Published in the International Symposium on Artificial Neural Networks, National Cheng King University, Tainan, Taiwan, December 15-17, 1994

Experiments on Chinese Character Recognition Using MVL Neural Models

Dr. C.J.Wang, Dr. V. Callaghan, K.H.Tang and J.S.Han
Department of Computer Science, University of Essex, Colchester CO43SQ, UK

E-Mail: cwang@essex.ac.uk, vic@essex.ac.uk

This paper reports on the results obtained from experiments on the application of Multiple Valued Logic (MVL) neural models to the recognition of handwritten Chinese characters. The MVL neural models investigated in this work are based on those proposed by Watanabe et-al 1990. It is shown that the recognition rate of the original model is too low to be useful for realistic applications. Improvements to the original model are proposed and experiments are described which show these modifications enable recognition rates to be increased from 17% to over 70% (or 95, if multiple choices are allowed. It is also shown that the modified model exhibits fast learning.

I. Introduction

With the advance of the electronic tablet technology the on-line character recognition is receiving an increasing amount of scientific and commercial attention. It provides an alternative means of input to the traditional keyboard which was principally designed for languages based on Latin characters. For a language that retains strong features of pictographic characters such as Chinese, inputting via a traditional keyboard is an extremely tedious task. Currently, the methods most commonly used for inputting Chinese characters are *pingying* (pronunciation) and *five-stroke* (encoding a character by a sequence of up to five gestures), both of which can be adapted to the traditional keyboards. The *pingying* input requires a speaker who is highly proficient in the Mandarin dialect and involves choosing from several to dozens of homonyms, whilst the user of the *five-stroke* method has to be trained to master with fluency the often ambiguous gesture encoding for thousands of Chinese characters. With both of these methods the aesthetics of Chinese handwriting as well as the user's personal style are lost. Also, they are difficult to use for the majority of Chinese speakers.

Tapper et al (1990) give a comprehensive description of the commercial products for handwriting input which are mostly based on feature or template matching, with speeds of the order 1 to 10 characters per second. Recognition rates, as cited in the advertising literature, usually exceed 95%. However, as they point out, these rates can only be achieved using careful writing produced by cooperative users. It is concluded that most of these systems are sufficiently fast to keep up with hand-writing, as Chinese characters, particularly those with many strokes, take a second or more to write.

However, two questions emerge that have not been properly addressed. First of all, what is the recognition rate that an *ordinary* user can achieve? Secondly, is it true that the recognition speed is sufficient for *ordinary* users? By the *ordinary* user, we mean a native Chinese written language user who is writing, for instance, a diary or a letter to friends or who is drafting a speech. The distinct features of ordinary users are the simplification of complicated characters and personalized and well practised gestures, both of which lead to a much faster pace. As far as such ordinary users are concerned, the existing commercial systems leaves much to be desired.

The work described in this paper is one in a series of attempts seeking simple yet effective techniques that can support an on-line pen user interface, allowing ordinary users to input handwriting Chinese characters. In order to be viable, such techniques should have a high recognition rate for free-style handwriting. Several techniques reported in the recent publications which are claimed to be capable of recognizing handwritten characters have been evaluated against more realistic situations than that quoted in the original investigation. The experiments on using the Multiple-Valued Logic Neural Networks (MVL Neural Model) for on-line handwriting Chinese characters are reported.

II. The MVL Neural Model

The MVL neural model being investigated is proposed by Watanabe et al (1990). It is reported that in solving the Xor problem, the MVL neural network is faster in training than perceptrons (Widrow & Winter 1988). It is also shown

Published in the International Symposium on Artificial Neural Networks, National Cheng Kung University, Tainan, Taiwan, December 15-17 1994

that a single MVL neuron may have non-linear separation of pattern space. Watanabe and Matsumoto (1992) further report that the multiple layer MVL neural net is capable of recognizing translated characters. Below is the definition of the MVL neuron extracted from Watanabe et al 1990. Figure 1 shows the configuration of a MVL neuron, where

x_i	– the ith input signal, $x_i \in \{-1, 0, 1\}$;
w_{pi}	– the weight for x_i when $x_i > 0$, $w_{pi} \in [0, R]$;
w_{ni}	– the weight for x_i when $x_i < 0$, $w_{ni} \in [0, R]$;
y	– the analog response of the neuron, $y \in [0, R]$;
z	– the output of the neuron, $z \in \{-1, 1\}$;
$\wedge$	– logical product, $x \wedge y = \min(x, y)$;
$\vee$	– logical sum, $x \vee y = \max(x, y)$;
h	– threshold;
e	– error; and
SGN	– sign function.

The output of the MIN MVL neuron is determined by the following equations:

$$y = (x_1 \cdot \alpha_1) \wedge (x_2 \cdot \alpha_2) \wedge \ldots \wedge (x_N \cdot \alpha_N) \tag{1}$$

where

$$\alpha_i = \begin{cases} w_{pi} & \text{for} \quad x_i > 0 \\ -w_{ni} & \text{for} \quad x_i < 0 \end{cases} \tag{2}$$

and $(x_1 \cdot \alpha_1) = w_{pi} \wedge w_{ni}$ for $x_i = 0$.

The threshold is the average of the maximum and minimum weights,i.e.

$$h = [\max(w_{si}) + \min(w_{si})] / 2 \tag{3}$$

where w_{si} represents both w_{pi} and w_{ni}.

In the training procedure, weights are modified by the following rules:

$$w_{pi}(k+1) = w_{pi}(k) + C \cdot w_{ni}(k) \cdot [d + (1 - z)/2] \cdot d \quad \text{for } x_i(k) = d \text{ and } w_{pi}(k) < R \tag{4}$$

$$w_{ni}(k+1) = w_{ni}(k) \cdot \{1 - C \cdot [d + (1 - z)/2] \cdot d\} \quad \text{for } x_i(k) = d \text{ and } w_{pi}(k) \geq R \tag{5}$$

$$w_{ni}(k+1) = w_{ni}(k) + C \cdot w_{pi}(k) \cdot [d + (1 - z)/2] \cdot d \quad \text{for } x_i(k) \neq d \text{ and } w_{ni}(k) < R \tag{6}$$

$$w_{pi}(k+1) = w_{pi}(k) \cdot \{1 - C \cdot [d + (1 - z)/2] \cdot d\} \quad \text{for } x_i(k) \neq d \text{ and } w_{ni}(k) \geq R \tag{7}$$

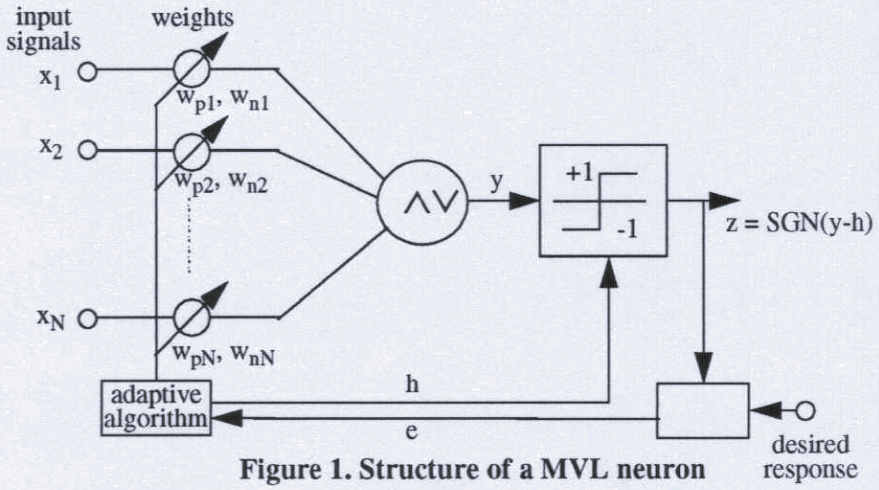

Figure 1. Structure of a MVL neuron

Published in the International Symposium on Artificial Neural Networks, National Cheng Kung University, Tainan, Taiwan, December 15-17 1994

where k is the discrete time index or training cycle number, d is the desired response and C is a learning constant. In Watanabe and Matsumoto 1992, the term '$[d+(1-z)/2]\cdot d$' is changed into '$[d+(1-z)/2]\cdot d$' and the relation '$\geq$' in the conditions of equations (5) and (6) is changed into '='. We shall refer to these two versions of MVL neural model as MVLN-1 and MVLN-2 respectively.

III. The Experiments on MVLN-1 and MVLN-2

A set of one hundred Chinese characters have been chosen in terms of their complexity, going from 1 stroke to 22 strokes. A collection of 12 samples of theses characters, written four times each by three persons, are used for the evaluation experiments. These experiments used three samples of each persons handwriting as training data; test were carried out on the fourth sample. The results of experiments on MVLN-1 is generally unsatisfactory, despite varying the weight range R, the learning rate C, and optimizing the bit map size, which is why a 12×12 is chosen, to achieve the best recognition rate. The Experiment on MVLN-2 is effectively the same. Although the performance varies slightly from each run, there is no significant difference between these two models. The best result is a 17% recognition rate, achieved on one person's handwriting samples which appear to be written fairly canonically. Cross person's recognition rate is effectively zero. However, the following findings should be noted:

1. Once the weight range is sufficiently large, a further increase of the range doesn't help the recognition rate but simply prolongs the training phase. Experiments with R=40, 60, 100, 180 and 256 show little difference in the recognition rate. However, the number of training cycles increased from an average of 118 to 758 over hundreds of runs;
2. The MVL neural model is sensitive to the learning rate. Experiments with C≥0.4 all failed to converge, and those with C= 0.1, 0.2 and 0.25 have no significant difference in the number of training cycles to converge, though it is slightly faster when C=0.25 than C=0.1.

Analysis of MVLN-1 and MVLN-2

There are essentially two problems in these two MVL neural model that contribute to the above findings. First of all, an MVL neuron is activated on a very strong condition of the input pattern matching the weight setting. This can be seen from the equations (1), (2), (3) and z=SGN(y-h), which state that a neuron will fire if and only if all the input signals fall in the direction of the stronger connection. For instance, if $w_{pi} > w_{ni}$, then x_i has to be 1 and if $w_{pi} < w_{ni}$, then x_i must be -1, for all i. The chance that a neuron fires when any of its input signals falls in the other direction is extremely slim. In fact, it is practically impossible because the learning algorithm tends to bring w_{pi} to R and w_{ni} to 0 if x_i is mostly 1, or the other way round if x_i is mostly -1. This trend of polarizing the weights when the network converges explains why a small range is as good as a large range. It facilitates no error tolerance – a neuron will only fire when the input pattern is exactly the same as the weights dictate and never fire on similar input patterns. Hence, the low recognition rate.

The second problem is the *swinging effect* of the learning algorithm. The weight adaptation rules as expressed in equations (4) to (7) can be expanded into 8 cases as shown in Table 1. Referring to Table 1, cases 1 and 2 indicate no weight adaptation since the neuron's response z is the same as the desired one which is inactive; cases 7 and 8 indicate that although the neuron's response is the same as the desired one which is active, the weights are still

Cases	d	z	x_i	Δw_{pi} (Condition)		Δw_{ni} (Condition)	
1	-1	-1	-1	0	($w_{pi} < R$)	0	($w_{pi} \geq R$)
2	-1	-1	1	0	($w_{ni} \geq R$)	0	($w_{ni} < R$)
3	-1	1	-1	Cw_{ni}	($w_{pi} < R$)	$-Cw_{ni}$	($w_{pi} \geq R$)
4	-1	1	1	$-Cw_{pi}$	($w_{ni} \geq R$)	Cw_{pi}	($w_{ni} < R$)
5	1	-1	-1	$-2Cw_{pi}$	($w_{ni} \geq R$)	$2Cw_{pi}$	($w_{ni} < R$)
6	1	-1	1	$2Cw_{ni}$	($w_{pi} < R$)	$-2Cw_{ni}$	($w_{pi} \geq R$)
7	1	1	-1	$-Cw_{pi}$	($w_{ni} \geq R$)	Cw_{pi}	($w_{ni} < R$)
8	1	1	1	Cw_{ni}	($w_{pi} < R$)	$-Cw_{ni}$	($w_{pi} \geq R$)

Table 1. The change of weight in each learning cycle for MVLN-1

Published in the International Symposium on Artificial Neural Networks, National Cheng Kung University, Tainan, Taiwan, December 15-17 1994

adapted to reinforce this response; and case 3 to 6 indicate the weight adaptation to change the neuron's response to the desired value. As can be seen, the amount of the weight modification is proportional to the value of the weight. This lead to an uneven pace of weight changes. In addition, the weight adaptation rules in equations 4 and 6 define that the amount of change in weight w_{pi} is proportional to the value of w_{ni} and vice versa. These rules cause the weights whose values are close to the maximum value R to swing up and down in the learning phase, which could lead the network not to converge if the constant C is relatively large. This explains why the MVL neural model is sensitive to the learning constant C as observed in our experiments. It contradicts the intuition that the learning pace should slow down as the network approaching the convergence state.

IV. The MVLN-3

The MVLN-3 model improves on the MVLN-1 and MVLN-2 by two features. First, the learning algorithm is revised to stabilize the weight adaptation. Second, the activation of the neuron allows a controlled error tolerance.

The Learning Algorithm

The learning algorithm of the MVLN-3 is expressed in equations (8) to (11). This learning algorithm is more in line with the principle of the perceptron's learning rule (Widrow & Winter 1988), which the original MVL neural model is based on (Watanabe et al 1990).

$$w_{pi}(k+1) = w_{pi}(k) + CR[(d - z)/2]x_i(k) \quad \text{for } x_i(k) > 0 \text{ and } 0 < w_{pi}(k) < R \tag{8}$$

$$w_{ni}(k+1) = w_{ni}(k)\{1 - C[(d - z)/2]x_i(k)\} \quad \text{for } x_i(k) > 0 \text{ and } w_{pi}(k) = 0 \text{ or } R \tag{9}$$

$$w_{ni}(k+1) = w_{ni}(k) + CR[(d - z)/2]x_i(k) \quad \text{for } x_i(k) < 0 \text{ and } 0 < w_{ni}(k) < R \tag{10}$$

$$w_{pi}(k+1) = w_{pi}(k)\{1 - C[(d - z)/2]x_i(k)\} \quad \text{for } x_i(k) < 0 \text{ and } w_{ni}(k) = 0 \text{ or } R \tag{11}$$

The effect of the weight adaptation, as shown case by case in Table 2, truly conforms to the intuitive explanation of Watanabe et al (1990), which states:

> *The principle of the above equations is that, according to the desired response d, the weights $w_{pi}(k)$ and $w_{ni}(k)$, which correspond to the inputs $x_i(k)$ with sign + and -, respectively, are to be adjusted with the coefficient C. If a weight having already the maximum value R has to be increased, then the weight on the opposite side is decreased.*

Therefore, we believe that equations (8) ~ (11) should be one formulation implied by Watanabe et al's original proposal.

The Error Tolerant Activation

In recall, a confidence level is given, which determines the percentage of input signals that must fall in the desired direction for the neuron to be activated. This is realized by counting the qualified input signals. A signal x_i is qualified if its associated weight is greater than or equal to the threshold h as defined in equation (3).

Cases	d	z	x_i	Δw_{pi} (Condition)		Δw_{ni} (Condition)	
1	-1	-1	-1	0		0	
2	-1	-1	1	0		0	
3	-1	1	-1	Cw_{pi}	$(w_{ni} = 0)$	-CR	$(w_{ni} < R)$
4	-1	1	1	-CR	$(w_{pi} < R)$	Cw_{ni}	$(w_{pi} = 0)$
5	1	-1	-1	$-Cw_{pi}$	$(w_{ni} = P)$	CR	$(w_{ni} < R)$
6	1	-1	1	CR	$(w_{pi} < R)$	$-Cw_{ni}$	$(w_{pi} = R)$
7	1	1	-1	0		0	
8	1	1	1	0		0	

Table 2. The change of weight in the improved MVL neural model

Published in the International Symposium on Artificial Neural Networks, National Cheng Kung University, Tainan, Taiwan, December 15-17 1994

Let M ∈ [0, 1] be the given confidence, z = SGN(K - M), where

$$K = \frac{1}{N_i}\sum_{i=1}^{N} SGN(x_i\alpha_i - h) \qquad (12)$$

and α_i is as defined in equation (2). With this activation mechanism, a controllable error tolerance can be facilitated.

Results of the Experiment on MVLN-3

In order to compare the performance between MVLN-1 and MVLN-3, the same experiments are conducted, which not only use the same data set but also use the same parameters and system configuration such as the 12×12 bit map to represent a character etc. The results of the experiments on MVLN-3 with confidence of 0.92, 0.95 and 0.97 are listed in Table 3 in comparison with those of MVLN-1. The figures given for the MVLN-3 recognition rate consist of a triple n_1:n_2:n_3 where n_1 is the percentage of exact match, n_2 is the percentage of the second match, and n_3 is the percentage of the third or more match. The observations of the experimental results are discussed as below.

The Effect of the Revised Learning Algorithm

The experiments show that the weight adaptation in MVLN-3 is stable. It converges quickly with the high learning rates with which MVLN-1 has failed. In all the experiments conducted, the number of learning cycle remains a constant for each individual training example, which of course depends on the learning rate C and the confidence level. The *swinging effect* observed in MVLN-1 learning phase is prevented.

The Recognition Rate and Mismatch Rate

From Table 3, it can be seen that the recognition rate as well as mismatch rate of MVLN-3 is significantly higher than those of MVLN-1. The exact recognition rate can be as high as 77%; the recognition rate can be up to 86% if the user is allowed to choose between two best matching characters. The higher recognition rate is undoubtedly due to the error tolerance activation mechanism. This can be verified by the fact that the MVLN-3 is capable of recognizing characters after being trained with a single example of each character while the MVLN-1's recognition rate is negligible in the same case. However, since the error tolerant activation mechanism incorporates no geometrical constraints on the errors tolerated, it inevitably leads to a higher mismatch rate. On the other hand, the representation of Chinese characters by a 12×12 bit map also tends to simplify several complicated characters, which usually have similar radicals and/or over 10 strokes, into very similar patterns. For instance, the characters in Figure 2 (a) are mistaken for one and so are those in Figure 2 (b).

(a) (b)

Figure 2. Example of mismatched characters

Training data	Recall data	Recognition Rate (%)				Mismatch Rate (%)			
		MVLN-1	MVLN-3			MVLN-1	MVLN-3		
			0.92	0.95	0.97		0.92	0.95	0.97
A1	A4	1	55:12:14	34:5:3	17:0:0	0	9	11	12
A1, A2	A4	9	59:14:23	56:14:12	44:9:6	2	3	12	13
A1, A2, A3	A4	17	63:12:20	62:12:16	59:11:11	4	5	8	10
B1	B4	0	41:3:0	17:0:0	4:0:0	0	5	2	0
B1, B2	B4	2	67:9:7	54:3:1	27:1:0	0	12	10	4
B1, B2, B3	B4	8	66:16:10	59:11:3	47:1:0	1	7	14	8
C1	C4	0	22:1:0	5:0:0	1:0:0	0	0	0	0
C1, C2	C4	1	67:7:4	46:1:0	19:0:0	0	10	8	0
C1, C2, C3	C4	5	77:9:7	60:3:4	34:1:1	0	3	4	4

Table 3. Results of experiments on MVLN-1 and MVLN-3

Published in the International Symposium on Artificial Neural Networks, National Cheng Kung University, Tainan, Taiwan, December 15-17 1994

It can also be seen from Table 3 that raising the confidence level may not necessarily reduce the mismatch rate but can significantly reduce the recognition rate. This suggests that the mismatch rate has to be reduced by other means. For example, representing the characters in a bigger bit map size may help reduce the mismatch rate. A better approach would be to use multiple MVLN-3 neurons to cover sub-areas of the entire bit map and feed the output of these MVLN-3 neurons to a MVLN-1 neuron. With such an arrangement, the geometric features of handwriting can be captured to certain degree. This technique is successfully used in the RAM-based neural network model (Aleksander & Morton 1990).

On-line learning and recognition is experimented using MVLN-3. In tens of tests on arbitrarily chosen native Chinese speakers with different cultural backgrounds, two to three samples of handwriting would give an average of over 70% recognition rate. This result is achieved with *ordinary* users rather than careful and cooperative ones.

The time for learning three samples of a handwritten Chinese character is in the order of milliseconds. Under the current simulation, the time to recognize a character is linear to the size of the dictionary. In our experiments with a hundred words in the dictionary, the time for recognition is in the order of tens of milliseconds. For a dictionary of three thousand characters, the estimated time is well below one second and this performance is independent of the complication of the character and the style of writing.

It should be noted that in all the experiments of MVLN-1, -2 and -3, only a single MIN MVL neuron is used to recognize a character. Using multiple layers of MVL neurons, e.g. a layer of MIN MVL neurons trained for several typical variations of the same character, followed by a MAX MVL neurons, the recognition rate can be further increased. This technique applies to all the MVL neural models.

V. Conclusions

The experiments reported are by no means complete and thorough. However, they do reveal the performance characteristics of MVL neural model and reflect the analysis of the learning and recall mechanisms. We believe that, in principle, MVLN-3 is inherently one formulation of the MVL neural model proposed by Watanabe, except that the error tolerant activation mechanism is inspired by the RAM-based neural network model WISARD (Aleksander & Morton 1990).

The recognition rate achieved with the single layer MIN MVL neural model is considered encouraging, because the data set used for the experiments is a difficult set of mixed Chinese characters and very little preprocessing of the input bit map has been exploited. Yet, the MVL neurons can learn to recognize personal style of handwriting with just a few examples.

The potential pattern recognition capability of the MVL neural model has not been fully explored. Further experiments are needed on incorporating techniques such as segmenting the input bit map, introducing noise in the training samples and using MIN-MAX two layer network structure etc. which may further increase the recognition rate to a satisfactory level. Combined with the fast learning of the MVL neural model, this approach has much potential for on-line handwriting recognition or adaptive handwriting scanning.

References:

1. Tappert, C. C., Suen, C. Y. & Wakahara, T., "The State of the Art in On-Line Handwriting Recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 8, August 1990, pp. 787-808.
2. Watanabe, T., Matsumoto, M., Enokida, M. & Hasegawa, T., "A Design of Multiple-Valued Logic Neuron", Proc. of the 20th International Symposium on Multiple-Valued Logic, 1990, pp. 418-425.
3. Widrow, B. & Winter, R., "Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition", IEEE Computer, Vol. 21, No. 3, March 1988, pp. 25-39.
4. Watanabe, T. & Matsumoto, M., "Layered MVL Neural Networks Capable of Recognizing Translated Characters", Proc. of the 22nd International Symposium on Multiple-Valued Logic, 1992, pp. 88-95.
5. Aleksander, I. & Morton, H., "An Introduction to Neural Computing", Chapman and Hall, 1990

Published in the International Symposium on Artificial Neural Networks, National Cheng Kung University, Tainan, Taiwan, December 15-17 1994

Appendix – A Sample of the Handwriting Used in the Experiments