Artificial Neural Networks in Medicine and Biology

A philosophical introduction

Opening lecture at the ANNIMAB-1 conference, Göteborg, May 13-16, 2000

Helge Malmgren

Department of Philosophy, Göteborg University

Abstract

Artificial neural networks (ANNs) are new mathematical techniques which can be used for modelling real neural networks, but also for data categorisation and inference tasks in any empirical science. This means that they have a twofold interest for the philosopher. First, ANN theory could help us to understand the nature of mental phenomena such as perceiving, thinking, remembering, inferring, knowing, wanting and acting. Second, because ANNs are such powerful instruments for data classification and inference, their use also leads us into the problems of induction and probability. Ever since David Hume expressed his famous doubts about induction, the principles of scientific inference have been a central concern for philosophers.

1. Introduction

The present lecture, while also serving as a brief tutorial on artificial neural network models, will deal with certain methodological and philosophical questions which arise when using such models. *Artificial Neural Networks*, or **ANNs** for short, is a heterogenous and loosely delimited set of mathematical techniques which were mainly developed during the second half of the 20th century. "Neural" in "artificial neural networks" stands for the fact that the techniques bear some similarities to the way we believe that real neurons and neural networks process information. "Artificial" can be said to denote three different things, first, that the models often grossly simplify or even distort what we know about neural mechanisms, second, that many of them are mainly used for other purposes than modelling real neurons, and third, that they are often implemented in non-neural structures.

A few words might be in place concerning the definition of ANN research and about the question, when did such research actually begin? In a wide sense, the term "artificial neural network models" includes every piece of neurological or neurophysiological theory. A better, more narrow definition confines the concept of an ANN to models with *less* emphasis on actual physiology and structure of the brain and *more* emphasis on the speculative search for abstract unit properties and global network structures which might produce interesting information-processing characteristics and which could potentially also explain important mental phenomena. With such a definition, much of the history of neurology and neurophysiology is left out. The beginning of a rich tradition of real ANN research can then be dated to around 1950. However, there are still several forerunners. It is not unreasonable, for example, to classify the work of René Descartes on the reflex arc and its superordinate control as a piece of early artificial neural network research. Ivan Pavlov's speculations in the early 20th century about the mechanisms of classical conditioning – where he comes close to discovering the principle which was later formulated by Donald Hebb – clearly belong to the same tradition. I should also mention that Sigmund Freud formulated a network model of the brain already in 1895. In his "Project for a Scientific Psychology", he tries to understand the overall relations between perception, memory and motivation in terms of three interacting systems of neurons. The

between perception, memory and motivation in terms of three interacting systems of neurons. The references to Descartes and Freud are not only historically interesting; they also point to two areas which later model builders have too often left out of their thinking, namely, *the hierarchical organisation of the brain* and *the importance of motivation for cognitive functioning.* Just to mention one example where these two areas are obviously relevant: how many neural network models of *dreaming* and its role in the overall functioning of the brain have you met with in the recent literature? Not so many, I presume. Hopefully we will see more large-scale brain models, integrating what is known about hiearchical structure and motivation in a not too distant future.

When Hebb, Rosenblatt and others started thinking about neural networks in the mid-20th century, their goals were usually set somewhat lower than the understanding of the brain as a whole. However, these early pioneers in the field were still motivated by a desire to understand the basic principles which the human mind-brain follows, especially when it is engaged in cognitive activities such as perception, concept formation and remembering. I will say more about these early attemps later. When new powerful ANN models were developed and made easily available in the 1980's, they also came to be used as tools for data handling in their own right, independently of their adequacy as brain models. This meant that the ANN tradition actually split into two, where the one which sticks most closely to the original motivations is today often called "Computational Neuroscience". However, I prefer the somewhat broader term "Brain Theory" (coined by Michael Arbib). The other half of the tradition does not have an accepted common name; it could be called "Applied ANNs", "ANNs for Data Handling", or "Adaptive Statistical Methods" (ASMs). The last-mentioned label is more comprehensive than the others, and emphasises the continuity with other fields of statistics (in the sense of inferential models).

Having subdivided the field of artificial neural network research in this way, I want to emphasise that deep connections exist between the two subfields. Certainly, the appeal and the success that ANN techniques have had in handling empirical data is to a large part due to their similarities with biological neuronal nets. I am thinking of the inherent nonlinearity of most ANN models, their parallel architecture and speed of calculation, the way ANNs can use distributed representations and the resulting relative robustness against noise and structural damage, the content addressability of associative nets and the computational power inherent in the recurrent nature of data processing in the more advanced models. As long as the performance of trained human beings on such tasks as recognition, prediction and control is superior to the available automated methods – and that this is so in many fields, including most of clinical medicine, is quite clear – there is all the reason in the world to continue to let our knowledge of the human brain inspire the development of new data handling techniques. But the converse also holds: attempts to theoretically understand the human brain cannot be successful if they are not backed by an extensive knowledge of the mathematical theory of adaptive systems. We will not, for example, understand the mechanisms of amnesia before we have a plausible theory of how temporal structures can be stored and retrieved in massively recurrent neural nets.

In the interface between the two subfields of ANN research, i.e. brain theory and adaptive statistical methods, we also find the most interesting philosophical issues. Why is it that the human mind-brain, working in an informal and intuitive manner, is still superior to formalised methods in so many important areas of life? And why is it that we, who rely in every moment of our life on our abilities to learn from the past and predict the future, have not yet been able to formulate

valid general principles of scientific inference and induction? These issues will hopefully form the basis for stimulating cooperative efforts by philosophers, mathematicians and neuroscientists during the decades to come, and I will try to cast a few rays of light on them later in this lecture.

2. ANN basics

An artificial neural network may be described as a set of neurons or *nodes* $\mathbf{X_i}$, each transforming its total or *net input* $\mathbf{x\_in_i}$ into an output or *activity* $\mathbf{x_i}$ according to an *activation function* (or *transfer function*) $\mathbf{f(x\_in_i)}$. Each node $\mathbf{X_i}$ sends its output to other units $\mathbf{X_j}$ through *connections* each having a certain effectivenesS or *weight* $\mathbf{w_{ij}}$. The net input to any unit $\mathbf{x_j}$ is usually modelled as a sum of all the outputs $\mathbf{x_i}$ from other units (and, in recurrent nets, from itself), weighted by the weights $\mathbf{w_{ij}}$ of the respective connections. Formally, then:

$$\mathbf{x\_in_j} = \sum_i \mathbf{x_i\, w_{ij}} \qquad (1)$$

$$\mathbf{x_j} = \mathbf{f(x\_in_j)} \qquad (2)$$

These basic principles are illustrated in the following diagram of an example node.

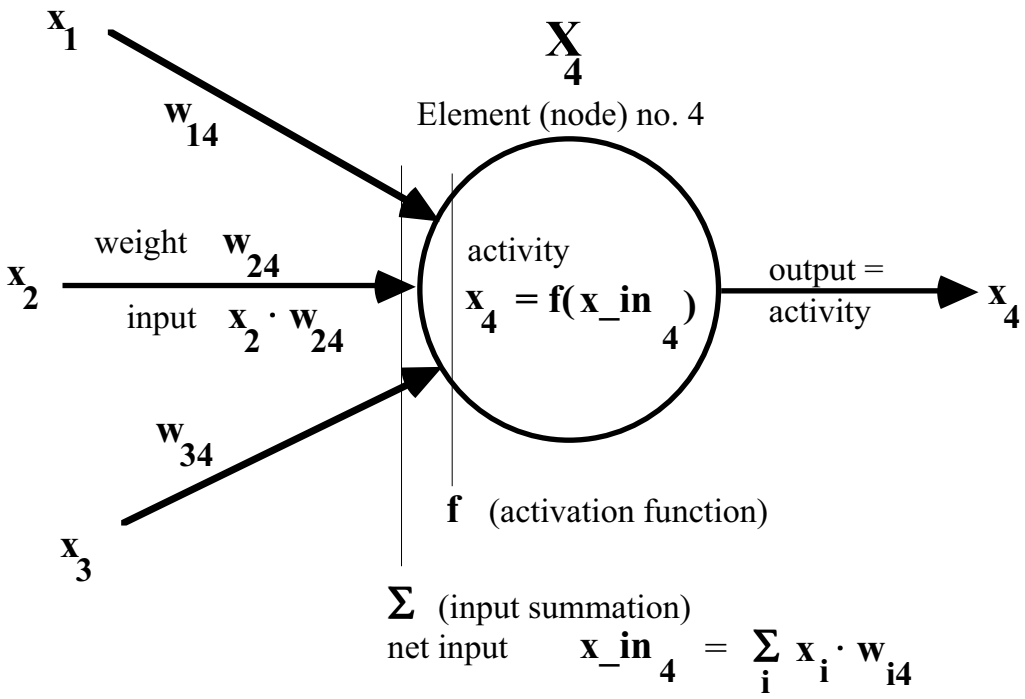

Figure 1. A common design of a node in an artificial neural network

To characterise an ANN one must also add a description of its connectivity or *architecture*; here the most important distinction is that between *feed-forward* and *feed-back* (or *recurrent*) nets. In the former, the information is thought to pass just once through the net, starting in an *input layer* of units and ending in an *output layer*. Between the input and the output layers, *hidden layers* of neurons may exist, which as a rule enhances the computational power of the ANN. Recurrent nets have a more complicated dynamics, with signals going back and forth between the nodes for some time. Most recurrent nets are designed so that this oscillatory behaviour ends in a stable signal pattern, which is then taken as the output of the net. In recurrent nets which process temporal patterns, a possible choice is of course to let temporal sequences of activities represent such patterns both on the input and the output side. The properties of such nets are however much less well understood.

A feed-forward net without hidden layers can look like this (only two of the connection weights are written out):

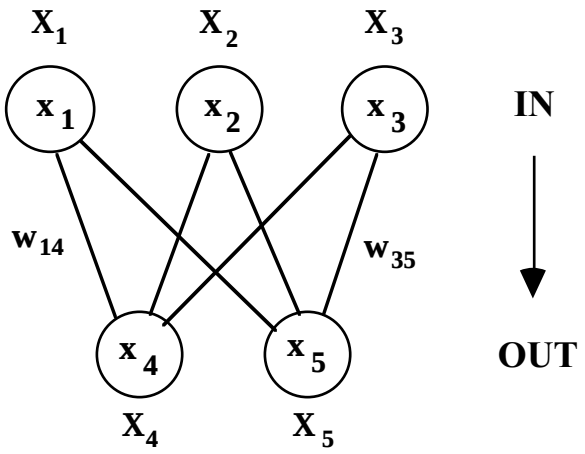

Figure 2. A one-layered feed-forward ANN

As I mentioned, any ANN node is associated with its own pre-defined activation function; in the simplest case the latter is *linear*, but more often a non-linear function is chosen. *Threshold* units which switch from one activity level to another at a certain threshold θ, and units with *sigmoid* (squashing) activation functions, are common choices. A sigmoid function is smooth and strictly monotonous function with a lower and upper bound; an example is provided in the following figure.

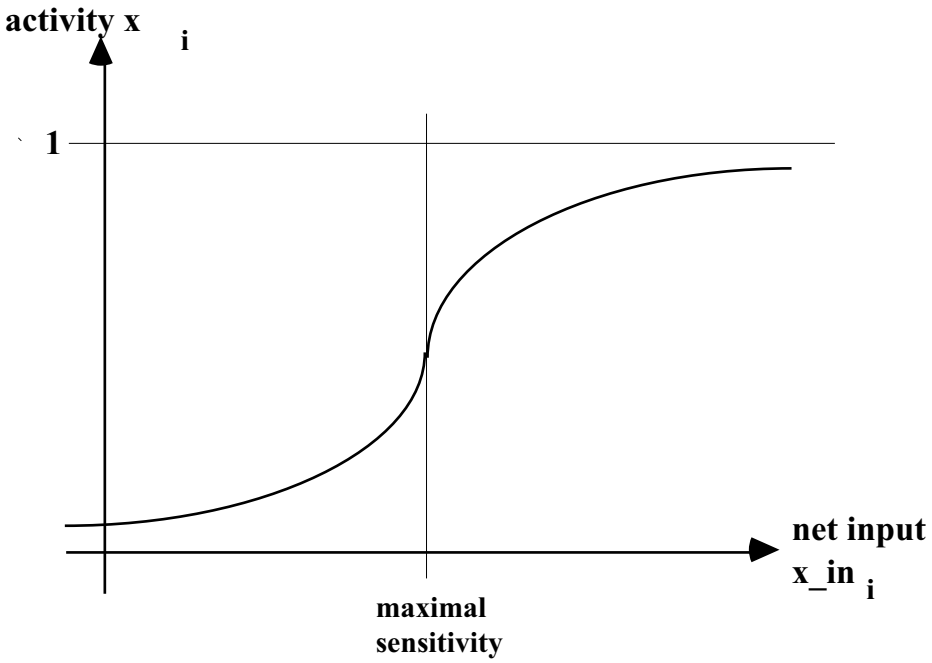

Figure 3. A sigmoid transfer function

Sigmoid functions are widely used in ANNs because they have nice computational and mathematical properties. It is also interesting to note that most biological neurons are sigmoid units in the sense that their frequency response on input has a region of maximum sensitivity somewhere between a threshold and a point of saturation. If, instead one looks at the single neuronal response on input it is a threshold, all-or-none affair. So, sigmoid and threshold transfer functions can both be said to model important aspects of biological reality.

Finally, I should mention another class of non-linear transfer functions which are commonly involved in ANN models, namely, the localised functions or *radial basis functions* (RBFs). These are functions – for example, Gaussians – which have a maximum value at a certain point in the input space but tend to vanish far away from this point. Different neuronal units are embodying different such functions, and together they can learn to encode the full input space in a way which simplifies the classification task. I will not go into the details of these networks here, but Richard Dybowski will say a good deal about RBF networks and their even more advanced cousin, the *Support Vector Machine*, in the lecture which follows this one.

Most importantly, any ANN has a *learning rule*. The learning rule, or learning algorithm, is the function according to which the connection weights are changed as a response to the "experience" of the net. It is the learning rule which gives an ANN its most important adaptivity. The weights and their changes is often seen as a model of the brain's long-term memory, while the activation of a unit or a net is sometimes regarded as the analogue of short-term memory.

A commonly used learning rule is the *Hebb rule*, which means that at each pulse of information processing, the weight $\mathbf{w_{ij}}$ changes in proportion to the activities $\mathbf{x_i}$ and $\mathbf{x_j}$ in the pre- and postsynaptic neurons $\mathbf{X_i}$ and $\mathbf{X_j}$,

$$\Delta w_{ij} = k \cdot x_i \cdot x_j \qquad (3)$$

Donald Hebb's original motivation for the rule was of course the fact that it offers a neat explanation of associative phenomena such as for example Pavlovian conditioning. Let us look at such an explanation, using an extremely simplified ANN model. Suppose that **A** is a neuron coding for the conditioned stimulus *CS* (in Pavlov's famous experiment, the sound of a bell), **B** is a neuron coding for the unconditioned stimulus *UCS* (the sight of food) and **C** finally the neuron responsible for the unconditioned response *R* (salivation). The units take on activity 1 if and only if the input exceeds threshold 0.5; else they have an activity level of zero. An activity level of 1 in **A** or **B** represents presence of the respective stimulus, while activity 1 in **C** represents release of the response R.

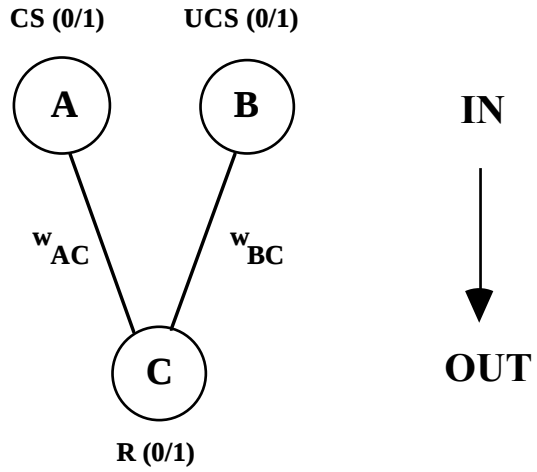

Figure 4. A model of classical conditioning

Initially, the weight between **A** and **C** is 0, while the weight between **B** and **C** is 1, representing the unconditional connection between the sight of food and the salivation response. Hebbs rule holds for both weights with the incremental factor k = 0.1.

Suppose now that we let the dog hear the bell at the same time as she sees the food. This means that the activities in units **A** and **B** are set to 1 at the same time. Since the weight between **B** and **C** is 1, the net input to **C** will be 1, so **C** will take on activity 1. That means that the condition for Hebbian learning is fulfilled not only for weight $\mathbf{w_{BC}}$, but also for weight $\mathbf{w_{AC}}$. Hence the connection between **B** and **C** will be strengthened with an amount of 0.1. After five exposures for the experimental condition, $\mathbf{w_{AC}}$ will have the value of 0.5, and stimulation of **A** alone will be able to release the response. This is classical conditioning. Note that in the model, the neuron **B** can be said to serve as a *teacher* for the neuron **A**, presenting the response which A is to learn.

Research during the last three decades has shown that a Hebb-like mechanism exists in the central nervous system in the form of so-called *long-term potentiation* (LTP) of synapses. Hence in spite of its simplicity, Hebbian or Hebb-like rules might be central to the explanation of human associative memory. However, the Hebb rule is certainly not the whole truth about memory in the brain, and as a tool for data handling, it has many limitations. So it has limited use in artificial neural networks for statistical purposes.

Another learning rule of basic importance is the *delta rule* which means that the weight change Δw_{ij} is dependent on the difference between the *actual output* x_j and a *desired output* d_j; formally:

$$\Delta w_{ij} = k \cdot x_i \cdot (d_j - x_j) \qquad (4)$$

The delta rule is but the simplest example of learning rules which can drive the performance of an ANN with non-threshold elements deliberately close to a fixed goal. It has been much used in single-layer, linear feed-forward nets and in that famous network, the *single-layer perceptron*, which has a threshold transfer function instead of the linear one. The perceptron was designed for *supervised pattern classification*, which means that the ANN is trained on a set of input patterns until for each pattern it gives the output which was pre-defined as the "correct" class label for that input. In the case of pattern classification with the single-layer perceptron, the delta rule degenerates into the *perceptron learing rule:*

$$\Delta w_{ij} = k \quad \text{if the response is 0 but should be 1}$$
$$\Delta w_{ij} = -k \quad \text{if the response is 1 but should be 0}$$
$$\Delta w_{ij} = 0 \quad \text{if the response is correct}$$
$$(k > 0) \qquad (5)$$

The single-layer perceptron classifies by drawing dividing straight lines within the input space. (With one-dimensional inputs it simply sets a dividing point on the input line, while in higher dimensions it draws hyperplanes). This is easily seen from the facts that (1) the threshold function for the output unit switches from 0 to 1 when its net input reaches a certain value and (2) the net input is a linear function of the contributions from other neurons. Hence a weight change only changes the placement and orientation of the dividing point, line or hyperplane within the input space. Therefore it is not difficult to find classification tasks which are too difficult for it. If two separate regions of the input space belong to the same class while some points in the region between them belongs to another, the simple perceptron cannot learn to classify all inputs correctly. A classical example is the so-called XOR problem. An even simpler example (for one-dimensional inputs) would be the classification of people's lengths in two classes, "normal-range" and "other". And here is a third example.

Suppose that in a sample of 10 healthy people (blue) and 10 patients with a certain disorder (red), the results on two laboratory tests X_1 and X_2 are distributed as follows.

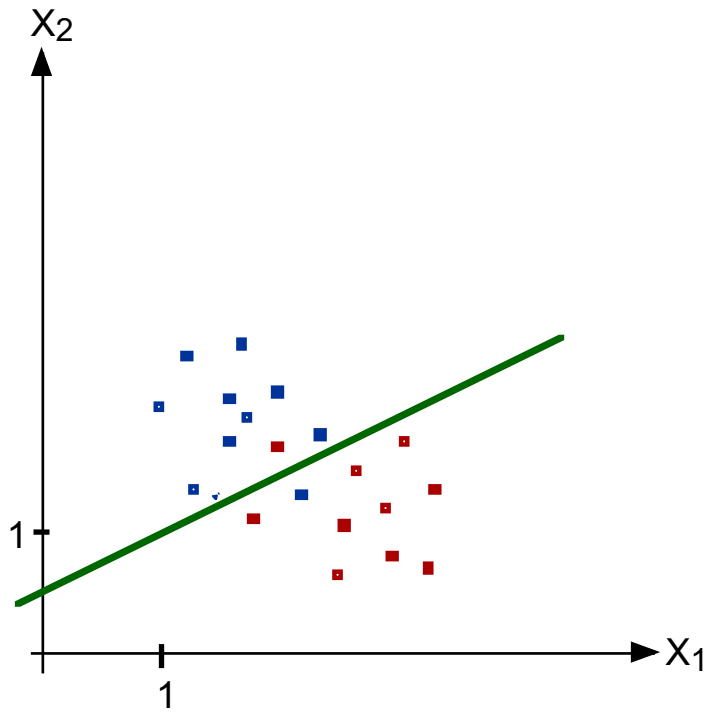

Figure 5. A linearly non-separable classification task

The green line represents the best solution attainable by a simple perceptron. This best solution involves a misclassification of two patients. I will come back to the example later.

Similarly, a linear feedforward net can use the delta rule to find the "best" formula for the input-output relation in a data set, but it basically performs what in statistics is called *linear regression* and if the input bears a non-linear relationship to the input, the net cannot capture the non-linearity. More about that in a while.

However, more complicated ANNs which have at least two layers of connections (at least three layers of units), and which use sigmoid activation functions and an advanced version of the delta rule (*back propagation of error*), perform much more complicated mathematical operations and in principle, they can find a solution for any consistent classification task. In our fictious example with classification of patients, a two-layer perceptron with sigmoid activation will find a solution which might look like the following figure:

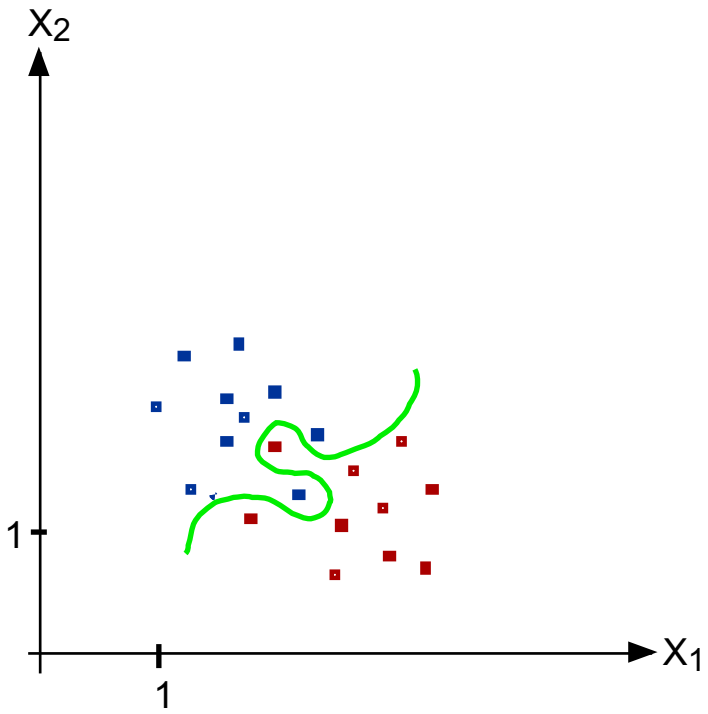

Figure 6. A non-linear solution of the classification task

More generally, they can learn to approximate almost any mathematical function (input-output relation) on a closed subspace of the input space. They can do this by finding a decomposition of the function in question in terms of a sum of sigmoid functions, or in terms of even more complex structures. It is this power, and the efficiency and speed of their learning algorithms in comparison with traditional ones, which make these ANNs superior in many contexts to most traditional statistical methods, and it is basically this power which underlies their recent successes as data managing instruments in medicine, biology and other areas.

Now, there are surely other non-linear methods in statistics than artificial neural networks. Indeed, there are even several alternative methods with the same universal power as the multi-layer perceptron. For example, function approximation with polynomials has been known for long to be a universal method. I will not go into detailed comparison here between ANNs and those other methods, but it should be mentioned that (1) even though several of the simpler neural nets involve no improvements over traditional statistics because they are equivalent to known methods, this is not the case for the more advanced neural nets; (2) the ANNs have not only enriched the repertoire of non-linear methods. Several of them (including the multi-layer perceptron with sigmoid functions and back propagation) also compare very favourable with most traditional such methods in terms of efficiency and speed of the algorithms; (3) the research traditions dealing with new ANN methods and new statistical *non-ANN* methods tend to fuse today, producing in-between creatures such as the Support Vector Machine. Altogether, the use of either classical ANNs or such new ANN-like models in situations which call for non-linear methods is often well motivated.

I will now talk about some advantages, and some disadvantages, of the inherent power of artificial neural networks and of the new related models.

3. Do we need such powerful models?

It could be asked: Do we as medical and biological scientists need more powerful statistical methods than the traditional ones, and if so, why? The correct answer to these questions is, I think, very simple. Medicine and biology deal with very complicated systems much of the workings of which are, as a rule, unknown to us. If we use a certain statistical method for such a system *just* because it is based on a simple (for example, linear or Gaussian) model which we can handle mathematically in an exact way, we run a great risk that the method throws away information which is actually needed to really understand the system. So, we do need powerful models.

In many situations this is not just a theoretical insight; on the contrary, the experienced clinician can often see with his trained eye that the statistical analysis leaves out something essential. Such an intuition can be objectively verified to the extent that there is an independent criterion of success, as for example when the clinician's X-ray diagnosis is shown by a patho-anatomic criterion to be better than that of a certain exact but crude image-analytical system. When there is no simple such criterion it is of course more difficult to prove scientifically that a certain model is inadequate. A case in point is the recent statistically based classification of mental disorders, DSM-III and its successor DSM-IV. (DSM means Diagnostic and Statistical Manual of Mental Disorders.) These systems are certainly based on a huge amount of data, but the statistical methods and theoretical models used are very simple. For example, it is not possible to ascribe two mental disorders to the same patient at the same time! But evidently, a patient may for example have both a dementia and a confusional syndrome (a delirium) at the same time. If, as a clinician, you know a subfield of the mental disorders fairly well you can easily point to a lot of such cases were the crude classification system of the DSM throws away information which is essential for prognosis and treatment planning.

Examples can be multiplied to show that many of the traditional statistical methods used in medicine and biology are very often not up to their task. This is of course nothing new, and especially it is no news for the statisticians, who certainly are developing more complicated methods all the time to meet the needs. One should not blame the misuse of statistics on the statisticians. But perhaps one could say that the sudden rise in popularity of ANN models during the 80's and 90's is an indicator that the size and importance of the problem was underestimated even by them.

The question which I have tried to answer briefly in this section is also adressed by Richard Dybowski in his talk, so here I leave it in his competent hands.

4. The coding problem

These recent successes of ANNs as data managing instruments in medicine, biology and other areas are mainly due to the enormous computational power of the multi-layer, nonlinear ANNs. But with power comes responsibility, and if the power of an ANN is not controlled in a very careful way, problems and paradoxes will appear. Neural network models should never be used without thinking carefully about the needs of the situation. Two of the most important problem areas have to do with *not coding the data in the best way* and with *using too powerful networks*. I will now illustrate the first of these statements statement with realistic data and using a very famous neural network, the *Self-Organising Map* (SOM). The SOM was designed in the 1980's by Teuvo Kohonen, whom we will have the honour to listen to later today.

A Self-Organising Map is a device which classifes patterns according to their intrinsic similarity and maps the result onto a spatial (usually two-dimensional) structure. It has two layers of units: the input layer, and the competitive (or Kohonen) layer. If the network is to be used for the classification of a set of n-dimensional input patterns (vectors), the input layer has **n** units. Typically, the output layer contains $\mathbf{m} \cdot \mathbf{m}$ units arranged in a square lattice, all receiving connections from each input node. These connections are randomised from the start and then modified by the training procedure which will be described shortly. Figure 7 shows a SOM network with 15 inputs (plus a bias input) and $\mathbf{8} \cdot \mathbf{8}$ outputs. The network was simulated using the commersial software NeuralWorks.

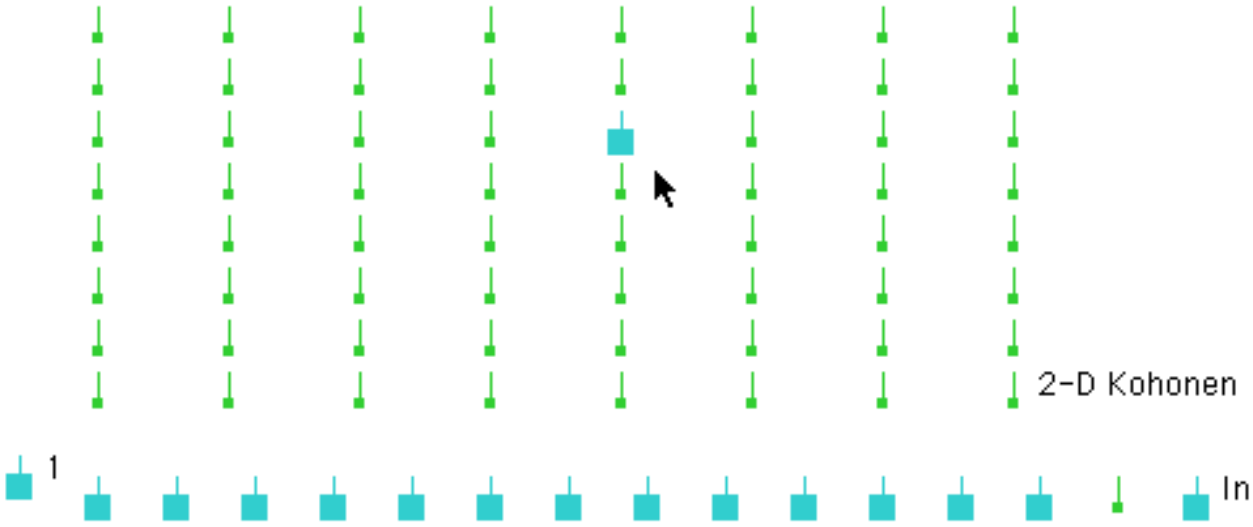

Figure 7. An 8 · 8 SOM network with 15 inputs (connections not shown)

The signal dynamics of the SOM network is as follows. After the randomisation of weights in the first layer of connections, each node $\mathbf{Y_i}$ in the competitive layer is associated with an n-dimensional weight vector $\mathbf{w_i}$. When an input vector **v** is presented to the net, it is compared with all the weight vectors $\mathbf{w_i}$, and the competitive node $\mathbf{Y_s}$ whose vector has the smallest Euclidean distance to **v** is declared *The Winner*. The activity of the winner is then set to 1 while all the other competitive nodes are inactivated. This is also shown in Figure 7, where the arrow points to the winner.

To see better what this seemingly esoteric Euclidean algorithm has to do with networks of real neurons, let us first imagine that the inputs are normalised. The net input to any Kohonen unit $\mathbf{Y_i}$ is the inner (scalar) product of the input vector **v** and the weight vector $\mathbf{w_i}$. For normalised vectors this product is equal to the cosine of the angle between the vectors, and hence maximal when the Euclidean distance between the vectors is minimal. Hence the unit whose weight vector is closest to the input vector will have the greatest net input, and if the transfer function is sigmoid (as it is in biological neurons) this unit will also be maximally activated. Further suppose that the neurons in the second layer are organised in a mutually inhibitory (competitive) fashion so that the maximally activated unit suppresses the activities of all the others. This is nothing but the required winner-takes-all mechanism. It is quite probable that there are neuronal structures in the brain which work in this manner. But for computational purposes, the original Euclidean distance model is of course simpler

manner. But for computational purposes, the original Euclidean distance model is of course simpler to work with, and it gives essentially similar results (for vectors of approximatively the same length, that is).

The learning, or weight update, rule of the SOM now prescribes that the weight vector of the winner unit $\mathbf{Y_s}$ be moved a little closer to the input in question. In this way $\mathbf{Y_s}$ will be an even more certain winner next time. The decisive trick with the SOM, however, is to move also the weight vectors of the *spatial neighbours* of $\mathbf{Y_s}$ somewhat closer to the input vector **v**. In this way, these neighbours will become more probable winners for inputs which are similar to **v**. It is not difficult to see that with repeated applications of this rule, a topological map of output units will result where *near-lying* units will be the winners for *similar* input vectors.

In the following example, a set of twenty-three 15-dimensional data points have been classified by means of an **8 · 8** SOM net. The data represent 15 variables from a psychological test (the Rorschach) given to six neuropsychological patients (A, B, C, E, F and G) at different stages of their disease. They all had Korsakov's Amnestic Disorder, mixed with other organic mental disorders, and due to different causes. Data from the same patient have been given the same colour in the picture of the resulting map. A1, A2,... represent the tests with patient A, and so on.

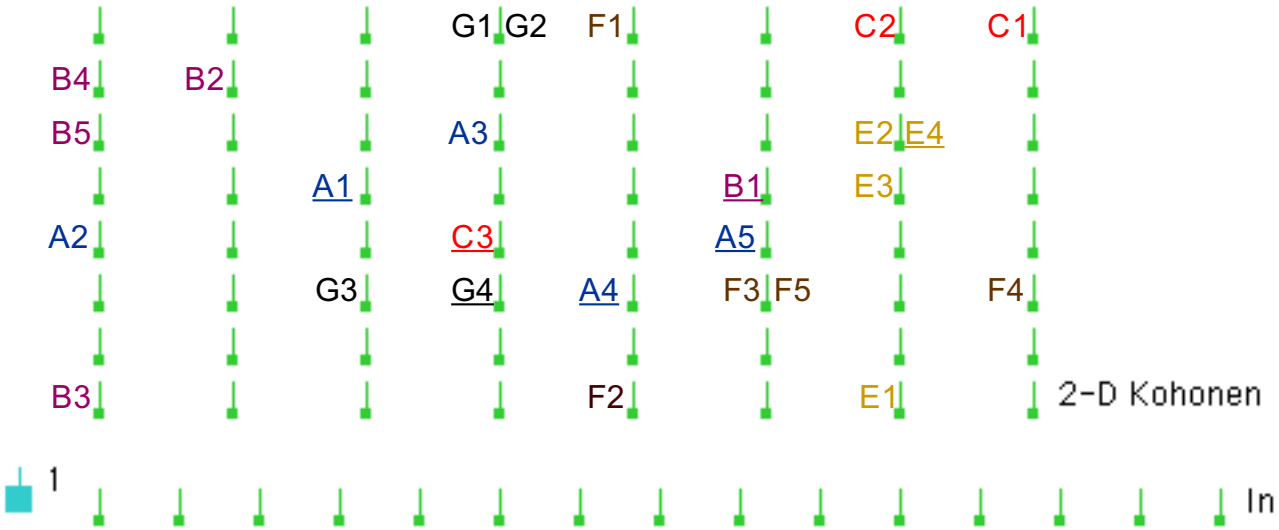

Figure 8. The SOM network with (scaled) Rorschach data

There is a clear tendency for the SOM network to place the tests from the same patient near to each other. (Note that the map is actually a toroid. This means that the upper and lower row are close to each other, as are the leftmost and the riughtmost column. Hence F1 and F2 are close, and so on.) However, the placement of the tests is also influenced by the patient's clinical condition. For example, patient C was much less disturbed at the last test (C3) than at the first two (C1 and C2), while B1 was much less disturbed at the first test (B1) than at the following ones. The underlined data represent the 7 tests with the lowest clinical score of global organic disturbance. Generally spoken, these less disturbed patients tend to cluster in the middle of the map (as seen from this side).

The data used here are not the original ones, since they have been scaled. The scaling

was an ad-hoc procedure to fit all values in the same diagram, multiplying or dividing the values of a majority of the variables by a factor 10. What happens if we run the SOM without scaling? The Euclidean distance between inputs is surely determined mostly by the largest components of the input, so it would be no surprise if there are substantial changes. Here is the result:

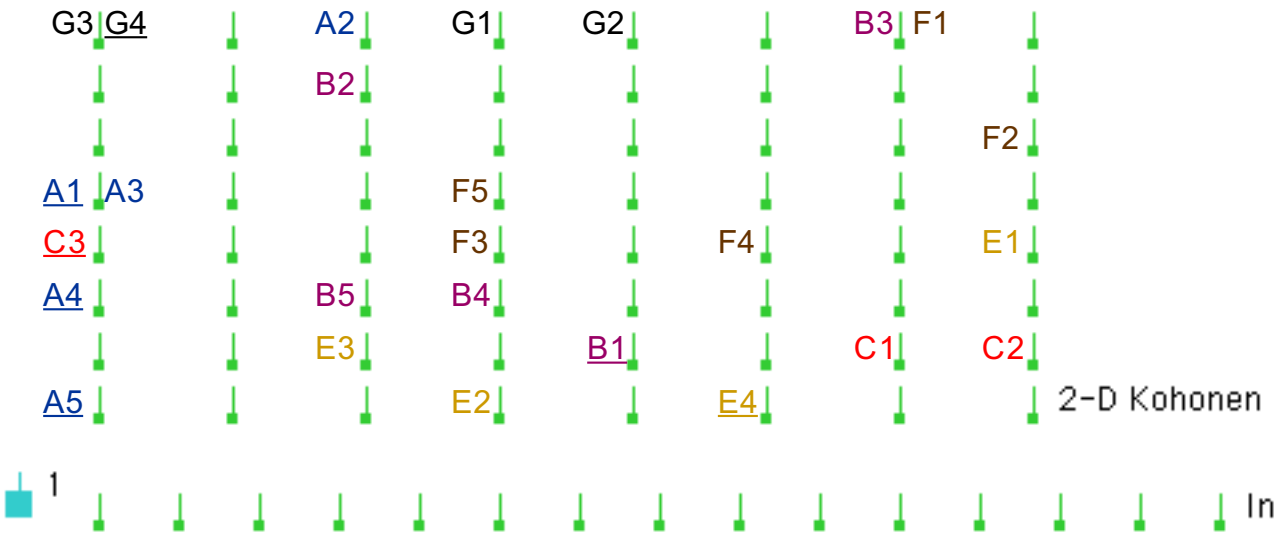

Figure 9. The SOM network with the original Rorschach data

Note that when comparing the two diagrams, no importance should be given to the absolute placement of the input vectors in the diagram. It is only the relative placement that matters. One can see that much of the structure from the previous example is kept; especially, there is still a tendency towards an intra-patient clustering. However, there are also important changes; an example is the move of B1 out from the "healthy" cluster. I would interpret this change as due to loss of information about the global level of organic mental disorder, since B1 was clearly the most "healthy" test in the series. That B1 is placed so close to B4 and B5 and so far away from A4 and A5 indicates that the SOM now "sees" more of the intra-patient similarities and less of the levels of disorder.

The morale of this example is that the coding of the data is all-important when using an artificial neural network for data analysis. If no structure turns up at the first attempt, a suitable re-coding of data may help. On the other hand, with an unsuitable coding, clusters and categorisations may be coding artifacts.

It might be objected that given data are given data and may not be handled at will. To this I would like to reply that the real world does not come in the form of data; it is we who choose to code dimensions of the real world as data. The patient before you does not present as a set of data in a fixed form, and the first mark of an intelligent clinical analysis is that a suitable form is choosen for the description of the situation and the patient. True, part of the input to the clinician is given as fixed data, for example the laboratory values. But these, too, were once coded by a human being (the lab doctor).

From a neurophysiological standpoint it might also be objected that our nervous system presents the world to us as data with a fixed code. That is true, but only in the sense that the

transducers of the body set certain limits to which kinds of information is passed to the brain. Already on the perceptual level of processing, the data form is not fixed. If it was, how could we ever see the same world from different aspects? Indeed, the nervous system continuously presents the world to us under alternative descriptions, among which we can choose using superordinate principles.

5. Too much power, or: the generalisation problem

It is commonplace today that if you don't have a lot of data and use a too powerful network for a classification or function approximation task, your result may come to suffer from a lack of generalisibility. Most ANN textbooks abund with warnings to this effect. I will first briefly go into the rational background of such warnings, and then – as a kind of counterweight – discuss some circumstances in which the use of very powerful networks with few data may be the best choice after all.

Suppose that you are investigating a process in nature which takes the form of an input-output relationship between two variables **x** and **y**. Also suppose that your measurements are known not to be not exact, but instead confounded by random noise. Your 10 measurement pairs turn out to be approximatively compatible with a linear input-output relationship between **x** and **y:**

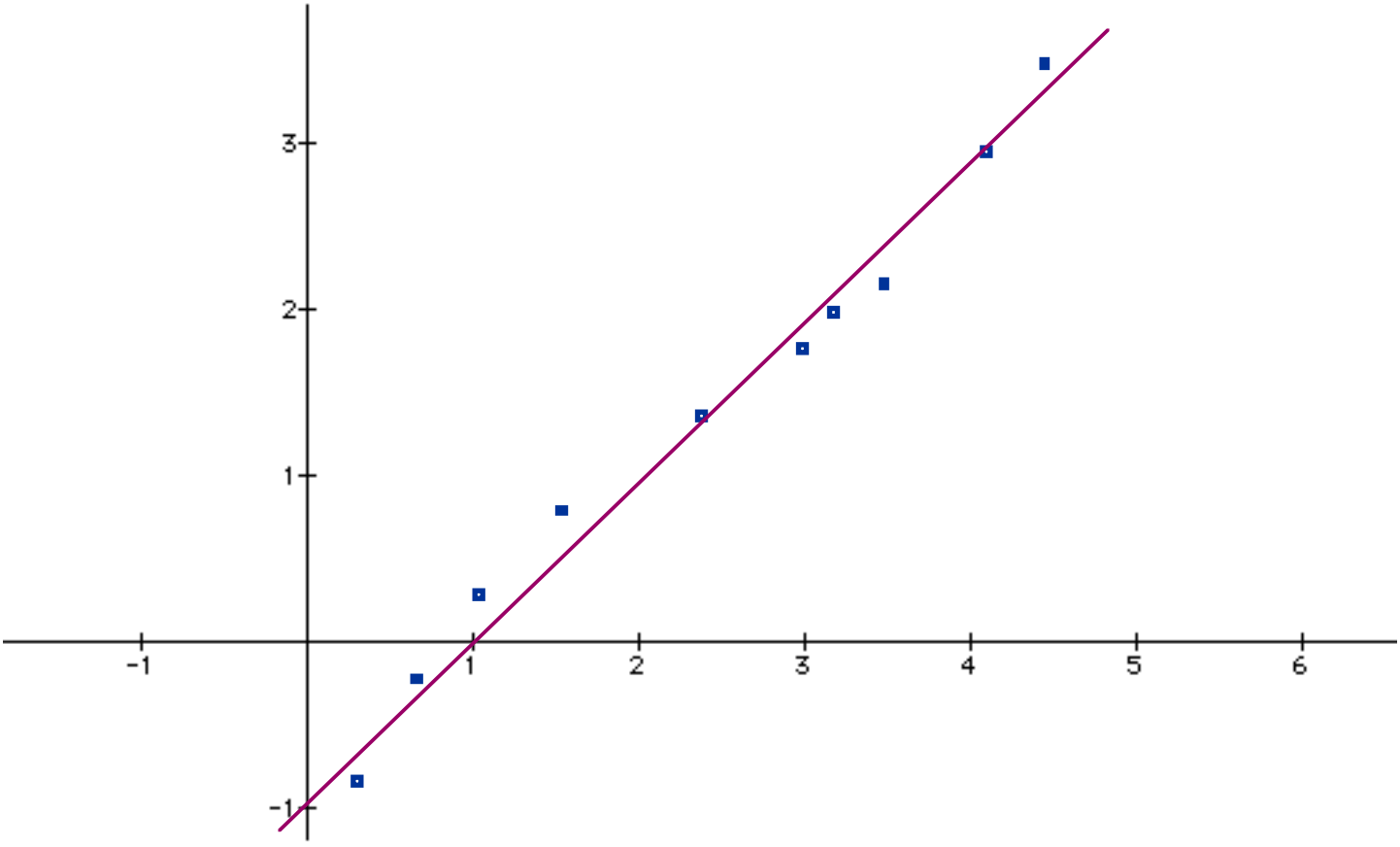

Figure 10. An approximatively linear relationship

but when you look closely at the data you wonder whether a cubic equation would not give an even

better fit. This turns out to be the case; the actual equation for the best-fitting third-degree curve is

$$= \frac{(x-3)^3}{8} + \frac{x^2}{10} + 1 .$$

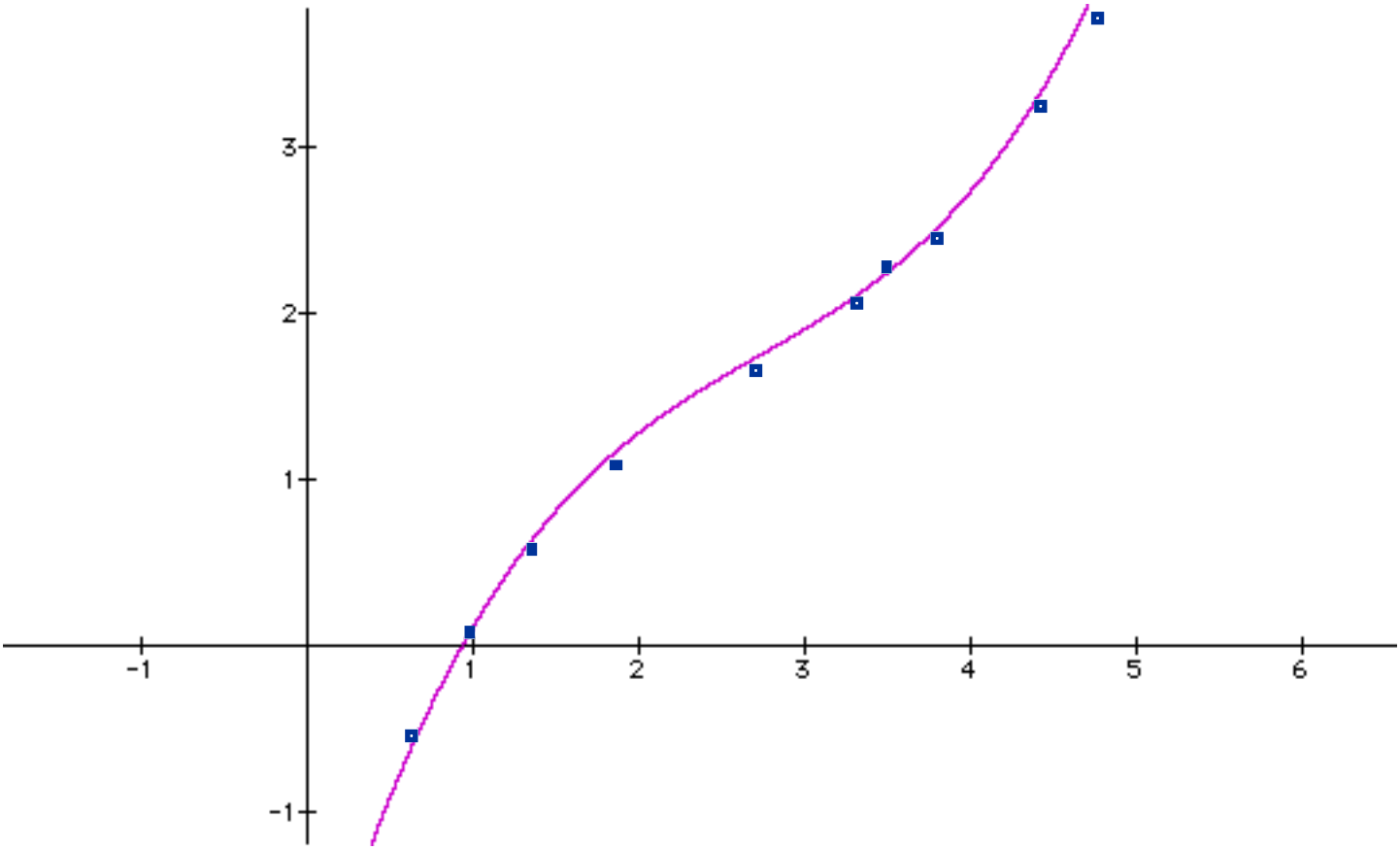

Figure 11. A cubic approximation of the relationship

However, when you compare the two hypotheses thus generated (the linear and the cubic one) with the next 10 measurements, you notice to your surprise that the linear hypothesis performs better. How can that be so? Why is the worse-fitting equation a better predictor?

The answer is easy if you suppose that the process you are studying is *really* a linear one, and that the deviations are *really* due to random noise. If this is so, no prediction can be expected to be better than the predictions from the linear hypothesis, and one simply has to rest content with the uncertainty inherent in the latter. All other strategies means mistaking manifestations of the noise for properties of the process. A similar argument can be made if the linear process in question is observed with a small error but has a substantial inherent random element, and of course in all kinds of mixtures of these situations (random process – large random observation error).

It might be illuminating to compare the scientist's situation here with a coin-tossing game with a biased coin. Suppose that in a large series of tosses you find that the coin tends to give heads two times out of three, but also that tails were slightly more frequent than heads on the few trials on days with snowstorm. Since you believe (for good reasons) that the coin is not sensitive to the

weather, the latter result does not make you choose a complicated strategy with bets on tails on days with snowstorm, and bets on heads on other days. You reason, correctly, that such a strategy will almost certainly lose in a longer run compared to the simple strategy of always betting on heads.

Hence, it is sometimes demonstrably better to act on simpler premisses than on more complicated ones. This, by the way, might be part of the reason why intuitive reasoning is so often superior to discursive thought. Maybe discursive thought tends to get too much entangled in details which intuitive thought sees are irrelevant?

But let us return to the case of mathematical modelling. Here is an argument related to the classification task described above. Let us have another look at Figures 5 and 6. We imagined that with the multi-layer perceptron, we could classify all the patients correctly. But if there is a substantial random error in the observations, and/or the values are randomly distributed, the straight line may actually be the best possible classification principle. This can even be proved mathematically in case of normal distributions. If these are symmetric and equal, the classification with the lowest expected number of misclassifications is defined by a straight line halfway between the means. Disregarding for the moment the problems involved in empirically estimating these means, the result looks rather like the following:

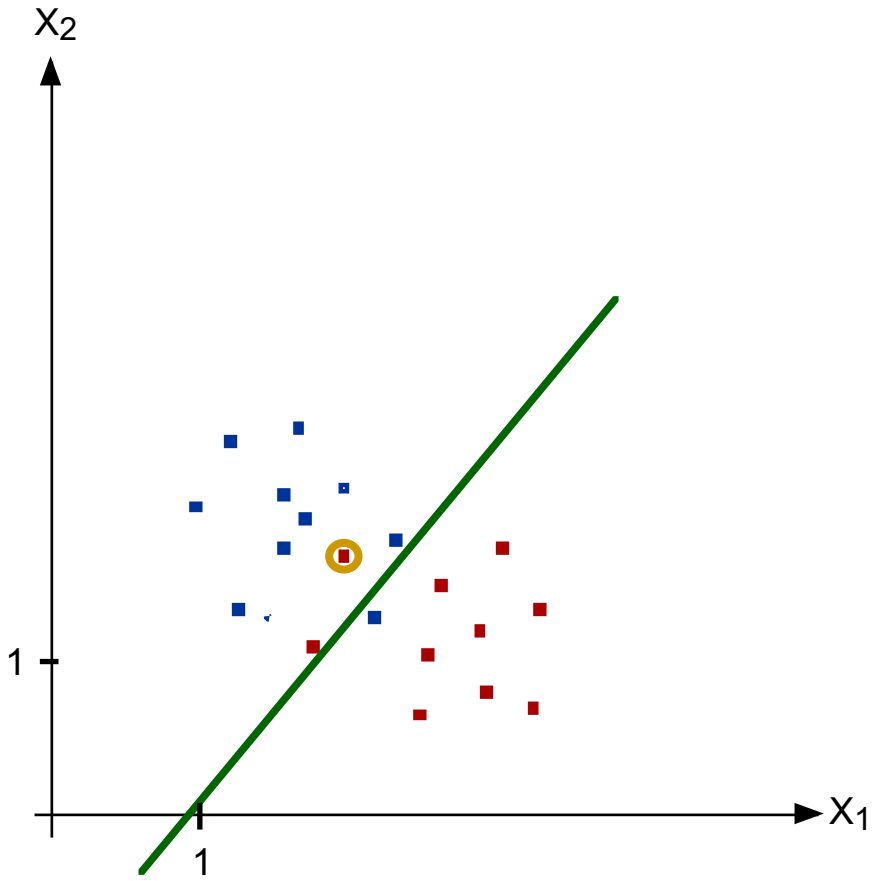

Figure 12. The best discrimination of two equal and symmetric Gaussians

The misclassifications in Figure 12 are then simply the result of unlucky circumstances: one healthy person (blue) happens to have values which make disease probable, and two patients (red) have values which makes disease improbable. In other words, when you see another person with the

encircled (red) value, you should classify her as blue (i.e., healthy), not as red!

However, what has been said holds only to the extent that the process which the scientist is studying is random and/or contaminated by substantial random noise. If we have reasons to believe that we are observing a deterministic process in a fairly noiseless way, and if we want to make as correct predictions as possible about the process, our equations must be framed so as to fit all the data almost exactly. Very often, we do have good reasons to suppose that we are dealing with this kind of situation. An example in point is given by astronomical observations of comets and asteroids, where the astronomer usually does not hesitate to fit a parabola or an ellipse to very few observed points rather than to the best regression line. This example also illustrates the importance of *specific background knowledge:* because the astronomers think know that planets move in approximatively elliptic trajectories (and not, for example, according to seventh-degree polynomial curves), they can make predictions from *very* few data.

If we now transfer this argument to the field of artificial neural networks, the result cannot possible be a univocal recommendation always to choose as simple networks as possible. *The power of the net must be adapted not only to the expected level of noise and other random elements, but also to what we know in beforehand about the specific nature of the underlying process*. If we believe that we are observing a deterministic process fairly exactly, and if we have a good hunch in beforehand about the mathematical form of the true hypothesis about it, we should not hesitate to try to fit the data almost exactly to an equation using a powerful neural network (or a comparable method). Else we run the risk of oversimplifying and underpredicting.

What about the common situation when we have only vague background knowledge about the nature of the process and the amount of noise present? From the above arguments about generalisation it seems to follow that it may often be worth while to experiment with powerful nets to see whether the results generalise to a validation set of data. If they do not, it is probable that the choosen method was too powerful and that it coded noise along with projectible relationships. If the results do generalise to the validation set, there is at least a possibility that the net has captured the true form of the process behind the data.

Note that the morale of this argument is *not* that we should not hesitate before we use very powerful ANN models. On the contrary, taking the textbook warnings about too powerful methods seriously is an essential first condition for the successful use of artificial neural networks as data analysis instruments. But it is equally essential to try to understand the conditions under which we may be allowed to use strong models with few data. I would like to argue that the delimitation of these conditions cannot be a purely formal issue, in the sense that we should search for a mathematical or statistical criterion which could be applied to the data as such to decide, from the data themselves, how complex a model we should choose. Any data set, however simple it looks, can logically be the result of noise, and any data set, however complex, can logically (though maybe not physically) be the result of noiseless measurements of deterministic process. How probable it is in a given case that the one or the other holds cannot – and should not – be decided from the structure of the data alone. In this decision, our apriori knowledge of the process which we are investigating comes into play in an all-important manner.

6. Bayesian inference and the problem of a priori probabilities

Probability theory offers a framework for integrating data and a priori knowledge. Let **p(A|B)** denote the relative or conditional probability of **A**, given that **B** is true. Bayes Theorem then says that **P(A|B)** can be expressed as

$$\mathbf{p(A|B) = p(B|A) \cdot p(A)/p(B)} \qquad (6)$$

where, of course, **p(B|A)** is the conditional probability of A given B, and **p(A)** and **p(B)** are the unconditional, or absolute, probabilities for A and B. Analogous formulations can be given for the case of continuous probability distributions, where the probabilities **p(A)** of events in a finite or denumerable set are replaced by probability densities **f(x)** over a continuum of points. Now, if **d** is a data set and **H** is a hypothesis, we may interpret **p(H|d)** as the probability of the hypothesis given these data. Bayes Theorem applied to this situation gives

$$\mathbf{p(H|d) = p(d|H) \cdot p(H)/p(d)} \qquad (7)$$

p(H|d) is here also spoken of as an a posteriori probability, since it the probability of H after we have weighed in the data. In contrast, **p(H)** is now called an a priori probability since it is the probability of **H** "before" the data. Note that the label "a priori" should not be taken to imply that our knowledge of **H** before the data **d** have been collected is wholly independent of all data, but only that it is a priori in relation to the very data set **d** (i.e., it is not conditional on these data).

Now the point of using Bayes' theorem in scientific inference is that **p(d|H)**, or the *likelihood* of **d** given **H**, can often be calculated. In the case when **d** is a singular case of a universal hypothesis, **d** follows logically from **H** and hence **p(d|H)** = 1. In most cases of statistical inference, **p(d|H)** differs from 1 but can be calculated exactly from probability theory. If, for example, **H** is the hypothesis that a coin gives heads with a probability of 2/3, and **d** is the event that two consecutive flips of the coin turned out to be heads, then the likelihood **p(d|H)** of this is = 4/9. Similarly, if **H** is the hypothesis that a Gaussian distribution with mean **m** and standard deviation **s** holds for a certain event, then the relative probability density **f(d|H)** of the event **d**: *the observed mean in a sample of size N is* = ***m'***, can be calculated exactly using Bayes formula.

Before I go deeper into what happens when you use Bayes' theorem for inference, let me point out that a lot of classical statistical inference stops before it gets that far. One influential school of statistical inference is called the Neyman-Pearson, or the *objective* school. The "objective" way of doing statistical inference is *estimation using the maximum likelihood principle*. This mean that if you have certain data **d** and a set of alternative hypothesis $\mathbf{H_1}$, $\mathbf{H_2}$,... $\mathbf{H_N}$ are given, you should choose that hypothesis $\mathbf{H_i}$ for which the likelihood of **d** given $\mathbf{H_i}$, $\mathbf{p(d|H_i)}$ or $\mathbf{f(d|H_i)}$, has the largest value. Suppose for example that you do not know the amount of bias of a biased coin and that you consider all the alternative hypotheses $\mathbf{H_x}$ that it gives heads with probability **x**. Also suppose that in a series of 10 throws, heads come up in 3 cases. Then the *ML* principle (the principle of maximum likelihood) says that you should choose the hypothesis $\mathbf{H_{0.3}}$, since that is the one which makes the observed data most likely. Similarly, if the mean of a certain variable in an observed

sample is **m'** and you consider all the alternative hypotheses $\mathbf{G_m}$ that you are dealing with a Gaussian process with mean **m**, then the ML principle says that out of these you should choose $\mathbf{G_{m'}}$. Again, if you look at a classification task where the decision has to be taken whether an observed object belongs to class **A** or class **B**, the likelihoods of the observation given these two hypotheses can often be calculated. Indeed (looking again at Figure 12), given that the populations **A** and **B** are distributed in a symmetric and equal way except for their different means, the *ML* principle can be used to demonstrate that the straight line halfway between the sample means divides the input space into a region where the likelihood of the input given **A**, **f(d|A)**, is higher than the likelihood **f(d|B)**, and another region where the opposite holds.

So the *ML* principle is a very versatile instrument. However, proponents of the so-called Bayesian school of statistical inference use to point out that the rationale for choosing the hypothesis with the largest likelihood is not clear. Indeed, they argue, it can be shown that in many situations *ML* is not the best principle. The core of the Bayesian message is that for accepting a hypothesis **H**, we want the probability of **H** given the data to be as high as possible. But since the probability of **H** given the data, **p(H|d)**, is something else than the likelihood **p(d|H)**, maximising the latter is not the same as maximising the former. The exact reason why this is so is of course Bayes' theorem, which shows that **p(H|d)** depends not only on **p(d|H)** but also, for example, on **p(H)**. If we are dealing with two alternative hypotheses $\mathbf{H_1}$ and $\mathbf{H_2}$ whose a priori probabilites differ, it may happen that $\mathbf{p(d|H_1)}$ is larger than $\mathbf{p(d|H_2)}$ but $\mathbf{p(H_2|d)}$ is larger than $\mathbf{p(H_1|d)}$. A simple case in point would be the situation where it is rather improbable that you are dealing with a biased coin with bias 2/3 – say, $\mathbf{p(H_1)} = 0.1$ – while it is quite probable that the coin is unbiased, $\mathbf{p(H_2)} = 0.9$. Although in this case $\mathbf{p(d|H_1)}$ is 4/9 while $\mathbf{p(d|H_2)}$ is only .25, Bayes' therom shows that $\mathbf{p(H_1|d)}$ is only 16/97 while $\mathbf{p(H_2|d)}$ is 81/97. Note, by the way, that this result can be derived by dividing out the term **p(d)** in the two applications of the theorem:

$$\mathbf{p(H_1|d) = p(d|H_1) \cdot p(H_1)/p(d)} \qquad (8)$$

$$\mathbf{p(H_2|d) = p(d|H_2) \cdot p(H_2)/p(d)} \qquad (9)$$

which in the present case gives

$$\mathbf{p(H_1|d)/p(H_2|d)} = \mathbf{[p(d|H_1) \cdot p(H_1)]/[p(d|H_2) \cdot p(H_2)]} = 16/81$$

Together with the information that $\mathbf{H_1}$ and $\mathbf{H_2}$ are the only two hypotheses available, i.e., $\mathbf{p(H_1|d)} + \mathbf{p(H_2|d)} = 1$, this leads to the desired result. This result, by the way, is consolling for common sense according to which we should not infer that a coin from our pocket is biased just because it turns up heads twice in a row.

What has been said can also be expressed as follows: the *ML* principle does correspond to the most probable hypothesis, given the data, if the considered alternative hypotheses are all equally probable a priori. If they are not, the *ML* principle and Bayesian reasoning may give different result. Applying this to the data in Figure 12, we can now see that if some hypotheses about the mean of the two Gaussians are much more probable a priori than others, the straight line which we have drawn in it need not be the most probable one given the data. Suppose for example that there are strong a priori reasons to believe that the true mean of the "healthy" population is smaller in both dimensions than the observed sample mean. (Our reasons may be that several previous studies have tended to show that.) Then Bayes' theorem says that the best dividing line, which need not any longer be a straight line, should be drawn further down and to the left.

The wider implications of all this should be clear. Bayesian reasoning is of course not confined to simple choices between a set of similar explanatory hypotheses such as two possible biases of a coin, or a set of possible means of Gaussians. Weighing in the a priori probabilities of all available alternative hypotheses, including those of different form and complexity, could in principle also solve the problem of model power. If it is a priori very much more probable that a certain relationship is linear in nature than that it is cubic, the fact that the data fit a cubic solution best does not entail that the relationsship is probably really cubic. For this reason using a non-linear neural network may not be motivated at all, since the results it can possibly give will probably not be useful. However, if we have good reasons a priori to believe that the realtion is really cubic, then a good fit to a certain cubic relation could imply not only maximum likelihood for the data given this hypothesis, but also high probability of the hypothesis given the data. Hence with this background knowledge, we should use a non-linear network.

A lot more will be said during this conference about neural networks from a Bayesian perspective, and also about the proper choice of power of one's ANN. Richard Dybowski will for example speak about so-called Bayesian neural networks, and Georg Dorffner will go into the issue of the predictive error of an ANN in relation to its power. The last few words in my lecture will instead be devoted to a few purely philosophical reflections.

The argument from Bayes theorem against the *ML* principle is logically impeccable in the sense that the derivations follow strictly from probability theory. Why, then, do the proponents of the "objective" school do not accept the argument? The reason is not that they do not believe in Bayes' theorem, or that they never apply it in statistics – of course they do – but that they object to the generalised application of it to epistemic probabilities - i.e., as a universal principle for the generation of empirical knowledge. Most of their arguments are versions of one basic question, namely, "Where do the a priori probabilities come from?". I will end my lecture with some reflections on this argument.

Try to think of the whole of science as a process where Bayes' theorem is used at each step to update our knowledge state,

$$\mathbf{p(H|d)} = \mathbf{p(d|H)} \cdot \mathbf{p(H)/p(d)} \qquad (7)$$

where, as said above, **p(H)** is the probability of **H** "before" the data **d** and **p(H|d)** is its probability "after" **d**. If we believe that this is the scheme which all empiricial justification of hypotheses must follow, and if we believe that science is wholly an empirical affair, then we must suppose that the

apriori probability **p(H)** is due to earlier applications of the same rule. But then, what about the first application of it? Where did the first **p(H)** come from?

This is what in the philosophy of science is often called *the problem of a priori probabilities*. As mentioned, proponents of the so-called objective statistical school use it as a pretext for retreating to the maximum likelihood principle. Even most Bayesian theorists usually think that the problem is so serious that they give up the idea of using Bayes' theorem as a general rule for the updating of *knowledge*. However, they claim, it can still be used as a rule for the rational updating of *belief*. This means that they interpret **p(H)** not as our degree of knowledge that **H** is true – the epistemic probability of **H** – but as our *degree of belief* in **H**. It is still rational, they argue, to update degrees of beliefs using Bayes' theorem. Hence the universal application of the theorem in science and statistics is saved, although to the price of sacrificing knowledge for belief. This is why the Bayesian school of statistics is so often called the "subjective" school.

If, now, one is a philosopher and looks at the situation in statistics at some intellectual distance, one can find it very unsatisfying. On the one hand, we have the objective school, using an inference principle (*ML*) which leaves us with the most probable hypothesis only under very restricted conditions. These conditions are not even spelled out in the theory. On the other hand, we have a theoretically much more consistent approach which, however, abstains from modelling scientific knowledge and rests content with being a theory of subjective belief. It is sometimes pointed out that the practical consequences of belonging to the one or the other school are often rather the same. This is true in the sense that with growing amounts of empirical data of relevance for a certain hypothesis, these data play a larger and larger role in the determination of its posterior probability while the prior play a correspondingly smaller role. But the difference is certainly practically relevant in many situations, especially of course when the differences between the priors of the alternative hypotheses is large and/or when the data set is small – as is the case in many of the situations where we consider using non-linear methods. Last but not the least, the situation is intellectually very unsatisfying. Does scientific knowledge rest either on subjective belief, or on inferential principles which are demonstrably misleading?

Is there any hope of reaching a third solution which is both logically consistent and can serve as a model of scientific *knowledge*? I think so, and I think that the road to go is to deny that subjectivism is a necessary consequence of Baysianism. I will not go into my arguments for this here, but only want to point out that it entails *either* denying that all scientific inference proceeds according to the Baysian scheme, *or* allowing some non-inferential knowledge about scientific hypotheses, *or* both. These are deep philosophical issues, and all what I claim to have shown today is they they are highly relevant to the theory and practice of artificial neural networks.