An Optimal Dimension Expansion Procedure for Obtaining Linearly Separable Subsets

Yuen-Hsien Tseng and Ja-Ling Wu
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, R.O.C.

Abstract

In this paper, we study the necessary and sufficient condition for linearly separable subsets and then propose an optimal dimension expansion procedure that makes any mapping to be performed by perceptrons learnable by error-correction procedure. For n-bit parity check problems, it is shown that only one additional dimension is augmented to make them solvable by single-layer perceptrons. Other applications such as for decoding error-correcting codes are also considered.

1. Introduction

The theorem that error-correction procedure can learn linearly separable subsets *in finite steps* has been proved early before 1965 [1]. However, for subsets that are not linearly separable, perceptrons are not guaranteed to converge, nor can we be sure that the pattern set being trained is linearly separable if it did not converge after some long learning steps. Furthermore, Minsky and Papert [2] showed that single-layer perceptrons is inadequate to solve the parity check problems, which on the other hand is quite simple by modern digital circuits. These factors contribute partly to a situation that single-layer perceptrons do not receive much attention until recently.

In this paper, we study the necessary and sufficient condition for linearly separable subsets and then propose a optimal dimension expansion procedure to make any mapping linearly separable. Such a procedure provides an immediate advantage that we do not really need a hidden layer to perform certain mappings, and are thus free to worry about getting trapped in local minima while applying backpropagation-like learning processes.

To begin with, suppose we have a finite set C of distinct patterns, $C = \{X_i \mid X_i \in \mathbf{R}^n, i=1, 2, ..., K\}$. Let the patterns of C be classified in such a way that each pattern in C belongs to only one of r categories. Or we can think that each pattern in C maps to one of r distinct values. Consider the case of r=2 and assign the two distinct values to be 1 and –1. We collect those patterns mapping to 1 in C^+ and the remaining in C^-. Formally, define

$$C^+ = \{k \mid X_k \rightarrow 1, \text{ for all } X_k \in C\} \quad (1a)$$

$$C^- = \{k \mid X_k \rightarrow -1, \text{ for all } X_k \in C\} \quad (1b)$$

CH 3065-0/91/0000-2461 $1.00 © IEEE

where '–>' denotes "is mapped to". The two subset C^+ and C^- defined in the above is linearly separable [1] if and only if there exists a weight vector $W \in R^n$ and a threshold $t \in R$ such that

$$W^T X_k - t > 0, \qquad \text{for all } k \in C^+ \tag{2a}$$

and $$W^T X_k - t < 0, \qquad \text{for all } k \in C^- \tag{2b}$$

If each pattern X_k in C maps to a bipolar pattern, say Y_k, $Y_k \in \{1, -1\}^m$, k = 1, ..., K, we write

$$C_i^+ = \{k \mid X_k \text{ -> } y_{ki} = 1, \text{ for all } X_k \in C\} \tag{3a}$$

$$C_i^- = \{k \mid X_k \text{ -> } y_{ki} = -1, \text{ for all } X_k \in C\} \tag{3b}$$

where y_{ki} is the *ith* component of Y_k, and i = 1, 2, ... , m.

In the rest of this paper, the scalar values are denoted by lower-case letters, while vectors and matrices are represented by capital letters. In the next section, we develop the dimension expansion procedure for obtaining linearly separable subsets. In section 3, applications of the procedure are shown. The final section presents our concluding remarks.

2. The Optimal Dimension Expansion Procedure

The behavior of a single-layer perceptrons with n inputs and m outputs can be expressed as

$$z_{ki} = \text{sgn}\,(a_{ki}) \tag{4}$$

$$a_{ki} = X_k^T W_i - t_i \tag{5}$$

where W_i is the *ith* column of weight matrix W, t_i is the *ith* component of threshold vector T, i = 1, 2, ..., m, and sgn is the sign function: sgn(a) = 1 if a > 0, sgn(a) = –1 if a ≤ 0. (Note the binary Hopfield model's motion equation can also be described by (4) and (5). Hence the discussion below also includes recurrent networks.) For the clarity of the ensuing discussion, vectors X_k and W_i are augmented by one additional dimension (referred to as index 0) to accomodate the thresholds in the above equations, i.e., $x_{k0} = -1$ and $w_{i0} = t_i$, and also for simplicity the dimension of X_k and W_i are still regarded as n, for all k and i. Now, rewrite Eqns. (4) and (5) in a more compact form

$$Z_k = \text{Sgn}\,(W_i^T X_k) \tag{6}$$

and we say that X_k is mapped to Z_k, X_k –> Z_k, by the perceptrons.

Now let us concentrate on some output unit i. By multiplying each X_k with y_{ki}, and let $U_k(i) = y_{ki} X_k$, we can construct a K×n matrix U(i) with each row the transpose of $U_k(i)$, k=1, 2, ..., K. Hence the inequalities in (2) can be rewritten as

$$U(i) W_i > 0 \tag{7}$$

Let B_i be a K×1 vector and each of its component be positive, denoted as $B_i \in (R^+)^K$. Then by definition, C_i^+ and C_i^- is linearly separable if and only if there exists a B_i such that

$$U(i)W_i = B_i \qquad (8)$$

, in other words, C_i^+ and C_i^- is linearly separable if and only if *B_i belongs to the intersection of the column space of U(i) and $(R^+)^K$*, where the column space of U(i) is a n-dimensional subspace contained in or expanding the K-dimensional vector space.

We now consider the solution of W_i. Provided that such B_i exists, the solution for W_i in (8) is exactly

$$W_i = (U(i)^T U(i))^{-1} U(i)^T B_i \qquad (9)$$

for $K \geq n$ and U(i) being of full rank [3]. For $K < n$, W_i can be obtained by setting $w_{ji} = 0$, removing *jth* column vector of U(i) which is not linearly independent to other column vectors, and solving the resultant equations.

Note that a sufficient condition for (8) to hold is that one of the column vector of U(i) belongs to $(\mathbf{R}^+)^K$. This condition will be used in the below algorithm for dimension expansion.

Since, as we have mentioned, error-corretion procedure can only learn linearly separable subsets, one has to make them separable before applying the learning procedure if they are not. This is done by transforming a mapping $X_k \rightarrow Y_k$, $X_k \in \mathbf{R}^n$, $Y_k \in \{1, -1\}^m$, $k = 1, 2, ..., K$, into $X^*_k \rightarrow Y_k$, where $X^*_k{}^T = [X_k^T \; D_k^T]$ is an expanded vector from X_k and D_k. The vector D_k is obtained by the following algorithm.

step 1 *Let $i = 1$; $D_k = 0$; (initially set D_k to be a vector of length 0)*

step 2 *$X^*_k{}^T = [X_k^T]$, $k = 1, 2, ..., K$;*

step 3 *$C_i^+ = \{k \mid X^*_k \rightarrow y_{ki} = 1, \text{for all } X^*_k\}$*

$C_i^- = \{k \mid X^*_k \rightarrow y_{ki} = -1, \text{for all } X^*_k\}$

step 4 *If C_i^+ and C_i^- is not linearly separable, then do*

step 4.1 *$D_k^T = [D_k^T \; c_i y_{ki}]$ $k = 1, 2, ..., K$, c_i : any positive number;*

(augment D_k by one dimension)

step 5 *$X^*_k{}^T = [X_k^T \; D_k^T]$; $i = i + 1$; if $i > m$ then stop else go to step 3.*

The above algorithm is tantamount to augment the matrix U(i) in (8) by a column vector in $(\mathbf{R}^+)^K$ so as to make (8) solvable. As to the test for linear separability, one can refer to [4]

This algorithm is similar to the one in [5], where it is proposed for recurrent networks and temporal mapping for bipolar input vectors (vectors in $\{1, -1\}^n$). it is conjectured in [5] that the algorithm adds only minimal number of units needed to legalize a mapping, i.e., to transform a mapping such that a single-layer perceptrons is able to perform. However, the following example shows there is the case that it may fail.

Example: An illegal mapping with 3 inputs and 2 outputs is shown below.

(1, 1, 1) -> (1, 1), (1, -1, -1) -> (1, 1), (-1, -1, 1) -> (1, 1), (-1, 1, -1) -> (1, 1),
(-1, 1, 1) -> (-1, -1), (-1, -1, -1) -> (-1,-1), (1, 1, -1) -> (1, -1), (1, -1, 1) -> (1, -1).

After running the expansion procedure, two bits are added to each of the input patterns and the resultant patterns are (assume c_i in step 4.1 is 1 for all i)

(1, 1, 1, 1, 1), (1, -1, -1, 1, 1), (-1, -1, 1, 1, 1), (-1, 1, -1, 1, 1),
(-1, 1, 1, -1, -1), (-1, -1, -1, -1, -1), (1, 1, -1, 1, -1), (1, -1, 1, 1, -1)

respectively. It can be shown that a perceptron with 5 inputs and 2 outputs is able to perform the mapping. However, if the second output bit is examined and tested for linear separability before the first one, then only one bit is addded to make the mapping legal. In this case, the result is

(1, 1, 1, 1), (1, -1, -1, 1), (-1, -1, 1, 1), (-1, 1, -1, 1),
(-1, 1, 1, -1), (-1, -1, -1, -1), (1, 1, -1, -1), (1, -1, 1, -1).

This example shows that the number of bits added to legalize a mapping has to do with the sequence that the output bits are examined. So an optimal expansion procedure would try all possible sequences of examining the output bits and see which results in minimum dimensions.

Remark 1: the maximum augmented dimension in the above algorithm is m, the number of output units, i.e., the dimension of X^*_k is no larger than n+m, no matter how many pattern pairs there are.

Remark 2: Suppose X^*_k is of dimension n+m, then X^*_k is just the concatenation of X_k and Y_k, i.e., $X^*_k = [X_k^T, Y_k^T]$, if all c_i in step 4.1 are 1.

3. Applications

3.1 N-bit Parity Problem

Remark 1 implies that for the n-bit parity problem (where a n-bit input maps to an one-bit output indicating whether the input vector contains an odd number of 1 or not), by augmenting only one additional dimension, the perceptrons can learn to perform the parity checking, which is otherwise impossible for single-layer perceptrons to handle. For example, a 2-bit parity problem can be described by the following mapping:

(1, 1) -> 1, (1, -1) -> -1, (-1, 1) -> -1, (-1, -1) -> 1.

It is obviously that there exists no weight vector and threshold such that it is separable. After expanding the input vectors using the aforecited procedure, it becomes

(1, 1, 1) -> 1, (1, -1, -1) -> -1, (-1, 1, -1) -> -1, (-1, -1, 1) -> 1.

By randomly setting the initial weight and threshold to (0.2, -0.3, 0.1) and -0.1, respectively, and training the perceptrons according to the error-correction learning rule [1] (i.e., delta rule [6])

$$W_i(s+1) = W_i(s) + c\,(y_{ki} - z_{ki})X_k \qquad (10a)$$

$$t_i(s+1) = t_i(s) - c\,(y_{ki} - z_{ki}) \qquad (10b)$$

where s is the iteration index and c is the positive step size, the final weight and threshold after convergence are (-0.8, 0.7, 1.1) and 0.9, respectively (c = 0.5 in this example). Consequently, an addition in one more input unit saves n hidden units as required by the simulations in [6] [7].

3.2 An Error Control Code Decoder

Consider the problem of decoding a systematic code [8]. A codeword X_k in a systematic code is composed of information word Y_k and parity word P_k, in the form of $X_k^T = [Y_k^T P_k^T]$, where X_k is of dimension n, Y_k is m and P_k is r. At the receiving end, the codeword X_k is decoded into Y_k despite of error noise, say one bit error. A single-layer perceptrons is expected to perform such mapping $X_k \rightarrow Y_k$, for k = 1, 2, ... $(n+1)2^m$. (2^m codewords and n error pattern for each codeword, suppose we are concerning only correcting one bit error, and that the perceptrons learns both the correct codewords and the erroneous ones) However, if the mapping is not linearly separable, the mapping may fail even there is no noise. In this case, the received codeword X_k^T is expanded into $X^*_k{}^T = [Y_k^T P_k^T Y_k^T]$, and then the perceptrons is trained to perform the mapping. The mapping $X^*_k{}^T \rightarrow Y_k^T$ is legal from remark 2.

4. Conclusion

We have shown a necessary and sufficient condition for linearly separable subsets. This condition is quite clear and useful under the geometrical interpretation. Based on the geometrical interpretation, we develop an optimal dimension expansion procedure to linearly separate the ensemble subsets. Two possible applications of the proposed procedure are illustrated. For those mappings whose domains contain finite and fixed points, the expansion procedure allows a single-layer perceptrons to perform the mapping exactly after learning the training set by error-correction procedure, without resorting to multilayered perceptrons and much time-consuming backpropagation algorithms.

Reference

[1] Nils J. Nilsson, *Learning Machines: Foundations of trainable Pattern-Classifying Systems*, McGraw-Hill, 1965.

[2] M.Minsky and S.Papert, *Perceptrons*, Cambridge, MA:MIT Press, 1969.

[3] Gilbert Strang, *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, 1986.

[4] Singleton, R.C., "A Test for Linear Separability as Appied to Self-organizing Machines", in *Self-or0ganizing Systems–1962*, pp.503-524, Spartan Books, 1962.

[5] Esther Levin, "A Recurrent Neural Network: Limitation and Training", Neural Networks, Vol.3, No.6, pp.641-650, 1990.

[6] D.E.Rumelhart, G.E.Hinton, and R.J.Williams, "Learning Internal Representaions by Error Propagation" in Parallel Distributed Processing, D.E.Rumelhart and PDP research group, Cambridge, MA:MIT Press 1986.

[7] Eric B.Baum, "Neural Net algorithms that Learn in Polyniomial Time from Examples and Queries", IEEE Trans. on Neural Networks, Vol.2, No.1, Jan. pp. 5-19, 1991.

[8] Richard E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, 1983.