WORKING NOTE AC87

Alan Creak
23 September 1993

DESCRIBING MODULAR NETWORKS.

I am writing down these ideas on modular networks partly so that I don't forget them and partly to encourage discussion. I take it as self-evident that if we're going to study modular networks we need to define what we're talking about, and agree on terminology. This can only be achieved satisfactorily by first defining the field of modular networks, and then attempting a systematic analysis, taking into account what appear to be the most significant characteristics of the networks and ensuring that all possibilities are covered.

What is a modular network ? The topic arose from Mark Scaletti's interest in modular neural networks(1), which were imagined as large neural networks composed by joining together several smaller networks, each with its own function. It has become clear to me, though, that it is more satisfactory to separate the properties of the component networks from those of the connections between them, so that the components are best treated as "black boxes". That being so, it doesn't matter if the components are not themselves neural networks. What we're doing really is thinking about big neural networks in which each "neuron" (which so far I've called a component network) can do very complicated things – and there's no more reason why these "neuron"s should be implemented by neural networks than there is for the simple neurons within a conventional network. In fact, unless it turns out that there are analogies between the intermodule connections and elements of the nervous system, it is probably sensible to define a modular network as a *network of modules*, and forget about the neural part entirely.

It would be nice to find a way to classify any network that might come along, but I don't think that's feasible as yet, except in a very simple way, because of the wide variety of possibilities. (An obvious distinction is that between feedforward and feedback networks, and Mark Scaletti has also distinguished between homogeneous and heterogeneous networks(1), but it isn't obvious that either of these is immediately fruitful in pointing out useful properties of the system. Indeed, Mark has decided against even attempting a classification on the grounds that we don't have enough understanding of what happens in modular networks to justify it. I argue rather that even a bad classification is better than none at all, because it offers a framework into which one can fit new knowledge, and it shows up gaps in knowledge, thereby suggesting potentially useful lines of enquiry. Of course, the new knowledge may not fit – but then it will be clear why the classification is bad, and we can make a new and better one.) Even if we can't manage a classification, though, we can surely attempt to formalise a network description in terms of its components – which will probably require a classification of the components. That's what I'm trying to write down here.

As a final comment in this introductory section, I observe that the notion of modularity is clearly inherently recursive. A simple multilayer perceptron can be seen as a modular network in which each layer is a distinct module. (It is stretching definitions a little to consider a layer as a modular *network* with the individual neurons as modules, because these are not connected together.) I don't think the possibility of hierarchies matters; the same principles should work at any level, provided we scrupulously avoid peering inside the components of our modular network. Indeed, the potential hierarchy is a good thing, and should be preserved with care, for it guarantees that the same techniques can be used to analyse systems at several levels if it should prove necessary.

DEFINITIONS.

A *module* is a processor with two *connections*, one for input and one for output.

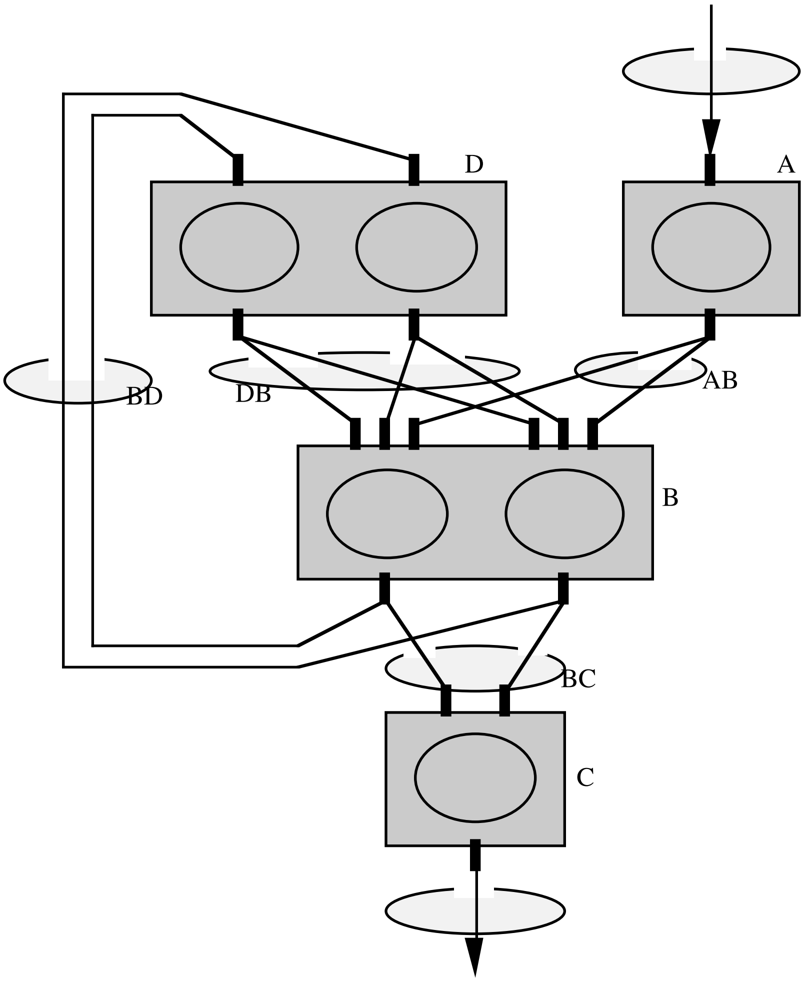
A connection consists of one or more *channels*, each carrying an elementary stream of information.

A channel accepts a single input sequence of values or produces a single output sequence of values. Connections are linked, output to input, by *links*.

Each link conveys a signal, typically represented as a set of parallel components, from one module to another. It is not thought of as a computing element, but may perform such transformations on signals as are needed in order to effect the communication between the modules. I shall later suppose too that some adjustable parameters may be associated with a link so that the network can adapt to circumstances. A link consists of a set of *lines*. I shall also find it convenient to suppose that all lines in a link carry related signals, so that the link can be thought of as a single multichannel communications medium; there may therefore be no link, one link, or several links between two modules. Channels are not necessarily uniquely associated with links; the sets of channels used by different links which meet at a connection may intersect.

Each line connects a single output channel, perhaps through some function, to a single input channel. The pattern of connections made by the lines, and the functions to which their signals are subjected, together determine the behaviour of the links of which they form the parts. Several lines may meet at the same input or output channel.

As a simple example of the terminology, consider this simple Elman network(2).



There are four modules.

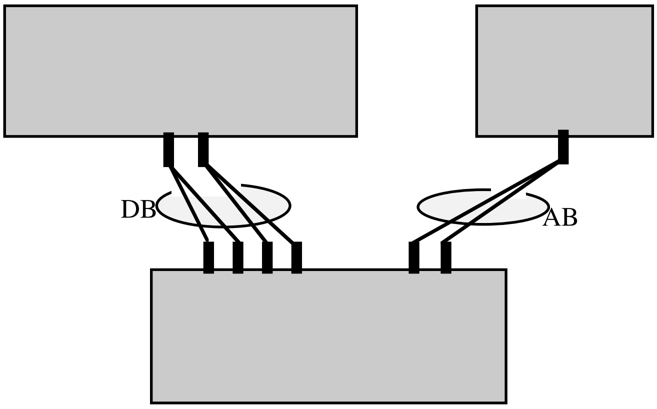
Module	Input channels	Output channels
A	1	1
B	6	2
C	2	1
D	2	2

There are also four links, apart from the input and output connections.

Link	Lines	Channels (in, out)	Remarks
AB	2	1, 2	Diverging lines
BC	2	2, 2	Parallel lines
BD	2	2, 2	Parallel lines; shares B's output connection with BC
DB	4	2, 4	Diverging lines; *doesn't* share B's input connection

No link incorporates any function; I have supposed, as is normally assumed, that the variable coefficients are components of the neurons, so they need not be implemented in the network. Clearly I could have made the other choice, and you may like to think about it again after contemplating the potentially recursive nature of the modular network definition; putting the coefficients in the network and leaving the functions in the modules is by no means a silly separation, and leads to a more elegant analogy between modular networks and simple neural networks.

I have drawn the diagram to bring out the underlying neural structure, but strictly speaking that's cheating; the internal structure of the modules is none of the network's business. The effect is most obvious in the representation of the input links to B. These could just as well be represented in this way :



Whether or not the appearance of shared connections suggested by the original representation is misleading is an interesting question, which I shall not address.

SIGNIFICANT FEATURES.

In order to attempt a description of any sort we must identify one or more properties of the objects concerned which will be satisfactory terms on which to base the description. A satisfactory property in this sense is one which is well defined, easy to determine, and significant in differentiating between objects in useful ways. (The colour of the box satisfies the first two criteria, but fortunately fails on the third.) In addition, it is neat and tidy if the properties chosen are essentially independent. It isn't *necessary* that they should be independent, but dependencies may mean that the significances of some properties may be affected by the values of others, which makes observations harder to interpret. What are the available features for modular neural networks ?

Mark Scaletti has identified three of these in his report(1) : the characteristics of the component networks, the characteristics of the connections between them, and the nature of any external control which is not effected by the component networks but must be imposed on the macroscopic network to make it operate. Others can certainly be identified : how, if at all, the system is to be trained, whether it is intended to handle time-dependent input or output, and so on.

I shall concentrate here on the first two properties mentioned : the characteristics of the component networks and the connections between them. These appear to satisfy the desirable criteria I mentioned, while the others are less easy to determine or appear to be dependent on the chosen two. It is worth remarking that the two categorisations of networks mentioned earlier – feedforward and feedback networks, and homogeneous and heterogeneous networks – fit into the description quite naturally. I readily agree that my choice may be less than optimal, but it's somewhere to start, and should at least give a view of the field from which we can identify something better.

THE COMPONENT MODULES.

Having declined to interest myself in the internal nature of the module, I have to regard it as characterised entirely by the signals accepted and produced. Even then, I must tread carefully, for I cannot necessarily use the module's function as a part of its specification. While it is undoubtedly an interesting fact about the module, it is of the nature of the modular networks under discussion that at least some of their modules may change in function from time to time as they learn new things. Here I am writing primarily of the precise mathematical function; it may well be that the function expressed in qualitative terms ("This module converts the input signals from the sensors into a binary representation") could be more reliable – but it is not obviously as useful.

What is left ? The *types* of the input and output signals – descriptions of the characteristics of their components. Whatever the machinery inside the module, it is likely to impose some restrictions on the sort of input which it can accept, and on the sort of output which it can generate. I now review a number of properties in terms of which a description of the signal might be expressed.

Channels : The first, and rather obvious, step is to count the channels. So far as I know, the result will always be an integer; while one can imagine signals arriving distributed over a flat continuous resistive sheet, I am ready to stipulate that such networks are not, as yet, of interest. That is not to say that the number of channels is constant. Many people work with networks which grow(3), and in which one can discern modules with varying connectivity. Commonly, though, they grow by proliferating their existing channels, thereby preserving the type of the connection, so it is usually (I know of no exception) possible to describe the signal transported in terms of an arbitrary number of channels of some specific type.

It is possible, though perhaps unusual, for channels of a connection to carry information of different types. The input connection of module B in the Elman network receives both input and recurrent information through different links, and one could regard layers of a back-propagation network as modules which receive both input signals and error signals, each arriving by its own set of channels and behaving in quite different ways.

Discrete or continuous values : The signal which is accepted or generated by a channel is a scalar, but it may be a continuous (real) number, perhaps bounded in some way, or it may be one value from a discrete set (commonly, though not necessarily, a set of two values). It seems rather unlikely that a channel will be able to change its nature from one to the other of these possibilities. I shall make the further realistic assumption that the values are represented by some scalar physical quantity, such as voltage, so that there will always be an order among the possible input or output values.

Discrete or continuous in time : Modules may accept input or produce output continuously in time, or they may only sample input or change their output at discrete moments, being oblivious to external events at other times. (I regard natural neurons as continuous in nature, even though they may receive and produce pulse trains, because the pulses may appear at any instant.) This property has nothing to do with digital simulation of continuous processes, for it is perfectly possible (though it may take a long time) to produce a simulation which is as good an approximation to continuous behaviour as may be desired for any sort of module behaviour that might be of interest. As in the previous example, this characteristic is a consequence of the module's machinery, and is most unlikely to change.

Persistent or spasmodic in time : I have included this characteristic just in case it's important; I am not entirely sure that it can be separated from the previous one. It embodies the distinction between input or output which is continuously valid, even though it may only change at discrete instants, and signals which are only valid during specific periods. I don't know of any genuine illustrative examples, but that may be simply because it's usually hidden in the code of a simple module. I can *imagine* an example, which may become real in Kim's work : a module to interpret Morse input signals, in which the character output is only valid when a complete character has been received. Like the previous two examples, this is a characteristic of the module's machinery, so is unlikely to be adjustable.

Function implemented : I remarked earlier that the function implemented by a module – the relationship between its input and output signals – is not necessarily a satisfactory feature for classification purposes, because it might change. It could equally be argued that the function might not change, and often doesn't, in which case the objection is groundless.

I can sidestep the question if I can demonstrate that the function is of no interest anyway. Unfortunately, I can't. If we ever manage to get anywhere near calculating global properties of networks, then the properties of the modules are evidently going to be important. For example, one could hardly pronounce on the stability of a network without knowing the properties of its modules.

I *can* demonstrate that the function is in some sense a different sort of property from those I have listed above. Consider Mark's distinction(1) between homogeneous and heterogeneous networks; the modules of a homogeneous network have all the earlier properties in common, but are likely to differ in their functions. Whether or not the distinction is useful, it recognisably identifies a real property of a modular network, and that is sufficient to make my point.

The functions can therefore not be dismissed as insignificant. It may be, though, that we can manage with less than a complete description of the function. To prove that a system is stable, it may only be necessary to know that the functions implemented by its parts are of certain sorts; full specification is not essential. (For example, $A\sin(\alpha\theta + \beta)$ is bounded for all real values of $(\alpha\theta + \beta)$.) It may therefore be sufficient to know something about the class of function implemented by the network; but I don't know just what that means.

In summary, I am reduced to rejecting the modules' functions as classification features because I don't know enough about how they might be significant. Any advances on this position are welcome.

THE COMMUNICATION BETWEEM THEM.

The other characteristic of the network is the parts which lie between the modules. Considering the intermodular links, I discern two major factors in their contribution to the network. First, they join the modules together to build up the whole network; and, second, each link may itself implement some function, mapping from the output of one module to the input of another, and possibly transforming signals of one class into another, as outlined in the previous discussion. These two factors are quite different in nature, so I shall discuss them separately.

Global properties.

By their very existence, the intermodular links define the macroscopic network topology. They define how the modules can communicate, which covers important network characteristics such as whether they should be considered feedback or feedforward, or both. I shall not discuss this topic further here, for two reasons. First, it is in the area of general systems theory, and is (no doubt) exhaustively treated in textbooks on the subject, and, second, in this area I find myself agreeing with Mark that, despite an abundance of examples, it is just not clear what's important.

Local properties.

When looking at a single link between two modules, we are on much firmer ground. Generally, every intermodular link joins the output, or part thereof, of one module to the input, or part thereof, of another (or, just possibly, the same one). We already know what sort of input and output signals we can have, so we can use this classification as a basis for discussing the links. I shall not attempt an exhaustive enumeration of possibilities, because I haven't time, but I shall raise some general points which seem to me to be of interest.

I think that there are two aspects of the link which must be discussed. One is essentially topological : how are the input channels connected to the output channels ? The other is more analytical : what functions are computed in the link ?

Taking the topological question first, I begin by establishing that it *is* a question. Why shouldn't it be a question ? (that's a metaquestion) – because we could, if we so desired, stipulate that lines in the links between modules should always be parallel, with one-to-one lines between the outputs of the source and the inputs of the destination. I think that position is untenable, simply because it doesn't do what we want it to do. A theory of modular networks must surely aim at generality, and a restriction of the sort suggested runs against that aim. More practically, if we want (eventually) to use the theory as a guide to designing modular networks, it must be able to cope with the networks we have, not those which it might be nice to have. Generally, whatever signal is communicated between the modules represents something or other, and it is quite likely that different sorts of module will require different representations. (That is perhaps more obvious if you suppose that one of the communicating parties is neural and the other isn't.) I shall assume that makes the point.

It is easy enough to describe the topology of a link by listing the lines, which is a matter of selecting from the complete set of possible combinations between input and output channels. I can think of no reason for rejecting any subset of the possible lines (except the empty set), though in most cases lines are likely to be established according to some pattern. Parallel, one-to-many and many-to-one links are examples. Many-to-many links may be complete (as between layers of multilayer perceptrons), or restricted to local "receptive fields". A pattern of some sort may be very important if the number of source or destination channels varies, as is possible if the component modules can grow or contract.

In any destination channel which is the target of a many-to-one connection, a combining function must be defined to determine how the several signals fix the value presented at the destination module's input channel. There are many possibilities. Obvious candidates with numeric values are sum, product, maximum, minimum; for logical values, possibilities include and, or, at least *N*, exactly *N*; and neither list even pretends to be exhaustive. Notice that the combining function is distinct from the functions discussed below, as it is an essential consequence of topological freedom, not an arithmetic, logical, or communications requirement.

These other functions constitute the second aspect of the link which I shall discuss. There are many reasons why we might wish the links to engage in some sort of computation. One may wonder whether such functions should be implemented as properties of the intermodular links or by providing separate coupling modules. I have taken the view here that they should be regarded as properties of the links so that the task of connecting together a lot of modules becomes simply one of identifying the link properties. I don't think that there is any point of principle involved, and if it proves more convenient to formulate the modular network by introducing additional modules I shall be quite happy.

The most obvious need for computation in a link is simply to match the characteristics of its two ends. If the input is continuously valued but spasmodic in time, and the output is discrete and continuous, matching them is not trivial. It may also be desirable to match numerical values : a module with unbounded output may feed one with bounded input, when some scaling, squashing, or other function must be applied.

A second reason for wishing to incorporate functions into the links is to allow the modular network itself to change in function, perhaps to learn some required behaviour or to adapt to varying circumstances. Again, there is no obvious reason to restrict the range of functions which may be used, but the analogy with neural networks suggests adjustable coefficients, perhaps combined with an additive function if the links are many-to-one. An advantage of allowing such functions is that it leads naturally to a recursive view of networks, with modules in larger networks acting as "neurons" in the obvious way.

In any block diagram, the links between the blocks can have different sorts of meaning(4). In a modular network, links are always likely to denote directed information channels of some sort, but different sorts of information may be present in different places. (I use "likely" rather than "certain" because it's a matter of convention. There are things about modular networks which could be shown as lines on diagrams, but are not customarily presented in that way. An example is the simultaneous adjustment of coefficients in different modules which is required in Tim's network, and perhaps in Waibel's network(5).) The two main sorts of information which turn up in many modular networks are data directly pertaining to the current problem, and control information. There may also be different sorts of data – so we might distinguish the forward data stream in a back-propagation network from the backward error stream. It seems likely that the best policy is to keep each link homogeneous; if there are different sorts of input channel, then arrange them into as many multiple links as are needed. I think it is desirable to regard each link as a single data stream which may be complex and embody many channels, but which nevertheless represents a single idea.

CONCLUSION.

It seems to me that I've found quite a lot to talk about – certainly more than I expected when I started to write it. Careful analysis has its value, and incidentally makes a lot of work.

If there's anything in what I've written, though, it at least provides some sort of vocabulary we can use to describe a network. It isn't yet adequate to perform any sort of detailed calculations – that's the bit I gave up on, and in any case we probably can't say enough about the behaviour of the individual modules – but a description is better than nothing, and the analysis has shown some considerations which will be important if we wish to construct quite general modular networks.

A final statement, perhaps not quite amounting to a conclusion, is that it's all part of general systems theory, or should be. Someone had better find out about it.

REFERENCES.

1 : M. Scaletti : *Modular neural networks for vision processing*, Auckland University Computer Science Report #76 (Auckland University Computer Science Department, September, 1993).

2 : J.L. Elman : "Finding structure in time", *Cognitive Science* **14**, 179 (1990).

3 : V. Honavar, L. Uhr : *Generative learning structures and processes for generalized connectionist networks*, Technical Report #91-02 (Department of Computer Science, Iowa State University, January, 1991).

4 : G.A. Creak : *Semantics of block diagrams*, unpublished Working Note AC69, January, 1990.

5 : A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang : "Phoneme recognition using time-delay neural networks", *IEEE Trans. Acoustics, Speech, Signal Processing* **37**, 328 (1989).