APPLY NEURAL NETWORKS TO MORSE CODE RECOGNITION

Cheng-Hong Yang, Cheng-Huei Yang[1], Yuan-Long Jeang, Gwo-Jia Jong, and Tsong-Yi Chen

Dept of Electronic Engineering, National Kaohsiung University of Applied Sciences
[1]Dept of Electrical Engineering, National Cheng Kung University

ABSTRACT

Computer has been successfully used as a tool in a variety of fields. However, most applications of computer products are designed for able persons, and are inaccessible to the disabled. Since the unadapted computer keyboard is not a suitable communication tool for physically disabled persons with such disabilities as muscle atrophy, cerebral palsy, and severe handicap. In this paper, the Morse code is selected as a communication adaptive device for entering Mandarin chanjei symbols into a computer. Although a stable typing rate is strictly required for an accurate recognition of Mandarin chanjei Morse code, maintaining this rate is a challenge for the disabled. Therefore, a suitable adaptive automatic recognition method is needed. The method presented here is divided into five stages: space recognition, tone recognition, learning process, adaptive processing, and character recognition. Experimental results revealed that the proposed method generated high recognition rate.

Keywords: Morse code, Adaptive signal processing, Mandarin chanjei symbols, Neural Networks.

1. INTRODUCTION

As technological advances make dramatic gains, these adaptive tools, combined mainly with computer software and hardware, will gradually play a more important role in the lives of the disabled. Unfortunately, the traditional computer keyboard cannot be a useful communication tool for physically disabled persons. Consequently, many computer assisted key-in systems have been developed for the disabled to overcome this barrier, e.g., the head mouse, mini-keyboard, king-keyboard, trackball, joystick, alternative keyboard, keyguard, and touch screen [1]. Also, many researchers have focused on a reduced set of switches for these input devices with an efficiency rate approaching one key press per selected character. To help persons whose hand coordination and dexterity are impaired by disabilities such as muscle atrophy, cerebral palsy, or other severe handicaps, a substitute keyboard is needed. The most applied widely adaptive tool, Morse code, has been shown to be an excellent candidate for this communication adaptive device [2-8].

Morse code is very simple and can be transmitted by using just a single switch. It can be very useful, under circumstances where a disabled person still retains good hand coordination and is able to operate a single switch. To permit the disabled persons to use Chinese version computer applications, we successfully edited a new Chinese chanjei Morse code [9]. To accurate Morse code recognition, a stable typing rate is strictly required. However, this restriction is a major hindrance to disabled persons, especially for the most severely physically-disabled. Therefore, a suitable adaptive automatic recognition method of the keyed-in Morse code is needed. In this paper, an advanced recognition method, which combines the variable degree variable step LMS algorithm [10] with the learning vector quantization (LVQ) method is suggested to increase prediction power. The results show that the proposed method provided high recognition rate.

2. METHODS

Based on the definition of Morse code, the tone ratio (dot to dash) has to be 1:3. This means that the duration of a dash is required to be three times that of a dot. In addition, the silent ratio (dot-space to character-space) also has to be 1:3. Unfortunately, it is a difficult task to maintain these precise intervals, especially when a user suffers from a physical disability and is unable to control body parts reliably [11, 12]. In fact, a Morse code time series is generally unstable in speed and/or in rate. Two major methods have been proposed in the literature to solve this problem. One method (SAM) is designed to increase the precision of Morse code [11], and the other is an automatic Morse code key-in system [12, 13]. In SAM, the users receive audible feedback for each dot and dash attempted to be transmitted. Basically, this technique is accepted in the training of able persons, but unsuitable for some disabled persons. Two switches (or a single-pole double-throw center-off switch) are used in the automatic Morse code key-in system. A dot or a dash is indicated by one of the two switches (or positions). When a switch is held down, it repeats the dot or the dash for every time unit. If both switches are held up for a period of time that is longer than one time unit, a space is generated. By using this technique, the time unit is required to be controlled manually. The test results for these two methods revealed that the silent intervals were much more difficult to master than the tone intervals. Thus, neither SAM nor the automatic key-in system can generate precise silent intervals.

The Morse code recognition method proposed in this paper is divided into five stages: space recognition, tone recognition, learning process, adaptive processing, and character recognition. A block diagram of the Morse code recognition process is shown in Fig. 1. Initially, the input data stream is sent individually to either tone recognition or space recognition depending on switch-down time (tone element) or switch-up time (space element). In tone recognition, the tone element value is first recognized as either a dot or a dash, and then sent to the learning process, which is used to recalculate weights and node thresholds. Simultaneously, in the tone buffer section, the recognized tone element (dot or dash) and each successive tone element are saved in a dot-dash buffer and a tone element buffer. Next, in the space recognition stage, the space element value is recognized as being either a dot-dash space or a character space. If a character space is obtained, then the value(s) in the tone buffer is (are) sent to character recognition. To account for a variable switch-down and switch-up speed, both, the space element value and tone element value, have to be adjusted. If the space element value is recognized as a character space, it is divided by a constant (3.0) before being fed into the adaptive processing stage. Otherwise, the space element value feeds directly into the adaptive processing stage. The tone element value is sent into the tone base adjustment. Once this occurs, the character can be identified in the character recognition.

n: the total number of Morse code elements in character x_i.
$m_j(x_i)$: a dot or dash recognized from $e_j(x_i)$.
$s_j(x_i)$: a character space or dot-dash space recognized from $b_j(x_i)$.

2.1 Space Recognition

The space recognition stage is employed to detect the spaces existing between whole characters as well as the space between isolated Morse code elements which comprise a unique character. Thus, if a data stream of characters composed of Morse code elements is entered, these elements must then be identified as being either spaces between whole characters or spaces between isolated elements of a character. S_1 is the initial silent_base value.

Followings, the procedure for this character detection operation is shown:

1. initiate j=1.
2. if $b_j(x_i)$ < silence_base , then go to step 3, otherwise go to step 4.
3. $b_j(x_i)$ is a dot-dash space. Let j=j+1 and go to step 2.
4. $b_j(x_i)$ is a character space. Then a sequence of tone durations between the character space is obtained. Go to step 1.

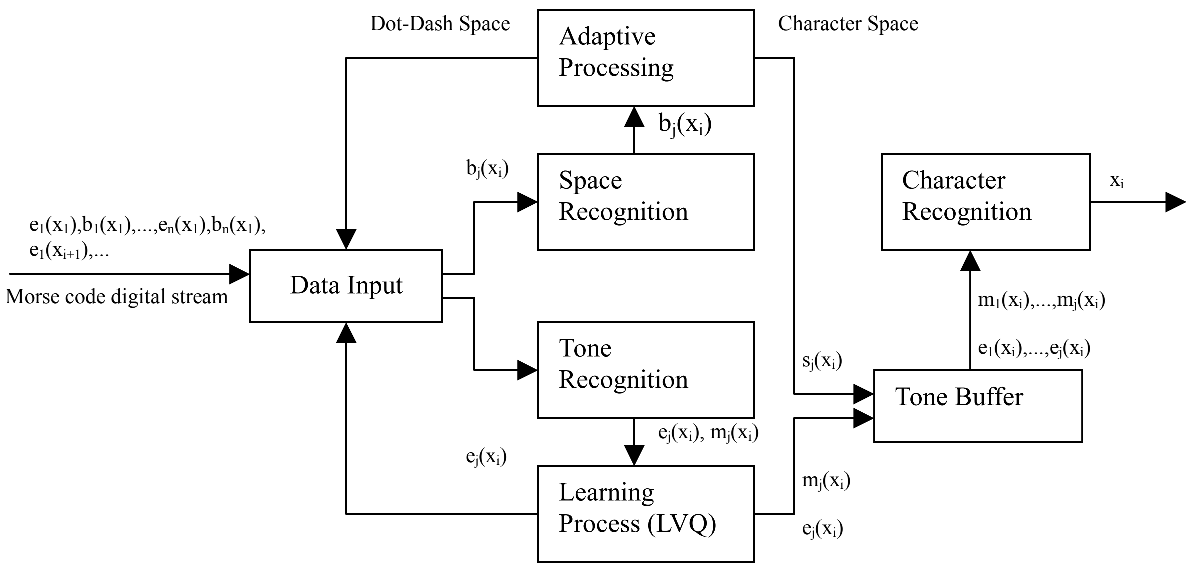

Fig. 1. Block diagram of the Morse code recognition system.

A Morse code character, x_i, is represented as follows:

$$e_1(x_i),\ b_1(x_i),\ \ldots,\ e_j(x_i),\ b_j(x_i),\ \ldots,\ e_n(x_i),\ b_n(x_i)$$

where

$e_j(x_i)$: *j*th tone duration in the character x_i.
$b_j(x_i)$: *j*th silent duration in the character x_i.

Unfortunately, the first character, x_i, cannot be immediately isolated because of the absence of an initial value S_1. Subsequently, the initial space length S_1 is obtained by extracting the first nine values of silent elements entered as reference values; afterward, all values taken are arranged in descending order, and the relationship among each value is then compared. If a value is found to be twice larger than any other value, this value is designated as being long (L), and the smaller values are designated as being short (S). Once all relationships have been established, the average of the nine reference values can be calculated and assigned to the initial space length, S_1. An illustration of this process is given below.

Let's assume, for example, that the following Morse code digital stream has been received: 542 287 244 254 327 196 142 2532 596 1143 211 1353 175 437 831 384 867 367 753 1344 677, in which odd position data are defined as tone elements while even position data, underlined, are defined as silent elements. The first nine silent values are arranged, in descending order, as follows: 2532, 1353, 1143, 437, 384, 367, 287, 254, and 196. After sorting, the first three values (2532, 1353, and 1143) are designated as L (since they are at least twice larger than the followings values) and the rest are designated as S. Afterward, the sum of long silent values (L) is divided by 3, and the sum of short silent values (S) is calculated. S_1 is the average value of the sum of long and short values. This process is illustrated by the following equation:

$$S_1 = (\text{sum of long}/3 + \text{sum of short})/\text{number of elements}$$

$$S_1 = [(2532+1353+1143)/3 + 437+384+367+287+254+196]/9 = 400.11$$

Once the initial S_1 value has been determined, it can be sent into the adaptive processing stage as the initial value of x_1. Meanwhile, the character detection equation can be used to calculate a subsequent S_1 value based on this obtained S_1 value to recognize spaces within elements. After a space element has been recognized, the S_1 value can be recalculated. If the result shows L, the space element is divided by 3, and the obtained value is only then sent into the adaptive processing; otherwise, the space element is directly sent into the adaptive processing stage to obtain a new S_1. Whenever an S_1 is obtained, the data stream is separated into elements and spaces. After the Morse code elements of a character have been isolated from a data stream, the elements can be recognized in the character recognition stage. The next step is to look at how tones are detected and processed.

2.2 Tone Recognition

Initially, the largest (tone_max) and the smallest (tone_min) values of the first nine data are taken as parameters of the range of the data stream. Then, based on this range, the data is treated by normalization to obtain an input value within a range of 0 to 1.

$$\text{tone\_float} = (\text{tone element value} - \text{tone\_min}) / (\text{tone\_max} - \text{tone\_min})$$

The obtained value, tone_float, and the target output value (based on dash and dot) are then sent into the neural network to be learned. These processes are repeated until a user-defined accuracy is reached, i.e. the error is acceptably low. In this study, the learning process is considered completed when the difference between the learning recognition output value and target output value is less than 0.0001. Moreover, the initial tone_base value (to be used in the character recognition stage) has to be determined. The same first nine data are selected again to perform this calculation. From the target output value, each data element can be recognized as being either dot or dash. The initial tone_base value is calculated as follows:

$$\text{tone\_base} = [(\text{sum of dash data} / 3) + \text{sum of dot data}] / \text{number of elements}$$

After being learned in the neural network, the data to be recognized are treated using the same process. The values are then sent into the neural network to be recognized and to determine whether the recognized value is a dash or dot. If the recognized value is a dash, it is compared with the tone_max value. If the dash is revealed to be larger than the tone_max value, then the tone_max value is substituted by the recognized value. If, however, the incoming value is more than 1.5 times higher than tone_max value, the incoming value is ignored in order to prevent that outlying values erroneously influence the recognition process. If the recognized value is a dot, it is compared with the tone_min value, and the tone_min value is then changed to the recognized value. In addition, the tone_base value has to be recalculated. If the tone element is recognized as a dash, then the tone_base value is reset to the average value of one third of the recognized value plus the old tone_base value. If the tone element is recognized as a dot, then the tone_base value is reset as the average value of the recognized value plus the old tone_base value. These procedures are repeated until all of the data have been recognized.

As presented in the previous example, the values of dash elements are: 542, 327, 596, 831, 876, 753, 677; dot values are: 244, 142, 211, 175. Based on these values, the tone_max value is 876, and the tone_min value is 175, the first data value is 542. Following the normalization treatment, the tone_float value is obtained.

$$\text{tone\_float} = (542\text{-}175 / (876\text{-}175) = 0.524$$

This value, 0.524, and the target output value are sent into the neural network to be learned, and the learning process is repeated until all of the data have been learned. The value of the initial tone_base is thus:

$$\text{tone\_base} = [(542+327+596+805+876)/3+244+142+211+175]/9 = 203.26$$

Once the learning process is finished, the recognition process can be initiated. The recognition process is similar to the learning process, except that the target output value does not need to be sent into the process. Instead, it is taken directly to predict the output value for the comparison to be used to obtain the recognition result. The example from above illustrates this process. As before, the first data value received is 542. Following the normalization treatment, the tone_float value is calculated to be 0.524. Then, this value is sent into the neural network to be recognized. The result shows that the first data, 542, is a dash since it is smaller than the tone_max value. The tone_max value does not change although the tone_base has to be reset to 192 (= [203+(542/3)]/2) in this example.

2.3 Character Recognition

According to the definition of Morse code, the tone ratio (dot to dash) must be 1:3. Thus, all characters in the Morse code table can be represented by a combination of the numbers 1 and 3, signifying a dot or a dash. As shown in Fig. 2, '金' can be encoded as '.---' or (1, 3, 3, 3).

日	月	金
• •	• • •	• — — —
(1, 1)	(1, 1, 1)	(1, 3, 3, 3)

Fig. 2. A simple representation of Morse code table characters

Now that we have a means for easily translating Morse code elements, it is time to see how characters are recognized. Once a character space value has arrived in the tone_buffer, it is a signal for the tone buffer elements to be sent to character recognition. If the recognized character set can be directly matched to a code set from the Morse code table, then it is immediately translated from the Morse code table. Otherwise, it has to be translated by the following minimum distance calculation. First, each tone element value in an unknown tone element stream is divided by the tone_base of the previous tone element set. Then, the distances between each tone value and the code elements in each character of the Morse code table are calculated. The character with the minimum Euclidean distance to the tone value is chosen as the value for the unknown character. The procedure for the shortest Euclidean distance method is the following. First, each tone element, $e_j(x_i)$, is divided by the tone_base, $e_j(x_i)$/tone_base, for j=1~n. Then, the roots of the sum of the square distances between the new tone element, $e_j(x_i)$, and the character in the Morse code table are calculated. The character in the Morse code table that has the shortest Euclidean distance is recognized as the unknown character. Assume, for example, the tone element values in an unknown character are the following: 234, 756, and 212, in which the tone_base is 247. After division by the tone_base, the three tone elements are 0.95, 3.06, and 0.86. The character with the shortest Euclidean distance in the chanjei Morse code table is character '中' (i.e., 1, 3, 1). Thus, character '中' is selected as the Morse code match for the unknown character.

2.4 Adaptive Processing

The variable degree variable step LMS (VDVSLMS) algorithm used in the system presented here serves to change the standard 'space' length [10]. The average of space $b_j(x_i)$ (j = 1, n-1) in x_i is the ith input data of the algorithm. The VDVSLMS algorithm utilizes the current data to compute a new weight vector using the weight update recursion of the standard LMS algorithm with step size μ. The new weight vector together with the current data are then utilized to update again the desired weight vector using the standard LMS algorithm weight update recursion with step size μ. Each adaptive weight, $W(n)$, is adjusted according to the equation

$$W(n+1) = W(n) - \alpha_2(n)\hat{\nabla}(n)$$

where

$$\alpha_2(n) = 2\mu(1 - \mu X^T(n)X(n))$$

The subscript on the α(n) is used to indicate the degree, and

$$\hat{\nabla}(n) = -2\varepsilon(n)X(n)$$

is an estimate of the gradient

$$\varepsilon(n) = d(n) - X^T(n)W(n)$$

where $d(n)$ is the scalar desired signal. μ is the step-size parameter that controls the speed of convergence as well as the steady-state and/or tracking behavior of the adaptive filter. The step size μ has a value of 0.02 in our system.

2.5 Learning Vector Quantization Method

In this paper, we propose a network structure to be a fully connected learning vector quantization (LVQ) method. There are two input nodes, one hidden layer (eight nodes), and two output nodes in total. The learning process of learning vector quantization neural networks is formalized in the following processes: the learning process and the recall process [14]. In the learning process, iterations are repeated until convergence in terms of the selected error criterion is reached. The error criterion

$$RMSE = \sqrt{\frac{\sum_j (T_j - O_j)^2}{N}}$$

which is used, the root mean square error (RMSE), is defined as follows:

where T_j is the target output activation, O_j is the actual output activation at output unit j, and N is the number of output nodes.

In the recalling process, input data calculations are repeated until all of the data have been processed. The input layer, output layer, and the classification of this system are shown in Table 1. If the output result shows $Y_1 > Y_2$, the recognized value is a dot, otherwise ($Y_2 > Y_1$), it's a dash.

Table 1. Input-output relationship table

Input X_1	Input X_2	Target output Y_1	Target output Y_2	Result
1	0	0	1	Dash
1	1	1	0	Dot

3. EXPERIMENTAL RESULTS AND DISCUSSION

Two groups of expert testing data, EXP1 and EXP2, were tested in order to investigate the efficiency of the proposed method. EXP1 testing data, number from Exp101 to Exp115, are collected from 15 abled peoples who are trained for a long period of time by typing 100 identical characters. EXP2 testing data, numbered from Exp201 to Exp215, are collected from 15 experts in the military wireless service by typing 100 identical characters. The experimental results are shown in Table 2. The average number of matches for the EXP1 and EXP2 are 88.47 and 90.73, respectively. As it was expected, the experts showed a little higher number of matches than the nonexperts. The experimental results indicated that the different initial S_1 turned into different recognition rate.

Table 2. The recognition result for two types of test problems.

Problems	Number of matches	Problems	Number of matches
Exp101	92	Exp201	95
Exp102	91	Exp202	97
Exp103	94	Exp203	95
Exp104	93	Exp204	94
Exp105	86	Exp205	83
Exp106	87	Exp206	85
Exp107	90	Exp207	92
Exp108	87	Exp208	93
Exp109	88	Exp209	88
Exp110	85	Exp210	96
Exp111	86	Exp211	86
Exp112	88	Exp212	89
Exp113	91	Exp213	90
Exp114	83	Exp214	89
Exp115	86	Exp215	89
Average	88.47	Average	90.73

The incorrect recognition might be generated in two main errors: character separation errors and character recognition errors. If the space between 'dot' and 'dash' within a character has unusual longer length, that will be mistaken as the space between characters. Once an incorrect character separation is generated, the character will be split into two characters so that the character recognition will be split into two characters so that the recognition will be affected.

The character recognition error is due to the typist's personality. If the typing speed is unstable, such as longer or shorter than the standard length, a character will be mismatched in the recognition. Usually, every one has his own typing speed. The system should provide adequate adjustment for the length of dot or dash. Because one types for a long period of time, his typing might cause errors by wearies. For example, it begins with 300ms to 100ms for the length of dash to dot, but it might change to 900ms to 300ms after a long period of typing. However, according to experience, a person's typing rate is generally constant over a short period, the person's present typing rate is similar to the typing rate of the previous several words. Therefore, in order to increase the recognized rate, the tone code element in the Morse code table has to be adjusted by a format which is designed for the individual. In addition, the adjustment should be based on the previous typing speed. It means that tone length has to be renew after each character has been recognized.

In this study, the defect of the new developed method is only adjusting space values and sometimes it produced some mistakes during the adaptive process. Thus, to have better performance, more efforts and adjustment should be considered in the process, such as in addition to modify space values, tone values should be adjusted within the adaptive process. The process to modify tone values might use statistic method or similar method as the adjustment of space. Either of these two methods should provide better results.

4. CONCLUSIONS

Morse code is a simple, speedy, and low cost communication method using a series of dots, dashes, and intervals with which each character can be translated into a predefined sequence of dots and dashes (the elements of Morse code). A stable typing rate is strictly required for Morse code to be used effectively as a communication tool. However, this restriction is a major hindrance to disabled persons, especially for persons who suffer from a severe physical handicap. Therefore, a suitable adaptive automatic recognition method is needed. The method was applied to 30 test problems. Experimental results showed that the proposed method obtained great recognition rate. In the future study, we expect to apply this method to the people with physical impairment.

ACKNOWLEDGEMENTS

This work was supported by the National Science Council, R.O.C., under contract NSC 88-2614-E-151-001.

REFERENCES

[1] R. Bower, *et al.* (Eds), The Trace Resourcebook-Assistive Technology for Communication, Control, and Computer Access, Trace Research & Development Center, Universities of Wisconsin-Madison, Waisman Center, 1998.

[2] D.K. Anson, Alternative Computer Access: A Guide to Selection, F.A. Davis, Philadelphia, 1997.

[3] C.-H. Luo and C.-H. Shih, Adaptive Morse-coded single-switch communication system for the disabled, Int. J. of Biomed. Comput. 41, pp. 99-106, 1996.

[4] C.–H. Shih and C.-H. Luo, A Morse-Coded recognition system with LMS and matching algorithms for persons with disabilities, Int. J. of Medical Informatics 44, pp. 193-202, 1997.

[5] S.P. Levine, J.R.D. Gauger, L.D. Bwera, and K.J. Khan, A comparison of mouth stick and Morse code text inputs, AAC augmentative and alternative Communication 2, 51, 45-51, 1996.

[6] L.N. Goble, and H.A. Colle, High-speed Morse code training, Proceedings of the IEEE 1985 National Aerospace and Electronics Conference, NAECON, 944-951, 1985.
[7] D.W. Lywood and J.J. Vasa, Computer-terminal operating and communication aid for the severely handicapped, Medical and Biological Engineering, 12, pp. 693-695, 1974.
[8] D.A. Shannon, W.S. Satewen, J. Miller, and B.S. Cohen, Morse-code controlled computer aid for the non-vocal quadriplegic, Medical Instrumentation, 15, pp. 341-343, 1981.
[9] Yang, Cheng-Hong, Developing a New Chinese Chang Jei Morse Code for the Disabled Persons, Chinese Journal of Medical and Biological Engineering, Vol. 18, No. 3, 189-194, 1998.
[10] M.A. Khasawneh. and K.A. Mayyas, A Newly Derived Variable Degree Variable Step Size LMS Algorithm, Int. J. Electronics, Vol. 79, No. 3, pp. 255-264, 1995.
[11] L.T. Hauck, SAM: an improved input device, Johns Hopkins APL Technical Digest 13, pp.490-493, 1992.
[12] J.J. French, F. Silverstein, and A.A. Siebens, An Inexpensive Computer Based Morse Code Communication System, RESNA ’86, pp.259-261, 1986.
[13] R. Trace and D. Center, Two Switch Autorepeat Morse Code, Waisman Center, University of Wisconsin-Madison, 1984.
[14] F. Limin, Neural Network in Computer Intelligence, McGraw-Hill, New York, 1994.