

A Novel, Fast-Learning Neural Network Used in Handwritten Character Recognition

Chia-Lun J. Hu, Professor
Electrical Engineering Department
Southern Illinois University at Carbondale
Carbondale, IL 62901, U.S.A.
cjhu@siu.edu

ABSTRACT

Key Words: **Pattern Recognition, Image Processing, Novel Neural Network, Superfast Learning**

As we published in the last few years [1-7], when the given input-output training vector pairs satisfy a PLI (positive-linear-independency) condition, the training of a hard-limited neural network to learn this mapping can be achieved non-iteratively with very short training time and very robust recognition when any **untrained** patterns are tested in the recognition mode. The key feature in this novel pattern recognition system is the use of **slack constants** in solving the connection matrix when the PLI condition is satisfied. Generally there are infinitely many ways of selecting the slack constants for meeting the training-recognition goal, but there is only **one** way to select them if an optimal robustness is sought in the recognition of the **untrained** patterns. This particular way of selecting the slack constants carries some special physical properties of the system -- the **automatic feature extraction** in the learning mode and the **automatic feature competition** in the recognition mode. Physical significance as well as mathematical analysis of these novel properties are to be explained in detail in this article. Real-time experiments are to be presented in an **unedited movie**. It is seen that in the system, the training of 4 hand-written characters is close to real time (<0.1 sec.) and the recognition of the **untrained** hand-written characters is >90% accurate.

I. Introduction

Most neural network learning schemes are derived from learning systems which are generally **iterative** in nature. That is, under a given mapping consisting of a set of training input-output vector pairs, the connection matrices are changed step by step until

the input-output relations of the network match the training pairs in the given mapping. On the other hand, as we studied in the last few years [1-7], for a **one-layered, hard-limited, perceptron (OHP)**, the connection matrix to meet a given input-output mapping can actually be obtained **non-iteratively in one step** if the given mapping satisfies a certain **positive-linear-independency** (or PLI) condition. Whenever the given mapping satisfies this condition (for most practical pattern recognition applications, this condition is satisfied as discussed in section II.) generally, there exist infinitely many solutions for the connection matrix and we can select an **optimum** solution such that in the recognition mode, the recognition of any **untrained** patterns becomes **optimally robust**. A simple pattern recognition scheme for recognizing hand-written English alphabets is then designed along this line and implemented on a moderately fast personal computer. It is seen that the training of 4 hand-written letters takes less than 1/10 second, and the recognition of any **untrained**, hand-written letter is in **real-time** and it is very **robust**.

Similar derivations of basic equations can be found in many places (see for example, Refs. [8-12]), but they are not used in similar ways towards designs and applications as those described in this paper. The reason for this difference is that the PLI condition was not known and was not used in the solution of the basic equations in 1960's to 1980's. This PLI condition (first published in 1990 [13], [14] by this author) is the foundation of the novel analysis and the novel design discussed in this article. The application of this PLI condition (using slack constants) to the solution of the basic equations which leads to the design and analysis of the neural-network pattern-recognition systems has **not** been found in the literature. Review of the basic theories is discussed in section II. Detailed analysis of automatic feature extraction and automatic feature competition are given in section III. It is seen that

the main approach we took in this study is, first, the derivation of the **novel OHP non-iterative learning scheme**, and then, the derivation of the **automatic feature extraction scheme**, and finally, its application to the design of a very robust and very fast learning pattern recognition system. The improvement of robustness and learning speed over other systems is shown numerically at the end of section IV.

II. Basic Analysis

As we published in the last few years [1-7],

Thm 1 For a one-layered, hard-limited perceptron (OHP):

- if (a) there are M mapping pairs to be learned, $\{U_m \rightarrow V_m, m=1 \text{ to } M\}$
 (b) each U is an N-dimension analog vector, each V is a P-bit binary vector, $2^P \geq M$,

then (A) the i-th row A_i (an N-vector) of the $P \times N$ connection matrix A must satisfy the following simultaneous strict inequalities.

$$A_i \bullet Y_{mi} > 0, \quad m=1 \text{ to } M, \quad i \text{ fixed}, \quad (1)$$

where $Y_{mi} = v_{mi} U_m$ and $v_{mi} (= \pm 1)$ is the i-th bit of V_m . " \bullet " means dot product between two N-vectors. Y_{mi} are the input vectors dichotomized according to the i-th output bit.

(B) the if-and-only-if condition that solution A_i exists in (1) is the following.

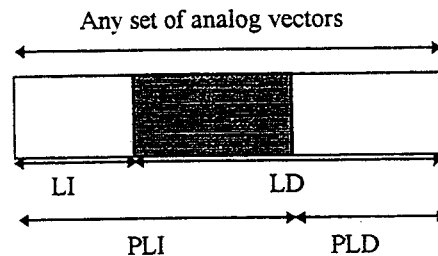
All $\{Y_{mi}, i \text{ fixed}, m=1 \text{ to } M\}$ are positively, linearly independent (PLI) (2)

PLI means that NO non-negative real numbers p_m , except all zeros, exist to satisfy the following linear dependent relation among all Y_{mi} 's.

$$\sum_{m=1}^M p_m Y_{mi} = 0, \quad (i \text{ fixed}, p_m \geq 0) \quad (3)$$

It is seen from the PLI definition here that PLI is a much broader class than LI (linear independency) as shown in Fig. 1. That is, any set of vectors which is linearly independent or LI must also be PLI, but

some LD vector may still be PLI as shown by the shaded area in Fig. 1¹



LI: linearly independent
 LD: linearly dependent
 PLI: positively, linearly independent.
 PLD: positively linearly dep.

Fig. 1

Due to the if-and-only-if nature of the PLI condition, if a given mapping $\{U_m \rightarrow V_m, m=1 \text{ to } M\}$ violates (2), then no matter what learning rule one uses, the OHP just cannot learn, because the connection matrix (or the solution of A_i in (1)) just does not exist. On the other hand, when (2) is satisfied for any given mapping, the OHP can definitely learn, and the connection matrix A in (1) can be obtained in one step. Because when solution exists, (1) can be solved with any elementary method by replacing " >0 " with " $=q_{mi}$ ", where $q_{mi} > 0$ is a positive slack constant, i.e.,

$$A_i \bullet Y_{mi} = q_{mi}, \quad m=1 \text{ to } M, \quad i \text{ fixed}. \quad (1')$$

In most pattern recognition applications, if the patterns to be distinguished are not very closely related to one another, then most likely, all pattern vectors U_m 's are LI (each training vector U_m here represents a distinct standard class pattern to be distinguished.) Then, by Fig. 1, they must also be PLI. Therefore (2) is satisfied and the solution of the

¹ Because, for a set of analog N-vectors $\{Y_m\}$ (subscript "i" dropped), if there exist only certain mix-sign real coefficients, but not same-sign p_m 's, to satisfy (3), then $\{Y_m\}$ is LD but it is PLI (corresponding to the shaded area in Fig. 1). On the other hand, any LI $\{Y_m\}$ must also be PLI, because "no real coefficients satisfying (3)" includes "no positive real coefficients satisfying (3)".

connection matrix in (1) for an OHP definitely exists for most pattern recognition applications.

Solving (1) by using slack constants q_{mi} is a **non-iterative** way of achieving the goal of supervised learning. It is not only very fast in learning (compared to the traditional **iterative** learning), but also, it allows us to achieve the optimal robustness in recognition by adjusting q_{mi} properly as explained in [6] and [7]. The conclusion of these two papers is repeated here.

To achieve an optimal robustness in recognition for an OHP, first, all q_{mi} 's must be set = one arbitrary positive constant q_i for all m 's as shown in (4).

$$A_i \bullet Y_{mi} = q_i, \quad m=1 \text{ to } M, \quad i \text{ fixed.} \quad (4)$$

Since q_i is also adjustable, if we select all q_i in such a way as shown in the following section, we can further improve the robustness and optimize it. This particular way of adjustment happens to carry another physically significant property, namely, the **automatic feature extraction**. This is to be explored in detail in the following section.

III. Automatic Feature Extraction and Feature Competition

If we notice from the definition under (1) that $Y_{mi} = v_{mi}U_m = U_m/v_{mi}$ because $v_{mi} = \pm 1$, we can write (4) in a matrix form:

$$AU = QV \quad (5)$$

where A is the $P \times N$ connection matrix. Each row of A is A_i . U is the $N \times M$ pattern matrix. Each column of U is the input U_m . Q is a $P \times P$ diagonal matrix. Each element of Q is q_i . V is the $P \times M$ binary matrix. Each column of V is the output V_m in the given mapping. If we apply the Sgn or the sign-operator to (5) (Sgn applied to a matrix is equivalent to Sgn applied to each element in the matrix), we obtain the **matrix control equation in the learning mode** of the OHP:

$$\text{Sgn}(AU) = V \quad (6)$$

The matrix Q does not appear in (6) because $\text{Sgn}(q_i v_{mi}) = v_{mi}$ where $q_i > 0$ and $v_{mi} = \pm 1$. If the dimension N of the pattern vector is much larger than the number M of the patterns to be classified, then this $N \times M$ U matrix is a slender, tall matrix.

We can rearrange the rows of U in such a way that we can partition U in many $M \times M$ sub-matrices U^k , $k=1$ to K such that

each U^k is **nonsingular** and all U^k 's are arranged in a **descending order** of $\|U^k\|$'s where $\|U^k\|$ is the absolute value of the determinant of U^k . (7)

Then, we can also rearrange the columns of A and partition the columns of A accordingly. By partition-multiplication of matrices, we then see that (5) becomes

$$\sum_{k=1 \text{ to } K} A^k U^k = QV \quad (8)$$

where we have set the **residual part** A^R in A (corresponding to the un-partitioned part in U) = 0.

Now, if we impose another condition on q_i 's in Q such that all q_i 's = K , the number of the nonsingular sub-matrices in U , then we can solve each A^k in (8) individually from

$$A^k U^k = V, \quad k=1 \text{ to } K \quad (9)$$

The solution of A^k definitely exists and it is unique ($A^k = V (U^k)^{-1}$) because U^k is **nonsingular**. The unknown connection matrix A in (6) is just a concatenation of all the A^k 's solved from (9). When A is obtained, the **noniterative learning** is done. We then arrive at the following important result.

(R1) Each U^k in the input pattern matrix U can be considered as a **feature matrix** of all the standard class patterns used in the training. Because if we set all $U^j=0$, except $j=k$, and substitute the concatenated U into the left-hand-side of (6), AU will be just equal to **one term**, $A^k U^k$. But we still obtain V at the output of this OHP because of (9). We do **not** need all the terms in $AU = \sum A^k U^k$ to obtain the output V in (6). Consequently, each U^k **alone** must carry **enough characteristics** of the input class patterns that allows the OHP to differentiate them according to the targeted output V . This U^k is therefore the k -th feature matrix and the m -th column of U^k is the k -th feature vector of the m -th class pattern used in training.

(R2) All features (or all column vectors U_m^k , $m=1$ to M) in U^k are selected with maximum discriminations because when U^k 's are selected according to (7) all U_m^k are spanning with

maximum discrimination from one another in the M-space of U_m^k . This is so because, if only the orientations of U_m^k are allowed to be changed in the M-space, then the maximum discrimination occurs only when these vectors are **mutually orthogonal** to each other. This can occur only when the **volume of the parallelepiped** spanned by these M vectors is a maximum. Since, by N-dimension geometry, the volume of a parallelepiped in an M-space is = the absolute value of the determinant $\|U^k\|$, **maximum $\|U^k\|$ we select according to (7) means maximum discrimination among the features we select in the N-space.**

(R3) This feature selection process is an **automatic global selection process**. That is, each feature may not be a physically connected part in the N-dimension space because of the selection process (7). But it **automatically selects the most distinguish parts** in the training class patterns $\{U_m, m=1 \text{ to } M\}$ for comparison because of the **maximum discrimination** property just explained in (R2).

(R4) Sorting out all **nonsingular sub-matrices U^k** in the input matrix U according to (7) is therefore an **automatic feature extraction process** in this novel, **neural network pattern recognition system**.

(R5) To **emphasize the importance of the selection of U^k 's** according to (7), we can use the following **control equation (instead of (6)) in the recognition mode for reaching more robustness in recognition.**

$$S = \text{Sgn}[\sum_k w_k A^k T^k] \quad (10)$$

where A^k (solved from (9)) is the sub-matrix learned in the **noniterative learning**. T^k is an $M \times 1$ feature vector of the test pattern T (an $N \times 1$ vector) where elements of T are re-arranged according to (7). w_k is a set of weight coefficients defined below.

$$w_k \equiv \|U^k\| \equiv \text{absolute value of the determinant of the sub-matrix } U^k \quad (10w)$$

S is a $P \times 1$ binary output vector in the recognition mode. If $S = \text{any } V_m$ in the giving mapping, then the test input T is recognized as the m-th pattern learned in the learning mode. Otherwise, T is not recognizable by this system.

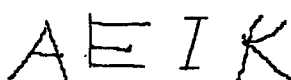
It is seen that the addition of the weight coefficients w_k here carries the flavor of **automatic feature competition**. Because the **most distinguish features**

in the class patterns found in the training are assigned to have the **highest weight** in the recognition of any **unknown patterns**. Consequently this would add more robustness in the recognition.

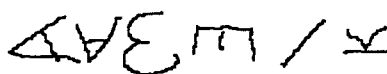
(R6) The learning and the recognition described above can all be done with a personal computer if some soft ware (e.g., **Visual Basic™**) is available for digitizing any patterns we draw for training as well as for recognition. This would then be a **true experiment**, not a **simulated experiment**, for testing all the above theoretical results. This is to be explained in the following.

IV. Experiments on a Handwriting Recognition Scheme

Based on the above non-iterative learning theory, a pattern recognition scheme is designed for recognizing some hand-written English alphabets. The designed system is implemented on a personal computer IBM Value Point 433 DX PC and is used in **real-time experiments**. It is seen that the training time for training 4 hand written alphabets is less than 1/10 of a second, and the recognition is instantaneous and the recognition of any **untrained hand-written alphabets** is more than 90% accurate. A typical set of training and testing patterns is shown in Fig. 2. The whole learning-recognition experiment is recorded on an unedited, real-time, video-movie for presentation in the conference.



Typical training patterns



Typical test patterns

Fig. 2

V. Conclusion

Because of the PLI condition we derived for solving the control equation of the OHP, it is possible to solve the unknown connection matrix of the supervised learning in a **non-iterative** manner by using positive slack constants p_{mi} . Adjusting p_{mi} properly one can then reach **optimal robustness**

when **untrained patterns** are tested in the recognition (or test) mode. This particular way of adjusting the slack constants happens to carry an intriguing physical property – the **automatic feature extraction and feature competition** – in this non-iterative learning system. This then allows us to design a very fast learning and very robust recognition system which is different from most of the conventional learning-recognition systems. The learning is very fast because it is done with **non-iterative means**. The recognition is very robust because an **optimal robustness scheme** is used. From some **real-time** (not simulated) **experiments** done with this system, we see that the learning is close to real-time and the recognition of the **untrained patterns** is in real-time. These experiments also showed that the recognition of the **untrained hand-written patterns** is above 90% accurate. These experimental results were recorded in life in **real-time**, and are shown in an **unedited video movie** to be presented in the conference.

References

- [1]. Hu, C. J., "Rotation invariance in a noniteratively trained neural network pattern recognizer," **Intelligent Engineering Systems Through Artificial Neural Networks**, 403-407, vol. 4., 1994, ASME Press.
- [2]. Hu, C. J., "Pattern Recognizer Using Real-Time Noniterative Learning," **Proc. World Congress on Neural Networks (WCNN '95)**, pp II-195 to 198, Washington, D.C., July 17, 1995.
- [3]. Hu, C. J., "Design of a parallelly cascaded, two-layered perceptron consisting of hard-limited neurons," **Proc. SPIE Appl. Sci. Artificial Neural Networks**, Orlando, FL, April 9, 1996.
- [4]. Hu, C. J., "Ultra-fast learning in a hard-limited neural network pattern recognizer," **Proc. IS&T/SPIE-96**, San Jose, CA, Jan. 28-Feb.2, 1996
- [5]. Hu, C. J., "Robustness of a neural network pattern recognizer using noniterative learning," **Proc. of World Congress on Neural Network '96**, pp. 416-419, San Diego, CA, Sept. 15-18, 1996.
- [6]. Hu, C. J., "Pattern recognition using an ultra-fast, noniteratively learned neural network." presented at **Conference on Information Sciences and Systems**, John Hopkins University, 3-20, 1997
- [7]. Hu, C. J., "Feature competition and feature extraction in a noniteratively learned neural network pattern recognition scheme," **Proc. SPIE, Optical Pattern Recognition**, Orlando, FL April 15, 1998.
- [8] Duda, R.O., Hart, P.E., **Pattern Classification and Scene Analysis**, John Wiley, 1973
- [9] Nilsson, N.J., **Learning Machines**, McGraw Hill, 1965
- [10] Minsky, M., and Papert, S., "**Perceptrons**," MIT Press, 1969
- [11] Cover, T. M., "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition," **IEEE Trans. on Electronic Computers**, 326-334, June 1965
- [12] Tou, J.T., Gonzales, R.C., **Pattern Recognition Principles**, Addison-Wesley, 1974.
- [13] Hu, C. J., "Characteristics of a novel, geometrical supervised learning scheme," **Proceedings of the 1990 ISSM International Conference on Parallel and Distributed Computing Systems**, New York, NY, 21-23, Oct. 10-12, 1990
- [14] Hu, C. J., "A novel geometrical learning scheme," **Proceedings of Appls. of Art. Neural Netws., SPIE**, Vol 1294, pp.426-431., Orlando, FL., April 18-20, 1990.