SUPERCOMPUTER COMPUTATIONS RESEARCH INSTITUTE

NEURAL NETWORKS AND CONFIDENCE LIMIT ESTIMATES

by

Bernd A. Berg and Ion-Olimpiu Stamatescu

swy809

FSU-SCRI-98-08

January 1998

THE FLORIDA STATE UNIVERSITY
TALLAHASSEE, FLORIDA

Neural Networks and Confidence Limit Estimates

Bernd A. Berg^{1,2,5}, and Ion-Olimpiu Stamatescu^{3,4,5}

¹ Department of Physics, Florida State University, Tallahassee, FL 32306, USA

² Supercomputer Computations Research Institute,
Florida State University, Tallahassee, FL 32306, USA

³ FEST, Schmeilweg 5, D-69118 Heidelberg, Germany

⁴ Institut für Theoretische Physik, Universität Heidelberg,
D-69120 Heidelberg, Germany

⁵ ZIF, Wellenberg 1, D-33615 Bielefeld, Germany

Abstract. We give a brief review of neural networks. This includes views and features about the general concept, the modelling problem for Neural Networks and their treatment with statistical physics methods. Their use as Expert Systems is illustrated for the application of Multilayer Perceptrons to High Energy Physics data analysis. Finally, we focus on recent statistical insight, which under certain conditions allows to obtain rigorous confidence limit estimates from a Neural Network analysis.

1 Views and Features of Neural Networks

There are various points of view concerning neural network models (in the following we shall generally use the shortening NN, sometimes ANN when we want to stress the “artificial” aspect):

- of cognition science, as models for cognitive processes;
- of artificial intelligence, as models for information processing devices;
- of physics, as statistical mechanics models;
- of neurophysiology, as models for brain processes.

While until the eighties the dominant paradigm in Artificial Intelligence (AI) and Cognition was the so called “physical-symbol-systems”, the neural networks have themselves a long history, starting with the *perceptron* concept (Rosenblatt 1962). It was, however, only in the last decades, that developments in neurophysiology, on one hand, and in the predictability and characterization of artificial networks, on the other hand, have permitted to set up a systematic modeling frame. There is very much literature, of course, in connection with neural networks and related problems (to quote only a few books: McClelland and Rumelhart 1986, Kohonen 1989, Churchland 1989, Amit 1989, Boden 1990, Bechtel and Abrahamsen 1991; Hertz, Krogh and Palmer 1991, Wasserman 1993, Domany et al. 1994, ...).

From the point of view of cognition neural networks correspond to a new paradigm, called "connectionism". In contradistinction with the "physical-symbol-systems" paradigm (corresponding to the "standard AI"), which is a top-down approach, "connectionism" represent a bottom-up approach in which "intelligent information processing" need not be *designed* but *results* from the activity of big assemblies of multiply interconnected, simple units. A cognitive science view is to consider NN as modeling "the micro-structure of cognitions" (McClelland et al. 1986). This makes NN adequate for modeling associative recognition of visual patterns (with the neurons coding the pixels), as well as for modeling semiotic processes, e.g., learning languages (with neurons coding words). In this connection the interesting features of neural networks appear to be:

- the "self - organization" aspect,
- the inherent associative character of their information flows,
- the natural way of accounting for multiple simultaneous constraints,
- fuzziness,
- stability under small disturbances,
- easiness in adaptation to new requirements, etc.

Of course, there are also disadvantages, some of them ensuing from just these advantages, e.g., the fuzziness. For applied AI neural networks represent devices which adapt themselves to the given problem. It is this aspect, together with their flexibility and their special capacity of treating problems with many constraints that have promoted ANN to important information processing devices. However, just because of this it is much more difficult to prove existence of solutions, to asses the time of convergence and to estimate the possible errors and their consequences. Some results have been obtained hereto, many of them due to application of methods from statistical mechanics.

The statistical physics interest for ANN is related to the phase structure of their state space. Some of these phases are characterized by a strongly degenerated ground state structure, typical for frustrated systems. It is just this aspect that is related to the pattern classification and associative recognition capabilities of these systems used as information processing devices. This approach has introduced a new perspective in AI, namely the statistical one. Before entering these matters, we want to sketch the "neurophysiology connection", since this is not only important for itself: in fact a good part of the interest in ANN is based on the assumed relation between their functioning and that of the brain.

Detailed information about the brain and its activity is provided by the modern neurophysiology (see, e.g., Kandel et al. 1995). The brain is a highly interconnected system (for the about 10^{11} neurons there are between a few and as many as about 10^5 synapses per neuron - one quotes an average synapse number of 10^4). We find:

- mostly threshold neurons (with essentially two states: active/inactive; the threshold behavior is however only approximative); but there are also linear response neurons;
- local as well as distant connections, which can be bundled, divergent or convergent;
- non-modifiable and modifiable synapses, and the modifiable synapses can show non-associative and associative modulation, with short term and long term effects;
- excitatory (mostly axo-dendritic), inhibitory (mostly axo-somatic) and modulatory synapses (axo-axonic: local modulation of the synapse from a second to a third neuron);
- slow and fast synapses (in fact, a wide distribution of synaptic delays);
- electric and chemical synapses, three important types of ions channels among very many others involved in the synaptic transmission, with a large variety of functions;
- biased topological mapping intermixed with functionally determined arrangements (e.g., visual field mapping and orientation columns in the visual cortex);
- various hierarchies of neurons and group structures with alternate interconnectivity and projection from one group into another, sometimes one-directional, sometimes with feedback.

Many of these characteristic features are believed to be represented in the construction of ANN. However, we would like to stress the observation – which is apparent from the above – that together with some general, more or less invariant features (like the predominance of threshold behaviour, high interconnectivity, synapse plasticity) there is a strong diversity of features, such that one could say that *for practically each problem there seem to be at least two – but typically many more – mechanisms or implementations*. To the extent that we want to use the “brain paradigm” for ANN modeling it is very important to try to assess correctly which part of the variability hinted at above is functional, which part represents stabilizing fluctuations (and therefore in a different sense is again functional) and which is uncontrollable, mostly harmless “noise”. This is to say, that both the use of NN for modeling brain activity *and* the use of neuro-physiological “hints” for developing ANN is not straightforward.

In what follows we shall first briefly describe the modeling problem for ANN and the main results hereto, then consider an application to automatic analysis of noisy experimental data and see how can we – in this particular example – control the inherent fallibility in the functioning of an ANN by turning it into limits for confidence levels for the results.

2 The Modeling Problem for Neural Networks

Typical problems for ANN modeling are:

- classify, memorize and retrieve patterns;
- reproduce associative structures between complex stimuli;
- develop optimal behavior under complex constraints.

These problems are related and in fact one can characterize the capacities of ANN by restricting oneself to one of these problems, namely that of storing and associative retrieving of patterns. This is at the same time one of their primary uses as information processing devices (notice that pattern classification and generalizations questions can be put in the same context – of associative recognition).

In establishing a neural network model we must define:

- (a) the architecture, *i.e.* the type of units and their connections,
- (b) the functioning, *i.e.* the rules controlling the activity of the network, and
- (c) the learning (or training) algorithms, by which the desired response or behavior is ensured.

2.1 Architecture.

The main type of neuron is the non-linear response unit. It is characterized by some kind of “sigmoid” function g , which describes the output V (one may think in terms of voltage) produced by submitting the neuron to a certain input h :

$$V = g(h) \quad (1)$$

e.g.

$$g(h) = c \tanh(a(h - b)) + d \quad (2)$$

In the limit of a very steep sigmoid we recover the two-state unit, usually termed Ising neuron if the $\sigma = \{\pm 1\}$ representation is used to describe the states *active/inactive* or McCulloch-Pitts neuron (McCulloch and Pitts 1943) if the $s = \{1, 0\}$ representation is used instead. There is, of course, a simple transformation rule between the various representations: $\sigma = 2s - 1$. However, different functioning or training laws prefer different representations in the sense that they take a more simple and intuitive form in the preferred representation.

These units¹ introduce a local non-linearity by transforming a more or less linear excitation potential into a threshold controlled response. They make synapses on other units and the connectivity characteristics (fully or partially connected nets, hierarchical or sheet-type organization of the connections, symmetric or asymmetric connections, etc.) defines the architecture of the network.

¹ There are, of course, also other kind of units, as linear response, multi-state units, etc. We shall not discuss them here, since they either cannot be used in networks to produce complex functions – like the former – or do not bring in genuine new kind of behavior – like the latter.

2.2 Functioning.

The functioning of the network is defined in the following way: the synapses made by the neurons $\{j\}$ on a neuron i produce signals $C_{ij}V_j$ depending on the outputs of j , V_j and on the "synaptic weights" C_{ij} . All these signals then add to an input for the neuron i

$$h_i = \sum_{j=1}^N C_{ij} V_j \quad (3)$$

which then responds according to (1). Here N is the number of neurons making synapses on i .

The biological similitude is most easily seen in the McCulloch-Pitts representation: firing of neuron j propagates and generates a presynaptic potential C_{ij} at the neuron i (C_{ij} can be positive or negative, according to whether the synapse ij is excitatory or inhibitory). The signals from all synapses at i add to form a post-synaptic potential:

$$h_i = \sum_{j=1}^N C_{ij} s_j, \quad s_j = 0, 1 \quad (4)$$

This potential then may trigger a firing of neuron i according to the rule:

$$s_i = \theta(h_i - \theta_i) \quad (5)$$

with θ_i a threshold characterizing neuron i . In the Ising representation this reads:

$$h_i = \sum_{j=1}^N C_{ij} \sigma_j, \quad \sigma_j = \pm 1 \quad (6)$$

and correspondingly:

$$\sigma_i = \epsilon(h_i - \theta_i) \quad (7)$$

(of course the synapses and thresholds have to be re-adapted when changing the representation).

This deterministic procedure can be replaced by a stochastic process, e.g. by introducing a noise component in the thresholds θ_i , e.g.:

$$\theta_i = \theta_i^0 + \beta \omega_i, \quad \overline{\omega_i} = 0, \quad \overline{\omega_i \omega_j} = \delta_{ij}$$

(a construction akin to the Langevin equation of brownian motion and therefore called sometimes Langevin dynamics) or by replacing the rule (7) with a probabilistic one, e.g.:

$$\sigma_i = 1 \text{ with probability } p = \frac{1}{1 + e^{-2\beta(h_i - \theta_i)}} \quad (8)$$

(the so called Glauber dynamics) or

$$\sigma_i = 1 \text{ with probability } p = \min [1, e^{-\beta(h_i - \theta_i)}] \quad (9)$$

(the so called Metropolis dynamics) – in both cases $\sigma_i = -1$ with probability $1 - p$. In (8) and (9) $1/\beta$ represents a kind of “temperature”. The zero – temperature limit is the deterministic rule (5), (7).

The process can go synchronously (at each step all neurons fire simultaneously) or asynchronously, in particular, serially². Nontrivial time dependence may be introduced by assigning synapse delays d_{ij} and letting the post-synaptic signal decay in time with some monotonically decreasing function $e(t)$ such that:

$$h_i(t) = \sum_j C_{ij} \sigma_j e(t - t_j - d_{ij}), \quad e(t) = 0 \text{ for } t < 0 \quad (10)$$

The synapses (the weights C_{ij}) may be taken symmetrical or asymmetrical. The Hopfield model is defined by using the Ising representation and assuming symmetric weights

$$C_{ij} = C_{ji}, \quad C_{ii} = 0$$

for the potential (6) in a fully interconnected network with N neurons. Figure 1 illustrates the connectivity of a Feed - Forward Net or multi-layer perceptron versus the connectivity of the Hopfield model. In a fully interconnected, symmetric network (*Hopfield network*) the activity patterns change each step but typically get locked after few steps in some “attractor”: a fixed activity pattern which from now on will reproduce itself at each step. This happens for a whole set of starting activity patterns which do not differ too much from the “attractor”. In a *feed - forward* network the neurons in the input sheet activate through the synapses relating them to the next sheet an activity pattern in the latter (this can be the output sheet or some intermediary sheet – in the latter case the procedure is repeated until it reaches the output sheet). Again, whole classes of input patterns would lead each time to equivalent output patterns. We see therefore directly the pattern recognition or pattern classification properties. Optimization, control and other problems can be implemented using the same basic functioning.

2.3 Learning.

The final step in setting up an ANN model is the definition of the procedure to tune the synapses (weights) C_{ij} to solve the given problem – e.g., to recognize classes of patterns. Of course, we can fix *by hand* the weights: for instance, if we know the patterns which we want to be produced in a Hopfield net we can just use a simple transformation (“the pseudo-inverse”) to fix the weights such that exactly these patterns will be attractors in the

² For a continuous-response function $g(h)$ one can also consider a continuous dynamics using time-derivatives.

Fig. 1. Fully connected Hopfield Net and Feed - Forward Net. Neurons are depicted as circles.

functioning of the network as described above. However, in problems like that of classification, of self-organization or of optimization *we do not know the attractors*³. Interesting about the neural networks is that not only we do not need to know how to implement a given structure but we do not even need to know what structure to implement: we only need to know the problem, that is, *what we want the network to do*. Then we simply let it work and use an algorithm which will tune the weights such that after some time the desired behavior is reached: the terms "learning" or "training" have therefore very intuitive meanings in connection with these algorithms.

The perceptron learning rule, which is the basic supervised training algorithm, is given here for an array of input Ising neurons $\{\sigma_j\}$ projecting onto a single output Ising neuron σ . It uses for each given input the difference between the desired output and the actual one to continuously adjust the weights. If we load the input array with the patterns $\{\sigma_j^\mu\}$, $\mu = 1, 2, \dots$ the learning rule is

$$C_j \rightarrow C_j + \epsilon \sum_{\mu} (\eta^\mu - \sigma^\mu) \sigma_j^\mu \quad (11)$$

with η^μ the *desired* output to the input σ_j^μ which has lead to the *actual* output σ^μ of the output neuron (ϵ is a parameter used for further tuning of the algorithm). In problems of classification and generalization a one-layer perceptron will achieve good results for linearly separable "pattern" spaces. For nonlinear classification problems multi-layer networks are used. For feed-forward multi-layer networks the formal algorithm of "back-propagation" (see, e.g., Rumelhart, Hinton and Williams in McClelland and Rumelhart 1986) can be shown to have good convergence properties. All these are "supervised training" algorithms, where the weights ("synapses") change proceeds in accordance to the correctness of the behavior. We shall consider continuous response neurons and multilayer perceptrons in section 4, after quoting a number of results from the statistical mechanics of the Hopfield networks in the next section.

³ Moreover, there is no evident biological mechanism implementing such a procedure, hence the heuristics based on the brain analogy are not effective.

From the point of view of the biological learning the most evident algorithm is the so called "Hebb rule" which asserts that there is a feedback from a firing neuron onto all the synapses which contributed positive presynaptic potentials to its firing and this feedback increases the corresponding weights. This is an "unsupervised training" algorithm which for a perceptron amounts simply to remove σ^μ in eqn.(11). For a fully connected net this rule takes the form

$$C_{ij} \rightarrow C_{ij} + \epsilon \sum_{\mu} \sigma_i^{\mu} \sigma_j^{\mu} \quad (12)$$

which can be interpreted as a self-organization rule, the input being iterated. The neurophysiological mechanism of "potentiation" could be thought of staying at the basis of the Hebb rule, but on one hand the neurophysiology of the synapse plasticity is very complicated and on the other hand the simple rule (12) must usually be further refined to provide good learning properties. Generally, ANN will always find themselves between the alternatives of optimization (as far as formal procedures can be obtained by just mathematical reasoning) and of biological motivation (which, under circumstances, may be vital, since not all problems allow easy formalization). More refined networks and learning rules include "sparse coding", "resilient propagation" or even "evolutionary algorithms" governing the development of the architecture itself (see, e.g., the articles of Horner and Kühn, and of Menzel in Ratsch et al. 1997). Finally, one can also consider the problem of "unspecific reinforcement", where the evaluation gives only the average error for a whole set of training steps (see Stamatescu 1996).

If one sees the rules for functioning (information processing) as "program", then one can say that ANN are essentially self-programming devices: they develop their own programs according to the requirements to which they are subjected during training.

3 The Physics of Neural Networks

As already noted, the present development of neural networks is to a large extent due to their opening to concepts and methods of statistical physics. The basis of this opening, besides the fact that ANN represent typical systems with many degrees of freedom, simple units and high interconnectivity, is that the questions one wants to have answered are of statistical character. If we think of biological networks (brains!), each is involved in another set of similar experiences and confronted with a different set of similar problems. We are not interested to find out which *certain* set of patterns can be stored by which *certain* network, but, say, which average number of more or less randomly occurring patterns can be stored as function of some general characteristics of a class of networks. It is also these kind of question which statistical physics method can help answering in connection with the AI applications of ANN. In this frame computer simulations, which have already a good tradition in

statistical physics, provide a kind of "intermediate phenomenology" useful both for the heuristic interpretation and for testing.

The attractors of the Hopfield model, which has been mostly studied, can be shown to represent minima of the energy function:

$$E = \frac{1}{2} \sum_{i,j} \sigma_i C_{ij} \sigma_j + \sum_i \eta_i \sigma_i \quad (13)$$

The external field η is related to the thresholds θ introduced in section 2. Then the rule (7) is an algorithm for minimizing the energy E and the stochastic procedures (8), (9) are algorithms for minimizing the free energy F obtained from the partition function

$$Z = e^{-F} = \sum_{\sigma_1=\pm 1} \sum_{\sigma_2=\pm 1} \dots \sum_{\sigma_n=\pm 1} e^{-\beta E}$$

This gives us at hand the whole arsenal which was or is developed in connection with spin glasses and other multi-connected statistical mechanics systems. The questions approached in this way concern:

- attractors, basins, stability, fluctuations and noise;
- collective phenomena, symmetries and order parameters, phase transitions, multi-scale phenomena;
- convergence questions for learning and for combined dynamics (activity and learning); ...

Summing up these problems under the general question of the structure of the state space, which bears a direct relation to that of pattern storing and retrieving, the general result (see, e.g., Amit 1989) is illustrated in figure 2, where α is the number of patterns divided by the specific number of synapses – the average number of synapses per neuron). Storing many patterns is equivalent to asking for multiple minima of the energy function E . This is typically achieved by a mixed structure of excitatory and inhibitory synapses, which provides many degenerated ground states serving as attractors for the network activity. Given a network and a number of random patterns which we want to store we use an algorithm of type (12) by which the patterns $\{\sigma_i^\nu\}$ are learned one after the other:

$$C_{ij}^\nu = C_{ij}^{\nu-1} + \epsilon (1 - \delta_{ij}) \sigma_i^\nu \sigma_j^\nu \quad (14)$$

(in fact a refined, optimized form of such an algorithm is used). We then find:

1. a paramagnetic phase above some temperature T_g in which the activity has no fixed points;
2. a spin glass phase between temperatures T_g and T_M characterized by many meta-stable states: the activity of the network visits these states for shorter or longer times but does not find the patterns learned under (14);

Fig. 2. The phases of the Hopfield model

3. a phase of retrieval states below T_M : the stored patterns can be retrieved as fixed points, they are even
4. true global minima of the free energy below T_c ;
5. finally, below T_{sp} spurious fixed points appear as mixtures of the learned patterns.

As can be seen from these considerations there exists a maximal value α_c above which retrieval is no longer possible ($\alpha_c \simeq 0.14$ for $T = 0$).

New developments put forward the question of non-trivial time structures. On one hand one wants to implement the synapse delay structure observed in biological systems and obtain in this way memorization and retrieval of time series. These are realized as limiting cycles of the activity and typically involve asymmetric synapses (again, biologically justified). On the other hand a more fundamental question is whether only the average firing rates carry information or there exist well timed events in the brain. That is, the timing of spikes may be relevant. While there is little secured biological information about the latter question, strong motivation for studying such aspects come from the cognitive problems known as "binding": how do we recognize separate features in a complex structure ("pattern segmentation") and how do we realize that a number of separately recognized parts build one complex pattern ("feature linking")? Problems which are of course of interest also from the point of view of developing powerful information processing devices. The study of such non-trivial time evolutions needs an enlargement of the frame first set by statistical physics – which essentially deals with equilibrium phenomena. These developments depart in fact from the Hopfield model as introduced above. This model still remains paradigmatic for the behavior of neural networks, however, and the amount of results obtained for it represents a good knowledge basis for the study of general ANN.

Finally, an important question is that of characterizing the performance of ANN beyond average behaviour: typical fluctuation, worst case, etc, which is relevant for ANN applications. A first step is to quantify the effect of occasional "wrong behavior" (incorrect recognition, misinterpretation, false association) from the point of view of the reliability of the result. A particular example will be studied in the next sections.

4 Multilayer Perceptrons and Data Analysis

Feed-forward ANN have become an increasingly popular tool for solving feature recognition and function mapping problems in a wide range of applications. In this section we illustrate this for High Energy Physics (HEP) data analysis. To date, the most commonly used architecture is the Multilayer Perceptron (MLP), which generalizes the perceptron concept of Rosenblatt 1962. The introduction of "hidden layers" (sheets of neurons which do not have direct connection to the exterior) permits tackling the problem of non-linear classification, that is, where the classification cannot be represented by a hyperplane in the pattern space. In fact it can be shown, that for most non-linear problems one hidden layer of non-linear neurons is sufficient⁴. Notice that linear response neurons cannot be used for this job, since then the hidden layers can be transformed away by a linear transformation. For a perceptron with L input neurons, 1 hidden layer with l neurons and 1 output neuron, with continuous, non-linear response functions (sometimes also called analogue neurons) the network function is defined by

$$Y_k = g \left[\sum_{i=1}^l c_i g \left(\sum_{j=1}^L C_{ij} d_j^k + \Theta_j \right) + \theta \right] \quad (15)$$

where $g(x)$ is a non-linear neuron activation function, e.g. (8)

$$g(x) = 1/[1 + \exp(-2x)]. \quad (16)$$

The forward propagation from one layer with 3 nodes to the next layer with 3 nodes is depicted in the right part of figure 1. Also included in the definition of the network function are the activation thresholds θ , Θ_i . The subscript $k = 1, \dots, N$ labels the input data, $\{d_j^k\}$. The forward-weights c_i of the hidden layer contract to a single node, and the input $\{d_j^k\}$ leads to the output Y_k . Usually the MLP is used in the analogue mode (continuous response function g). In this way for each data point k a single output number $0 < Y_k < 1$ is obtained, which we shall interpret after discussing the involved issues in more detail.

In experimental HEP a single data point d_i^k , $i = 1, \dots, L$ is described by the kinematic variables recorded by the detector and already processed by

⁴ See, e.g. Hertz, Krogh and Palmer 1991.

some track re-construction software or hardware. For instance, according to the Standard Model each top quark of a produced $t\bar{t}$ pair preferably decays into a b quark and a W boson. The all-jets channel consists of events in which both W bosons decay into light quarks observed as jets. Such jet events may be described by kinematic parameters like d_1^k centrality, d_2^k aplanarity, d_3^k average number of jets, d_4^k sum of non-leading transverse jet energies, and possibly others. Fed with such parameters (from the input layer), the task of the MLP would be to decide whether an event k is a $t\bar{t}$ signal or background. In the particular case of searching for $t\bar{t}$ in the all-jets channel, the signals are obscured by an enormous background consisting of events from other QCD and semi-weak processes. However, the method discussed does by no means depend on this example or any particular use of kinematic variables.

Quite commonly one faces a situation where the probability densities ρ_s for signals and ρ_b for background overlap. An example with two kinetic variables is depicted in figure 3, for which the optimal dividing line (cut) between signals and background is explicitly known. Plotted are 160 signals and 160 background data, generated by Monte Carlo (MC). From these data best cuts at constant $x = 0.798$ or $y = 0.619$, respectively are calculated and also depicted in the figure. Obviously it is non-trivial to find a reasonable approximation to the theoretical line from the data, even if the number of data can be significantly enlarged. The situation gets worse in realistic HEP applications, where normally a data point is described by $L > 2$ variables and a $L - 1$ -dimensional hypersurface ought to be determined. Although the parametric form of the optimal hypersurface between signal and background is typically unknown, theory may still allow to generate a large number of artificial signals and background events through a MC event generator like for instance HERWIG (Marchesini et al. 1992). In the following such data are called *training sample*. The AI aspect of a MLP manifests itself in the fact that it can convert the information of the training sample into a parametric form for an estimated optimal hypersurface, dividing signals and background. This is achieved by training the MLP, what amounts to minimizing the function

$$E = \sum_{k=1}^N (Y_k - \delta_{bk})^2 \quad \text{where} \quad \delta_{bk} = \begin{cases} 0 & \text{for } k \text{ signal,} \\ 1 & \text{for } k \text{ background,} \end{cases} \quad (17)$$

with respect to the MLP weight parameters $\{C, c\}$ and threshold parameters $\{\Theta, \theta\}$ defined in (15). The choice $\delta_{bk} = 1$ for background and 0 for signal is, of course, definition and could be interchanged without altering the essence of the problem. Important is that the number of data N exceeds by far the number of free parameters $\{C, c\}$ and $\{\Theta, \theta\}$. Otherwise over-fitting may result, what means that the trained MLP reflects then primarily properties of the particular training sample and not generic properties of their probability densities.

Due to the non-linearity of the MLP it is non-trivial to find the global minimum of E (17). However, for practical purposes this is not necessary either,

Fig. 3. Data with overlapping signal $\rho_s(x, y)$ and background $\rho_b(x, y)$ probability densities.

instead any solution will be acceptable that allows later on to disentangle signals from background. Of course, better minima will lead to an increased efficiency. For practical purposes improved variants of backpropagation algorithm (see, e.g. Hertz, Krogh and Palmer 1991, Wasserman 1993, Menzel in Ratsch et al. 1997) work well and are distributed in the JETNET package (Peterson et al. 1994). Backpropagation is a gradient descent minimization procedure, which in its simplest form reads:

$$\Delta c_i = -\eta \frac{\partial E}{\partial c_i} = \eta \sum_{k=1}^N \delta^k V_i^k, \quad \delta^k = g'(h^k)(\delta_{bk} - Y_k) \quad (18)$$

$$\Delta C_{ij} = -\eta \frac{\partial E}{\partial C_{ij}} = \eta \sum_{k=1}^N \Delta_i^k d_j^k, \quad \Delta_i^k = g'(H_i^k) c_i \delta^k \quad (19)$$

η being an optimization parameter. Here $V_i^k = g(H_i^k)$ are the outputs of the hidden neurons when the input d_j^k is set in the input layer and

$$H_i^k = \sum_{j=1}^L C_{ij} d_j^k, \quad h^k = \sum_{i=1}^l c_i V_i^k \quad (20)$$

are the corresponding inputs to the hidden, respectively output neurons. Since for large networks simple gradient descent may be rather time consuming one uses various improvements over this rule, e.g. conjugate gradient or other methods. Although in principle 1 hidden layer is sufficient for reproducing any smooth hypersurface in a classification problems, this tells nothing about

the optimal architecture. Presently, a general theory which determines the optimal number of hidden layers and their nodes does not exist and application are mainly influenced by previous practical experience. To avoid overfitting, equation (15) with a small number of nodes is a good starting point. Subsequently, one may employ more sophisticated MLP networks and find smaller E -values (17) through trial and error.

A trained network is characterized by two efficiency functions: the probability $F_s(Y)$ for tagging signals and the probability $F_b(Y)$ for tagging background events, where Y is a cut-off parameter on the Y_k values of (15). A Data point k is *tagged* by the MLP, when it returns $Y_k < Y$. Suppose, a particular network implements for some Y the optimal cut of figure 3. By counting we find 108 tagged signals and 50 tagged background events, hence the efficiencies $F_s = 108/160 = 0.675$ and $F_b = 50/160 = 0.3125$. Here and in the following we omit the arguments Y , when we deal with efficiencies for a fixed choice of Y . Quite generally

$$0 < F_b < F_s < 1$$

is the (reasonable) range for the efficiencies and in the following we assume that their error is negligible. In practice, there are of course errors due to the finite size of the training sample and, possibly more severe, due to uncertainties in the method of generating the training sample. Their relevance is best explored by repeating the entire approach with appropriate changes. See the next section.

Once the network is trained, it may be applied to N real data and it will return

$$0 \leq N^Y \leq N \quad (21)$$

tagged data. From this the mean value for the number of signal in the N events is estimated to be

$$N_s = \frac{N^Y - F_b(Y) N}{F_s(Y) - F_b(Y)}. \quad (22)$$

Notable with this equation is that the r.h.s. depends on Y , whereas the l.h.s. does not. In practice this amounts to a stability criterium, namely the cut-off Y should be picked in a region where the r.h.s. of (22) defines a flat function of Y . In the next section we investigate the confidence limits of the thus estimated signal likelihood $p = N_s/N$.

5 From MLP Output to Confidence Limits

We summarize some recent work (Berg and Riedler 1997), which rigorously solves the problem of determining signal confidence limits from the output (21) of a trained MLP with known efficiencies. The approach works for all methods (including conventional cuts) which tag signals versus background with different, known efficiencies.

5.1 Generalized Clopper–Pearson approach

For the situation where signals are observed directly, Clopper and Pearson (1934) (for textbooks see Brandt 1983 or van der Waerden 1971) derived a method, which relies on the binomial distribution and allows to calculate rigorous confidence bounds on the signal probability $p = N_s/N$. The probability to observe N_s signals within N measurements is given by the binomial probability density

$$b(N_s|N, p) = \binom{N}{N_s} p^{N_s} q^{N-N_s}, \quad q = 1 - p. \quad (23)$$

We are faced with the inverse problem: if N_s signals are observed, what is the confidence to rule out certain p ? Assume that probabilities p_-^c and p_+^c are given (the superscript c stands for “confidence”) and we like to find the following bounds (no superscripts) on the (unknown) exact signal likelihood p : The *largest* p_- and the *smallest* p_+ such that for every (hypothetical) p , $p < p_-$ with probability p_-^c (or less) and $p > p_+$ with probability p_+^c (or less). For instance, with $p_-^c = p_+^c = 0.159$ this guarantees $p_- \leq p \leq p_+$ with the standard one error bar confidence probability of 68.2%. In fact, for almost all p the actual confidence turns out to be better, but (due to their definition) the $p_{\pm}$ bounds cannot be improved without leading, for some (hypothetical) p -values, to violations of the imposed probabilities $p_{\pm}^c$. Clopper and Pearson proved that the thus defined lower p_- and upper p_+ bounds are given as solutions of the equations

$$p_-^c = \sum_{k=N_s}^N b(k|N, p_-) \quad \text{and} \quad p_+^c = \sum_{k=0}^{N_s} b(k|N, p_+) \quad (24)$$

with the additional convention $p_- = 0$ for $N_s = 0$ and $p_+ = 1$ for $N_s = N$ (for a graphical illustration see Berg and Riedler 1997).

We are interested in the more involved situation of the previous section where a NN (or similar device) yields statistical information by tagging signals and background data with different, known, efficiencies: F_s and F_b . Applying the network to all N data results in N^Y tagged data, composed of $N^Y = N_s^Y + N_b^Y$, where N_s^Y are the tagged signals and N_b^Y are the tagged background data. Of course, the values for N_s^Y and N_b^Y are not known. Our task is to determine confidence levels for the signal likelihood p from the sole knowledge of N^Y . We proceed by writing down the probability density of N^Y for given p and, subsequently, generalizing the Clopper–Pearson method.

First, assume fixed N_s . The probability densities of N_s^Y and N_b^Y are binomial and thus the probability density for N^Y is given by the convolution

$$P(N^Y|N_s) = \sum_{N_s^Y + N_b^Y = N^Y} b(N_s^Y|N_s, F_s) b(N_b^Y|N_b, F_b), \quad N_b = N - N_s. \quad (25)$$

Summing over N_s removes the constraint and the N^Y -probability density, with N, p fixed, is

$$P(N^Y|N, p) = \sum_{N_s=0}^N b(N_s|N, p) P(N^Y|N_s). \quad (26)$$

For given p_-^c, p_+^c and N^Y , we define confidence limits p_- and p_+ in analogy with equation (24)

$$p_-^c = \sum_{k=N^Y}^N P(k|N, p_-) \quad \text{and} \quad p_+^c = \sum_{k=0}^{N^Y} P(k|N, p_+) \quad (27)$$

The interpretation is as before. For instance, the confidence interval $[p_-, p_+]$ corresponding to the standard error bar is again obtained with $p_-^c = p_+^c = 0.159$. Bounds calculated with $p_-^c = p_+^c = 0.023$ ensure the standard two error bar confidence level of 95.4%, and so on.

5.2 Signal Probability Distributions

Distribution functions (see Brandt 1983 or van der Waerden 1971) are defined as integrals over the corresponding probability densities $\rho(p')$

$$F(p) = \int_0^p \rho(p') dp'.$$

Here the lower integration bound already specializes to probability densities defined for $0 \leq p \leq 1$ and the fact that $\rho(p')$ is a probability density implies $0 \leq F(p) \leq 1$ with $F(0) = 0$ and $F(1) = 1$.

Equation (27), or of course (24) when applicable, can be used to sandwich the *a-posteriori* signal probability distribution $F(p) = F(p|N, N^Y)$ between lower and upper bounds. Namely, it is easy to see that

$$\begin{aligned} F_1(p) &= 1 - p_+^c(p) = \sum_{k=N^Y+1}^N P(k|N, p) \leq F(p) \leq \\ F_2(p) &= p_-^c(p) = \sum_{k=0}^{N^Y} P(k|N, p). \end{aligned} \quad (28)$$

Using $N = 26000$ and $N_s = 130$, figure 4 depicts these functions for the examples $N^Y = 142$ and 194. Upper and lower bounds are seen close together, such that $F(p) = (F_1(p) + F_2(p))/2$ would be a reasonable working approximation. The corresponding probability densities are the derivatives with respect to p . Their numerical calculation is straightforward when analytical expressions for the derivatives of the binomial coefficients in equation (26) are used. Figure 5 exhibits the results, $P_1(p|N, N^Y)$ and $P_2(p|N, N^Y)$. At $p = 0$ the probability densities have δ -function contributions. Whereas in our example for good detection efficiency ($N^Y = 194$) the effect is practically negligible, it is quite remarkable for bad detection efficiency ($N^Y = 142$ in our example):

Fig. 4. *A-posteriori* signal probability distributions (upper and lower bounds).

Fig. 5. *A-posteriori* signal probability densities (corresponding to upper and lower bounds of the distribution functions in figure 4).

$$P_i(p) = F_i(0) \delta(p) + \dots, \quad (i = 1, 2) \text{ with } F_1(0) = 0.136 \text{ and } F_2(0) = 0.156.$$

In typical applications the efficiencies F_s and F_b may not be known exactly either. Instead, a number of training sets ($j = 1, \dots, J$) may exist, each giving somewhat different efficiencies F_s^j, F_b^j . In this situation a bootstrap type of approach (see, e.g., Efron 1987) can be applied. We can linearly combine

different probability densities to an ultimate one

$$P_i(p) = J^{-1} \sum_{j=1}^J P_i^j(p|N_j^Y), (i = 1, 2),$$

and determine $P_i(p)$ as discussed. Finally, to involve conjectured *a-priori* likelihoods may in many situations be unavoidable and, actually, be quite successful. In our case: When a clear, positive signal identification is possible, we find practically no difference between a Bayesian maximum entropy and our approach. However, our example of weak signal identification shows that *a-priori* likelihoods are better avoided when a rigorous alternative exists.

6 Conclusions

NN are biologically motivated, but their relevance for cognitive processes has remained questionable. Essential ingredients for the latter may well be missing. On the other side, modelling of ANN has led to a number of interesting results. Statistical physics methods allow to demonstrate the existence of distinct phases. In particular, in agreement with expectations, a phase of retrieval states is found. For many practical applications, like pattern matching or data analysis in particle and astrophysics (to name just two), multilayer perceptrons have become indispensable tools. Many features of NN are of a statistical nature. However, this does not mean that the results cannot be used for quantitative analysis. For instance, when an ANN sorts data with exactly known efficiencies, rigorous confidence limits can be derived.

References

- Amit, D.J. (1989): *Modeling Brain Function* (Cambridge Univ. Press, Cambridge)
- Berg, B.A. and Riedler, J. (1997): Signal Confidence Limits from a Neural Network Data Analysis. hep-ex/9703001, to appear in Comp. Phys. Commun..
- Boden, M., ed. (1990): *The Philosophy of Artificial Intelligence* (Oxford Univ. Press, Oxford)
- Bechtel, W. and Abrahamsen, A. (1991): *Connectionism and the Mind* (Blackwell, Cambridge)
- Brandt, S. (1983): *Statistical and Computational Methods in Data Analysis* (North-Holland, Amsterdam)
- Churchland, P.M. (1989): *A Neurocomputational Perspective* (MIT Press, Cambridge)
- Clopper, C.J. and Pearson, E.S. (1934): The Use of Confidence or Fiducial Limits Illustrated in the Case of the Binomial. *Biometrika* **26**, 404–413
- Domany, E., van Hemmen, J.L. and Schulten, K., eds. (1994): *Models of Neural Networks* (Vols. 1 and 2, Springer, New-York)
- Hertz, J., Krogh, A. and Palmer, R.G. (1991): *Introduction to the Theory of Neural Computation* (Addison-Wesley Pub. Comp.)

- Efron, B. (1982): *The Jackknife, the Bootstrap and Other Resampling Plans* (Society for Industrial and Applied Mathematics, Philadelphia)
- Kandel, E.K., Schwartz, J.H. and Jessel, T.M. (1995): *Essentials of Neuroscience and Behavior* (Appleton and Lange, Norwalk)
- Kohonen, T. (1989): *Selforganization and Associative Memory* (Springer, Berlin)
- Marchesini, G., Webber, B.R., Abbiendi, G., Knowles, I.G., Seymour, M.H. and Stanco, L. (1992): HERWIG 5.1 – A Monte Carlo Event Generator for Simulating Hadron Emission with Interfering Gluons. *Comp. Phys. Commun.* **67**, 465–508
- McClelland, J.L., D.E. Rumelhart, D.E., eds. (1986): *Parallel Distributed Processing* (Vols. 1 and 2, MIT Press, Cambridge)
- McCulloch, W.S., Pitts, W. (1943): A Logical Calculus of the Ideas Immanent in Nervous Activity, *Bull. Math. Biophys.* **5**, 115–133
- Peterson, C., Rognvaldsson, T and Lönnblad, L. (1994): JETNET 3.0 – A Versatile Artificial Neural Network Package. *Comp. Phys. Commun.* **81**, 185–220
- Ratsch, U., Richter, and Stamatescu, I.-O., eds. (1997): *Aspects of Intelligence and Artificial Intelligence* (Springer, Heidelberg)
- Rosenblatt, F. (1962): *Principles of Neurodynamics* (Spartan Books, Washington DC)
- Stamatescu, I.-O. (1996): “The Neural Network Approach” in *Proceedings of the International Conference on the Philosophy of Biology, Vigo, 1996*
- van der Waerden, B. (1971): *Mathematische Statistik* (Springer, Heidelberg)
- Wasserman, Ph.D. (1993): *Advanced Methods in Neural Computing* (Van Nostrand Reinhold)

DISCLAIMER

This report was prepared as an account of work sponsored in part by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This document is available upon request in alternate formats for individuals with print-related disabilities. Contact the Publications Department at (904) 644-1010 for more information.