Music Chord Recognition Using Artificial Neural Networks

Neshadha Perera
Texas Tech University

Saluka Ranasinghe Kodituwakku
University of Peradeniya

Proceedings of the International Conference on Information and Automation, December 15-18, 2005, Colombo, Sri Lanka.

Music Chord Recognition Using Artificial Neural Networks

M. A. P. Neshadha Perera*, S. R. Kodithuwakku+
*Department of Statistics And Computer Science
University of Peradeniya, Sri Lanka
Email: salukak@pdn.ac.lk
+Department of Statistics And Computer Science
University of Peradeniya, Sri Lanka
Email: neshadha@gamil.com

Abstract—**Musical Instrument Digital Interface (MIDI) is an industrial standard for storing and transmitting musical data among various digital musical instruments. A MIDI file contains a sequence of musical data which can be processed to extract important information about the musical score it contains. In this research, a musical score presented as a MIDI file is used to recognize music chords corresponds to it with the help of an artificial neural network designed to follow the multiple adaptive linear neuron (MADALINE) network model.**

I. Introduction

Musical Instrument Digital Interface (MIDI) is an industrial standard for store musical data and share musical data and configuration data among various electronic musical instruments of different brands and different models. Since its introduction in January 1983, it has become very popular among many musicians and music producers. After the introduction of multimedia capabilities into the modern personnel computer, MIDI technology started to play a vital role in multimedia world. This leads to develop software tools which can be used to produce music scores much easily and quickly and these musical scores were able to be played on any MIDI compatible device or any multimedia personnel computer.

A MIDI file is a collection of MIDI events organized into virtual MIDI tracks. A typical MIDI file could consist of one or more MIDI tracks in which MIDI events correspond to different categories and different channels can be hold. In addition, a MIDI file contains META data, which describes the characteristics of the MIDI file such as its time signature and tempo information.

A music chord is a harmonic combination of musical notes, usually consist of three notes. Figure 1 illustrates the C Major chord. The test bed used in this research was a Yamaha PSR-2000 model electronic keyboard. This particular model can produce automatic accompaniment for 35 distinct chord types based on a single scale. Therefore the device can respond to total number of 420 different chord types.

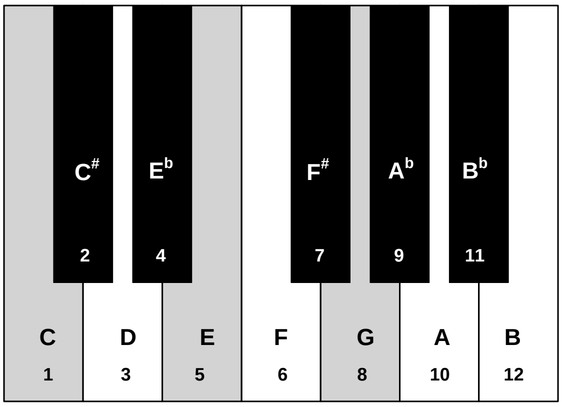

Fig. 1 Notes in C Major Chord

A multiple adaptive linear neuron model used here to identify music chords is a collection of perceptrons arranged in a linear network as shown in the fig. 5.

A MIDI file was segmented at specific time intervals and an artificial neural network is used to recognize musical chords correspond to each time segment in the provided MIDI file. The neural network consist of 24 linearly arranged perceptrons, each can identify a particular chord.

II. Analyzing MIDI file

A. Determination of MIDI file parameters

By extracting necessary Meta data, length of the MIDI file, number of MIDI ticks per measure and the speed at which the MIDI file should be played can be recognize. A MIDI tick is the smallest time quantity in a musical score. Usually a MIDI tick corresponds to a one MIDI event. A measure is the smallest period or the segment of time which is analyzed to recognize chords. These measures are equal in size and correspond to rhythmic cycles. Once these details are extracted, a suitable step size can be determined by following relations:

MIDI Ticks per Measure = Resolution × Beats Per Measure

$$\text{Number of measures} = \frac{\text{total number of MIDI ticks}}{\text{ticks per measure}}$$

B. Segmenting MIDI file

MIDI file is filtered to get only the NOTE_ON MIDI messages and NOTE_OFF MIDI messages. Each NOTE_ON message should terminated with a corresponding NOTE_OFF message. The time gap between these two messages gives the duration of a particular MIDI note. Figure 2 illustrates this concept in graphical form.

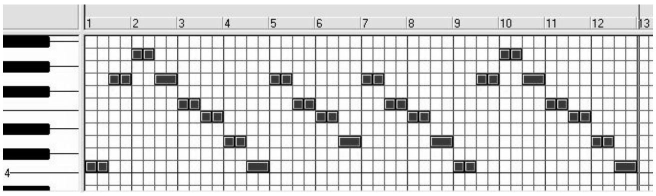

Fig. 2 MIDI notes in a music score in graphical form

Each note in MIDI has a specific note number ranging from 0 to 127. These note numbers are divided by 12 using modulo division and get the remainder to determine the relative position of a particular note in an octave. The numbers shown in fig. 1 are the corresponding position numbers of each note in an octave. These 12 notes become the inputs of the neural network which is used to recognize chords. The input is set to +1 if the note exists in the measure and is set to -1 if the note dose not exists in the measure. The following table shows the results of this process applied to measure 1 of the music score presented in fig. 2.

TABLE 1 ANALYZED MIDI NOTES OF MEASURE 1

Actual MIDI Note number	Relative note number	Input to Neural Network
48	0	+1 (Exist)
49	1	-1 (Not exist)
50	2	-1 (Not exist)
51	3	-1 (Not exist)
52	4	-1 (Not exist)
53	5	-1 (Not exist)
54	6	-1 (Not exist)
55	7	+1 (Exist)
56	8	-1 (Not exist)
57	9	-1 (Not exist)
58	10	-1 (Not exist)
59	11	-1 (Not exist)

III. MADALINE NETWORK

A. Individual Neurons

Individual neurons in the network were responsible for identifying specific chords. Each neuron is trained first using Widow-Hoff learning algorithm given in [3] with the sample data set given in table 3 and 4. Each neuron consist of 12 inputs corresponds to 12 semitone notes in an octave. Hard limiter shown in (1) is used as the activation function of the neuron.

$$f(x) = \begin{cases} +1; x > 0 \\ -1; x \le 0 \end{cases} \tag{1}$$

The bias term and bias weight are set to zero so that the influence of bias term is cancelled. The structure of a single neuron with its training component is presented in fig. 3.

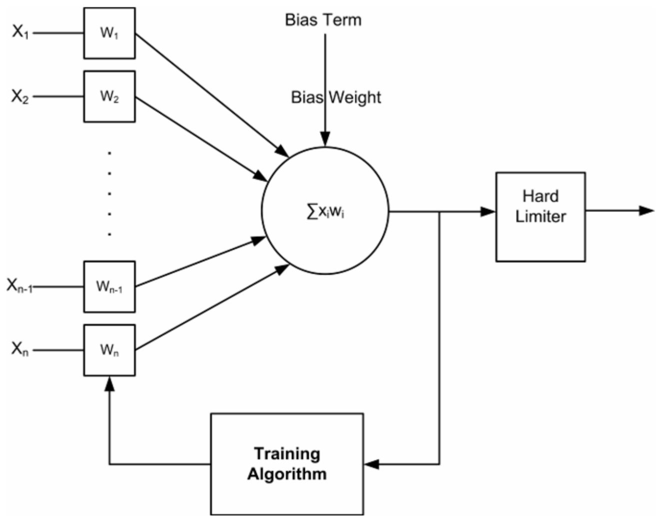

Fig. 3 Single neuron structure with training component.

The initial weight set is set to +1 and the training process is used to find the best weight set to identify the particular chord.

B. MADALINE Network structure

The neural network consist of 24 neurons shown in fig. 3 arranged linearly as a MADALINE network described in [3], so that they all share the same set of inputs. The structure of the MADALINE network is given in fig. 4. Outputs of each neuron are connected to an output selector in which only one neuron is selected as the neuron producing the output. The table 2 shows the chords identified by each neuron. Neurons are trained for these chords before the system is used to recognize chords.

TABLE 2 CHORDS IDENTIFIED BY EACH NEURON

Neuron	Chord	Neuron	Chord
1	C Major	13	$F^{\#}$ Major
2	C Minor	14	$F^{\#}$ Minor
3	$C^{\#}$ Major	15	G Major
4	$C^{\#}$ Minor	16	G Minor
5	D Major	17	A^{b} Major
6	D Minor	18	A^{b} Minor
7	E^{b} Major	19	A Major
8	E^{b} Minor	20	A Minor
9	E Major	21	B^{b} Major
10	E Minor	22	B^{b} Minor
11	F Major	23	B Major
12	F Minor	24	B Minor

C. *Output Selection*

It is expected to activate only one neuron at a time. However, if more than one neuron produces +1 as the output, the output selecting algorithm choose the best output by considering the value before the non linearity of each activated neuron. It calculates the error signal based on this value exactly same as it was calculated during the training and then the neuron with the least error is selected as the output neuron.

If in any case two error values are also become equal, the chance is given to the user to select the appropriate output chord.

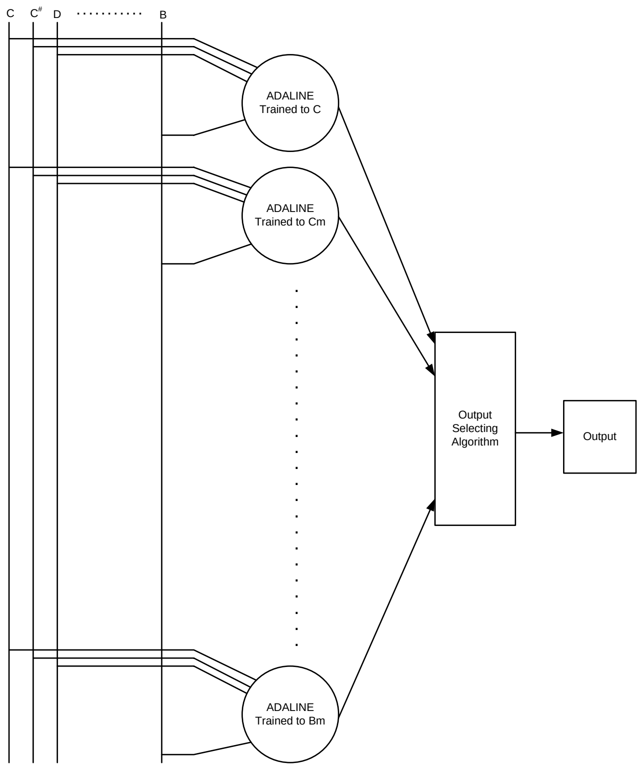

Fig. 4 MADALINE network structure

D. *Training the Network*

Each neuron is trained to recognize its corresponding chord before using to recognize music chords. Neurons are trained with LMS algorithm given in [3] one at a time with the same training data set. The number of iterations is limited to 10000 and 3 different learning rates 0.01, 0.001 and 0.1 are used with 3 different tolerance levels 1%, 0.1% and 1% respectively. However, when recognizing chords, weight set generated with the second combination (learning rate = 0.001 and tolerance= 0.1%) is used. Once the net is trained, it was then tested with the data set used for training to check its accuracy. If the net contains errors, it was re-trained to remove the errors in identification of chords.

The training data set consist of two subsets as theoretical sample data and real world sample data.

1) *Theoretical sample data*

Theoretical test data set consist of minimum notes required to complete a music chord. For the used 24 chords, the theoretical data set is given in the table 3.

TABLE 3 THEORETICAL TRAINING DATA SET

	Chord Name	**Note Combination**	**Symbol**
1	C Major	1 – 5 – 8	C
2	C Minor	1 – 4 – 8	Cm
3	$C^{\#}$ Major	2 – 6 – 9	$C^{\#}$
4	$C^{\#}$ Minor	2 – 5 – 9	$C^{\#}$m
5	D Major	3 – 7 – 10	D
6	D Minor	3 – 6 – 10	Dm
7	E^{b} Major	4 – 8 – 11	E^{b}
8	E^{b} Minor	4 – 7 – 11	E^{b}m
9	E Major	5 – 9 – 12	E
10	E Minor	5 – 8 – 12	Em
11	F Major	6 – 10 –	F
12	F Minor	6 – 9 – 1	Fm
13	$F^{\#}$ Major	7 – 11 – 2	$F^{\#}$
14	$F^{\#}$ Minor	7 – 10 – 2	$F^{\#}$m
15	G Major	8 – 12 – 3	G
16	G Minor	8 – 11 – 3	Gm
17	A^{b} Major	9 – 1 – 4	A^{b}
18	A^{b} Minor	9 – 12 – 4	A^{b}m
19	A Major	10 – 2 – 5	A
21	A Minor	10 – 1 – 5	Am
21	B^{b} Major	11 – 3 – 6	B^{b}
22	B^{b} Minor	11 – 2 – 6	B^{b}m
23	B Major	12 – 4 – 7	B
24	B Minor	12 – 3 – 7	Bm

2) *Real world sample data*

The real world sample data set consist of chords actually applied to the famous nursery rhyme “Twinkle twinkle little star…” The music score shown in fig. 2 corresponds to the musical score representation of this sample song.

The real chords in fig. 2 are then adjusted by shifting the note values to match the other chords and real world sample data set is obtained. Table 4 shows the real world sample data set. Note that the real world sample data set consist of all partial inputs. That is all the chords are 2-note chords.

TABLE 4 REAL WORLD SAMPLE TRAINING DATA

	Chord Name	Note Combination	Symbol
1	C Major	1 – 8	C
3	$C^{\#}$ Major	2 – 9	$C^{\#}$
5	D Major	3 – 10	D
7	E^{b} Major	4 – 11	E^{b}
9	E Major	5 –1 2	E
11	F Major	6 – 1	F
13	$F^{\#}$ Major	7 – 2	$F^{\#}$
15	G Major	8 – 3	G
17	A^{b} Major	9 – 4	A^{b}
19	A Major	10 – 5	A
21	B^{b} Major	11 – 6	B^{b}
23	B Major	12 – 7	B

E. Recognizing Chords

The segmented MIDI file is fed into the neural network one measure at a time and result of the network for each measure is then queued in a Array-List data structure. These chords are then transformed into MIDI System Exclusive messages and transferred back to MIDI device in real time. MIDI message transformation is done according to the manufacturers standards given in [4] and final MIDI message formats are given in table 6. Message format is given in table 5.

TABLE 5 MIDI SYSTEM EXCLUSIVE MESSAGE FORMAT FOR CHORD DATA TRANSMISSION IN YAMAHA PSR-2000

Byte	Value	Description
1	F0h	System Exclusive Status Byte
2	43h	YAMAHA manufacturer ID
3	7Eh	Style Message ID
4	02h	Type (Fixed)
5	WWh	Chord root
6	XXh	Chord Type
7	YYh	Bass Note
8	ZZh	Bass type
9	F7	Endof Exclusive Message

These MIDI system exclusive messages are only applicable with Yamaha PSR-2000 model or any other compliant model from the same manufacturer.

TABLE 6 FINAL OUTPUT MIDI MESSAGES FOR EACH CHORD TYPE

	Chord	System Exclusive Message (In Hexadecimal Format)
1	C	F0 43 7E 02 31 00 31 00 F7
2	Cm	F0 43 7E 02 31 08 31 08 F7
3	$C^{\#}$	F0 43 7E 02 41 00 41 00 F7
4	$C^{\#}$m	F0 43 7E 02 41 08 41 08 F7
5	D	F0 43 7E 02 32 00 32 00 F7
6	Dm	F0 43 7E 02 32 08 32 08 F7
7	E^{b}	F0 43 7E 02 23 00 23 00 F7
8	E^{b}m	F0 43 7E 02 23 08 23 08 F7
9	E	F0 43 7E 02 33 00 33 00 F7
10	Em	F0 43 7E 02 33 08 33 08 F7
11	F	F0 43 7E 02 34 00 34 00 F7
12	Fm	F0 43 7E 02 34 08 34 08 F7
13	$F^{\#}$	F0 43 7E 02 44 00 44 00 F7
14	$F^{\#}$m	F0 43 7E 02 44 08 44 08 F7
15	G	F0 43 7E 02 35 00 35 00 F7
16	Gm	F0 43 7E 02 35 08 35 08 F7
17	A^{b}	F0 43 7E 02 26 00 26 00 F7
18	A^{b}m	F0 43 7E 02 26 08 26 08 F7
19	A	F0 43 7E 02 36 00 36 00 F7
20	Am	F0 43 7E 02 36 08 36 08 F7
21	B^{b}	F0 43 7E 02 27 00 27 00 F7
22	B^{b}m	F0 43 7E 02 27 08 27 08 F7
23	B	F0 43 7E 02 37 00 37 00 F7
24	Bm	F0 43 7E 02 37 08 37 08 F7

IV. RESULTS AND DISCUSSION

Training with the sample data set with three different learning rates (μ) and tolerance values finishes at different number of iteration as mentioned in the table 7. However, none of the combinations exceed the maximum number of iterations.

The trained MADALINE model is tested back using the same training data set. During this test, the two note inputs obtained from the real world example song, showed multiple outputs along with the correct output. However, some data sets did not give satisfactory results. Therefore, to maintain the accuracy of the model, these data were removed and the net was re-trained only with remaining data sets.

Even though this discourages the usage of real world sample data for training the neural network, it highlights the theoretical background of the chord identification. This also verifies the accuracy of the industrial approach used today to identify chords in many digital musical instruments. Most digital musical instruments require at least three notes to identify a chord correctly.

The learning rate has an affect on number of iteration required to train each neuron in the neural net. When the learning rate is small, the training becomes a precise but slow process. However, table 4 shows some contradictory results. The reason for this would be the high probability of getting into the compliance range of the target output or the tolerance level when leaning rate is small. When the learning rate is high, the output could easily miss the chance to be in the tolerance level. It could be in either much higher level or in a much lower level. Therefore it takes more iteration to converge towards the zero error.

In addition to the hard limiter function, couple of other candidate functions such as continuous, Gaussian and sigmoid are used to test their suitability in this application. However, none of these functions converged towards the zero error even within 10,000 iterations. Therefore, the hard limiter is remains as the non linearity function in this model.

The MIDI messages corresponds to identified chords are generated with the aid of a lookup table. Since each chord has a specific message as listed in table 6, using a lookup table saves a lot of calculations and eventually it speeds up the generating process.

In general, the idea of using a neural network to identify musical chords in a MIDI file can be accomplished successfully. In addition, this system is open for modifications and new additions. Many things can be done to improve the system, especially to increase the accuracy of chord identification with partial inputs.

TABLE 7 TRAINING ITERATIONS WITH DIFFERENT LEARNING RATES AND TOLERANCE VALUES

Neuron	Number of Iterations Taken For Training		
	With μ= 0.01, Tolerance=1%	With μ= 0.001, Tolerance=0.1%	With μ= 0.1, Tolerance=1%
1	542	110	362
2	1011	1047	1227
3	470	542	362
4	831	435	1119
5	650	650	506
6	1263	1335	615
7	686	506	326
8	1191	903	363
9	686	218	434
10	1119	651	687
11	290	434	506
12	1155	327	1155
13	182	362	470
14	1263	903	1119
15	398	470	542
16	1155	831	1335
17	434	326	398
18	1191	867	1239
19	578	434	542
20	903	1515	867
21	542	434	470
22	651	723	831
23	902	470	686
24	1299	651	759

V. CONCLUSION

The MADALINE model can be used to identify musical chords in a music score with a reasonably higher accuracy. Since the MADALINE approach is based on a neural network, it can take the advantage of learning and identify method rather than using only prior knowledge to identify chords. Therefore, this model can be easily extended for new additions without loosing its accuracy or efficiency.

Apart from its expendability, this model shows a capability of identify musical chords even with partial inputs. That is even if there are only two notes found in a particular time segment; this model can identify the appropriate chord most of the time. When there are multiple matching results, it eliminates the results with high error and keeps the result with lowest result. However, this method can be replaced with a fuzzy controller to decide the best chord when there are multiple candidates.

Another possible future modification would be a set of multi layer neural networks. Each ADALINE neurons can be replaced by a multi layer neural network to increase the accuracy of detecting chords. With this method, the model can be optimized to identify chords even with partial inputs.

The other major improvement can be done in segmenting the MIDI file. In this model, the MIDI file is segmented based on time intervals. But to decide exactly at which point, a chord change should occur, a fuzzy based system can be used.

ACKNOWLEDGMENT

Author would like to consider this as great opportunity to thank Dr. G. E. M. D. C. Bandara, Head of the department of production engineering, University of Peradeniya, for the great knowledge given in the neural network field and his supervisor Dr. S. R. Kodithuwakku for his guidance given throughout this research. Last but not least the authors thank goes to all of his colleagues and family members for their remarkable support given in this research.

REFERENCES

[1] Yo-Ping Huang, Guan-Long Guo, and Chang-Tien Lu, Using back propagation model to design a MIDI music classification system, 2004.

[2] Barbara Tillmann, Hervé Abdi, and W. Jay Dowling, Musical style perception by a linear auto-associator model and human listeners, 2004

[3] Stamatios V. Kartalopoulos, Understanding neural networks and fuzzy logic – basic concepts and applications, IEE press, 1996

[4] Yamaha Corporation, Yamaha PSR-2000/PSR-1000 data list, 2001