ISSN 0105-8517

Some New Thoughts on Neural Networks With Complex Connection Matrices

Miklos Szilagyi
Jens Christian Mikkelsen
Kjeld H. Mortensen

DAIMI PB – 322
July 1990

COMPUTER SCIENCE DEPARTMENT
AARHUS UNIVERSITY
Ny Munkegade, Building 540
DK-8000 Aarhus C, Denmark

Some New Thoughts on Neural Networks With Complex Connection Matrices

Miklos Szilagyi *
Jens Christian Mikkelsen †
Kjeld H. Mortensen ‡

July 1990

Abstract

In the literature neural networks are usually understood in the domain of real activation functions and connection weights. The concept of neural networks was generalized by one of us [Szi88] to include complex connections between complex units. In the present note this idea is further developed. A mathematical model is presented, an expression for the network's energy as well as a complex learning rule are proposed.

1 The Model

The model we have used to describe complex networks is the ordinary Hopfield network of N neurons [Hop84], which in the real case is described by N differential equations, also called the network equations:

$$\frac{du_j}{dt} = \sum_{k=1}^{N} w_{jk} f_k(u_k(t)) + I_j(t), \tag{1}$$

*Permanent address: Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ 85721, USA.
†E-mail address: jcm@daimi.dk.
‡E-mail address: khm@daimi.dk.

where $j = 1, 2, \ldots, N$. Here u_j are the unit input activations, f_j are the output activation functions and I_j are the external inputs. The weight matrix constituted by the w_{jk}'s is symmetric, so that $w_{jk} = w_{kj}$.

To make a model of a complex network, we will simply allow all of the values (except for the time t) to be complex. The only thing changed is the symmetry demand, which becomes a requirement that the weight matrix is hermitian: $w_{jk} = w_{kj}^*$.

Now let's try to split up the network equation into real and imaginary parts. We have

$$\begin{aligned} u_j(t) &= x_j(t) + iy_j(t) && (2)\\ w_{jk} &= p_{jk} + iq_{jk} && (3)\\ f_j(x+iy) &= g_j(x,y) + ih_j(x,y) && (4)\\ I_j(t) &= J_j(t) + iK_j(t), && (5) \end{aligned}$$

where $j, k = 1, 2, \ldots, N$, $t, x, y, p_{jk}, q_{jk} \in I\!R$, $x_j, y_j, J_j, K_j \in I\!R \to I\!R$ and $g_j, h_j : I\!R^2 \to I\!R$.

We can then rewrite the network equations as:

$$\begin{aligned} \frac{d(x_j(t) + iy_j(t))}{dt} &= \sum_{k=1}^{N} (p_{jk} + iq_{jk})(g_k(x_k(t), y_k(t)) + ih_k(x_k(t), y_k(t)))\\ &\quad + J_j(t) + iK_j(t)\\ &= \sum_{k=1}^{N} ((p_{jk}g_k(x_k(t), y_k(t)) - q_{jk}h_k(x_k(t), y_k(t)))\\ &\quad + i(p_{jk}h_k(x_k(t), y_k(t)) + q_{jk}g_k(x_k(t), y_k(t))))\\ &\quad + J_j(t) + iK_j(t), && (6) \end{aligned}$$

which we can separate into two differential equations, one for the real and one for the imaginary part:

$$\begin{aligned} \frac{dx_j}{dt} &= \sum_{k=1}^{N} (p_{jk}g_k(x_k(t), y_k(t)) - q_{jk}h_k(x_k(t), y_k(t))) + J_j(t) && (7)\\ \frac{dy_j}{dt} &= \sum_{k=1}^{N} (p_{jk}h_k(x_k(t), y_k(t)) + q_{jk}g_k(x_k(t), y_k(t))) + K_j(t). && (8) \end{aligned}$$

From now on we shall write these network equations somewhat simpler as

$$\boxed{\begin{aligned} \frac{dx_j}{dt} &= \sum_{k=1}^{N} (p_{jk} g_k - q_{jk} h_k) + J_j \\ \frac{dy_j}{dt} &= \sum_{k=1}^{N} (p_{jk} h_k + q_{jk} g_k) + K_j, \end{aligned}} \tag{9}$$

($j = 1, 2, \ldots, N$). Thus, we have described the network of N complex neurons by a system of $2N$ coupled differential equations in real variables.

2 The Energy of the Network

The energy function we would like to propose looks like

$$\boxed{E = -\frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} f_j^* w_{jk} f_k - \mathrm{Re}\,(\sum_{j=1}^{N} f_j^* I_j).} \tag{10}$$

or written in vector form (with $\vec{f}$ and $\vec{I}$ as column vectors):

$$E = -\frac{1}{2} \vec{f}^{+} W \vec{f} - \mathrm{Re}\,(\vec{f}^{+} \vec{I}). \tag{11}$$

Furthermore, since the matrix A with $f_j^* w_{jk} f_k$ as its element at row j and column k is hermitian, we have

$$(f_j^* w_{jk} f_k)^* = f_j w_{jk}^* f_k^* = f_k^* w_{kj} f_j, \tag{12}$$

and $\mathrm{Im}\,(A)$ must be antisymmetric with zeros in the main diagonal. Therefore,

$$\mathrm{Im}\,(\sum_{j=1}^{N} \sum_{k=1}^{N} f_j^* w_{jk} f_k) = 0, \tag{13}$$

so E is all real and we can alternatively choose to write (10) as

$$E = \mathrm{Re}\,(-\frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} f_j^* w_{jk} f_k - \sum_{j=1}^{N} f_j^* I_j). \tag{14}$$

2.1 Proof of Energy Relationship

We shall prove that the energy proposed is a non-increasing function of time. We see that

$$\begin{aligned}\mathrm{Re}\,(f_j^* w_{jk} f_k) &= \mathrm{Re}\,((g_j - ih_j)(p_{jk} + iq_{jk})(g_k + ih_k)) \\ &= g_j p_{jk} g_k + h_j q_{jk} g_k + h_j p_{jk} h_k - g_j q_{jk} h_k \\ &= p_{jk}(g_j g_k + h_j h_k) - q_{jk}(g_j h_k - g_k h_j) \qquad (15)\end{aligned}$$

$$\begin{aligned}\mathrm{Re}\,(f_j^* I_j) &= \mathrm{Re}\,((g_j - ih_j)(J_j + iK_j)) \\ &= g_j J_j + h_j K_j \qquad (16)\end{aligned}$$

which means that we can write (14) as

$$E = -\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}(p_{jk}(g_j g_k + h_j h_k) - q_{jk}(g_j h_k - g_k h_j)) - \sum_{j=1}^{N}(g_j J_j + h_j K_j). \quad (17)$$

Now let's prove that

$$\frac{\partial E}{\partial g_j} = -\frac{dx_j}{dt} \wedge \frac{\partial E}{\partial h_j} = -\frac{dy_j}{dt}. \quad (18)$$

First we see that by arguments similar to the ones in [Szi89] page 34, we have

$$\frac{\partial}{\partial g_j}\sum_{j=1}^{N}\sum_{k=1}^{N} p_{jk} g_j g_k = 2\sum_{k=1}^{N} p_{jk} g_k, \quad (19)$$

provided that $p_{jk} = p_{kj}$. But we know that the weight matrix is hermitian, so this must be the case. Similarly, we have, of course:

$$\frac{\partial}{\partial h_j}\sum_{j=1}^{N}\sum_{k=1}^{N} p_{jk} h_j h_k = 2\sum_{k=1}^{N} p_{jk} h_k, \quad (20)$$

while on the other hand:

$$\frac{\partial}{\partial h_j}\sum_{j=1}^{N}\sum_{k=1}^{N} p_{jk} g_j g_k = \frac{\partial}{\partial g_j}\sum_{j=1}^{N}\sum_{k=1}^{N} p_{jk} h_j h_k = 0. \quad (21)$$

Since the weight matrix is hermitian, we have $q_{jk} = -q_{kj}$. Thus,

$$\begin{aligned}
\sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk}(g_j h_k - g_k h_j) &= \sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk} g_j h_k - \sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk} g_k h_j \\
&= \sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk} g_j h_k - \sum_{k=1}^{N}\sum_{j=1}^{N} q_{kj} g_j h_k \\
&= \sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk} g_j h_k + \sum_{k=1}^{N}\sum_{j=1}^{N} q_{jk} g_j h_k \\
&= 2\sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk} g_j h_k. \qquad (22)
\end{aligned}$$

When we differentiate this with respect to g_j and h_j, respectively, we'll get

$$\frac{\partial}{\partial g_j}\sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk} g_j h_k = \sum_{k=1}^{N} q_{jk} h_k \qquad (23)$$

and

$$\frac{\partial}{\partial h_j}\sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk} g_j h_k = \frac{\partial}{\partial h_j}\sum_{k=1}^{N}\sum_{j=1}^{N} q_{kj} g_k h_j = \sum_{k=1}^{N} q_{kj} g_k = -\sum_{k=1}^{N} q_{jk} g_k. \qquad (24)$$

We should now be well equipped to look if the energy as written in (17) fulfils (18). For the first half of (18) we have by (9), (19), (21), (22) and (23) that

$$\begin{aligned}
\frac{\partial E}{\partial g_j} &= -\frac{1}{2}(2\sum_{k=1}^{N} p_{jk} g_k + 0 - 2\sum_{k=1}^{N} q_{jk} h_k) - J_j \\
&= -(\sum_{k=1}^{N}(p_{jk} g_k - q_{jk} h_k) + J_j) = -\frac{dx_j}{dt}. \qquad (25)
\end{aligned}$$

For the second half of (18) we correspondingly have by (9), (20), (21), (22) and (24) that

$$\begin{aligned}
\frac{\partial E}{\partial h_j} &= -\frac{1}{2}(0 + 2\sum_{k=1}^{N} p_{jk} h_k + 2\sum_{k=1}^{N} q_{jk} g_k) - K_j \\
&= -(\sum_{k=1}^{N}(p_{jk} h_k + q_{jk} g_k) + K_j) = -\frac{dy_j}{dt}. \qquad (26)
\end{aligned}$$

Using these two results, we can now look at the derivative of the energy E with respect to time t:

$$
\begin{aligned}
\frac{dE}{dt} &= \sum_{j=1}^{N}\left(\frac{\partial E}{\partial g_j}\frac{dg_j}{dt} + \frac{\partial E}{\partial h_j}\frac{dh_j}{dt}\right) \\
&= -\sum_{j=1}^{N}\left(\frac{dx_j}{dt}\frac{dg_j}{dt} + \frac{dy_j}{dt}\frac{dh_j}{dt}\right) \\
&= -\sum_{j=1}^{N}\left(\frac{dx_j}{dt}\left(\frac{\partial g_j}{\partial x_j}\frac{dx_j}{dt} + \frac{\partial g_j}{\partial y_j}\frac{dy_j}{dt}\right) + \frac{dy_j}{dt}\left(\frac{\partial h_j}{\partial x_j}\frac{dx_j}{dt} + \frac{\partial h_j}{\partial y_j}\frac{dy_j}{dt}\right)\right) \\
&= -\sum_{j=1}^{N}\left(\frac{\partial g_j}{\partial x_j}\left(\frac{dx_j}{dt}\right)^2 + \frac{\partial h_j}{\partial y_j}\left(\frac{dy_j}{dt}\right)^2 + \frac{dx_j}{dt}\frac{dy_j}{dt}\left(\frac{\partial g_j}{\partial y_j} + \frac{\partial h_j}{\partial x_j}\right)\right). \qquad (27)
\end{aligned}
$$

If we add the constraint that $\frac{\partial g_j}{\partial y_j} = -\frac{\partial h_j}{\partial x_j}$, the last line becomes:

$$
\frac{dE}{dt} = -\sum_{j=1}^{N}\left(\frac{\partial g_j}{\partial x_j}\left(\frac{dx_j}{dt}\right)^2 + \frac{\partial h_j}{\partial y_j}\left(\frac{dy_j}{dt}\right)^2\right). \qquad (28)
$$

Provided that $\frac{\partial g_j}{\partial x_j} \geq 0$ and $\frac{\partial h_j}{\partial y_j} \geq 0$, we see that $\frac{dE}{dt} \leq 0$. So, we must demand the following of the activation functions f_j to ensure that the energy is a non-increasing function of time:

1. $\frac{\partial g_j}{\partial x_j} \geq 0$ and $\frac{\partial h_j}{\partial y_j} \geq 0$.
2. $\frac{\partial g_j}{\partial y_j} = -\frac{\partial h_j}{\partial x_j}$.

(29)

Note that if f_j should be holomorphic, we know by the Cauchy-Riemann differential equations that:

$$
\frac{\partial g_j}{\partial x_j} = \frac{\partial h_j}{\partial y_j} \wedge \frac{\partial g_j}{\partial y_j} = -\frac{\partial h_j}{\partial x_j}, \qquad (30)
$$

so in that case it'll suffice to demand that $\frac{\partial g_j}{\partial x_j} \geq 0$.

2.2 Complex Energy

When one looks at (14), one is certainly tempted to investigate the complex number

$$E_C = -\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} f_j^* w_{jk} f_k - \sum_{j=1}^{N} f_j^* I_j, \tag{31}$$

and, in particular, $E_I = \mathrm{Im}(E_C)$. By (10) we easily see that

$$E_I \;=\; -\sum_{j=1}^{N} \mathrm{Im}(f_j^* I_j). \tag{32}$$

Note that, when the pattern constituted by the I_j's is identical to the output pattern, we have $f_j = I_j$ for all $j = 1, 2, \ldots, N$, so $I_j f_j^*$ is real. This means that E_I becomes zero then. It is, however, also possible for E_I to become zero without $f_j = I_j$ being true, e.g. if $I_j = 0$.

Other than this it is hard to see what E_I can be used for, but it could be interesting to keep it in mind till later.

3 Learning Rule

The energy surface has to be shaped in order to make the network learn. I.e. we are going to alter the entries in the weight matrix by means of a learning rule which is to be used during the training of the network. Each vector in the training set have to become an attractor. This is accomplished when minima are created in the energy surface such that if we present a vector which "looks like" one in the training set, the network completes it and ends up in the right attractor. The topology of the basins of attraction corresponding to each attractor determine the capability of the network to discriminate between individual inputs coming from the environment.

The learning rule we have developed is very much like the Hebbian learning rule as stated in [Szi89] page 31. Our version is:

$$\boxed{\Delta w_{jk} = \eta f_j f_k^*,} \tag{33}$$

or more compactly

$$\Delta W = \eta \vec{f} \vec{f}^{+}, \tag{34}$$

where $\vec{f}$ is a column vector of outputs and η is the learning rate which takes different values depending on the learning mode:

$$\eta \in \begin{cases} I\!R_{+} & \text{in learning mode} \\ \{0\} & \text{no learning} \\ I\!R_{-} & \text{anti-learning mode} \end{cases} \tag{35}$$

Anti-learning takes place when $\vec{I} = \vec{0}$, i.e. with no influence from the environment, and is used in order to improve the performance of the neural network. (This is also compared with REM[1] sleep in [KS].)

The weight matrix W is hermitian and should continue to be such during learning. ΔW is hermitian because

$$(\Delta w_{jk})^* = (\eta f_j f_k^*)^* = \eta f_j^* f_k = \Delta w_{kj}. \tag{36}$$

It then follows that W stays hermitian.

The energy increase ΔE must be non-positive. This can easily be verified from (11) and (34):

$$E + \Delta E = -\frac{1}{2}\vec{f}^{+}(W + \Delta W)\vec{f} - Re(\vec{f}^{+}\vec{I}),$$

which implies that:

$$\Delta E = -\frac{1}{2}\vec{f}^{+}\Delta W \vec{f} = -\frac{1}{2}\eta(\vec{f}^{+}\vec{f})^2 \leq 0, \tag{37}$$

when $\eta \in I\!R_{+}$.

In connection with this learning rule we performed some simple experiments with content-addressable memory. In this case the weight matrix is constructed analytically as described in [Szi89] p. 33:

$$w_{jk} = \sum_{s=1}^{p} V_j^{(s)} V_k^{(s)^*}, \tag{38}$$

[1]Rapid eye movements - appear during dream sleep.

where $\vec{V}^{(s)}$ are the output vectors (training patterns) and p is the number of patterns/vectors to be stored in the neural network. This, of course, follows from (33). The activation function used was $f(x+iy) = \tanh(x) + i\tanh(y)$ which is described in the next section.

The configuration was 24 neurons in the network and the training set contained 3 random vectors from $\mathbb{C}^{24}$. The performance of this network was succesful and it could reconstruct very noisy patterns. Often vectors with over 50% of the entries altered could be recognized[2].

4 Finding a Proper Activation Function

Finding the right activation function is not easy. We have the two requirements for this function as stated in (29) and these are not making the conquest any simpler.

The apparently obvious choice is to transfer the mostly used activation functions from the real to the complex domain. We have studied three, viz. $\tanh(z)$, $1/(1+\exp(-z))$, and $\operatorname{atan}(z)$. All of these have singularities in the complex plane which are $z = i(n+\frac{1}{2})\pi$, $z = -i(2n+1)\pi$, and $z = \pm i$, respectively ($n \in \mathbb{Z}$). None of the functions are limited in the neighbourhood of these singularities, and such properties are unwanted because we then cannot guarantee that the network will not end up in an undefined state. But the crucial point is that the functions do not satisfy our requirements in (29). More specifically, they all violate $\frac{\partial g_j}{\partial x_j} \geq 0$. (The three functions are holomorphic so they do fulfil 2. in (29).)

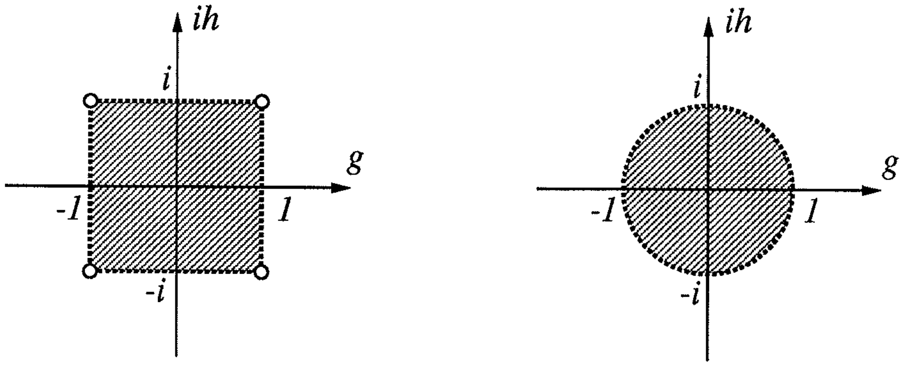

Figure 1: Output domains of two activation functions.

Another function one could think of is $f(x+iy) = \tanh(x) + i\tanh(y)$. It satisfies all of our requirements, but has the disadvantage of being

[2]This depends on the orthogonality of the original training vectors.

too simple. The function returns values confined to a box as shown in figure 1. Most often the output values end up in one of the four corners of the box. So, all entries in the training vectors should in this case be chosen as pointing to one of the corners.

In a Hopfield-type network, it is often the case that the outputs end up on the rim of the output domain (depending on the type of activation function). This could be an advantage to real valued networks which are bounded to values on the real axis. Take e.g. the circle-shaped domain in figure 1. Here each coordinate in the training vectors have some kind of a direction[3] which could be useful in the context of continous interpretations instead of just discrete as in the real valued network.

Our efforts until now haven't yielded a more compound activation function. But the search continues because it is very important to develop one that would show the real strength of complex-valued neural networks.

References

[Hop84] J.J. Hopfield. Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons. *Proceedings of the National Academy of Sciences*, volume 81, pages 3088–3092, May 1984.

[KS] R. Kühn and G. Sawitzki. Spinning Brain - An Interactive Program for Associative Recall of Visual Patterns. Document file for a public domain MacIntosh program.

[Szi88] Miklos Szilagyi. Neural Networks with Complex Connection Matrices. Technical Report DAIMI PB - 260, Computer Science Department, Århus University, August 1988.

[Szi89] Miklos Szilagyi. Neural Networks. Lecture Notes DAIMI FN - 27, Computer Science Department, Århus University, July 1989.

[3] In stable networks there would be both a variable direction and length.