SYNTHESIS FOR SYMMETRIC WEIGHT MATRICES OF NEURAL NETWORKS

M. Saubhayana and R. W. Newcomb

Microsystems Laboratory
Department of Electrical and Computer Engineering
University of Maryland, College Park
College Park, Maryland 20742, USA

ABSTRACT

A synthesis method to guarantee symmetric weight matrices for a class of neural networks (which includes the Hopfield neural network as a special case) is proposed. This fills in a gap in the Li-Michel-Porod's synthesis and guarantees asymptotic stability for a given set of linearly independent equilibrium points under Lyapunov's stability criteria.

1. INTRODUCTION

In this paper, we propose an alternative method to the synthesis of Li, Michel and Porod [1, pp. 981] whereby we obtain symmetric weight matrices for the neural networks of generalized Hopfield class [1]. Asymptotic stability of sets of given equilibrium points is guaranteed by using these symmetric weight matrices. In [1], the synthesis method is proposed using only given initial equilibrium points but the method requires the singular value decomposition and symmetric condition on some intermediate matrices, which are not always obtained. This paper proposes an alternative synthesize which guarantees the symmetric weight matrices by augmenting with an additional equilibrium point and using simple linear transformation to give more degrees of freedom to the synthesis.

The proposed synthesis method first transforms a set of desired linearly independent equilibrium points to the proper form. Then using an augmentation vector makes the set of transformed equilibrium points to a direct sum of 2 by 2 blocks of non-zero entries. At this point, the system can be separated to several 2 by 2 sub-systems. Solving a nonlinear equation in each sub-system results in the desired corresponding components to give a weight matrix.

The generalized Hopfield neural network and its brief stability analysis are given in Section 2. The proposed synthesis method and a numerical example are discussed in detail in Section 3. Then the Summary Section closes the paper.

2. GENERALIZED HOPFIELD NEURAL NETWORK

2.1 Model

The generalized Hopfield neural network model can be implemented by electrical circuit components such as resistors, nonlinear capacitors and transistors and can be drawn in vector-matrix form as shown in Fig. 1 where voltage controlled sources are used.

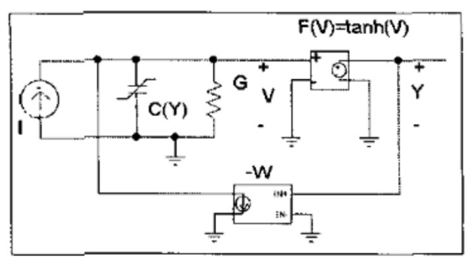

Figure 1. Generalized Hopfield neural network

The mathematical model of the above circuit can be written using Kirchhoff's current law at the input node of each voltage-controlled voltage source. For example, the ith voltage-controlled voltage source equation is:

$$c_i(y_i)\frac{dv_i}{dt} = \sum_j w_{ij}y_j - g_i v_i + I_i \tag{1}$$

$c_i(y_i)$ and g_i are shunt nonlinear capacitance and conductance at the input of the ith voltage-controlled voltage source, v_i is its ith input voltage (capacitance voltage), y_i is its ith output voltage, I_i is the ith bias current, and w_{ij} is the neural network weight as the g_m of a voltage-controlled current source. The output of each voltage-controlled voltage source is written in term of its input voltage as follows:

$$y_i = f_i(v_i) \tag{2}$$

0-7803-7761-3/03/$17.00 ©2003 IEEE

Usually the function $f(\cdot)$ is a sigmoidal function, which is monotonically increasing, odd and equal to zero only when the input is zero. One example of the sigmoidal function is the hyperbolic tangent function, which we use in our example.

2.2 Stability Analysis of the Generalized Hopfield Neural Network

The main objective of the neural network is to be able to move its states from a possible initial point to a pre-determined possible equilibrium point. This means the equilibrium point should be asymptotically stable. This section will be devoted to stability analysis.

Noting (2), from (1) select y_i as its state (call it x_i, that is, set $x_i = y_i$) and as in [1], define $h_i(x_i) = \frac{1}{c_i(x_i)} \cdot \frac{df_i}{dv_i}(v_i)\Big|_{v_i = f_i^{-1}(x_i)}$. We write state equations as follows:

$$\dot{x}_i = h_i(x_i)\left[\sum_j w_{ij}x_j - g_i f_i^{-1}(x_i) + I_i\right] \quad (3)$$

with equilibrium points as solutions of:

$$0 = \left[\sum_j w_{ij}x_j - g_i f_i^{-1}(x_i) + I_i\right] \quad (4)$$

Using $s_i(x_i) = g_i f_i^{-1}(x_i)$, (3) can also be written in matrix form as in [1]:

$$\dot{X} = H(X)[WX - S(X) + I] \quad (5)$$

Based on (3), we write a scalar energy-like function as:

$$Q(X) = -\frac{1}{2}X^T WX + G\int_0^x f^{-1}(z)dz - IX \quad (6)$$

However, this may not be a Lyapunov function, though it is guaranteed to be if W is symmetric. Assuming W is symmetric, using LaSalle's theorem [3], the equilibrium points of (3) are asymptotically stable in $\Omega(\varepsilon) = \{X \in R^n \mid |X_i| \le (l_M - \varepsilon)\}$, in which the case of the hyperbolic tangent activation function has $l_M = 1$ and $\varepsilon > 0$ is arbitrary small. Since we wish the weight matrix W to be symmetric for $Q(X)$ to be a Lyapunov function, this paper will concentrate on how to construct a symmetric W matrix from a provided set of linearly independent equilibrium point vectors.

In generalized Hopfield neural networks, $H(X)$ can be any positive definite matrix. This situation does not alter the above stability analysis.

3. SYNTHESIS OF SYMETRIC WEIGHT

3.1 Synthesis Idea

At an equilibrium point, (4) can be rewritten in the matrix form as:

$$0 = WY - GV + I \quad (7)$$

The method starts with a given set of n linearly independent equilibrium *n-vector* points, which are known in terms of input (V) and corresponding output (Y) of the voltage-controlled voltage sources (V and Y are related to each other by the activation functions). If this set of V's is not linearly independent, we can always get rid of some redundancy. Let $V_{d1},\ldots,V_{dn}$ and corresponding $Y_{d1},\ldots,Y_{dn}$ be the desired linearly independent equilibrium point vectors. These vectors have to satisfy (7). A series of elementary row operations can transform these desired equilibrium point vectors to a desirable form (call them V_{ti} and Y_{ti}). We transform such that there are two nonzero entries in proper locations and all zeros for the remaining entries. We also add to each of these vectors special vectors that have two zero entries, where V_{ti} is nonzero, and the other entries chosen wisely from entries of an augmenting vector V_x (and corresponding Y_x). Call these resulting *2n* vectors V_{ei} and Y_{ei}. Then subtract from them the augmenting vectors to form the following equation:

$$WA - GB = 0 \quad (8)$$

where

$$A = [Y_{e1} - Y_x \mid Y_{e2} - Y_x \mid \ldots \mid Y_{en} - Y_x] \quad (9)$$

$$B = [V_{e1} - V_x \mid V_{e2} - V_x \mid \ldots \mid V_{en} - V_x] \quad (10)$$

From (7) we desire to calculate W. Recall that all pairs of V_{ei}, Y_{ei} and the pair V_x, Y_x have to satisfy (7). Thus the input bias I will be able to be calculated from (7) using the pair V_x, Y_x. The V_{ei}, Y_{ei}, V_x, Y_x will be constructed in such a way that A and B are arranged in non-zero 2 by 2 blocks on their main diagonal axis while the other entries are zero (basically, it will be a direct sum of *n/2* of 2 by 2 matrices) if n is even while if n is odd, there will be a one by one block as the last matrix. So, for n even, (8) will look like:

$$\begin{bmatrix} W_{11} & 0 & 0 & 0 \\ 0 & W_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & W_{\frac{n}{2}\frac{n}{2}} \end{bmatrix} \begin{bmatrix} A_{11} & 0 & 0 & 0 \\ 0 & A_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & A_{\frac{n}{2}\frac{n}{2}} \end{bmatrix} = \begin{bmatrix} G_{11} & 0 & 0 & 0 \\ 0 & G_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & G_{\frac{n}{2}\frac{n}{2}} \end{bmatrix} \begin{bmatrix} B_{11} & 0 & 0 & 0 \\ 0 & B_{22} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & B_{\frac{n}{2}\frac{n}{2}} \end{bmatrix} \quad (11)$$

Each sub-matrix will be 2 by 2 with those of A being non-singular (except for the last block that might be 1 by 1 depending on the value of n) and each block can be written as:

$$W_{ii} = G_{ii} B_{ii} A_{ii}^{-1} \quad (12)$$

Because we need symmetric W, the 2 by 2 W_{ii} has to be symmetric too. Consequently, the (1,2) and (2,1) components of W_{ii} have to be equal. Evaluating the right side of (12) for the corresponding (1,2) and (2,1) components we can force them to be equal to each other by selecting the two free components of V_x at the two rows that go with W_{ii}. There will be one equation with two unknowns in the two corresponding positions of V_x. So, we have some degree of freedom to select and can always find a solution as seen below. Also the corresponding components of I (bias current) are calculated form (7). This technique will be used for all W_{ii} to get all components of V_x and corresponding Y_x. At this point, we know V_{ei} so we can relate it back to the desired equilibrium point vectors V_{di}.

With the above idea, we can synthesize the symmetric weight matrix. In the next section, we will illustrate by a numerical example the step-by-step procedure.

3.2 Numerical Example

In this section, we give a numerical example in R^4. Let us start with any initial choices of $V_{d1},\ldots,V_{d4}$, which have to be linearly independent. We use the hyperbolic tangent as the activation function. We also assume as given the conductance matrix $G = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$. The following steps are used:

1. From a given equilibrium point set of vectors V_d, group them in pairs (if there are an odd number of vectors, leave the last one alone). Then use row operations (or a conversion similar to the one in step 8 equation 15 below) to transfer them in to a set V_t, in which each vector has two non-zero entries and zero entries for the rest. The non-zero entries are in the *1*st and *2*nd for the first pair, next non-zero pair in the *3*rd and *4*th locations and so on. This can be done because the desired equilibrium point vectors are linearly independent. Keep these row operations as an invertible transformation for future use. As an example, we take V_t as follows:

$$V_{t1} = \begin{bmatrix} 0.5 \\ 0.25 \\ 0 \\ 0 \end{bmatrix}, V_{t2} = \begin{bmatrix} -0.5 \\ 0.5 \\ 0 \\ 0 \end{bmatrix}, V_{t3} = \begin{bmatrix} 0 \\ 0 \\ 0.5 \\ 0.25 \end{bmatrix}, V_{t4} = \begin{bmatrix} 0 \\ 0 \\ -0.5 \\ 0.5 \end{bmatrix}$$

2. Define an augmentation vector $V_x = \begin{bmatrix} v_{x1} \\ v_{x2} \\ v_{x3} \\ v_{x4} \end{bmatrix}$, and corresponding $Y_x = \begin{bmatrix} y_{x1} \\ y_{x2} \\ y_{x3} \\ y_{x4} \end{bmatrix} = \tanh(V_x)$, in such a way that when we subtract these vectors from the vectors in the modified equilibrium vectors ($V_{e1},\ldots,V_{en}$ and $Y_{e1},\ldots,Y_{en}$) to compose the matrices A and B defined above (in (9) and (10)), we will have the proper form for (12). Therefore, as we want to get the direct sum of 2 by 2 boxes in A and B, we set:

$$V_{e1} = \begin{bmatrix} 0.5 \\ 0.25 \\ v_{x3} \\ v_{x4} \end{bmatrix}, V_{e2} = \begin{bmatrix} -0.5 \\ 0.5 \\ v_{x3} \\ v_{x4} \end{bmatrix}, V_{e3} = \begin{bmatrix} v_{x1} \\ v_{x2} \\ 0.5 \\ 0.25 \end{bmatrix}, V_{e4} = \begin{bmatrix} v_{x1} \\ v_{x2} \\ -0.5 \\ 0.5 \end{bmatrix}$$

At this point, we do not know V_x and Y_x yet as there are four free parameters.

3. From the above V_{ei}, we write A and B in direct sum form as follows:

$$A = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}$$

$$A_{11} = \begin{bmatrix} \tanh(0.5) - \tanh(v_{x1}) & \tanh(-0.5) - \tanh(v_{x1}) \\ \tanh(0.25) - \tanh(v_{x2}) & \tanh(0.5) - \tanh(v_{x2}) \end{bmatrix}$$

$$A_{22} = \begin{bmatrix} \tanh(0.5) - \tanh(v_{x3}) & \tanh(-0.5) - \tanh(v_{x3}) \\ \tanh(0.25) - \tanh(v_{x4}) & \tanh(0.5) - \tanh(v_{x4}) \end{bmatrix}$$

$$B = \begin{bmatrix} 0.5 - v_{x1} & -0.5 - v_{x1} & 0 & 0 \\ 0.25 - v_{x2} & 0.5 - v_{x2} & 0 & 0 \\ 0 & 0 & 0.5 - v_{x3} & -0.5 - v_{x3} \\ 0 & 0 & 0.25 - v_{x4} & 0.5 - v_{x4} \end{bmatrix}$$

We know the given G and we want the W to be symmetric. So we use (12) and get W_{11} and its components. In this case, we have:

$$W_{11} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 0.5 - v_{x1} & -0.5 - v_{x1} \\ 0.25 - v_{x2} & 0.5 - v_{x2} \end{bmatrix} .$$

$$\frac{1}{\det A_{11}} \cdot \begin{bmatrix} \tanh(0.5) - \tanh(v_{x2}) & -(\tanh(-0.5) - \tanh(v_{x1})) \\ -(\tanh(0.25) - \tanh(v_{x2})) & \tanh(0.5) - \tanh(v_{x1}) \end{bmatrix} \quad (13)$$

4. The key step is to set $w_{12} = w_{21}$ and solve for v_{x1} and v_{x2}. There is one nonlinear equation and two unknowns so we have flexibility to select solutions. Select $v_{x2} = 0.3$ and solve the nonlinear equation to get $v_{x1} = 0.494563$. Then we have $W_{11} = \begin{bmatrix} 2.15568 & -0.03521 \\ -0.03521 & 2.1522 \end{bmatrix}$. A solution is guaranteed because the domain of the hyperbolic tangent function is R and the range is a (-1, 1), which always has at least one intersection point with the linear affine function on the left of (13).

5. With the same technique used with W_{11}, we select $v_{x4} = 0.3$ and solve for $v_{x3} = 0.494563$. Then we have $W_{22} = \begin{bmatrix} 2.15568 & -0.03521 \\ -0.03521 & 2.1522 \end{bmatrix}$.

6. At this point, we have $V_x = \begin{bmatrix} 0.494563 \\ 0.3 \\ 0.494563 \\ 0.3 \end{bmatrix}$ and

$$V_{e1} = \begin{bmatrix} 0.5 \\ 0.25 \\ 0.494563 \\ 0.3 \end{bmatrix}, V_{e2} = \begin{bmatrix} -0.5 \\ 0.5 \\ 0.494563 \\ 0.3 \end{bmatrix},$$

$$V_{e3} = \begin{bmatrix} 0.494563 \\ 0.3 \\ 0.5 \\ 0.25 \end{bmatrix}, V_{e4} = \begin{bmatrix} 0.494563 \\ 0.3 \\ -0.5 \\ 0.5 \end{bmatrix}$$

For all of these V_e vectors, corresponding $Y_e = \tanh(V_e)$ vectors and I vector (calculated in the next step), it should be verified that they all satisfy (7).

7. Find the bias vector (I) from $I = -WY_x + GV_x$. It is $I = \begin{bmatrix} 0.012446 \\ -0.010841 \\ 0.012446 \\ -0.010841 \end{bmatrix}$.

8. Because, we start with V_{ti}, and we get V_{ei} from the above synthesis, we need some conversion between them. A conversion can be written as:

$$V_{ti} = C.V_{ei} \quad (14)$$

where

$$C = \begin{bmatrix} V_{t1} & V_{t2} & V_{t3} & V_{t4} \end{bmatrix} \begin{bmatrix} V_{e1} & V_{e2} & V_{e3} & V_{e4} \end{bmatrix}^{-1} \quad (15)$$

This technique is seen to be easily extended to higher order systems.

9. The last step is to use the inverse of the initial row operations to convert V_{ti} back to V_{di} (reverse order to the initial transformation).

4. SUMMARY

In this paper, we give a synthesis method to obtain symmetric weight matrices in the generalized Hopfield neural network given any set of linearly independent equilibrium point vectors. A solution is guaranteed because there is always at least one intersection point between the hyperbolic tangent and a linear affine function in R^2. This is important because asymptotic stability is based on the Lyapunov stability criterion and is guaranteed by this symmetric weight matrix. An easy to follow step-by-step procedure is given in the paper.

5. REFERENCES

[1] Li J.H., Michel A. N. and Porod W., "Qualitative Analysis and Synthesis of a Class of Neural Networks". *IEEE trans. Circuits and Systems,* vol. 35, no. 8, pp. 976-986, August 1988.

[2] Hopfield J. J., "Neurons with graded response have collective computational properties like those of two-state neurons". *Proc. Nat. Acad. Sci., USA,* vol. 81, pp. 3088-3092, May 1984.

[3] Khalil H. K., *Nonlinear Control Systems.* Second edition, New Jersey: Prentice Hall, 1996.