Musical Instrument Recognition Using Artificial Neural Network

Mahmoud Shirazi, Alireza Khanteymoori, Bahram Sadeghi Bigham
Department of Computer Sciences and Information Technology
Institute for Advanced Studies in Basic Sciences (IASBS)
Zanjan, Iran
{m.shirazi, khanteymoori, b_sadeghi_b}@iasbs.ac.ir

Abstract— **This paper addresses musical sounds recognition produced by different instrument and focus on classification of instrument tones. Architecture of back-propagation and networks are applied as classifiers. The discrete Fourier transform vectors, mean, and variance extracted from each segment are used as parameters. The Music Instrument Sample Database (UIOWA) is used for this experiment. The number of instrument is 14. We use 16 different structures of neural networks for recognition these instruments and compare the results. Ezzaidi Hassan [1] obtained SR(14)=12/14 by MLP with 60, 120, and 240 units in the middle layer without impacting of training data set. We obtain SR(14)=13/14 using a different way for analyzing the music sounds.**

Keywords-component; Musical instrument recognition, Classification, Back-Propagation Neural Network, Multilayer Perceptron (MLP)

I. INTRODUCTION

Automatic instrument recognition is a subtask of musical content identification. Recently many studies have been oriented to musical signal analysis and processing in order to respond to the high demand of internet users and to countless multimedia applications. The demand includes audio indexing, automatic transcription, genre classification, singer identification and instrument recognition [1].

Various attempts have been made to construct automatic musical instrument recognition systems [1-4]. Researchers have used different approaches and scopes, achieving different performances. Most systems have operated on isolated notes, often taken from the same, single source, and having notes over a very small pitch range. There are many methods for instrument recognition, which use common spectro-temporal properties like cepstral coefficients or spectral envelopes.

In this paper, we use the artificial neural network for instrument recognition. The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations and also to use it. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical.

Artificial Neural Network (ANN) is a model that is inspired from human biological cells. The ANN main property is that it is able to learn from input (Unsupervised) or input-output (Supervised) and subsequently produce output for new input data.

The Multilayer Perceptron (MLP) neural networks have one input layer. The input data are given to the network from input layer. These networks have one or more middle layers. The encoded output data are produced by output layer.

A well known kind of the MLP is one with Back-Propagation (BP) neural network. In this paper, we use this kind of neural network that is learned from unsupervised learning algorithms.

Ezzaidi Hassan [1] proposed the use of different neural networks structures for music instruments recognition. He used a Multilayer Perceptron (MLP) neural network with back-propagation for classify 19 and 14 music instruments. The fastest and steepest descent with the momentum algorithm used in the training process to update the networks weight and biases in the negative direction of the gradient.

In this paper, we use different structures of neural networks for recognition 14 instruments and compare the results.

The rest of the paper is organized as following: The MLP with Back-Propagation neural network is discussed in the Section II generally. Two common kinds of the learning algorithms that we use in this paper are elaborated in Section III, and finally the experimental results are presented in Section IV.

II. BACK PROPAGATION NEURAL NETWORK

The output from a backpropagation neural network is computed using a procedure known as the forward-backward pass.

In forward pass the input layer propagates a particular input vector's components to each node in the middle layer.

Then the middle layer nodes compute output values, which become inputs to the nodes of the output layer. The output layer nodes compute the network output for a particular input vector.

The forward pass produces an output vector for a given input vector based on the current state of the network weights. The weights are adjusted to reduce the error by propagating the output error backward through the network. Because the desired output for each node is known, in backward pass the network computes error values for each node in the output layer. Therefore, the network computes the error for the middle layer nodes. The amount of error due to each middle layer node depends on the size of the weight assigned to the connection between the two nodes. Then the network adjusts the weight values to improve network performance using the Delta Rule. Finally, the network computes the overall error to test network performance. The training set is repeatedly presented to the network and the weight values are adjusted until the overall error is below a predetermined tolerance.

I. LEARNING ALGORITHMS

Supervised learning requires a training set that consists of input vector and a target vector associated with each input vector. The NN learner uses the target vector to determine how well it has learned, and to guide adjustments to weight values to reduce its overall error. The weight updating is generally described as:

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \tag{1}$$

Here $\Delta w_{ij}(n)$ is determined by learning algorithm and $w_{ij}(n)$ is initialized randomly. In this paper, the following two supervised algorithms have been employed and their prediction power and their performances have been compared.

A. Gradient Descent BP algorithm

In this algorithm, learning iteration consists of two phases-forward pass, which simply calculates the output(s) value of the NN for each training pattern; and backward propagation, which propagates from the output layer toward the input layer where weights are adjusted as functions of the back propagation error signal.

In this algorithm, the Sum Squared Error (SSE) is used as the objective function, and the error of output neuron j computes as follows:

$$\varepsilon(n) = \frac{1}{2}\left(\sum e_j^2(n)\right), e_j(n) = d_j(n) - y_j(n) \tag{2}$$

where $d_j(n)$ and $y_j(n)$ are respectively the target and the actual values of the j-th output unit.

The total mean squared error is the average of the network errors of the training examples.

$$E_{Av} = \frac{1}{N}\left(\sum_{n=1}^{N} \varepsilon(n)\right) \tag{3}$$

A weighted sum a_j, for the given input x_i, and weights w_{ij}, is computed as follows:

$$a_j = \sum_{i=1}^{n} x_i w_{ij} \tag{4}$$

Here n is the number of inputs to one neuron.

The standard sigmoid activation function with values between 0 and 1 is used to determine the output at neuron j:

$$y_j - f(a_j) = \frac{1}{1+e^{-a_j}} \tag{5}$$

Weights are updated according to following equation:

$$w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}(t) \tag{6}$$

$$\Delta w_{ij}(t) = \eta \frac{\partial E}{\partial w_{ij}} \tag{7}$$

η is learning rate.

B. Gradient Descent BP with mommentum algorithms

The convergence of the network by back propagation is crucial problem because it requires much iteration. To mitigate this problem, a parameter called "Momentum", can be added to BP learning method by making weight changes equal to the sum of fraction of the last weight change and the new change suggested by

$$w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}(t) + \alpha \Delta w_{ij}(t-1) \tag{8}$$

where α is momentum constant. The momentum is an effective means not only to accelerate the training but also to allow the network to respond to the (local) gradient [5].

I. EXPERIMENTAL RESULTS

This paper addresses musical sounds recognition produced by different instruments and focuses on the classification of instrument tones. We use different structures of neural networks for the recognition of 14 types of instruments and compare the results.

Ezzaidi Hassan [1] obtained SR [1] (19)=16/19 and SR(14)=12/14 by MLP with 60, 120, and 240 units in the middle layer without impacting of training data set. We obtain SR(14)=13/14 using a different way for analyzing the music sounds, and MLP with different parameters for the musical instrument recognition without impacting of training data set.

[1] *SR(14) is the score recognition for instruments identification with MLP networks for 14 instruments*

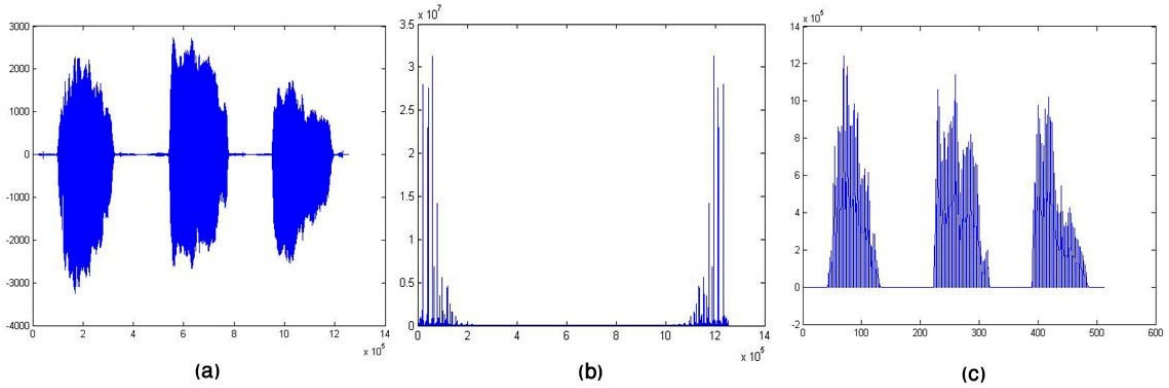

Fig. 1. (a) Sample of a signal (b) FFT output for signal that shown in (a) (c) Output of analysis step

A. Database

The collection of instruments used in this work is from the database of the University of Iowa [6], musical instrument samples. The collection is composed of 14 instruments which are as the following: Flute, Bass Flute, Bass Clarinet, Soprano Saxophone, Bb Clarinet, Bassoon, Alto Saxophone, French Horn, Tenor Trombone, Violin, Alto Flute, Bb Trumpet, Double Bass, and Piano.

The frequency of all samples is 44.1 KHz and the number of sample for each instrument is not equal in the collection above. Some of these instruments, like Bass Flute, have 12 samples and some of the others, like Piano, have 231 samples. We want to use a set of 50 samples for all these instruments as input in our work. Since each sample in this collection plays the repeated note, for increasing the number of entries, each sample is divided into several parts with equal lengths. Therefore, a set of 50 samples for all these instruments is extracted and used in this work. 80 percent of these samples are used for training and the rest of them are used to test the network.

B. Analysis of music

Here, we first read each of the samples as a signal. Then we analyze these samples for using as input of the neural network. In the analysis procedure we first calculate the Fast Fourier Transform (FFT) of the input vector. The FFT output is divided into 256 sections with the same length. Mean and variance of data in each section is calculated as the indicator for that. These indicators are used for constructing the output vector for this step. Obviously, the length of the output vector is 512. This output vector is the one used in the neural network. One sample of a particular signal, the FFT of this signal, and the output of analysis step are shown in Fig. 1.

C. Train and Test

We use a Multilayer Perceptron (MLP) neural network with back-propagation as a classifier. The number of the existing cells in the hidden layer is taken from the following numbers: 40, 80, 120, and 240.

Two learning algorithms (described in the Section III) are used for comparing the results. First one is the gradient descent back-propagation with adaptive learning rate (Traingdx), and the latter one is the fastest and steepest descent with the momentum algorithm (Traingdm). In both of the above cases, the multilayer network uses the sigmoid transfer function. The number of the iterations is set to 700 in the learning procedure. In the second case, that uses the momentum algorithm as the training function, the learning rate is fixed to 0.1 and the momentum constant is fixed to 0.9.

In this paper we want to use 16 different structures of MLP for comparing the results. These structures constructed from three parameters. These parameters consist of two learning algorithms (Section III), the number of middle layers (one or three), and the number of cells in each middle layer (40, 80, 120, and 240). In order to facilitate analysis of results, we are mixed the structures which using the same learning algorithm and the same number of middle layer. Therefore, we have 4 new structures that each of them contains four sub structures. The difference of these sub structures is the number of cells in each middle layer. As showed in Table I, each of these structures have one NO. In continuance of this paper we reference to each structure with its structure NO.

TABLE I. Different structures of MLP that used in this paper

Structure	Number of middle layer	Training function
I	1	Traingdx
II	3	Traingdx
III	1	Traingdm
IV	3	Traingdm

Identify applicable sponsor/s here. If no sponsors, delete this text box. (sponsors)

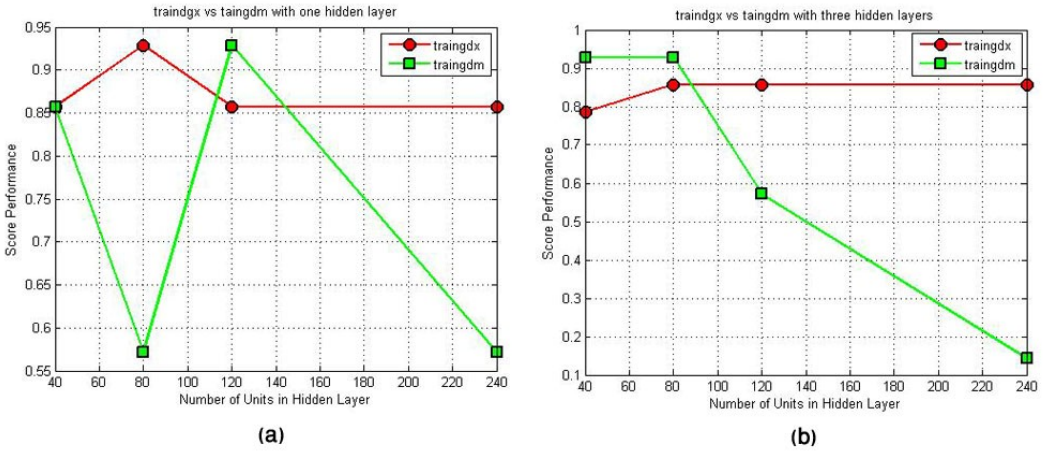

Fig. 2. Traingdx vs. Traingdm (a) With one middle layer (b) With three middle layers

According to the previous studies, two criterions namely Major Ratio (MAR) and Minor Ratio (MIR) were proposed to calculate the score performance. The MAR criterion considers an instrument as recognized if the score performance is better than all the other instruments and higher than 50%. The MIR criterion expresses an instrument as recognized if the score performance is only better than all the other instruments and lower than 50%.

Table II presented in this section illustrates the diagonal score recognition of the confusion matrix for structures numbers I to IV, respectively. The value of each cell in the Table III is SR(14) corresponds to the score performance with 14 considered instruments added to the score obtained from the MAR and MIR criterions, respectively.

As shown in Table II, with using Traingdx as the training function for a network with one middle layer (Structure I), the best score recognition (SR), which is obtained for 14 instruments, is 13/14 for the middle layer consisted of 80 units. For a neural network, that uses the Traingdx as a training function and has three middle layers (Structure II), the best SR is obtained for 80, 120, and 240 units in each middle layer. The best SR, which is obtained with using the Structure III, is calculated 13/14 for the middle layer consisted of 120 units. Finally, using the Structure IV, the best score performance is calculated 13/14 for a neural network that uses Traingdm as the training function and has three middle layers with 40 and 80 units in each one.

TABLE II. The score recognition of 14 instruments for multilayer perceptron neural network

Structure	Score Performance			
	40	80	120	240
I	12/14	13/14	12/14	12/14
II	11/14	12/14	12/14	12/14
III	12/14	8/14	13/14	8/14
IV	13/14	13/14	8/14	2/14

As can be seen from Table II with changing the number of units in middle layer in the Structures I and II the score performance does not change drastically, but this happens by changing the number of units in middle layers in the Structures III and IV.

In Fig. 2, these results are compared. In this figure the number of units in each middle layer is plotted versus the score performance to compare the effects of using two different types of training functions that is used in our work. The red and green lines are related to Traingdx and Traingdm respectively.

Fig. 2(a) compares the results obtained from using the Traingdx with the results gained from using Traingdm in a neural network that has one middle layer. Nevertheless both of these training functions give us the same score performance (13/14), there is an important difference between their behaviors. The score performance calculated by the Traingdx is almost constant for any number of units in the middle layer, while this quantity changes sharply when using the Traingdm.

For an MLP with three middle layers and each one consisted of 40 and 80 units, the score performance of Traingdm is better than the score performance of Traingdx, as was shown in Fig. 2(b). In spite of this fact, the Traingdm does not have a stable behavior that is not a good point about this training function.

Fig. 2 shows that for instrument recognition it is much better to use Traingdx instead of Traingdm, because the stability helps to reach an acceptable response sooner.

CONCLUSION

In this study, we are interested in the instrument identification task in the monophonic context. We use the Fast Fourier Transform (FFT) and mean and variance extracted from each segment that used as parameters. We obtained different score recognitions with different structures of neural network. The best score recognition that obtained in this study

is 13/14. Ezzaidi Hassan [1] obtained SR(14)=12/14 by MLP with 60, 120, and 240 units in the middle layer without impacting of training data set.

For instrument recognition, using the Traingdx as training function changing the number of units in the hidden layers, does not change the score performance substantially, but with the Traingdm as training function changing the number of units in the hidden layers, change the score performance substantially. Then training function has important role in instrument recognition.

REFERENCES

[1] E. Hassan, "Instruments recognition using neural networks and spectral information", Conference SETIT 2007 :Hammamet 25-29 March 2007.

[2] G. Mazarakis, P. Tzevelekos, and G. Kouroupetroglou, "Musical instrument recognition and classification using time encoded signal processing and fast artificial neural networks", Lecture Notes in Artificial Intelligence (LNAI), vol 3955, pp.246-255, 2006.

[3] E. Benetos, M. Kotti, and C. Kotropoulos, "Large scale musical instrument identification", in Proc. 4th Sound and Music Computing Conf., pp. 283-286, July 2007.

[4] E. Benetos, M. Kotti, C. Kotropoulos, "Applying Supervised Classifiers Based on non-negative Matrix Factorization To Musical Instrument Classification non-negative Matrix Factorization To Musical Instrument Classification", in Proc. IEEE Int Conf. Multimedia & Expo, July 2006.

[5] S. Haykin, "Neural Networks: A Comprehensive", second edition, Prentice-Hall, 1999.

[6] University of lowa's Music Instrument Samples, http://theremin.music.uiowa.edu/MIS.html