is a point worth illustrating. Consider the scenario presented in Figure 2. The predominant approach to compute how similar two music signals sound is

[2] Million Song Dataset
[3] MIR Toolbox, Chroma Toolbox, MARSYAS, Echonest API

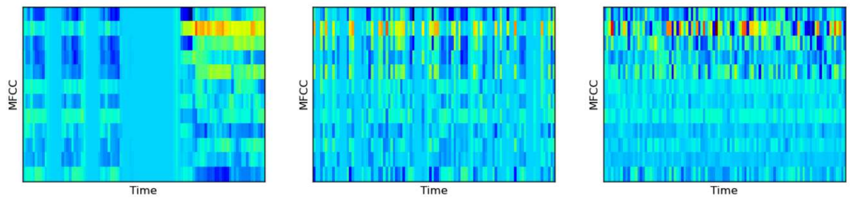

Fig. 2 *What story do your features tell?* Sequences of MFCCs are shown for a real music excerpt (left), a time-shuffled version of the same sequence (middle), and an arbitrarily generated sequence of the same shape (right). All three representations have equal mean and variance along the time axis, and could therefore be modeled by the exact same distribution.

to model their Mel-Frequency Cepstral Coefficients (MFCCs) with a Gaussian Mixture Model (GMM) and compute some distance measure between them, e.g. KL-divergence, Earth mover's distance, etc. [6]. Importantly though, representing these coefficients as a mixture of Gaussians reduces the observation to mean and variance statistics, discarding temporal structure. Therefore, the three MFCC sequences shown —a real excerpt, a shuffled version of it, and a randomly generated one— are identical in the eyes of the model. The audio that actually corresponds to these respective representations, however, will certainly not *sound* similar to a human listener.

This bears a significant consequence: any ambiguity introduced or irrelevant variance left behind in the process of computing features must instead be overcome by the pattern recognition machine. Previous research in chord recognition has explicitly shown that better features allow for simpler classifiers [12], and intuitively many have spent years steadily improving their respective feature extraction implementations [40, 45]. Moreover, there is ample evidence these various classification strategies work quite well on myriad problems and datasets [9]. Therefore, underperforming content-based MIR systems are more likely the result of deficiencies in the feature representation than the classifier used to make sense of it.

It is particularly prudent then, to examine the assumptions and design decisions incorporated into feature extraction systems. In music signal processing, audio feature extraction typically consists of a recombination of a small set of operations, as depicted in Figure 3: splitting the signal into independent short-time segments, referred to as blocks or frames; applying an affine transformation, generally interpreted as either a projection or filterbank; applying a non-linear function; and pooling across frequency or time. Some of these operations can be, and often are, repeated in the process. For example, MFCCs are computed by filtering a signal segment at multiple frequencies on a Mel-scale (affine transform), taking the logarithm (non-linearity), and applying the Discrete Cosine Transform (affine transformation). Similarly, chroma features are produced by applying a constant-Q filterbank (affine transformation), taking the complex modulus of the coefficients (non-linearity), and summing across octaves (pooling).

Considering this formulation, there are three specific reasons why this approach might be problematic. First, though the data-driven training of classifiers and other pattern recognition machines has been standard for over a decade in music informatics, the parametrization of feature extractors —e.g. choice of filters, non-linearities and pooling strategies, and the order in which they are applied—

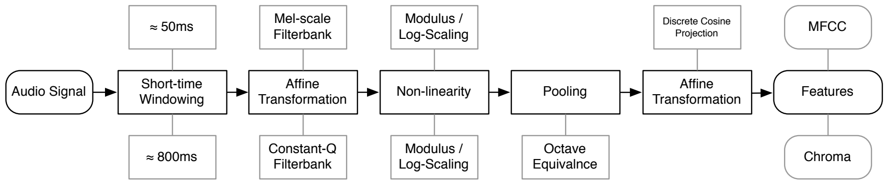

Fig. 3 *State of the art*: Standard approaches to feature extraction proceed as the cascaded combination of a few simpler operations; on closer inspection, the main difference between chroma and MFCCs is the parameters used.

remains, by and large, a manual process. Both feature extraction and classifier training present the same basic problem: there is a large solution space and, somewhere in it, a configuration that optimizes an objective function over a dataset. Though the music informatics community is privileged with a handful of talented researchers who are particularly adept at exploring this daunting space, crafting good features can be a time consuming and non-trivial task. Additionally, carefully tuning features for one specific application offers no guarantees about relevance or versatility in another scenario. As a result, features developed for one task — chroma for chord recognition [21] or MFCCs in speech [16]— are used in others they were not specifically designed for, e.g. structural segmentation [38] or music classification [41]. The caveat of repurposing features designed for other applications is that, despite potentially giving encouraging results, they are not optimized for this new use case. In fact, recent research has demonstrated that better features than MFCCs exist for *speech recognition* [26], the very task they were designed for, so it is almost certain that there are better musical features as well. Therefore, the conclusions to draw from this are twofold: continuing to manually optimize a feature representation is not scalable to every problem, and we may be unnecessarily constraining our search of the solution space.

Second, these information processing architectures can be said to be *shallow*, i.e. incorporating only a few non-linear transformations in their processing chain. Sound, like other real-world phenomena, naturally lives on a highly non-linear manifold within its time-domain representation. Shallow processing structures are placed under a great deal of pressure to accurately characterize the latent complexity of this data. Feature extraction can thusly be conceptualized as a function that maps inputs to outputs with an order determined by its *depth*; for a comprehensive discussion on the merits and mathematics of depth, we refer the curious reader to [3]. Consider the example in Figure 4, where the goal is to compute a low-dimensional feature vector (16 coefficients) that describes the log-magnitude spectrum of a windowed violin signal. One possible solution to this problem is to use a *channel vocoder* which, simply put, low-pass filters and decimates the spectrum, producing a piece-wise linear approximation of the envelope. It is clear, however, that with only a few linear components we cannot accurately model the latent complexity of the data, obtaining instead a coarse approximation. Alternatively, the *cepstrum* method transforms the log-magnitude spectrum before low-pass filtering. In this case, the increase in depth allows the same number of coefficients to more accurately represent the envelope. Obviously, powerful pattern recognition machines can be used in an effort to compensate for the deficiencies of

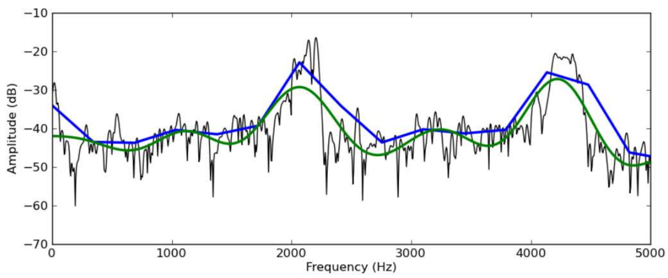

Fig. 4 *Low-order approximations of highly non-linear data*: The log-magnitude spectra of a violin signal (black) is characterized by a channel vocoder (blue) and cepstrum coefficients (green). The latter, being a higher-order function, is able to more accurately describe the contour with the same number of coefficients.

a feature representation. However, shallow, low-order functions are fundamentally limited in the kinds of behavior they can characterize, and this is problematic when the complexity of the data greatly exceeds the complexity of the model.

Third, short-time signal analysis is intuitively problematic because the vast majority of our musical experiences do not live in hundred millisecond intervals, but at least on the order of seconds or minutes. Conventionally, features derived from short-time signals are limited to the information content contained within each segment. As a result, if some musical event does not occur within the span of an observation —a motif that does not fit within a single frame— then it simply cannot be described by that feature vector alone. This is clearly an obstacle to capturing high-level information that unfolds over longer durations, noting that time is extremely, if not fundamentally, important to how music is perceived. Admittedly, it is not immediately obvious how to incorporate longer, or even multiple, time scales into a feature representation, with previous efforts often taking one of a few simple forms. *Shingling* is one such approach, where a consecutive series of features is concatenated into a single, high-dimensional vector [11]. In practice, shingling can be fragile to even slight translations that may arise from tempo or pitch modulations. Alternatively, *bag-of-frames (BoF)* models consider patches of features, fitting the observations to a probability distribution. As addressed earlier with Figure 2, bagging features discards temporal structure, such that any permutation of the feature sequence yields the same distribution. The most straightforward technique is to ignore longer time scales at the feature level altogether, relying on post-filtering *after* classification to produce more musically plausible results. For this to be effective though, the musical object of interest must live at the time-scale of the feature vector or it cannot truly be encoded. Ultimately, none of these approaches are well suited to characterizing structure over musically meaningful time-scales.

2.1 A Concise Summary of Current Obstacles

In an effort to understand why progress in content-based music informatics is plateauing, we have reviewed the standard approach to music signal processing and feature design, deconstructing assumptions and motivations behind various decisions. As a result, three potential areas of improvement are identified. So that each may be addressed in turn, it is useful to succinctly restate the main points of this section:

- **Hand-crafted feature design is neither scalable nor sustainable**: Framing feature design as a search in a solution space, the goal is to discover the configuration that optimizes an objective function. Even conceding that some gifted researchers might be able to achieve this on their own, they are too few and the process too time-consuming to realistically solve every feature design challenge that will arise.

- **Shallow processing architectures struggle to describe the latent complexity of real-world phenomena**: Feature extraction is similar in principle to compactly approximating functions. Real data, however, lives on a highly non-linear manifold and shallow, low-order functions have difficulty describing this information accurately.

- **Short-time analysis cannot naturally capture higher level information**: Despite the importance of long-term structure in music, features are predominantly derived from short-time segments. These statistics cannot capture information beyond the scope of its observation, and common approaches to characterizing longer time scales are ill-suited to music.

3 Deep Learning: A (Slightly) Different Direction

Looking toward how we might begin to address these specific shortcomings in modern music signal processing, there is a renaissance currently underway in computer science. *Deep learning* is riding a wave of promise and excitement in multiple domains, toppling a variety of long-standing benchmarks, but has yet to gain significant traction in music informatics. Here, we present the argument that by reframing music signal analysis as a deep learning problem, it may be possible to overcome every issue named previously. To first motivate this stance, it is particularly useful to break down the ideas behind the very name itself. We then formally define some basic elements of deep signal processing architectures, and illustrate how these methods are simply extensions of current systems. Lastly, we discuss how recent breakthroughs in *unsupervised* learning have greatly reduced the need for large collections of annotated data, thereby making deep learning practical for a wide range of applications.

3.1 Why *Learning*?

In broad terms, an overarching goal of information processing is the design of computational systems that model some observed behavior; for an input, x, find a function, $f(\cdot)$, that produces the desired output, y. Perhaps the simplest example of this notion is that of a line, defined by the equation $y = mx + b$. Collecting the parameters as $\Theta = [m, b]$, the space of all lines can be compactly represented by $y =$

$f_{Line}(x|\Theta)$. We can then borrow the language of objected oriented programming and say that the equation defines the function's *class*, whereas the parameters define its *instance*. Therefore, restating for clarity, a function is a specific instance of a general equation class.

Now, suppose that we wish to model the relationship between a collection of input data and corresponding outputs. Assuming these output values are known, it is possible to objectively measure how well a function approximates this relationship, which we will call its *fitness*. Consider for a moment that there exists a hypothetical space of all possible solutions and, in it, at least one function that optimally satisfies this measure. Thus, the goal of finding such a function can be thought of as a *search*, and, for the sake of discussion, we offer that "learning" and "searching" can be used interchangeably. Unfortunately, the space of all possible functions is effectively infinite and intractable to explore exhaustively, automatically or otherwise. However, conceptualizing a function in terms of classes and instances provides an elegant way of making this search significantly more manageable: a function's class, which can be purposefully designed, greatly reduces the space of all possible instances to a much smaller subspace that *can* be explored.

Given this view of finding objectively good functions, it is interesting to again consider conventional approaches to feature design. As addressed previously, feature extraction and classification are often treated as independent system components. Features are designed by leveraging an explicit idea about what the output should conceptually represent and the engineering acumen necessary to encode it. Therefore, these representations are, by definition, limited to those concepts we can both imagine and implement. Importantly though, the quality of a feature representation cannot be measured directly, but only after "fitting" a pattern recognition machine to it. Designing a function —both class and instance— capable of producing the optimal feature representation is especially difficult because there are two free variables: the equation and its parameters. The common strategy to deal with this problem is to design the feature representation manually, fit a classifier to the data, and repeat until performance improves. Not only is this arduous, but it is potentially unnecessary; there exist a variety of numerical optimization methods and search algorithms to automatically solve such problems. This is hardly a new observation, as demonstrated by the work of Cabral and Pachet [10]. Their proposed EDS system searches the space of feature extraction functions by defining a set of possible operations and using the genetic algorithm (GA) to find good configurations of these parts.

With this in mind, the distinction between function classes and instances is an integral one to make. By heavily restricting the space of possible solutions, automatically searching a subspace becomes feasible. Even more encouraging, adopting an objective fitness measure of a function that is differentiable with respect to its parameters allows for the application of gradient methods to find good instances [36], offering two significant benefits. Practically speaking, gradient methods will find optima of a fitness measure, whereas other search strategies either offer no convergence guarantees or are too computationally intensive. Once numerical optimization is feasible, this frees the researcher to focus less on the specific parameters of a system and more on the abstract design of the underlying model, using domain knowledge to steer it toward certain kinds of solutions. Therefore, learning is advantageous because it can simplify the overall design problem, accelerate research, yield flexible systems that can adapt to new data, and facilitate the discovery of new solutions not previously considered.

3.2 Why *Deep*?

As seen from the earlier discussion of shallow architectures, complex manifolds can be extremely difficult to describe with simple systems, and may require a large number of piece-wise components to closely model this data. Therefore, if we are to accurately model or characterize real-world signals, higher order systems are necessary. In the same way that learning and searching are functionally interchangeable in the previous section, here *depth* and *order* are synonymous. As a complementary attribute, the *breadth* of a function is defined by the number of parallel components in a transformation [5].

There is mathematical rigor behind the trade-off between depth and breadth of a processing structure [3,35], but a classic example adequately illustrates these principles. Early work in computer science focused on using processing structures to learn and implement Boolean logic operations. It was shown early on that the Perceptron, a simple non-linear model, could not achieve certain functions, namely the exclusive-or (XOR). Expressed symbolically, for two inputs p and q, the XOR is defined as the following operation:

$$p \oplus q = (p \wedge \neg q) \vee (\neg p \wedge q) \tag{1}$$

While it is true that one Perceptron cannot capture this behavior, a composite of *three* Perceptrons can. Each parenthetical proposition in Eq. (1) is within the capacity of a single Perceptron, and a third Perceptron is able to achieve the desired logic function by composing the two simpler statements. It is of utmost importance to appreciate that a second-order cascade of simple processing elements is more powerful than the first-order parts alone. This property, where a composed whole is greater than the sum of its parts, is known as *emergence*, and is characteristic of complex systems.

Hierarchical composition —realizing complex structures through the combination of simpler parts— is fundamental to the representational power of deep information processing architectures, but why does this matter for music? The answer, quite literally, is that music is *composed*: pitch and loudness combine over time to form chords, melodies and rhythms, which in turn are built into motives, phrases, sections and, eventually, construct entire pieces. This is the primary reason shallow architectures are ill-suited to represent high-level musical concepts and structure. A melody does not live at the same level of abstraction as a series of notes, but is instead a higher, emergent quality of those simpler musical objects. At its core, music is *structured*, and processing architectures capable of encoding these relationships are necessary to adequately characterize this information.

3.3 Simple Parts, Complex Structures

Before proceeding, we would first like to draw attention a slight matter of formulation. There are, roughly speaking, two equivalent perspectives to deep learning:

discriminative and probabilistic deep networks. For reasons that will become apparent shortly, we frame the following discussion in the context of the former due to strong parallels with digital signal theory; for the probabilistic perspective, we refer to [34].

The one constant spanning all deep networks is the idea that larger processing structures can be composed of smaller, adaptable components. One of the first fundamental building blocks in artificial neural network research is the non-linear affine transformation:

$$Y = h(W \cdot X + b) \tag{2}$$

Here, an input column vector, X^N, is transformed to an output column vector, Y^K, by taking the inner product with a matrix of coefficients $W^{K \times N}$, a vector of biases b^K, and typically a non-linear activation $h(\cdot)$, i.e. the hyperbolic tangent or sigmoid function. Recalling the previous discussion of classes and instances, many familiar transforms can be recovered from this one class of functions. For example, defining $W = \exp(-2\pi jkn/N)$, $b = 0$, and $h(\cdot) = \log_{10}(|\cdot|)$ yields the log-magnitude spectra of the Discrete Fourier Transform (DFT), or, dropping the non-linearity, W can be computed from a Principal Components Analysis (PCA) over a collection of data, to name two specific instances. In this sense, layers in a deep network can be viewed as a generalization of many common transformations used in music signal processing, and, vice versa, common processing structures could be generalized to fit inside the framework of deep networks, e.g. convolution, downsampling, standardization, etc.

Another interesting consequence of the observation that neural networks share a common formulation with the DFT is that they must also share a filterbank interpretation when applied to time domain signals. In this case, the matrix W can be viewed as a set of K, N-length Finite Impulse Response (FIR) filters, expressed by the following difference equation:

$$y_k[n] = h(b_{k,0} + w_{k,0}x[n] + w_{k,1}x[n-1] \ldots + w_{k,N-1}x[n-N+1]) \tag{3}$$

Adopting a digital signal processing perspective of such architectures provides alternate ways of understanding the role these different processing elements play in the network. For example, pooling, or simply the reduction of datapoints, can be seen as different forms of decimation in time or frequency, whereas a non-linearity like full-wave rectification (complex modulus) has a demodulation effect. Therefore, much of the intuition that goes into filter design and digital signal processing is also relevant to the design of deep networks. Interestingly enough, while conventional filter design typically requires filterbanks to be linear to make the analysis tractable, we have seen how the representational power of linear systems can be quite limited. Reformulating digital filters as deep networks unlocks the potential to build higher order, non-linear functions with adaptable, data-driven parameterizations.

3.4 The Natural Evolution of Deep Architectures

In the previous two sections, we have made the case that deeper processing structures are better suited to characterize complex data, and drawn attention to the

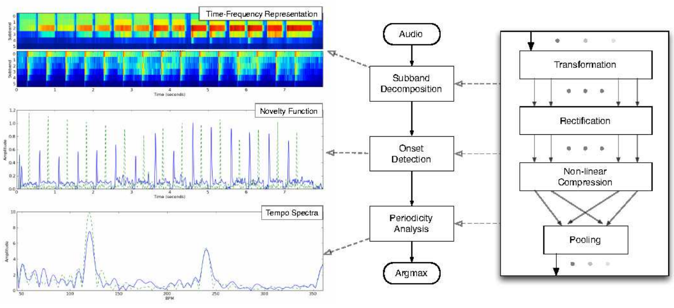

Fig. 5 *A complex system of simple parts*: Tempo estimation has, over time, naturally converged to a deep architecture. Note how each processing layer absorbs a different type of variance —pitch, absolute amplitude, and phase— to transform two different signals into nearly identical representations.

realization that the building blocks in deep learning share a common formulation with standard operations in digital signal theory. It should come as little surprise then that there are instances where deep signal processing structures have developed in the due course of research, and there are two worth illustrating here.

For some time, state of the art tempo estimation algorithms have been based on deep, non-linear processing architectures. The high-level intuition behind system design is relatively straightforward and, as evidenced by various approaches, widely agreed upon; first identify the occurrence of musical events, or onsets, and then estimate the underlying periodicity. The earliest efforts in tempo analysis tracked symbolic events [15], but it was soon shown that a time-frequency representation of sound was useful in encoding rhythmic information [47]. This led to in-depth studies of onset detection [2], based on the idea that "good" impulse-like signals, referred to as *novelty functions*, would greatly simplify periodicity analysis. Along the way, it was also discovered that applying non-linear compression to a novelty function produced noticeably better results [31]. Various periodicity tracking methods were simultaneously explored, including oscillators [19], multiple agents [22], inter-onset interval histograms [18], and tuned filterbanks [23].

Reflecting on this lineage, system design has, over the last two decades, effectively converged to a deep learning architecture, minus the learning, where the same processing elements —filtering and transforms, non-linearities, and pooling— are replicated over multiple processing layers. Interestingly, as shown in Figure 5, visual inspection demonstrates why it is particularly well suited to the task of tempo estimation. Here we consider two input waveforms having nothing in common but tempo; one is an ascending D Major scale played on a trumpet, and the other is a delayed series of bass drum hits. It can be seen that, at each layer, a different kind of variance in the signal is removed. The filterbank front-end absorbs rapid fluctuations in the time-domain signal, spectrally separating acoustic events. This facilitates onset detection, which provides a pitch and timbre invariant estimate of events in the signal, reducing information along the frequency dimension. Lastly, periodicity analysis eliminates shifts in the pulse train by discarding phase information. At the output of the system, these two acoustically different inputs have been transformed into nearly identical representations. The main takeaway, therefore, is that deep architectures are able to absorb variance in the data over multiple layers, turning one complex problem into a series of simpler ones.

More recently, multi-level wavelet filterbanks, referred to as scattering transforms, have shown promise for audio classification by capturing information over not only longer, but also multiple, time-scales [1]. Recognizing MFCCs as a first-order statistic, this second-order system yielded better classification results over the same observation length while also achieving convincing reconstruction of the original signals. The authors demonstrate their approach to be a multi-layer generalization of MFCCs, and exhibit strong parallels to certain deep network architectures, although the parameterization here is not learned but defined. Perhaps a more intriguing observation to draw from this work though is the influence a fresh perspective can have on designing deep architectures. Rather than simply propagating all information upwards through the structure, as is common in deep learning, the system keeps summary statistics at each timescale, leading to better performance.

3.5 Why now?

The major obstacle in deep learning has always been the issue of parameter optimization. Historically, deep networks relied almost entirely on *supervised* training, i.e. using a large amount of ground-truth data to explicitly direct a system to produce the correct outputs, and were infamously slow to converge. Large annotated datasets are quite difficult to curate, and as a result deep networks were prone to poor local minima and overfitting from insufficient training data.

In 2006, Hinton et al. showed that unsupervised training could leverage unlabeled data by encouraging an adaptable network to *generate* realistic data [27]. Building a deep network proceeds by adding a layer to the network, greedily training it to convergence, and then freezing those parameters; this repeats until reaching the desired architectural depth. Then, to address task-specific questions, the parameters of the network are "fine-tuned" in a supervised manner. In practice, unsupervised "pre-training" acts as a strong, data-driven prior that initializes parameters closer to a final solution, avoiding poor local minima and requiring less labeled data for training [4].

The core concept behind this approach is that of reconstructing previously observed data; if a model can project information into an alternate representation and back again with minimal error, salient characteristics must be encoded in that intermediary representation. Stated in a more intuitive manner, if you can take something apart and successfully put it back together, you are inherently discovering its most important components. Defining explicitly, the goal is to learn both a forward and inverse transform, f and f^{-1}, that minimizes the error between the input X_{in} and its reconstituted version X_{rec}:

$$\begin{aligned} Y &= f(X_{in} \mid \Theta_f) \\ X_{rec} &= f^{-1}(Y \mid \Theta_i) \end{aligned} \tag{4}$$

Recalling the discussion of Section 3.3, note that this is the same formulation of the forward and inverse DFT, which happens to achieve *perfect* reconstruction by way of two complementary affine transformations. Importantly though, the DFT is a versatile, but ultimately general, transform that projects a signal onto a

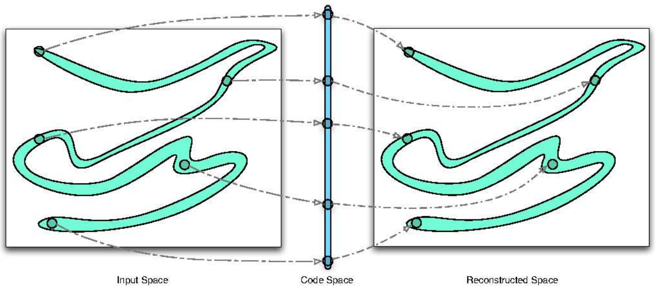

Fig. 6 *Learning where the data lives*: A hypothetical, non-linear manifold is contained within an over-complete input space. Though the data (green) are distributed irregularly, there exists some projection that can transform this data to an invertible representation without a loss of information.

pre-defined set of complex sinusoids, which may have little to do with the actual data being transformed. Learning invertible transforms is an effective unsupervised objective because it adapts a decomposition to the data being analyzed, thus discovering the lower dimensional manifold on which the information lives. As shown by the cartoon illustration given in Figure 6, sound can be thought of as being unevenly concentrated in dense regions of the space it occupies, i.e. a discrete digital representation. A simple thought experiment motivates this idea. If sound were uniformly distributed within this representation, the overwhelming majority of our acoustic experience would be white noise. We know from experience though that such signals are thankfully quite rare, and instead many naturally occurring sounds are even harmonic. Therefore sound must not be distributed uniformly, and that the space of *possible* sounds is over-complete compared to the range *actual* sounds that naturally occur. Unsupervised learning works by taking advantage of this very strong prior on the data, in that we can assume all sound is distributed similarly within its representation.

Complementary to the principles of unsupervised learning, there are also situational factors that strengthen the timeliness of these methods. First and foremost, there is an overwhelming amount of *unlabeled* data available for unsupervised methods to absorb, allowing labeled datasets to be reserved for fine-tuning and evaluation. Additionally, significant increases is computational power, and especially advances in parallel computing via GPUs, make many of these approaches not only feasible, but in some cases quite efficient. As evidenced by the recent work of [32], we are even starting to see attempts at large scale deep learning. In a similar vein, software libraries and toolkits [7, 14] are now available that leverage these computational gains to make deep learning more accessible to the entire research community.

4 Case Studies in Music Informatics

Though the MIR community has been somewhat hesitant to adopt feature learning and deeper architectures, there is an increasing effort to explore these research avenues. One of the earliest instances is that of sparse audio coding, which is relevant to the discussion as a data-driven approach to signal analysis. Sparse

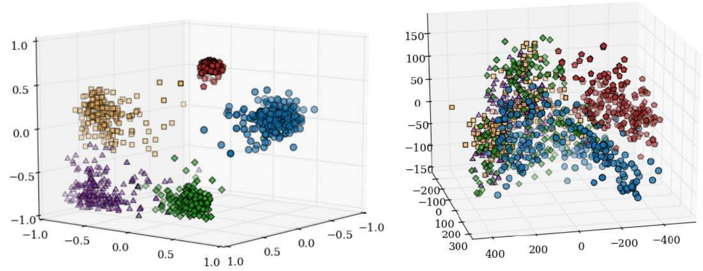

Fig. 7 *Learning a well-organized representation*: The representations of various instrument sounds are shown via NLSE (left) and PCA of MFCCs (right) for five instrument classes: Tuba (red pentagons), Oboe (green diamonds), Clarinet (purple triangles), Cello (blue circles) and Flute (yellow squares).

coding has been shown to yield useful representations for a variety of applications, such as instrument recognition [37] and polyphonic transcription [46], but is limited in practice by a slow inference process. Also, as mentioned earlier, previous work has leveraged search algorithms to optimize feature transformations for various tasks, such as music description or chord recognition [52, 10]. Deeper processing structures are also gaining popularity in music informatics, ranging from work on frame-wise representations, like in genre classification [25] or mood estimation [48], to slight reformulations of classic problems, as in artist identification [17] or chord recognition [28]. However, from this modest corpus of deep learning in music informatics, there are three notable instances worth highlighting here that aptly demonstrate the potential these methods hold to overcome the shortcomings outlined in Section 2.1.

4.1 Better Features, Simpler Classifiers

One of the recurring themes in this discussion is the emphasis placed on the organization and robustness of a feature space, or conversely the degree of variance absorbed in the course of feature extraction. While it is intuitively appealing that better features relax the stress placed on a classifier, this has been demonstrated explicitly in the context of instrument classification via deep learning [30]. The goal of this work is to project monophonic instrument sounds into an intuitive, i.e. low-dimensional and metric, representation. Extending the previous approach presented in [24], this is achieved by learning a Non-Linear Semantic Embedding (NLSE) via a convolutional neural network (CNN). The network is trained by making a copy of itself and defining the loss function as the Euclidean distance between the outputs of the two identical networks. This way, a feature space is learned where the distance between similar inputs is small and dissimilar inputs is large. Rather than projecting data to absolute positions in the output space, pairwise training allows the machine to discover an embedding that optimally preserves the relative relationship between two inputs.

For an input, the network operates on patches of constant-Q pitch spectra, such that translations of the kernels in the CNN are linear with respect to both pitch and time. As a baseline comparison, a PCA of MFCCs is also considered, keeping

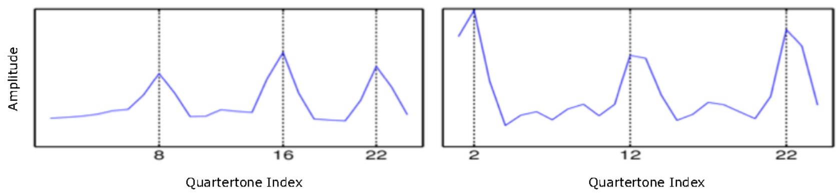

Fig. 8 *Finding meaning from features*: Automatically learning what attributes are informative for a given task may lead to unexpected insights, such as pitch intervals —quartal chords (left) and power chords (right)— informing genre classification, a task classically dominated by timbre features. (Reprinted with permission)

only the first 3 dimensions. To demonstrate the level of semantic organization in the feature space, a naïve k-nearest neighbor is used to classify points as different instruments. For the 5-class embeddings shown in Figure 7, overall classification accuracies for NLSE and PCA are 98% and 63%, and for ranked retrieval the mean-average precision (MAP) scores are 0.734 and 0.212, respectively. Whether or not a more powerful classifier could improve accuracy over the PCA features is irrelevant; the key takeaway from this example is that simple methods are sufficient when the feature space is well organized. Furthermore, the inherent structure of well organized representations makes it particularly attractive as a user interface for navigating and exploring large sound libraries.

4.2 Learning from Feature Learning

Another critical point addressed throughout this article is that feature learning creates the opportunity to discover useful attributes that might not be considered in the process of manual feature design. While this might be an easy argument to make for particularly abstract tasks —after all, what *are* the limits of introspection when determining why two songs are similar?— some research in feature learning has shown it is possible to discover unexpected attributes that are useful to well-worn problems. One such instance is the work by Henaff et al., who applied Predictive Sparse Decomposition (PSD) to the task of genre recognition, achieving state of the art results. As mentioned previously, sparse coding can be useful in classification tasks, but is generally impractical in most applications due to the slow inference process of computing optimal sparse codes. Using this observation as a starting point, PSD trains a logistic regressor to approximate, or *predict*, an input's sparse code, such that inference can be achieved by a single matrix multiplication. As opposed to the distributed coordinate space shown in the previous example, the forward projection learned via PSD is over-complete because sparse representations are more likely to be linearly separable in high dimensions, i.e. informative attributes are separated along the different axes of the space.

Using a linear SVM, the proposed system performs competitively with other state of the art methods (84.3%), while offering the benefit of fast inference. An unexpected consequence of this work is the realization that certain feature detectors, learned from constant-Q representations, seem to encode distinct pitch intervals. As illustrated in Figure 8, these components are indicative of different

Fig. 9 *Same notes, different chord?*: Here, the same collection of simultaneous notes —a G power chord— in bars 2 and 4 are heard as Major and minor, respectively, because of prior context —the B and B flat in bars 1 and 3.

uses of harmony. For example, power chords are common in rock and popular music, but unlikely in jazz. In the opposite scenario, quartal chords are common in jazz, but less so in rock and pop. This observation is especially profound, because the long-standing assumption in feature design for genre recognition is that spectral contour matters much more than harmonic information, as evidenced by the volume of publications that exclusively make use of MFCCs. Given the current state of affairs in music informatics, it is hardly a stretch of the imagination to assume that similar discoveries might occur in other classic problems, or inform new tasks that lack a good intuition toward feature design.

4.3 Encoding Longer Musical Structure

In the classic view of automatic chord recognition, a common definition of a *chord* is the "simultaneous sounding of two or more notes" [42]. As a result, most work operates on the assumption that frame-level observations of a music signal are sufficient to assign observations to particular chord classes. It was shown early on that this approach could be significantly improved upon by adding a musically motivated sequence model after the pattern recognition stage to smooth classification results [49]. This is necessary because, even in modern popular Western music, the explicit use of chords is rarely straightforward. In practice, real music signals often comprise complex tonal scenes that often only *imply* a particular chord, typically in the presence of salient vocal melodies that artfully utilize non-chord tones. Furthermore, certain chords can only be distinguished as major or minor, the typical problem formulation, by considering prior context. A characteristic example of this is given in Figure 9. Here, four monophonic notes proceed a G power chord in two scenarios; in the first bar, the second quarter note is a B natural, or the 3rd scale degree in G Major, whereas in the third bar, the second quarter note is a B flat, or the 3rd scale degree in G minor. Therefore, even though the chords in bars two and four are composed of the exact same notes, they are heard as having major and minor qualities, respectively, due to prior harmonic context.

In an effort to account for this behavior, the work presented in [28] adopts a slightly different view of chord recognition. Using a CNN to classify five-second tiles of constant-Q pitch spectra, an end-to-end chord recognition system is produced that considers context from input observation to output label. Figure 10 illustrates how receptive fields, or local feature extractors, of a convolutional network build abstract representations as the hierarchical composition of parts over time. Lower level behaviors are encoded in mid-level feature maps, which can be again combined with other mid-level features and encoded at the next level, and so on up the hierarchy. At the highest level, the abstract representation can be

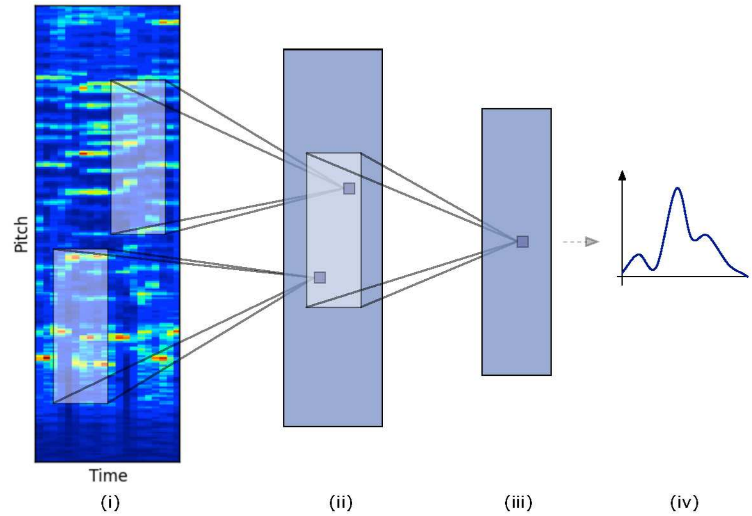

Fig. 10 *A hierarchy of harmony*: Operating on five-second CQT patches as an input (i), the receptive fields of a CNN encode local behavior in feature maps (ii) at higher levels. This process can then be repeated (iii), allowing the network to characterize high-level attributes as the combination of simpler parts. This abstract representation can then be transformed into a probability surface (iv) for classifying the input.

transformed into a probability density function (PDF) and the winning chord class taken as the $argmax()$ of the PDF. As an initial inquiry in CNN-based chord recognition, the system achieves results competitive with the state of the art (77.48%) on a large, publicly available corpus of chord annotations. This is particularly encouraging for a few reasons. First and foremost, building context into the feature representation greatly reduces the need for post-filtering after classification. Therefore, this accuracy is achieved by a causal chord recognition system, a potentially desirable property precluded by the application of the general Viterbi algorithm. Additionally, this is merely the first attempt in an otherwise unexplored research area, and there is ample room for improvement over these results. Variations in training strategy, the underlying processing model, choice of input representation or breadth of chord vocabulary open the door for several promising "next steps" in chord recognition.

5 The Future of Deep Learning in MIR

In this article, we have sought to better understand the current state of affairs in content-based music informatics and diagnose potential deficiencies in state of the art approaches, finding three specific shortcomings: hand-crafted feature design is not sustainable, shallow architectures are fundamentally limited, and short-time analysis alone fails to capture long-term musical structure. Several arguments then motivate the position that deep learning is particularly well-suited to address each of these difficulties. By embracing feature learning, it is possible to optimize a system's internal feature representation, perhaps even discovering it outright, while deep architectures are especially well-suited to characterize the hierarchical nature of music. It was further shown that the community is already beginning to naturally adopt parts of the broader deep learning landscape, and that these methods can be seen as the next logical step in the research trajectory. As we look toward exploring these methods further in the context of MIR, it is beneficial to outline domain-specific challenges and the impact such a conceptual reformulation might have on the discipline.

5.1 Outstanding Challenges

Reflecting on the entire discourse, there are a few legitimate obstacles to this line of research. The most immediate hurdle facing the adaptation of deep learning to music signal processing is merely a matter of literacy and the successful application of these methods to classic problems in MIR. There is an empirical sense of skepticism regarding neural networks among many in the various perceptual AI communities, including MIR, due in no small part to the exaggerated promises of very early research, and popular opinion has not evolved with the science. One step toward updating this perspective is through discussions like this one, by demystifying the proverbial "black box" and understanding what, how, and why these methods work. Additionally, reframing traditional problems in the viewpoint of deep learning serves as an established starting point to begin developing a good comprehension of implementing and realizing these systems. In a similar vein, the MIR community also possesses a mastery of digital signal theory and processing techniques, insight that could, and should, be applied to deep networks to better formulate novel or alternative theoretical foundations.

Another, more widely known problem is the practical difficulty behind getting such methods to "work," which takes a few different forms. Though the features, and more specifically the parameters, learned by the model are data-driven, the successful application of deep learning necessitates a thorough understanding of these methods and how to apply them to the problem at hand. Various design decisions, such as model selection, data pre-processing, and carefully choosing the building blocks of the system, can impact performance on a continuum from negligible differences in overall results to whether or not training can, or will, converge to anything useful. Likewise, the same kind of intuition holds for adjusting the various hyperparameters —learning rate, regularizers, sparsity penalties— that may arise in the course of training. The important thing to recognize though is that these are skills to be learned. Using deep learning presents a design problem not altogether different from the one with which we are familiar, but the approach is overtly more abstract and conceptual, placing a greater emphasis on high-level decisions like the choice of network topology or appropriate loss function.

That said, one of the more enticing challenges facing music informatics is that time-frequency representations, though two-dimensional, are fundamentally *not* images. When considering the application of deep learning to MIR problems, it is prudent to recognize that the majority of progress has occurred in computer vision. While this gives our community an excellent starting point, there are many assumptions inherent to image processing that start to break down when working with audio signals. One such instance is the use of local receptive fields in deep learning, common in CNNs and, more recently, tiled networks [33]. In these architectures, it is known that the strongest correlations in an image occur within local neighborhoods, and this knowledge is reflected in the architectural design. Local neighborhoods in frequency do not share the same relationship, so the natural question becomes, "what architectures *do* make sense for time-frequency representations?" As we saw previously, CNNs yield encouraging results on time-frequency representations of audio, but there are certainly better models to be discovered. This is but one open question facing deep learning in music signal processing, and a concerted research effort will likely reveal more.

5.2 Potential Impact

In addition to hopefully advancing the discipline beyond current glass ceilings, there are several potential benefits to the adoption and research of deep learning in music informatics. Though learning can discover useful features that were previously overlooked or not considered, this advantage is amplified for new challenges and applications that do not offer much guiding intuition. For tasks like artist identification or automatic mixing, it is difficult to comprehend, much less articulate, exactly what signal attributes are informative to the task and how an implementation might robustly capture this information. These problems can, however, be quantified by an objective function —these songs are by the same artist, or this is a better mix than that one— which allows for an automated exploration of the solution space. In turn, such approaches may subsequently provide insight into the latent features that inform musical judgements, or even lead to deployable systems that could adapt to the nuances of an individual.

Deep learning also offers practical advantages toward accelerating research. Rather than trying to compare the instances from one class of functions, evaluation can take place at the class level. This process has the potential to be significantly faster than current research approaches because numerical methods attempt to automatically optimize the same objective function we do by hand. Additionally, unsupervised learning is able to make use of all recorded sound, and the data-driven prior that it leverages can be steered by creating specific distributions, e.g., learn separate priors for rock versus jazz. Finally, music signals provide an interesting setting in which to further explore the role of *time* in perceptual AI systems, and has the potential to influence other time-series domains like video or motion capture data.

References

1. Andén, J., Mallat, S.: Multiscale scattering for audio classification. In: Proc. 12th Int. Conf. on Music Information Retrieval (ISMIR) (2011)
2. Bello, J.P., Daudet, L., Abdallah, S., Duxbury, C., Davies, M., Sandler, M.: A tutorial on onset detection in music signals. IEEE Trans. Audio, Speech and Language Processing **13**(5), 1035–1047 (2005)
3. Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning **2**(1), 1–127 (2009)
4. Bengio, Y., Courville, A.C., Vincent, P.: Unsupervised feature learning and deep learning: A review and new perspectives. arXiv/1206.5538 (2012)
5. Bengio, Y., LeCun, Y.: Scaling learning algorithms towards AI. Large-Scale Kernel Machines **34** (2007)

6. Berenzweig, A., Logan, B., Ellis, D.P., Whitman, B.: A large-scale evaluation of acoustic and subjective music-similarity measures. Computer Music Journal **28**(2), 63–76 (2004)
7. Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., Bengio, Y.: Theano: a CPU and GPU math expression compiler. In: Proc. of the Python for Scientific Computing Conf. (SciPy) (2010)
8. Bertin-Mahieux, T., Ellis, D.P.W.: Large-scale cover song recognition using the 2D Fourier transform magnitude. In: Proc. 13th Int. Conf. on Music Information Retrieval (ISMIR), pp. 241–246 (2012)
9. Bishop, C.: Pattern recognition and machine learning. Springer (2006)
10. Cabral, G., Pachet, F.: Recognizing chords with EDS: Part One. Computer Music Modeling and Retrieval pp. 185 – 195 (2006)
11. Casey, M., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., Slaney, M.: Content-based music information retrieval: Current directions and future challenges. Proc. IEEE **96**(4), 668–696 (2008)
12. Cho, T., Bello, J.P.: A feature smoothing method for chord recognition using recurrence plots. In: Proc. 12th Int. Conf. on Music Information Retrieval (ISMIR) (2011)
13. Chordia, P., Sastry, A., Sentürk, S.: Predictive tabla modelling using variable-length markov and hidden markov models. Journal of New Music Research **40**(2), 105–118 (2011)
14. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: A matlab-like environment for machine learning. In: BigLearn, NIPS Workshop (2011)
15. Dannenberg, R.: An on-line algorithm for real-time accompaniment. In: Proc. Int. Computer Music Conf., pp. 193–198 (1984)
16. Davis, S., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. Acoustics, Speech and Signal Processing **28**(4), 357–366 (1980)
17. Dieleman, S., Brakel, P., Schrauwen, B.: Audio-based music classification with a pretrained convolutional network. In: Proc. 12th Int. Conf. on Music Information Retrieval (ISMIR) (2011)
18. Dixon, S.: Evaluation of the audio beat tracking system Beatroot. Journal of New Music Research **36**(1), 39–50 (2007)
19. Edward, W., Kolen, J.F.: Resonance and the perception of musical meter. Connection science **6**(2-3), 177–208 (1994)
20. Flexer, A., Schnitzer, D., Schlueter, J.: A MIREX meta-analysis of hubness in audio music similarity. In: Proc. 13th Int. Conf. on Music Information Retrieval (ISMIR), pp. 175–180 (2012)
21. Fujishima, T.: Realtime chord recognition of musical sound: a system using common lisp music. In: Proc. Int. Computer Music Conf. (1999)
22. Goto, M., Muraoka, Y.: A real-time beat tracking system for audio signals. In: Proc. Int. Computer Music Conf., pp. 171–174 (1995)
23. Grosche, P., Müller, M.: Extracting predominant local pulse information from music recordings. IEEE Trans. Audio, Speech and Language Processing **19**(6), 1688–1701 (2011)
24. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proc. Computer Vision and Pattern Recognition Conf. (CVPR). IEEE Press (2006)
25. Hamel, P., Wood, S., Eck, D.: Automatic identification of instrument classes in polyphonic and poly-instrument audio. In: Proc. 10th Int. Conf. on Music Information Retrieval (ISMIR) (2009)
26. Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., Kingsbury, B.: Deep neural networks for acoustic modeling in speech recognition. IEEE Signal Processing Magazine (2012)
27. Hinton, G.E., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7), 1527–1554 (2006)
28. Humphrey, E.J., Bello, J.P.: Rethinking automatic chord recognition with convolutional neural networks. In: Proc. Int. Conf. on Machine Learning and Applications (2012)
29. Humphrey, E.J., Bello, J.P., LeCun, Y.: Moving Beyond Feature Design: Deep Architectures and Automatic Feature Learning in Music Informatics. In: Proc. 13th Int. Conf. on Music Information Retrieval (ISMIR) (2012)
30. Humphrey, E.J., Glennon, A.P., Bello, J.P.: Non-linear semantic embedding for organizing large instrument sample libraries. In: Proc. ICMLA (2010)
31. Klapuri, A., Davy, M.: Signal Processing Methods for Music Transcription. Springer (2006)

32. Le, Q., Monga, R., Devin, M., Corrado, G., Chen, K., Ranzato, M., Dean, J., Ng, A.: Building high-level features using large scale unsupervised learning. Proc. Int. Conf. on Machine Learning (ICML) (2012)
33. Le, Q.V., Ngiam, J., Chen, Z., Chia, D., Koh, P.W., Ng, A.Y.: Tiled convolutional neural networks. Advances in Neural Information Processing Systems **23** (2010)
34. Le Roux, N., Bengio, Y.: Representational power of restricted boltzmann machines and deep belief networks. Neural Computation **20**(6), 1631–1649 (2008)
35. LeCun, Y.: Learning invariant feature hierarchies. In: Computer Vision–ECCV 2012. Workshops and Demonstrations, pp. 496–505. Springer (2012)
36. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predicting Structured Data (2006)
37. Leveau, P., Sodoyer, D., Daudet, L.: Automatic instrument recognition in a polyphonic mixture using sparse representations. In: Proc. 8th Int. Conf. on Music Information Retrieval (ISMIR), (2007)
38. Levy, M., Noland, K., Sandler, M.: A comparison of timbral and harmonic music segmentation algorithms. In: Acoustics, Speech and Signal Processing (ICASSP), 2007 IEEE Int. Conf. on, vol. 4, pp. IV–1433. IEEE (2007)
39. Levy, M., Sandler, M.: Music information retrieval using social tags and audio. Multimedia, IEEE Transactions on **11**(3), 383–395 (2009)
40. Lyon, R., Rehn, M., Bengio, S., Walters, T., Chechik, G.: Sound retrieval and ranking using sparse auditory representations. Neural computation **22**(9), 2390–2416 (2010)
41. Mandel, M., Ellis, D.: Song-level features and support vector machines for music classification. In: Proc. 6th Int. Conf. on Music Information Retrieval (ISMIR) (2005)
42. Mauch, M., Dixon, S.: Simultaneous Estimation of Chords and Musical Context From Audio. IEEE Trans. Audio, Speech and Language Processing **18**(6), 1280–1289 (2010)
43. McFee, B., Lanckriet, G.: Hypergraph models of playlist dialects. In: Proc. 13th Int. Conf. on Music Information Retrieval (ISMIR) (2012)
44. Müller, M., Ellis, D., Klapuri, A., Richard, G.: Signal processing for music analysis. Jnl. Selected Topics in Signal Proc. **5**(6), 1088–1110 (2011)
45. Müller, M., Ewert, S.: Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In: Proc. 12th Int. Conf. on Music Information Retrieval (ISMIR). Miami, USA (2011)
46. Nam, J., Ngiam, J., Lee, H., Slaney, M.: A classification-based polyphonic piano transcription approach using learned feature representations. In: Proc.12th Int. Conf. on Music Information Retrieval (ISMIR) (2011)
47. Scheirer, E.D.: Tempo and beat analysis of acoustic musical signals. Jnl. Acoustical Soc. of America **103**(1), 588–601 (1998)
48. Schmidt, E.M., Kim, Y.E.: Modeling the acoustic structure of musical emotion with deep belief networks. In: Proc. Neural Information Processing Systems (2011)
49. Sheh, A., Ellis, D.P.W.: Chord segmentation and recognition using em-trained hidden markov models. In: Proc. 4th Int. Conf. on Music Information Retrieval (ISMIR) (2003)
50. Slaney, M.: Web-scale multimedia analysis: does content matter? Multimedia, IEEE **18**(2), 12–15 (2011)
51. Sumi, K., Arai, M., Fujishima, T., Hashimoto, S.: A music retrieval system using chroma and pitch features based on conditional random fields. In: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE Int. Conf. on, pp. 1997–2000. IEEE (2012)
52. Zils, A., Pachet, F.: Automatic extraction of music descriptors from acoustic signals using EDS. In: Proc. AES (2004)