Automatic Polyphonic Piano Music Transcription by a Multi-Classification Discriminative-Learning

Stefano D'Urso and Aurelio Uncini

INFOCOM Dept. - University of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome - Italy
aurel@ieee.org
http://infocom.uniroma1.it/aurel

Abstract. In this paper we investigate on the use locally recurrent neural networks (LRNN), trained by a discriminative learning approach, for automatic polyphonic piano music transcription. Due to polyphonic characteristic of the input signal standard discriminative learning (DL) is not adequate and a suitable modification, called multi-classification discriminative learning (MCDL), is introduced. The automatic music transcription architecture presented in the paper is composed by a pre-processing unit which performs a constant Q Fourier transform such that the signal is represented in both time and frequency domain, followed by a peak-peaking and decision blocks: the last built with a LRNN. In order to demonstrate the effectiveness of the proposed MCDL for LRNN several experiments have been carried out.

1 Introduction

Music Transcription is formally defined as the process of finding the score of a musical piece, that is, finding a parametric representation of an acoustic waveform (notes, intensity, starting times, durations, instruments and other sound features).

The 1^{st} attempt in Automatic Music Transcription (AMT) has been made by Moorer [1] in 1975. He was able to identify some of the problems that persist to this day as monophonic-polyphonic classification, onset-offset detection, octave errors, ghost notes, repeated notes, notes' length and reverberation. Several transcription systems have been developed after Moorer's one: although different, all of them follow a three steps procedure.

The 1^{st} phase is known as signal's pre-processing, and its aim is to obtain a time-frequency representation of the musical signal. The simplest one is the spectrogram, but it fails because of its incoherence with the logarithmic human's way of hearing sounds. In 1988, [2] proposed the Constant Q Transform (CQT). It emulates human ears, modelling the critical band scale and avoiding the fixed bandwidth (a Fourier Transform's feature): the constant factor Q, represents the ratio of frequency resolution. In 1996, Guillermain and Kronland-Martinet [3]

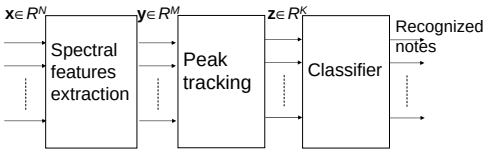

Fig. 1. General scheme of automatic music transcription system. The vector x represent the input time windowed signal; y is the array of its time-frequency representation; z is the array containing the principal spectral peaks

proposed the Wavelet Transform (WT) (also known as scalogram). It decomposes the signal in terms of shifts and dilations of an elementary function known as the mother wavelet. The result is then interpreted in the time-scale domain.

The 2^{nd} step is known as tracking phase and its aim is to track partials (locating sinusoids in the note). It is possible to find different approaches: Klapuri's [4] sinusoid tracks, Sterian and Wakefield's [5] Kalman filter algorithm or blackboard architectures [6]. The interpretation of tracking results is know as recognition phase.

In this work, we decide to pay attention on Automatic Transcription of Polyphonic Piano Music. We exploited most related works [7] [8], in particular, Matija Marolt's SONIC [8]. We introduced a novel neural network model made up of Locally Recurrent Neural Network (LRNN) trained by a discriminative learning algorithm opportunely modified for the task of multi classification of polyphonic music recognition.

This paper is organized as follows: in section 2 we illustrate the neural network model made up of LRNNs for AMT; in section 3 we consider the discriminative learning approach and its modified version, the MCDL; in section 4 we combine the LRNNs model with the MCDL technique. We finally related about experiments results.

2 A Neural Networks approach for AMT

The AMT should be considered a typical problem of dynamic pattern association, since we have to associate sounds to notes.

The increasing popularity of neural network models to solve pattern recognition problems has been primarily due to their low dependence on domain-specific knowledge the availability of efficient learning algorithms for practitioners to use: they provide a new suite of non-linear algorithms for feature extraction (using hidden layers) and classification. In addition, existing feature extraction and classification algorithms can also be mapped on neural network architectures for efficient (hardware) implementation.

Referring to Figure 1, for the feature extractions we used constant Q transform CQT [2] as signal's pre-processing phase. More details on its fast implementation are in [14].

In [8] Marolt tested several neural networks models (MLP, RBF, time-delay, Elmann, Fuzzy ARTMAP) and the one that best fitted with the SONIC architecture was the TDNN. Our motivation for RNNs relies on the fact that dynamic recurrent neural networks have proved to be really useful in many temporal processing applications as DSP and temporal pattern recognition. In particular, we'll make use of Locally Recurrent neural Network (LRNNs): in [9] it is possible to find the major advantages of LRNNs with respect to other models of RNNs as buffered MLP or fully RNNs.

We used Causal Recursive BackPropagation (CRBP) [9] as gradient-based learning: it is an on-line algorithm that implements and combines together the BackPropagation Through Time (BPTT) and the Real-time Recurrent Learning (RTRL); it can be efficiently implemented (respect to truncated BPTT) and has the advantage of being local in space and time (respect to RTRL that is not local in space).

In order to obtain improvements in terms of generalization capability and of learning speed we'll make use of the flexible spline activation function [10].

When working with IIR synapses it is important to assure stability: it is known that a static causal filter is asymptotically stable if and only if (iff) the poles of its transfer function lie inside the unit circle of the complex plane. We implemented the Intrinsically Stable Adaptation (ISA) [11] that makes possible to continually adjust the coefficients with no need of stability test or poles projection: the coefficients are adapted in a way that intrinsically assures the poles to be inside the unit circle.

The neural network model is depicted in Figure 2:

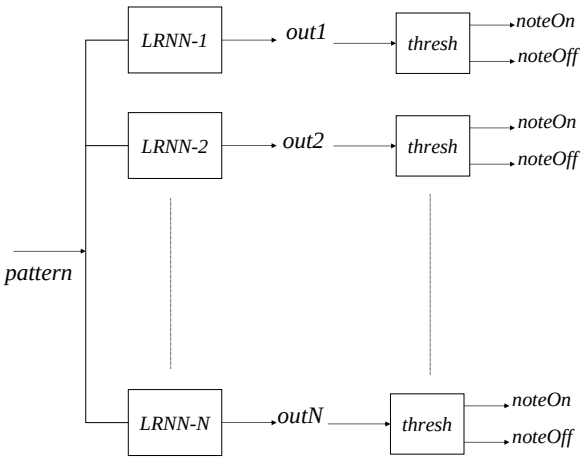

Fig. 2. Neural Network model for Automatic Music Transcription

Each LRNN is trained to recognize a note among the N chosen; the input pattern (the output of the signal's pre-processing phase), is FW through each

net: if the out of the single net is above a threshold, this means that the note is ON, OFF otherwise.

3 The Discriminative Learning for multi classification problems

3.1 The Discriminative Learning

One drawback of traditional approaches to pattern classification is that the estimation error does not immediately translate into correct recognition: the standard MLP uses a minimum mean square error (MMSE) criterion that doesn't necessarily minimize the classification error rates. In [12], Juang and Katagiri introduced a new formulation for the minimum error classification (MEC) problem called discriminative learning.

Let's consider a k-dimensional feature vector $\mathbf{x}$; a linear discriminant function could be defined as:

Each LRNN is trained to recognize a note among the N chosen; the input pattern (the output of the signal's pre-processing phase), is FW through each net: if the out of the single net is above a threshold, this means that the note is ON, OFF otherwise.

4 The Discriminative Learning for multi classification problems

4.1 The Discriminative Learning

One drawback of traditional approaches to pattern classification is that the estimation error does not immediately translate into correct recognition: the standard MLP uses a minimum mean square error (MMSE) criterion that doesn't necessarily minimize the classification error rates. In [12], Juang and Katagiri introduced a new formulation for the minimum error classification (MEC) problem called discriminative learning.

Let's consider a k-dimensional feature vector $\mathbf{x}$; a linear discriminant function could be defined as:

$$g(\mathbf{x}) = \mathbf{w}\mathbf{x}^T + w_0 \tag{1}$$

where $\mathbf{w}$ represent the weight vector and w_0 the threshold.

If we have a pattern recognition problem with M classes, we'll have M discriminant function, and so a classifier parameter set Λ:

$$\Lambda = \{\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2}, \ldots, \boldsymbol{\lambda_M}\} = \{\mathbf{w_1}, w_{01}, \mathbf{w_2}, w_{02}, \ldots, \mathbf{w_M}, w_{0M}\} \tag{2}$$

where $\boldsymbol{\lambda_i} = \{\mathbf{w_i}, w_{0i}\}$

Let the feature vector element $x_0 = 1$, each discriminant function could be written as:

$$g(\mathbf{x}) = \mathbf{w}\mathbf{x}^T + w_0 x_0 = \boldsymbol{\lambda_i}\mathbf{x}^T \tag{3}$$

This classifier uses the following decision rule:

$$C(\mathbf{x}) \in C_i \qquad \text{if } g_i(\mathbf{x}, \Lambda) = \max_j g_j(\mathbf{x}, \Lambda) \tag{4}$$

Thus, a feature vector $\mathbf{x}$ belongs to the class C_i that has the maximum value of the discriminant function; having linear discriminant functions brings hyperplan decision boundaries.

Despite the fact that learning with MMSE criterion does not necessarily lead to MEC, due to the computational efficiency, the determination of the classifier parameters set is usually formulated as a MMSE procedure using an objective function weighted by a discriminate function.

In order to derive the new objective criterion, the traditional discriminant formulation, have to be replaced with the following three-step procedure:

1. Determination of the form of the discriminant functions $g_i(\mathbf{x}, \Lambda)$
2. Determination of a quantity that indicates whether an input token x of the k-th class is to be misclassified according to the design rule of [7], implemented by the classifier parameter set Λ. This quantity is known as misclassification measure and one reasonable possibility is:

$$d_k(\mathbf{x}) = -g_k(\mathbf{x}, \Lambda) + \left[\frac{1}{M-1} \sum_{i,j \neq k} g_j(\mathbf{x}, \Lambda)^\eta \right]^{\frac{1}{\eta}} \tag{5}$$

3. Formulation of the minimum error objective. A general form of the cost function can be defined as:

$$\ell_k(d_k(\mathbf{x}) = d_k(\mathbf{x}, \Lambda) \tag{6}$$

   Note that the cost function ℓ_k and the misclassification measure d_k can be defined individually for each class k. Two of the several possibilities for the cost function are the exponential or the translated sigmoid: both of them are smoothed zero-one cost functions suitable for gradients algorithms. Clearly, a correct classification have no costs, instead a misclassification leads to a penalty that becomes a count of the classification error determined by the loss defined above.

Finally, for any unknown $\mathbf{x}$, it is possible to define an empirical average cost as:

$$J(\mathbf{x}, \Lambda) = \sum_{k=1}^{M} \ell_k(\mathbf{x}, \Lambda) a \tag{7}$$

where $a = \begin{cases} 1 & \text{if } \mathbf{x} \in C_k \\ 0 & \text{otherwise} \end{cases}$

This classifier performance function is the basis of the objective that we shall optimize with descent methods as the MMSE: $\Lambda_{t+1} = \Lambda_t + \epsilon \nabla J(\Lambda_t)$. The Probabilistic Descent Theorem [13] assures the convergence of Λ_t to a locally optimum solution Λ^*.

In [12] it is possible to find the complete MEC formulation for MLP. The MLP structure for an M classes classification problem is depicted in Figure 3: the network has M output neurons, each one of which models the discriminant function of the DL formulation. Since an input x belongs to the class with the highest discriminant function value among the M discriminant functions values, we expect that the correspondent output neuron of the MLP has the highest value among the M output neurons values.

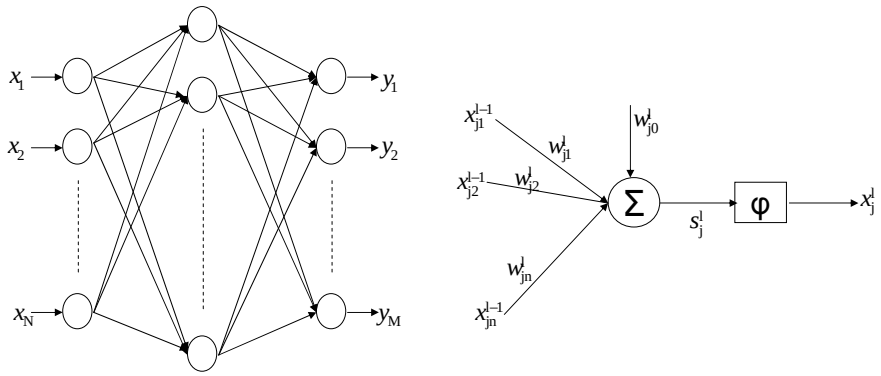

Fig. 3. MLP for M classes classification problem and neuron-j, level-l, n-input

4.2 Multi-Classification Discriminative Learning Algorithm

The DL model is attractive but it can't be applied directly to our architecture, in fact it is required that each input pattern belongs to a single class; instead, our model can contain input patterns that belong to different classes, or in other words, sounds made up of different notes (polyphony). We need to re-consider the basic idea of discrimination: we need a procedure that takes into account the fact that there's the possibility of a not complete separation between classes and that some input could lie in that common territory; we need to extend the MEC for MLP to MCDL for LRNN.

Intuitively, we could apply a sort of superimposition of effects procedure in which we consider a multiple classes input $\mathbf{x} \in C_p \subseteq \bigcup_{k=1}^{M} C_k$ as belonging separately to each one of the right p classes, and then combine the results. The application of this procedure to the standard weight adjustment rule for neuron in Figure 3:

$$\Delta w_{ji}^{l} = -\eta \frac{\partial \ell}{\partial s_j^l} \frac{\partial s_j^l}{\partial w_{ji}^l} = \delta_j^l x_i^{l-1} \tag{8}$$

leads to the following modification:

$$\Delta w_{ji}^{l} = \sum_{a \in \{C_p\}} \Delta w_{ji}^{l}(a) = x_i^{l-1} \sum_{a \in \{C_p\}} \delta_j^l(a) \tag{9}$$

where $\Delta w_{ji}^{l}(a)$ and $\delta_j^l(a)$ are respectively the weight adjustment rule and the delta rule for an input $\mathbf{x}$ that belongs to the a-th right class among the p, according to the superimposition of effects rule.
Let's consider the delta rule for the last layer:

$$\begin{cases} \delta_j(a) = -\eta \frac{\partial \ell(a)}{\partial s_j} = -\eta \frac{\partial \ell(a)}{\partial d_j} \frac{\partial d_j}{\partial y_j} \frac{\partial y_j}{\partial s_j} = -\eta \ell'(\partial d_a) \frac{\partial d_j}{\partial y_j} \\ \frac{\partial y_j}{\partial s_j^l} = \varphi'(s_j) = 1 \qquad \text{because of the final linearity} \end{cases} \tag{10}$$

$$d_a = -y_a + \left[\frac{1}{M-p} \sum_{k \notin \{C_p\}} y_k^{\eta} \right] \tag{11}$$

$$\frac{\partial d_a}{\partial y_k} = \begin{cases} -1 & k \in \{C_p\} \\ \frac{y_k^{\eta-1}}{M-p} \left[\frac{1}{M-p} \sum_{k' \notin \{C_p\}} y_{k'}^{\eta} \right] & k \notin \{C_p\} \end{cases}$$

Combining the results above:

$$\Delta w_{ji}^{l} = \sum_{a \in \{C_p\}} \delta_j(a) = \begin{cases} -\eta \mathbf{x}_i^{l-1} \frac{\partial d_a}{y_j} \ell'(d_a) & j \in \{C_p\} \\ -\eta \mathbf{x}_i^{l-1} \frac{\partial d_a}{y_j} \sum_{k \in \{C_p\}} \ell'(d_a) & j \notin \{C_p\} \end{cases}$$

where $\frac{\partial d_a}{y_j}$ is replaced by its right expression. If an input $\mathbf{x}$ belongs to no classes, no adjustment is made ($\Delta w_{ji}^{l} = 0$).

We also have to redefine the classifier decision rule: an input $\mathbf{x}$ belongs to a set of classes $\{C_p\}$, for each one of which, the discriminant function satisfies a quantification belonging condition rule as for example a thresh function or a domain integrity test.

5 The MCDL LRNN for ATM

The final model is depicted in Figure 4:

After a pre-training phase, in which each LRNN is trained singularly to recognize its target note, a discriminative over-training is made in order to assure generalization and relationship between nets (that is: which is other nets' behaviour when an input pattern is presented to the single net).

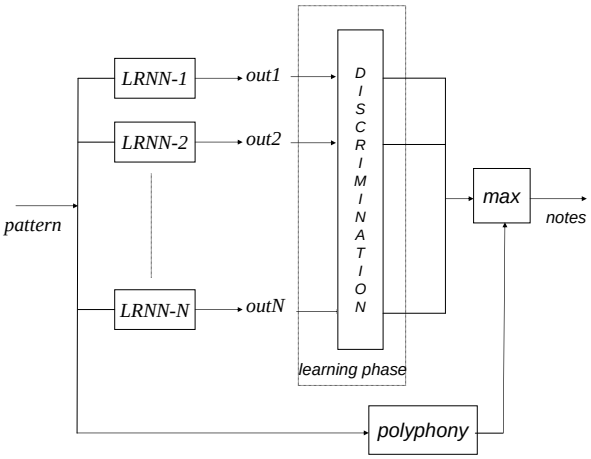

Fig. 4. MCDL model for Automatic Music Transcription

There's the necessity to introduce a polyphony net in order to establish how many outputs to take, that is, how many notes are playing (are ON) at a particular instant: if the polyphony net outputs m, the highest m outs are taken. We choose as polyphony net a LRNN that has to be trained independently from other LRNNs, since it has a different task from the others.

6 Experimental Results

The experimental results reported here are of two kinds: the first is a comparison between the model depicted in Figure 2 and a TDNNs model (similar to SONIC), in order to demonstrate the advantages in using LRNNs and the drawbacks of such approaches respect to discriminative ones; the second is mainly the description of the growing learning process of the MCDL model and its strength respect to the model in Figure.

All the networks used in these experiments had two layers, three hidden neurons with hyperbolic activation spline function, one output linear neuron and a variable number of input (generally 200) depending on the parameters chosen for the CQT analysis (frequency range and the resolution). We used MA-AR:2-1 for the first layer in LRNN and MA:2 for the first layer in TDNN.

All the wave samples used for training/testing are random pieces (16 bit - Mono - 11025Hz) generated using the Roland Virtual Sound Canvas VSC-88: these pieces has a maximum polyphony level of 5 and a minimum note length of 50ms.

6.1 TDNN model vs. LRNN model

We used nearly 1000 wave samples in the range [C3; B5] as training and testing sets. The LRNN model reveals its efficacy respect to the TDNN both on training

and testing sets, in particular, the LRNN model is able to identify correctly notes that are misclassified by the TDNN with octave errors.

One drawback with this scheme is the redundancy of training, in fact each network has to be trained independently with a specific training set. Moreover, there's the need to construct larger training sets in order to minimize the classification error on testing sets that are substantially different from the training sets (that is: testing sets that contain different music styles respect to the corresponding training sets): however, very large training sets makes the learning a real difficult task.

That's why there's the necessity to introduce a new model that is able to overcome these difficulties.

6.2 MCDL for ATM

We used single note wave files (with different lengths) as training set for all of the LRNNs in the scheme. After an individual pre-training phase in which every LRNN learns its target note from its single note wave files, a discriminative over-training is made (with the whole training set), in order to relate each LRNN to the other.

The model is able to classify exactly single notes and monophonic sequences, without the need to enlarge the training set. That's an interesting result, and reveals the strength of the DL respect to a standard LRNN model: it's impossible for a standard LRNN model to obtain this result, conditions being equal.

If we over-train again the monophonic-model obtained before, with growing levels of polyphony wave files, using the MCDL algorithm, we are able to obtain again correct classification, without wasting the preceding results.

Obviously, when working with high levels of polyphony, there's the necessity to reconsider the preceding over-trainings steps and vary the η parameter of discrimination: generally, a short value for η is used when polyphony grows.

7 Conclusions

In this paper, we presented a new approach to Automatic Music Transcription of Polyphonic Piano Music: the MCDL LRNN model. This model reveals its efficacy in most of typical Music Transcription problems combining the advantages of LRNNs (respect to other dynamic pattern recognition techniques) and discriminative learning (respect to other classification techniques).

The multi-classification technique is the real novelty of this approach: its usage makes the learning simpler and faster, making useless the build of separated training sets for each notes, giving coherence and generalization to the whole model.

Because of the novelty of this approach, several tests there have to be made besides Music Transcription context, in order to understand the exact of efficacy of this technique. Extension of this system to higher levels of polyphony,

shorter note's length and different pre-processing techniques have to be considered. Additionally, a post-processing block (a neural network or an hidden Markov model) may be considered to correct errors.

References

1. Moorer, J.A. (1975), On the Segmentation and Analysis of Continuous Sound by Digital Computer, Ph.D.Thesis, Department of Computer Science, Stanford University, Stanford
2. Brown J. C. (1988), Calculation of a Constant Q Spectral Transform, J. Acoust. Soc. Am., 1991
3. Guillermain P., Kronland-Martinet R. (1996), Characterization of Acoustic Signal through Continuous Linear Time-Frequency Representations, Proc. IEEE, vol.84, no.4, pp.561-585
4. Klapuri A. (1997), Automatic Transcription of Music, Master of Science Thesis, Tampere University of Technology
5. Sterian A., Wakefield G. H. (1998), A model-base approach to partial tracking for musical transcription
6. Kashino K., Nakadai K., Kinoshita T., Tanaka H. (1995), Application of Bayesian probability network to music scene analysis, Proceedings of International Joint Conference in AI, Workshop on Computational Auditory Scene Analysis, Montreal, Canada
7. Dixon S. (2000), On the Computer Recognition of Solo Piano Music, Proceedings of Australian Computer Music Conference, Brisbane, Australia
8. Marolt M., SONIC: Transcription of Polyphonic Piano Music with Neural Networks, Ph.D. Thesis
9. Campolucci P., Piazza F. Uncini A. (1999), On-line Learning Algorithm for Locally Recurrent Neural Networks, IEEE Trans. On Neural Networks, vol. 10, no. 2
10. L. Vecci, F Piazza, A. Uncini, "Learning and Approximation Capabilities of Adaptive Spline Activation Function Neural Networks", Neural Networks, Vol. XI, No.2, pp. 259-279, 1998
11. Campolucci P., Piazza F. (2000), Intrinsic Stability-Control Method for Recursive Filters and Neural Networks, IEEE Trans. On Circuits and Systems - II: Analog and Digital Signal Processing, vol.47, no.8
12. Juang B. H., Katagiri S. (1992), Discriminative Learning for Minimum Error Classification, IEEE Trans. On Signal Processing, vol. 40, no. 12
13. Katagiri S., Lee C. H., Juang B. H. (1991), New discriminative algorithm based on the generalized probabilistic descent method, Proc. 1991 IEEE Workshop Neural Netoworks for Signal Processing, Piscataway, NJ, pp. 299-308
14. J. C., Puckette M. S. (1992), An efficient algorithm for the calculation of a constant Q transform, J. Acoust. Soc. Am., 92(5):2698-2701