Dong Yu and Li Deng

Deep Learning and Its Applications to Signal and Information Processing

Today, signal processing research has a significantly widened its scope compared with just a few years ago [4], and machine learning has been an important technical area of the signal processing society. Since 2006, deep learning—a new area of machine learning research—has emerged [7], impacting a wide range of signal and information processing work within the traditional and the new, widened scopes. Various workshops, such as the 2009 ICML Workshop on Learning Feature Hierarchies; the 2008 NIPS Deep Learning Workshop: Foundations and Future Directions; and the 2009 NIPS Workshop on Deep Learning for Speech Recognition and Related Applications as well as an upcoming special issue on deep learning for speech and language processing in *IEEE Transactions on Audio, Speech, and Language Processing* (2010) have been devoted exclusively to deep learning and its applications to classical signal processing areas. We have also seen the government sponsor research on deep learning (e.g., the DARPA deep learning program, available at http://www.darpa.mil/ipto/solicit/baa/BAA-09-40_PIP.pdf).

The purpose of this article is to introduce the readers to the emerging technologies enabled by deep learning and to review the research work conducted in this area that is of direct relevance to signal processing. We also point out, in our view, the future research directions that may attract interests of and require efforts from more signal processing researchers and practitioners in this emerging area for advancing signal and information processing technology and applications.

Digital Object Identifier 10.1109/MSP.2010.939038
Date of publication: 17 December 2010

INTRODUCTION TO DEEP LEARNING

Many traditional machine learning and signal processing techniques exploit shallow architectures, which contain a single layer of nonlinear feature transformation. Examples of shallow architectures are conventional hidden Markov models (HMMs), linear or nonlinear dynamical systems, conditional random fields (CRFs), maximum entropy (MaxEnt) models, support vector machines (SVMs), kernel regression, and multilayer perceptron (MLP) with a single hidden layer. A property common to these shallow learning models is the simple architecture that consists of only one layer responsible for transforming the raw input signals or features into a problem-specific feature space, which may be unobservable. Take the example of a support vector machine. It is a shallow linear separation model with one feature transformation layer when kernel trick is used, and with zero feature transformation layer when kernel trick is not used.

Human information processing mechanisms (e.g., vision and speech), however, suggest the need of deep architectures for extracting complex structure and building internal representation from rich sensory inputs (e.g., natural image and its motion, speech, and music). For example, human speech production and perception systems are both equipped with clearly layered hierarchical structures in transforming information from the waveform level to the linguistic level and vice versa. It is natural to believe that the state of the art can be advanced in processing these types of media signals if efficient and effective deep learning algorithms are developed.

Signal processing systems with deep architectures are composed of many layers of nonlinear processing stages, where each lower layer's outputs are fed to its immediate higher layer as the input. The successful deep learning techniques developed so far share two additional key properties: the generative nature of the model, which typically requires an additional top layer to perform the discriminative task, and an unsupervised pretraining step that makes effective use of large amounts of unlabeled training data for extracting structures and regularities in the input features.

A BRIEF HISTORY

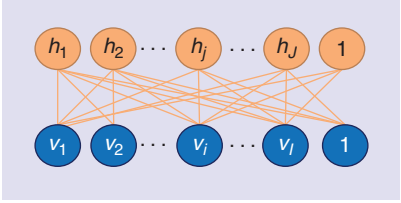
The concept of deep learning originated from artificial neural network research. Multilayer perceptron with many hidden layers is a good example of the models with deep architectures. Backpropagation, invented in 1980s, has been a well-known algorithm for learning the weights of these networks. Unfortunately backpropagation alone does not work well in practice for learning networks with more than a small number of hidden layers (see a review and interesting analysis in [1]). The pervasive presence of local optima in the nonconvex objective function of the deep networks is the main source of difficulty in learning. Backpropagation is based on local gradient descent and starts usually at some random initial points. It often gets trapped in poor local optima and the severity increases significantly as the depth of the networks increases. This difficulty is partially responsible for steering away most of the machine learning and signal processing research from neural networks to shallow models that have convex loss functions (e.g., SVMs, CRFs, and MaxEnt models) for which global optimum can be efficiently obtained at the cost of less powerful models.

The optimization difficulty associated with the deep models was empirically

alleviated when a reasonably efficient, unsupervised learning algorithm was introduced in 2006 by Hinton et al. [7] for a class of deep generative models that they called deep belief networks (DBNs). A core component of the DBN is a greedy, layer-by-layer learning algorithm that optimizes DBN weights at time complexity linear to the size and depth of the networks. Separately and with some surprise, initializing the weights of an MLP with a correspondingly configured DBN often produces much better results than that with the random weights [1], [5]. As such, deep networks that are learned with unsupervised DBN pretraining followed by the backpropagation fine-tuning are also called DBNs in the literature (e.g., [8] and [9]).

A DBN comes with additional attractive properties: 1) The learning algorithm makes effective use of unlabeled data; 2) It can be interpreted as Bayesian probabilistic generative models; 3) The values of the hidden variables in the deepest layer are efficient to compute; and 4) The overfitting problem that is often observed in the models with millions of parameters such as DBNs, and the underfitting problem that occurs often in deep networks are effectively addressed by the generative pretraining step.

The DBN training procedure is not the only one that makes deep learning possible. Since the publication of the seminal work of [7], numerous researchers have been improving and applying the deep learning techniques with success. Another popular technique is to pretrain the deep networks layer by layer by considering each pair of layers as a denoising auto-encoder [1]. We will provide a brief overview of the original DBN work and the subsequent progresses in the remainder of this article.



[FIG1] An RBM with *I* visible units and *J* hidden units.

A PRIME ARCHITECTURE OF DEEP LEARNING

In this section, we present a short tutorial on the most extensively investigated and widely deployed deep learning architecture, the DBN, as originally published in [7].

DBNs are probabilistic generative models that are composed of multiple layers of stochastic, latent variables. The unobserved variables can have binary values and are often called hidden units or feature detectors. The top two layers have undirected, symmetric connections between them and form an associative memory. The lower layers receive top-down, directed connections from the layer above. The states of the units in the lowest layer, or the visible units, represent an input data vector.

A DBN is built as a stack of its constituents, called restricted Boltzmann machines (RBMs) that we introduce next.

RESTRICTED BOLTZMANN MACHINE

An RBM is a special type of Markov random field that has one layer of (typically Bernoulli) stochastic hidden units and one layer of (typically Bernoulli or Gaussian) stochastic visible or observable units. RBMs can be represented as bipartite graphs as shown in Figure 1, where all visible units are connected to all hidden units, and there are no visible-visible or hidden-hidden connections.

In an RBM, the joint distribution p (v, h; θ) over the visible units v and hidden units h, given the model parameters θ, is defined in terms of an energy function E (v, h; θ) of

$$\mathrm{p}(\mathrm{v}, \mathrm{h}; \theta) = \frac{\exp(-\mathrm{E}(\mathrm{v}, \mathrm{h}; \theta))}{Z}, \quad (1)$$

where $Z = \sum_{\mathrm{v}} \sum_{\mathrm{h}} \exp(-\mathrm{E}(\mathrm{v}, \mathrm{h}; \theta))$ is a normalization factor or partition function, and the marginal probability that the model assigns to a visible vector v is

$$\mathrm{p}(\mathrm{v}; \theta) = \frac{\sum_h \exp(-\mathrm{E}(\mathrm{v}, \mathrm{h}; \theta))}{Z}. \quad (2)$$

For a Bernoulli (visible)-Bernoulli (hidden) RBM, the energy function is defined as

$$\mathrm{E}(\mathrm{v}, \mathrm{h}; \theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} v_i h_j - \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} a_j h_j, \quad (3)$$

where w_{ij} represents the symmetric interaction term between visible unit v_i and hidden unit h_j, b_i, and a_j are the bias terms, and I and J are the numbers of visible and hidden units. The conditional probabilities can be efficiently calculated as

$$p(h_j = 1|\mathrm{v}; \theta) = \sigma\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right), \quad (4)$$

$$p(v_i = 1|\mathrm{h}; \theta) = \sigma\left(\sum_{j=1}^{J} w_{ij} h_j + b_i\right), \quad (5)$$

where $\sigma(x) = 1/(1 + \exp(x))$. See a derivation in [1].

Similarly, for a Gaussian (visible)-Bernoulli (hidden) RBM, the energy is

$$\mathrm{E}(\mathrm{v}, \mathrm{h}; \theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} v_i h_j + \frac{1}{2}\sum_{i=1}^{I}(v_i - b_j)^2 - \sum_{j=1}^{J} a_j h_j. \quad (6)$$

The corresponding conditional probabilities become

$$p(h_j = 1|\mathrm{v}; \theta) = \sigma\left(\sum_{i=1}^{I} w_{ij} v_i + a_j\right), \quad (7)$$

$$p(v_i|\mathrm{h}; \theta) = \mathcal{N}\left(\sum_{j=1}^{J} w_{ij} h_j + b_i, 1\right), \quad (8)$$

where v_i takes real values and follows a Gaussian distribution with mean $\sum_{j=1}^{J} w_{ij} h_j + b_i$ and variance one. Gaussian-Bernoulli RBMs can be used to convert real-valued stochastic variables to binary stochastic variables, which can then be further processed using the Bernoulli-Bernoulli RBMs.

Taking the gradient of the log likelihood $\log p(\mathrm{v}; \theta)$ we can derive the update rule for the RBM weights as

$$\Delta w_{ij} = E_{\mathrm{data}}(v_i h_j) - E_{\mathrm{model}}(v_i h_j), \quad (9)$$

where $E_{\mathrm{data}}(v_i h_j)$ is the expectation observed in the training set and $E_{\mathrm{model}}(v_i h_j)$ is that same expectation under the distribution defined by the model. Unfortunately, $E_{\mathrm{model}}(v_i h_j)$ is intractable to compute so the contrastive

divergence (CD) approximation to the gradient is used where $E_{\text{model}}(v_i h_j)$ is replaced by running the Gibbs sampler initialized at the data for one full step [7]. Careful training of RBMs is essential to the success of applying deep learning to practical problems. A practical guide of the RBM training is provided in [6].

FROM RBM TO DBN

Stacking a number of the RBMs learned layer by layer from bottom-up gives rise to a DBN, an example of which is shown in Figure 2. The stacking procedure is as follows. After learning a Gaussian-Bernoulli RBM (for applications with continuous features such as speech) or Bernoulli-Bernoulli RBM (for applications with nominal or binary features such as black-white image or coded text), we treat the activation probabilities of its hidden units as the data for training the Bernoulli-Bernoulli RBM one layer up. The activation probabilities of the second-layer Bernoulli-Bernoulli RBM are then used as the visible data input for the third-layer Bernoulli-Bernoulli RBM, and so on. Theoretical justification of this efficient layer-by-layer greedy learning strategy is given in [7], where it is shown that the stacking procedure above improves a variational lower bound on the likelihood of the training data under the composite model. That is, the greedy procedure above achieves approximate maximum likelihood learning. Note that this learning procedure is unsupervised and requires no class label.

When DBN is applied to classification tasks, the generative pretraining can be followed by or combined with other, typically discriminative, learning procedures that fine-tune all of the weights jointly to improve the performance of the DBN. This discriminative fine-tuning is often performed by adding a final layer of variables that represent the desired outputs or labels provided in the training data. Then, the backpropagation algorithm can be used to adjust or fine-tune the DBN weights. For example, for speech recognition, the output layer can represent either syllables, phones, subphones, phone states, or other speech units used in the HMM-based speech recognition system.

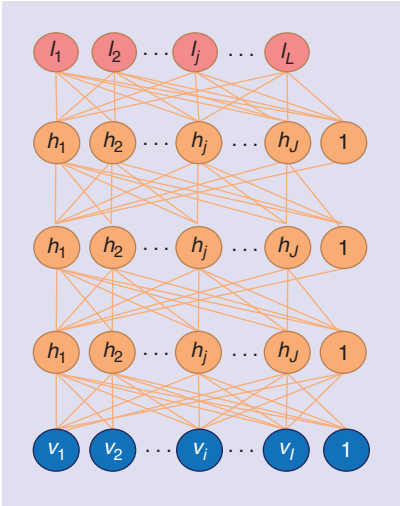
The learning procedure discussed above is typically expensive compared with the inference procedure, which can be efficiently carried out by a single forward pass. The inference procedure of DBN is analogous to the forward pass of the conventional MLP.

APPLICATIONS OF DEEP LEARNING TO SIGNAL PROCESSING AREAS

In the expanded technical scope of signal processing, the signal is endowed with not only the traditional types such as audio, speech, image and video, but also text, language, and document that convey high-level, semantic information for human consumption. In addition, the scope of processing has been extended from the conventional coding, enhancement, analysis, and recognition to include more human-centric tasks of interpretation, understanding, retrieval, mining, and user interface [4]. Many signal processing researchers have been working on one or more of the signal processing areas defined by the matrix constructed with the two axes of "signal" and "processing" discussed here. The deep learning techniques discussed in this article have recently been applied to quite a number of extended signal processing areas. We now provide a brief survey of this body of work in three main categories. Due to the limitation on the number of references, we have omitted some reference listings in the following survey.

SPEECH AND AUDIO

The traditional MLP has been in use for speech recognition for many years and when used alone, their performance is typically lower than the state-of-the-art HMM systems with observation probabilities approximated with Gaussian mixture models (GMMs). Recently, the deep learning technique was successfully applied to phone [8], [9] and large vocabulary continuous speech recognition (LVCSR) tasks by integrating the powerful discriminative training ability of the DBNs and the sequential modeling ability of the HMMs. Such a model as shown in Figure 3 is typically named DBN-HMM, where the observation probability is estimated using



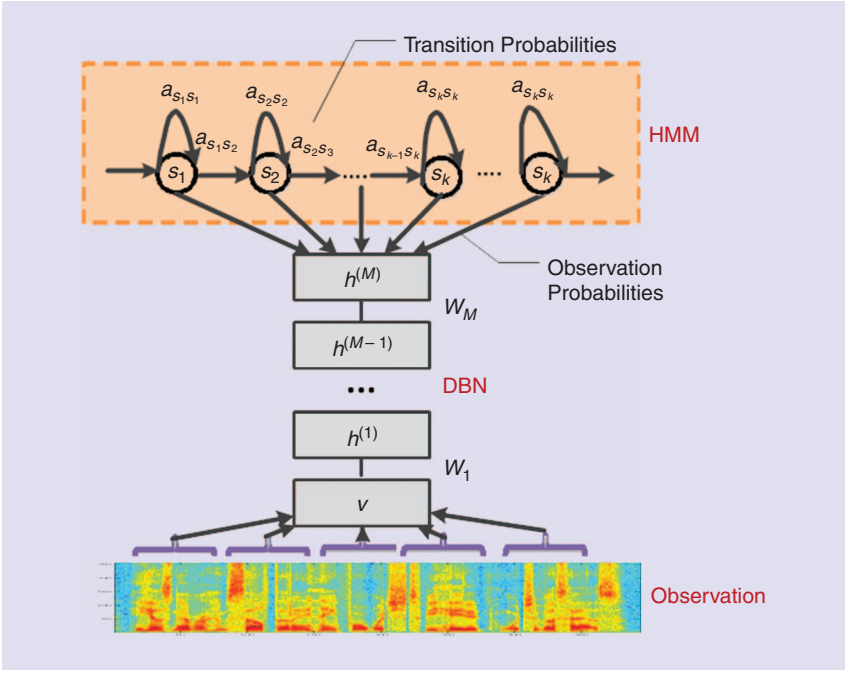
[FIG2] The DBN model used for classification. The hidden layers are generatively pretrained layer by layer by considering each pair of layers as an RBM. The output layer has labels from the supervised data.

the DBN and the sequential information is modeled using the HMM.

In [9], a five-layer DBN was used to replace the Gaussian mixture component of the GMM-HMM and the monophone-state was used as the modeling unit. Although the monophone model was used, the DBN-HMM approach achieved competitive phone recognition accuracy with the state-of-the-art triphone GMM-HMM systems.

The work in [8] improved the DBN-HMM used in [9] by using the CRF instead of the HMM to model the sequential information and by applying the maximum mutual information (MMI) training technique successfully developed in speech recognition to the DBN-CRF training. The sequential discriminative learning technique developed in [9] jointly optimizes the DBN weights, transition weights, and phone language model and achieved higher accuracy than the DBN-HMM phone recognizer with the frame-discriminative training criterion implicit in the DBN's fine-tuning procedure implemented in [9].

The DBN-HMM can be extended from the context-independent model to the context-dependent model and from the phone recognition to the LVCSR. Experiments on the challenging Bing mobile voice search data set collected



[FIG3] The DBN-HMM model for speech recognition. The observation probabilities are estimated using the DBN. The state values can be syllables, phones, subphones, monophone states, or triphone states and senones.

under the real usage scenario demonstrate that the context-dependent DBN-HMM significantly outperforms the state-of-the-art HMM system. Three factors contribute to the success: the usage of triphone senones as the DBN modeling units, the usage of the best available triphone GMM-HMM to generate the senone alignment, and the tuning of the transition probabilities. Experiments also indicate that the decoding time of a five-layer DBN-HMM is almost as that of the state-of-the-art triphone GMM-HMM.

In [5], the deep auto-encoder [7] is explored on the speech feature coding problem with the goal to compress the data to a predefined number of bits with minimal reproduction error. DBN pretraining is found to be crucial for high coding efficiency. When DBN pretraining is used, the deep auto-encoder is shown to significantly outperform a traditional vector quantization technique. If weights in the deep auto-encoder are randomly initialized, the performance is substantially degraded.

Another popular deep model is the convolutional DBN, which has been applied to audio and speech data for a number of tasks including music artist and genre classification, speaker identification, speaker gender classification, and phone classification, with strong results presented.

Other deep models have also been developed and presented. For example, the deep-structured CRF, which stacks many layers of CRFs, have been successfully used in the speech-related task of language identification, phone recognition, sequential labeling [15], and confidence calibration.

IMAGE AND VIDEO

The original DBN and deep auto-encoder were developed and demonstrated with success on the simple image recognition and dimensionality reduction (coding) tasks (MNIST) in [7]. It is interesting to note that the gain of coding efficiency using the DBN-based auto-encoder on the image data over the conventional method of principal component analysis as demonstrated in [7] is very similar to the gain reported in [5] on the speech data over the traditional technique of vector quantization.

In [10], Nair and Hinton developed a modified DBN where the top-layer model uses a third-order Boltzmann machine. They applied this type of DBN to the NORB database—a three-dimensional object recognition task. An error rate close to the best published result on this task was reported. In particular, it was shown that the DBN substantially outperforms shallow models such as SVMs.

Tang and Eliasmith developed two strategies to improve the robustness of the DBN in [14]. First, they used sparse connections in the first layer of the DBN as a way to regularize the model. Second, they developed a probabilistic denoising algorithm. Both techniques are shown to be effective in improving the robustness against occlusion and random noise in a noisy image recognition task. Another interesting work on image recognition with a more general approach than DBN appears in [11].

DBNs have also been successfully applied to create compact but meaningful representations of images for retrieval purposes. On the large collection image retrieval task, deep learning approaches also produced strong results.

The use of conditional DBN for video sequence and human motion synthesis was reported in [13]. The conditional DBN makes the DBN weights associated with a fixed time window conditioned on the data from previous time steps. The computational tool offered in this type of temporal DBN may offer the opportunity to improve the DBN-HMMs towards efficient integration of temporal-centric human speech production mechanisms into DBN-based speech production models.

LANGUAGE PROCESSING AND INFORMATION RETRIEVAL

Research in language, document, and text processing has seen increasing popularity recently by signal processing researchers, and has been designated as one of the main focus areas by the society's audio, speech, and language processing technical committee. There has been a long history of using (shallow) neural networks in language modeling (LM)—an important component in speech recognition, machine translation, text information retrieval, and in natural language processing. Recently, deep

networks have started attracting attention in the field of language processing and information retrieval.

Temporally factored RBM has been used for LM. Unlike the traditional N-gram model, the factored RBM uses distributed representations not only for context words but also for the words being predicted. This approach can be directly generalized to deeper structures.

In natural language processing, Collobert and Weston [2] developed and employed a convolutional DBN as the common model to simultaneously solve a number of classic problems including part-of-speech tagging, chunking, named entity tagging, semantic role identification, and similar word identification. A similar multitask learning technique with DBN is used in [3] to attack the machine transliteration problem, which may be generalized to the more difficult problem of machine translation.

Finally, we discuss a very interesting approach of using DBN and deep autoencoder for document indexing and retrieval [11], [12]. It is shown that the hidden variables in the last layer not only are easy to infer but also give a much better representation of each document (based on the word-count features) than the widely used latent semantic analysis. Using the compact code produced by deep networks, documents are mapped to memory addresses in such a way that semantically similar text documents are located at nearby address to facilitate rapid document retrieval. This idea is explored for audio document retrieval and some class of speech recognition problems with the initial exploration reported in [5].

SUMMARY AND FUTURE DIRECTIONS

We have introduced the basic idea of deep learning, the prevailing deep models such as DBN, and the popular and effective deep learning algorithms including the RBM and denoising autoencoder-based pretraining approaches. Literatures show that deep learning techniques have already demonstrated promising results in many signal processing applications.

Deep learning is an emerging technology. Despite the empirical promising results reported so far, much needs to be developed. In this section, we point out some important future directions.

- We need to better understand the deep model and deep learning. Why is learning in deep models difficult? Why do the generative pretraining approaches seem to be effective empirically? Is it possible to change the underlining probabilistic models to make the training easier? Are there other more effective and theoretically sound approaches to learn deep models?
- We need to find better feature extraction models at each layer. We have noticed that if we don't use the derivative and accelerator features in the DBN-HMM, the speech recognition accuracy is significantly reduced. This suggests that the current Gaussian-Bernoulli layer is not powerful enough to extract important discriminative information from the features. Recent work has shown that by using a three-way associative model called mcRBM, derivative and accelerator features are no longer needed to produce state-of-the-art recognition accuracy. There is no reason to believe mcRBM is the best first-layer model for feature extraction either. Theory needs to be developed to guide the search of proper feature extraction models at each layer.
- It is necessary to develop more powerful discriminative optimization techniques. Although the current learning strategy of generative pretraining followed by discriminative fine-tuning seems to work well empirically for many tasks, it failed to work for some other tasks such as language identification. For those tasks, the features extracted at the generative pretraining phase seem to describe the underlining speech variations well but do not contain enough information to distinguish between different languages. A learning strategy that can extract discriminative features for those tasks is in need. Extracting discriminative features may also greatly reduce the model size needed in the current deep learning systems.
- We need to develop effective and scalable parallel algorithms to train deep models. The current optimization algorithm, which is based on the mini-batch stochastic gradient, is difficult to be parallelized over computers. The current best practice is to use graphical processing units (GPUs) to speedup the learning process. However, single machine GPU processing is not practical for large data sets that is typical in large-scale, real-world speech recognition and similar applications. To make deep learning techniques scalable to thousands of hours of speech data, for example, theoretically sound parallel learning algorithms need to be developed.
- We need to search for better approaches to use deep architectures for modeling sequential data. The existing approaches, such as DBN-HMM and DBN-CRF, represent simplistic and poor temporal models in exploiting the power of DBNs. Models that can use DBNs in a more tightly integrated way and learning procedures that optimize the sequential criterion are important to further improve the performance of sequential classification tasks.
- Developing adaptation techniques for deep models is necessary. Many conventional models such as GMM-HMM have well-developed adaptation techniques that allow for these models to perform well under diverse and changing real-world environments. If deep models do not have effective adaptation techniques, it would be difficult for them to outperform the conventional models when the test set distribution is different from the training set distribution, which is common in real applications.

AUTHORS

Dong Yu (dongyu@microsoft.com) is a researcher and *Li Deng* (deng@microsoft.com) is a principal researcher, both at Microsoft Research, Redmond, Washington.

(continued on page 154)

Format, inherited in IM AF, enables simplicity in the file structure in terms of objects that have their own names, sizes, and defined specifications according to their purpose.

Figure 4 illustrates the IM AF file format structure. It mainly consists of "ftyp," "moov," and "mdat" type information objects/boxes. The "ftyp" box contains information on file type and compatibility. The "moov" box describes the presentation of the scene and usually includes more than one "trak" boxes. A "trak" box contains the presentation description for a specific media type. A media type in each "trak" box could be audio, image, or text. The "trak" box supports time information for the synchronization with other "trak" boxes described media. The "mdat" box contains the media data themselves described in the "trak" boxes. However, a "trak" box may also include a URL where from the media data could be imported. In this way the "mdat" box maintains a compact representation enabling consequently, efficient exchange and sharing of IM AF files.

Furthermore, in "moov" box some specific information is also included, such as the group container box "grco", the preset container box "prco" and the rules container box "ruco" for storing group, preset and rules information, respectively. The "grco" box contains, zero or more group boxes "grup" describing the group hierarchy structure of audio tracks and/or groups. The "prco" box contains one or more "prst" boxes that describe the predefined mixing information, in the absence of user interaction. The "ruco" box contains zero or more selection rules boxes "rusc" and/or mixing rules boxes "rumx" describing the interactivity rules related to selection and/or mixing of audio tracks.

FURTHER TECHNICAL DEVELOPMENTS

Due to the music industry's extreme interest in IM AF, there has already been a request for additional functionality to be incorporated in the IM AF standard, in terms of an amendment. This functionality is related to audio equalization (EQ) information aiming to offer users a complete music producer experience.

Audio EQ settings can be stored as presets in the preset-mix mode or specified by users in user-mix mode. Tweaking the EQ parameters directly in an IM AF player would be a similar process to the traditional producer-based music mixing environment. Alternatively, EQ settings could be applied automatically, based on a user's saved EQ profile, for masking reduction among different instruments and/or boosting or attenuating certain frequency bands of particular instruments, depending upon the user's preferences in music styles, personal taste, or mood.

AUTHORS

Inseon Jang (jinsn@etri.re.kr) is with the Realistic Acoustics Research Team at the Electronic and Telecommunications Research Institute (ETRI). She is a coeditor of the IM AF standard.

Panos Kudumakis (panos.kudumakis@eecs.qmul.ac.uk) is research manager at the Centre for Digital Music, Queen Mary University of London, United Kingdom. He has been an active member of the MPEG Committee since 1998.

Mark Sandler (mark.sandler@eecs.qmul.ac.uk) is a professor and director at the Centre for Digital Music, Queen Mary University of London, United Kingdom.

Kyeongok Kang (kokang@etri.re.kr) is a director at the Realistic Acoustics Research Team of the Electronic and Telecommunications Research Institute (ETRI).

RESOURCES

REFERENCES

[1] "Digital music report 2009: New business models for a changing environment," *Int. Federation of the Phonographic Industry (IFPI)*. London, Jan. 2009 [Online] Available: http://www.ifpi.org/content/library/dmr2009.pdf.
[2] S. Goel, P. Miesing, and U. Chandra, "The impact of illegal peer-to-peer file sharing on the media industry," *California Manage. Rev.*, vol. 52, no. 3, pp. 6–33, Spring 2010.
[3] L. Chiariglione, "The digital media manifesto-vision" [Online]. Available: http://manifesto.chiariglione.org/dmm.htm

SERVICES

[4] Korean Interactive Music Service [Online]. Available: http://www.audizen.com
[5] French Interactive Music Service [Online]. Available: http://www.iklaxmusic.com

STANDARD

[6] ISO/IEC 23000-12. (2010, July). *Information Technology—Multimedia Application Format (MPEG-A)—Part 12: Interactive Music Application Format* [Online]. Available: http://www.iso.org/iso/prods-services/ISOstore/store.html

CONFORMANCE AND REFERENCE SOFTWARE

[7] "Study of ISO/IEC 23000-12 FPDAM1 IM AF Conformance and Reference Software" [Online]. Available: http://phenix.it-sudparis.eu/mpeg/doc_end_user/current_document.php?id=29359, 93rd MPEG Meeting, Guangzhou, China, Oct. 2010, N11575.

[SP]

exploratory DSP continued from page 149

REFERENCES

[1] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[2] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. ICML*, 2008.
[3] T. Deselaers, S. Hasan, O. Bender, and H. Ney, "A deep learning approach to machine transliteration," in *Proc. 4th Workshop Statistical Machine Translation*, Athens, Greece, Mar. 2009, pp. 233–241.
[4] L. Deng, "Expanding the scope of signal processing," *IEEE Signal Processing Mag.*, vol. 25, no. 3, pp. 2–4, May 2008.
[5] L. Deng, M. Seltzer, D. Yu, A. Acero, A. Mohamed, and G. Hinton, "Binary coding of speech spectrograms using a deep auto-encoder," in *Proc. Interspeech*, 2010.
[6] G. Hinton, "A practical guide to training restricted Boltzmann machines," Univ. Toronto, Tech. Rep. 2010-003, Aug. 2010.
[7] G. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, pp. 1527–1554, 2006.
[8] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," in *Proc. Interspeech*, Sept. 2010.
[9] A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *Proc. NIPS Workshop Deep Learning for Speech Recognition*, 2009.
[10] V. Nair and G. Hinton, "3-D object recognition with deep belief nets," in *Proc. NIPS*, 2009.
[11] M. Ranzato, S. Chopra, Y. LeCun, and F.-J. Huang, "Energy-based models in document recognition and computer vision," in *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2007.
[12] R. Salakhutdinov and G. Hinton, "Semantic hashing," in *Proc. SIGIR Workshop Information Retrieval and Applications of Graphical Models*, 2007.
[13] G. Taylor, G. E. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," in *Proc. NIPS*, 2007.
[14] Y. Tang and C. Eliasmith, "Deep networks for robust visual recognition," in *Proc. ICML*, 2010.
[15] D. Yu, S. Wang, and L. Deng, "Sequential labeling using deep-structured conditional random fields," *J. Select. Topics Signal Processing* (Special Issue on Statistical Learning Methods for Speech and Language Processing), 2010.

[SP]