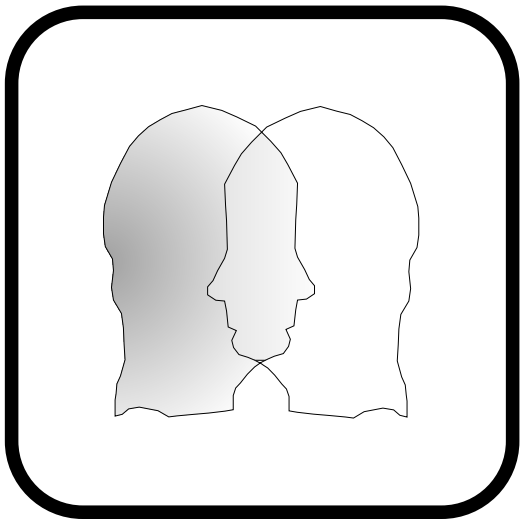
Reflex



Algorithms, data and hypotheses -Learning in open worlds-

Heinz Mühlenbein

Report number: 1995/04

Research Group for Adaptive Systems
SET.AS

Algorithms, data and hypotheses
-Learning in open worlds-

Heinz Mühlenbein

German National Research Center for Computer Science (GMD),
Schloß Birlinghoven, 53754 Sankt Augustin,
Germany.

Abstract

This paper contains an informal discussion about how to synthesize reasonable hypotheses from data. This is a fundamental problem for any system acting in the real world. The problem consists of three interconnected subproblems: fitting the past data to a hypothesis (model), selecting promising new data in order to increase the validity of the hypothesis, and selecting a hypothesis in a class of hypotheses (models). We argue that molecular electronics may be important for the development of such systems. First, it provides the computing power needed for such systems. Second, it can help in defining a new computational model urgently needed for the design of an artificial systems synthesizing hypotheses about processes of the real world.

1 Introduction

With the introduction of computers, automata have been playing a continuously increasing role in the natural sciences. In this paper I focus on special automata called *learning systems.* A learning system has a learning procedure by which it can develop methods that cannot be deduced trivially from its learning procedure. The learning system tries out hypotheses (methods) and selects the better ones. It has *a priori* a well defined *universe of hypotheses* from which it must choose those to be tried. If this universe is small, then the "inventiveness" of the machine is severely limited, and the value of the methods that it develops depends more on the astuteness of the programmer in choosing a universe containing good hypotheses than an ability of the learning system to pick the best hypothesis from among those in the universe. In order to give the learning system a "free hand", it should have a universe which, although well-defined, is so large and varied that the user of the system is not even acquainted with the forms of all the methods it contains.

Artificial learning systems need huge processing capabilities. New physical concepts of information processing have to be developed to meet these requirements. A promising research direction is molecular electronics. But I would like to mention a second reason, why molecular electronics might be interesting for the design of artificial learning systems. Learning systems, natural or artificial, face the problem of finding good hypotheses which explain the past data and which can be used for predictions. But a fundamental theorem states that hypotheses cannot be assigned a probability in the classical sense of being true [Popper, 1972]. There is a similar problem in quantum mechanics, the theoretical foundation of molecular electronics. Here also researchers are looking for an extension of classical probabilty theory. It is a general feature of quantum mechanics that one needs a rule to determine which of the alternative "histories" can be assigned probabilities [Gell-Mann & Hartle, 1992].

A rigorous treatment of the above problems

seems to be out of reach at this moment. Therefore I will concentrate on some important aspects of the general problem. The outline of the paper is as follows. In section 2 data fitting will be described from the mathematical point of view. In section 3 I extend the basic model. The general task of data modelling and its connection to Occam's razor is described in a Bayesian framework. Section 4 introduces the concept of complexity as defined in computer science and in probability theory. The problem of discrete vs. continuous representations is discussed in section 5. Some principal limitations of artificial automata are discussed in section 6. In the final section, two learning systems are discussed which have been implemented by my research group. One system models collective learning of populations in a similar way to *Darwinian evolution;* the other system models learning of an artificial organism equipped by something like a *brain.* The paper ends with a short discussion of the question: Can quantum mechanics make contributions towards finding a new computing paradigm needed for systems operating in the real world?

2 Approximation of functions

Many learning procedures can be formulated as approximation problems. Let X be the input space and Y the output space of an unknown process $s := X \longrightarrow Y$. The problem is to find an approximation $f : X \longrightarrow Y$ in a search space (the universe) F such that

$$f(x) \approx s(x) \qquad x \in X. \tag{1}$$

The approximation problem can be precisely defined if a norm is given in the search space F. In this case one looks for an ϵ-*approximation* such that

$$\|f - s\| < \epsilon. \tag{2}$$

In order to solve the problem, some information about s has to be used. I will investigate the case where the unknown solution can be computed using a finite *data set* D, where

$$D = \{(x_j, y_j = s(x_j)), j = 1 \ldots n\}.$$

Many learning procedures determine an approximation f by fitting the data according to some criterion. The most popular criterion is called *Least Mean Square Error (LMSE),* which minimizes the sum of the squared errors between the data and the model predictions

$$Err_n = \frac{1}{n}\sum_{j=1}^{n} \|y_j - f(x_j)\|^2 \longrightarrow MIN. \tag{3}$$

This minimization problem is investigated in different scientific disciplines. If the search space is a space of functions, then the problem belongs to *mathematical approximation theory.* If the search space consists of non-numeric elements such as rules or program components, then the problem belongs to *artificial intelligence.* It is called *program synthesis by examples.*

In mathematical approximation theory, many results have been obtained. A distinction is made between the *interpolation problem* and the *approximation problem.* In interpolation, the function f has to fit the data points exactly. In approximation one looks for a function f which approximates the unknown function s best according to some norm in the function space. By definition, the error of the best approximation function not more than that of the best interpolation function. But the best interpolation function can be constructed for many subspaces of functions, whereas no method has been discovered for constructing the best approximation function.

Most of the results from mathematical approximation have been obtained for the one-dimensional case only. I will summarize some well known results for the maximum norm and search spaces F consisting of polynomials. Here the best interpolation polymomial was computed

by Chebychev. The optimal interpolation points $\{x_i\}$ are given by the zeros of the Chebychev polynomials. It could be shown that the error of the best interpolation polynomial is only a factor of $O(log(n))$ larger than the error of the best approximation polynomial where n is the order of the polynomial. For the Euclidian norm, the optimal interpolation polynomial is defined at the zeros of the Legendre polynomials.

An extension of the classical mathematical approximation problem was developed by Traub [1980]. The generality and power of the extension is a result of the fact that *information and problem complexity* play a central role in this approach. In classical approximation theory, optimal algorithms are computed by making various technical assumptions about the class of algorithms and the class of problem elements. These assumptions are often not verifiable. Furthermore, depending on the assumptions, many different optimal approximations might exist. The concept of problem complexity deals with the meta-problem: how to find the best of the optimal approximations.

Problem complexity is a measure of the intrinsic difficulty of obtaining the solution to a problem regardless of how this solution is obtained. It can be defined with respect to a model of computation and a class of "permissible" information operators. Unfortunately, the determination of problem complexity is very difficult; it has been completely solved for only very few problems. I will not discuss this approach further. The interested reader is referred to Traub [1980].

The theory of optimal algorithms closes the gap between the mathematical and the statistical approach to the data-driven learning problem. In the statistical approach, the data is not reliable but corrupted by noise. Therefore the approximation problem consists of two subproblems: first to estimate the amount of noise and then to approximate the corrected data. I will discuss this problem with a simple example. Consider the problem of a non-parametric estimation of a regression function s from observations of the form $y_i = s(x_i) + \xi_i, \quad i = 1, ..n$, where $x_i = i/n$, and ξ_i are random variables such that

$$E(\xi_i) = 0, \quad E(\xi_i \xi_j) = \sigma^2 \delta_{ij} \quad \sigma^2 > 0. \qquad (4)$$

It is assumed that s is defined on [0,1] and can be represented as a Fourier series: $s(x) = \sum_{j=1}^{\infty} c_j \phi_j(x)$. Let the approximation be defined for $N \leq n$ as

$$f_N(x) = \sum_{j=1}^{N} \hat{c_j} \phi_j(x); \quad \hat{c_j} = \frac{1}{n} \sum_{m=1}^{n} y_m \phi_j(x_m). \qquad (5)$$

What is the optimal order N of the approximation? The answer depends on the optimality criterion to be used. The usual C_p-criterion leads to

$$N_{opt} = \arg_{N \leq n} \min(Err_N + 2\sigma^2 N n^{-1}), \qquad (6)$$

where

$$Err_N = \frac{1}{n} \sum_{i=1}^{n} (f_N(x_i) - y_i)^2.$$

For $\sigma \ll 1$ we have the mathematical approximation problem with the solution $N = n$. With a very large amount of noise, i.e. $\sigma \gg 1$ the optimal N can be much smaller. If we interpret N as an indicator for the complexity of the approximation model, we see that the C_p-criterion tries to balance model complexity with interpolation error.

In the next section I will treat the above problem in a general Bayesian framework. This model is able to handle noisy data as well as active data selection.

3 Data selection, model selection, and Occam's razor

In science, a central task is to develop and compare models to account for data. Two levels of

inference are involved in the task of data-driven modelling. At the first level of inference, one assumes that one of the models that was invented is true: that model is then fitted to the data. Typically a model includes some free parameters; *fitting the model to the data* involves inferring what values those parameters should probably take, given the data. This is the approach of mathematical approximation theory. The results of this inference are often summarised by the most probable parameter values and hopefully some error bars on those parameters. The second level of inference is the task of *model comparison*. Here, one wishes to compare the models in the light of the data, and assign some sort of preference or ranking to the alternatives.

Model comparison is a difficult task because it is not possible simply to choose the model that fits the data best: more complex models can always fit the data better, so the maximum likelihood model choice would lead us inevitably to implausible over-parameterised models which generalise poorly. *Occam's razor* states that unnecessarily complex models should not be prefered to simpler ones.

In this section I will survey the Bayesian approach to Occam's razor. This survey is based on MacKay [1992].

Let us write down Bayes' rule for the two levels of inference described above. Each model H_i is assumed to have a vector of parameters w. A model is defined by its functional form and two probability distributions: a prior distribution $P(w|H_i)$ which states what values the model's parameters might plausibly take; and the predictions $P(D|w, H_i)$ that the model makes about the data D when its parameters have particular values w. Note that models with the same parameterisation but different priors over the parameters are defined to be different models.

In **model fitting** it is assumed that one model H_i is true, and the model's parameters w are then inferred from given data. Using Bayes's rule, the **posteriori probability** of the parameter w is

$$P(w|D, H_i) = \frac{P(D|w, H_i)P(w|H_i)}{P(D|H_i).} \qquad (7)$$

In words:

$$Posterior = \frac{Likelihood * Prior}{Evidence}.$$

For model fitting, the normalising constant $P(D|H_i)$ is commonly ignored. It will be important in the second level of inference, and it is named *evidence* for H_i. For **model comparison** one wishes to infer which model is most plausible given the data. The posterior probability of each model is:

$$P(H_i|D) \propto P(D|H_i)P(H_i). \qquad (8)$$

The second term, $P(H_i)$, is a *subjective* prior over our hypothesis space which expresses how plausible we thought the alternative models were *before* the data arrived. This subjective part of the inference will typically be overwhelmed by the objective term, the evidence. Assuming that there is no reason to assign strongly differing priors $P(H_i)$ to the alternative models, models H_i are ranked by evaluating the evidence.

3.1 Model fitting

Let us now explicitly study the evidence in order to gain insight into how the Bayesian Occam's razor works. The evidence is defined as

$$P(D|H_i) = \int P(D|w, H_i)P(w|H_i)dw. \qquad (9)$$

For many problems, including interpolation, it is common that the integrand has a strong peak at the most probable parameters w^*. Then the evidence can be approximated by the height of the peak of the integrand times its width, $\Delta w = w - w^*$

$$P(D|H_i) \simeq P(D|w^*, H_i)P(w^*|H_i)\Delta w. \qquad (10)$$

If w is k-dimensional, and if the posterior is well approximated by a gaussian, the above equation can be computed. The facor Δw is given by the determinant of the gaussian covariance matrix (MacKay [1992]):

$$P(D|H_i) \simeq P(D|w^*, H_i)P(w^*|H_i)(2\pi)^{k/2}(detC)^{-1/2}, P(D|w, \beta, a, N) = P(\{y_i\}|\{x_i\}, w, \beta, A, N), \tag{11}$$

where

$$C = -\nabla\nabla logP(w|D, H_i).$$

Let us apply this framework to the noisy interpolation problem. For simplicity, let us assume that x and y are scalars. To define a linear interpolation model, a set of k fixed basis functions $A = \{\phi_j(x)\}$ is chosen. The interpolated function is assumed to have the form

$$y(x) = \sum_{j=1}^{k} w_j \phi_j(x).$$

The data set is modeled as deviating from this mapping under some additive noise process:

$$y_i = y(x_i) + \xi_i.$$

If the ξ have a zero-mean gaussian distribution whose standard deviation is σ_ν, then the probability of the data given the parameters is:

$$P(D|w, \beta, A, N) = \frac{exp(-\beta/2 Err_D(D|w, A))}{Z_D(\beta)} \tag{12}$$

where $\beta = 1/\sigma_\nu^2, Err_D = \sum(y(x_i) - y_i)^2$, and $Z_D(\beta) = (2\pi/\beta)^{N/2}$. Under these assumptions, finding the maximum likelihood parameters w^* is identical to minimizing the quadratic error Err_D. This is just the least mean square error ($LMSE$) criterion mentioned in section 2. It is well known that this may be an "ill-posed" problem. That is, the w that minimises Err_D is underdetermined and/or depends sensitively on the details of the noise in the data. Thus it is clear that to complete our interpolation model we need a prior R that expresses the sort of smoothness we expect the interpolant $y(x)$ to have. I will not discuss this extension here. Strictly one should write

since interpolation models do not predict the distribution of inputs $\{x_i\}$. But with the Bayesian framework this problem, often called *active learning* or *sequential design*, can also be addressed. There are two scenarios in which one would like to actively select training data. In the first, data measurements are expensive or slow, and the researcher wants to know where to look next so as to learn as much as possible. In the second scenario, there is an immense amount of data, and one has to select a subset of points that are the most useful.

For active data selection, objective functions have to be defined which measure the *expected informativeness* of candidate measurements. At least three different criteria are possible: maximizing the total information gain, maximizing the information gain in a region of interest, and maximizing the discrimination between two models. All these criteria depend on the assumption that the hypothesis space is correct. This is their main weakness. Paaß and Kindermann [1995] used the variance of the predictions of a population of models. Data is selected in areas where the variance is highest.

3.2 Model comparison

I now proceed with the second level of inference, model comparison. To rank alternative basis sets A, noise models N and regularisers R in the light of the data D, the posterior probabilities for alternative models $H = \{A, N, R\}$ are examined:

$$P(H|D) \propto P(D|H)P(H). \tag{13}$$

Assuming that there is no reason to assign strongly differing priors $P(H)$, alternative methods H are ranked just by examining the evidence $P(D|H)$.

A slightly different approach to the model selection problem uses the *minimal description length* of Rissanen [1992]. It is restricted to binary problems. Let C be an injective coding function from a discrete set X into the set of all binary strings B^*. Let $L(x)$ be the length of $C(x)$, i.e., the number of binary digits in C(x). A code C is said to be a prefix code, if

$$\sum_{x \in X} 2^{-L(x)} \leq 1. \tag{14}$$

Thus, a prefix code defines a distribution on X. Shannon's fundamental coding theorem states that for a given distribution P(x), all prefix codes must have a mean length bounded below by the entropy

$$\sum_x P(x)L(x) \geq -\sum_x P(x) log_2 P(x) \tag{15}$$

The lower bound can be reached only if the lengths satisfy the equality $L(x) = -log_2 P(x)$ for every x. In this sense one could call $-log_2 P(x)$ the Shannon complexity of x relative to the "model" P.

The above analysis can be extended to a whole class $M = \{P(y|x,\theta)\}$ where θ ranges over some subset of the k-dimensional Euclidean space. In this case the minimum description length criterion can be computed, which combines model complexity, the number of parameters and the precision of the data n.

$$MDL(y|x,k) = -log_2[P(y|x,\hat{\theta})] + \frac{k}{2} log_2 n \tag{16}$$

MDL has to be minimized over k to get the optimal model complexity.

3.3 Some remarks

Bayesian model selection is a simple extension of maximum-likelihood model selection: the evidence is obtained by multiplying the best fit likelihood by a model complexity factor. The evidence is a measure of a model's *plausibility*. The amount of CPU time required to run a model is not addressed. Choosing between models on the basis of how many operations they need can be seen as an exercise in *decision theory*. This needs further study.

The Bayesian framework does not lead to new learning procedures, but it is very useful in clarifying the many implicit assumptions hidden in the specific learning procedures.

The framework presented in this section is currently one of the most advanced methods for data-driven learning. Its application depends on many assumptions. The crucial question is whether these assumptions are fulfilled for an unknown data set.

I will now discuss other measures of the complexity of a problem.

4 Information, complexity and uncertainty

In computer science the complexity of a problem is measured by the length of the *shortest program* written in some standard language (e.g., a program for a Turing machine) by which the problem can be solved. This information is called the *algorithmic complexity* of the problem. Often this measure cannot be computed. It is therefore of limited practical use. Furthermore, it does not take into account how many operations have to be executed by the program to solve the problem. Such a measure is the *computational complexity*.

Computational complexity is a characterization of the time or space requirements for solving a problem by a particular algorithm. Both of these

requirements are usually expressed in terms of a single parameter that represents the size of the problem.

Definition: *The time complexity function $f(n)$ of an algorithm is the largest amount of time required to solve a problem of size n.*

It has been very useful to distinguish between two classes of algorithms by the rate of growth of their time complexity function. One class is called P. It consists of *polynomial time algorithms.* Here the time complexity can be expressed in terms of a polynomial. The second class of algorithms consists of *exponential time algorithms.* This class is called "non polynomial", NP. More precisely the NP classes of problems are defined as follows. If an individual problem has a solution, then the algorithm will find that solution in *exponential time.* But it must be possible to check in polynomial time that the proposed solution is indeed a solution.

NP problems arise in many contexts. A very popular problem is the "travelling salesman problem". Here one seeks a tour which visits each city exactly once for which the distance is a minimum. The number of possible tours grows exponential in the number of cities. In fact, this problem is not only NP, but what is named as *NP-complete.* This means that any other NP problem can be converted into it in polynomial time. It is commonly believed by computer scientists that it is impossible, with a Turing machine-like device, to solve an $NP - complete$ problem in polynomial time.

Conjecture: $P \neq NP$

This conjecture remains the most important unsolved problem in complexity theory. NP problems are the hard ones. For large problem sizes, they are *transcomputational.* This term was coined by Bremermann [1962]. An algorithm which needs more than 10^93 operations is transcomputational. It cannot run until completion on any real computational system. The exact number is not so important, but it shows that there are definite limits to the computational power of any system in our universe. This bound has implications for NP problems. The travelling salesman problem for instance has approximately 10^90 tours for 66 cities. In real life one is interested in good solutions for problems with more than 1000 cities.

Bremermann [1962] derived his bound by simple considerations based on quantum theory. It is surely an upper bound.

Bremermann's bound: *No data processing system, whether artificial or living, can process more than $2 * 10^{47}$ bits per second per gram of its mass.*

Bremermann derives the limit from the following considerations based on quantum physics. The phrase "processing x bits" means the transmission of that many bits over one or several communication channels within the computing system. Now assume that information is encoded in terms of energy levels within the interval $[0, E]$. Assume further that energy levels can be measured with an accuracy of only ΔE. The most refined encoding is defined in terms of markers by which the whole interval is divided into $N = E/\Delta E$ equal subintervals, each associated with the amount of energy ΔE. In order to represent more information with the same amount of energy, it is desirable to reduce ΔE. The extreme case is represented by the Heisenberg principle of uncertainty: energy can be measured to the accuracy of ΔE if the inequality

$$\Delta E \Delta t \geq h \tag{17}$$

is satisfied. This means that

$$N \leq \frac{E\Delta t}{h}.$$

Now by Einstein's formula

$$E = mc^2.$$

If we take the upper (most optimistic) bound of N we get

$$N = \frac{mc^2\Delta t}{h}.$$

Substituting numerical values for c and h, one obtains $N = 1.36 \cdot 10^{47} m\Delta t$.

Using this bound, Bremermann calculated the total number of bits processed by a hypothetical computer the size of the earth within a time period equal to the estimated age of the earth. He computed 10^{93} bits. This number is referred to as *Bremermann's limit*. Problems that require processing more than 10^{93} bits of information are called *transcomputational problems.* It is obvious that exponential time algorithms are already transcomputational for fairly small problem sizes.

Recent research in fuzzy sets and probability theory has taken a different approach in trying to define complexity. It is not absolutely defined, but relative to the knowledge of a given observer. A good survey about the different definitions is given by Klir and Folger [1988].

Two general methods of defining system complexity can be distinguished: one is based on *information*, the other on *uncertainty.* In the first one the complexity is proportional to the amount of information required to *describe the system.* In the second one, system complexity is proportional to the amount of information needed to *resolve any uncertainty* associated with the system.

To the neurophysiologist, for instance, the brain consists of a network of fibers and a soup of enzymes. Therefore the transmission of a detailed description of it requires much time and space. To a butcher, in contrast, the brain is simple, for he has to distinguish it from only about thirty other types of "meat".

Both definitions of complexity are relative to an observer and its knowledge. They are related to each other under the *closed world assumption.* With this assumption the universe of discourse can be divided into two sets: the set of events known as possible and the set of events known as impossible. The set of unknown events is assumed to be empty.

The above definitions have been primarily used for the purpose of developing computational methods by which systems that seem incomprehensible can be simplified to an acceptable level of complexity. There is a major problem with this approach. Even if one has found an algorithm that reduces the complexity of the given system, the computational complexity associated with the simplification algorithm has to be taken into account. If the resulting algorithm is transcomputational, it is of no practical use.

Another severe problem is the closed-world assumption. The real world is open for any system operating in it. For any system the set of unknown events is infinite. Unfortunately, the scientific understanding of *open worlds* is in its infancy. I believe that the development of a calculus for dealing with open systems is one of the most important problems in epistemology, probability theory, and also quantum physics. I will just mention the work of Jaynes [1992]. He raises the question of whether probability theory is a "physical" theory of phenomena governed by "chance" or "randomness" or whether it should be considered as an extension of logic, showing how to reason in situations of incomplete information. Jaynes [1992] remarks: "We then see the possibility of a future quantum theory in which the role of incomplete information is recognized: for any variable F, the dispersion $(\Delta F)^2 =< F^2 > - < F >^2$ represents only the accuracy with which the theory is able to predict the value of $F \cdots$ When ΔF is infinite, it means only that the theory is completely unable to predict F. The only thing that is infinite is the uncertainty of prediction."

In summary: The concepts of information, complexity, and uncertainty are used differently in different disciplines. In the future a common framework of these concepts is needed. Quantum mechanics can play a major role in this development. I will now turn to another important topic for any system, that of the representation.

5 Discrete vs. continuous representations

The theory of computing has been centered on the binary, all-or-none type. It has been, from the mathematical point of view, combinatorial rather than analytical. Rigid, all-or-none concepts have little connections to the continuous concept of real or complex numbers, on which mathematical analysis is based. John von Neumann [1948], one of the founders of today's computers, warned: "Formal logic is, by the nature of its approach, cut off from the best cultivated portions of mathematics, and forced onto the most difficult part of the terrain, into combinatorics." Therefore von Neumann predicted that a powerful theory of automata will differ from the present system of formal logic in two relevant aspects.

1. The actual length of "chains of reasoning", that is, of the chains of operations, will have to be considered.
2. The operations of logic will all have to be treated by procedures which allow exceptions. All of this will lead to theories which are less rigid than past and present formal logic.

Von Neumann continued: "There are numerous indications to make us believe that this new system of formal logic will move closer to another discipline which has been little linked in the past with logic. This is thermodynamics, primarily in the form it was received from Boltzmann."

In my opinion, von Neumann's predictions turned out to be right. The importance of the length of the chain of operations was first recognized in computer science. It lead to the theory of computational complexity discussed in the previous section. Boltzmann's thermodynamics approach, especially the concept of entropy, is becoming increasingly popular in the design of new learning systems. I like to call this new emerging field "quantitative artificial intelligence".

New learning systems now under development for robotics do not use just one learning procedure: they frequently employ different learning procedures at different levels of the system architecture. The learning systems try to process both discrete and continuous information. Some promising new architectures consist of three levels. An overview and a sematic description of the three levels is shown in the following table.

Semantics	Characteristics
plans relations objects	discrete processes discrete values
features	discrete processes continuous values
signals	continuous processes continuous values

At the most abstract level, there are *discrete values and discrete processes.* The idealization of this representation is that its members can be characterized abstractly as a set of discrete elements. At the lowest level, the information is in the form of *continuous values and continuous processes.* The constraints at this level are captured by Shannon's information and his sampling theorem. The idealization of this representation is that of a continuous function of a set of variables, e.g., $y = f(x,t)$. In between these extremes there is an intermediate level that can be characterized as requiring *continuous values of discrete processes.* For example, the rotation of the visual field can be characterized by rotational values of a single rigid body motion process. There is only one process, but the actual parameter values that describe that process are continuous.

After describing some advanced methods how to synthesize reasonable hypotheses from data, I will discuss the question: What are the principal limits of such an approach? I will show that there are limitations, which follow from the general *induction problem*, discussed intensively by Popper [1972].

6 Principal limitations of artificial automata

In 1943 McCulloch and Pitts proved this remarkable theorem: *Anything which can be defined at all logically, strictly and unambiguously in a finite number of words can also be realized by an artificial neural network.* At first this theorem seems to indicate that an artificial system is able to solve any clearly defined problem. But the content of the theorem has to be interpreted differently. This was shown by von Neumann [1948] who raised the two questions:

- Can the network be realized within practical limits, e.g., is the required number of connections less than the number of atoms in the universe?
- Can every existing mode of behavior be put completely and unambiguously into words?

Let us discuss both questions with a specific example, the classification of geometrical entities as performed by humans. There have been three approaches to this central problem of vision. I call them the *theoretical*, the *learning-from-example* and the *copy-the-brain* approach.

In the theoretical approach researchers try to find a computational calculus that solves the classification problem. Up to now, a calculus has only been developed for very restricted idealized geometric objects. The limitations of the learning-by-examples approach was already discussed by von Neumann [1948]. He argued as follows: There seems to be no difficulty in describing how an automata might be able to identify any two rectilinear triangles. The classification of more general kinds of triangles — triangles whose sides are curved, triangles that are indicated by shading, etc. can also easily be done. Next we want the system to recognize handwritten objects and letters. Von Neumann remarks: "At this point we should have the vague and uncomfortable feeling that a complete catalogue along such lines would not only be exceedingly long, but also unavoidably indefinite." These problems, however, constitute only a small fragment of the more general concept of identification of analogous geometrical entities. This, in turn, is only a microscopic piece of the general concept of analogy. "Nobody would attempt to describe and define within any practical amount of space and time the general concept of analogy which dominates human vision". Learning from examples just by enumeration is not effective for large problem domains. The number of examples goes to infinity.

Therefore, a bottom-up approach was tried — replicating the brain, which obviously solves the classification problem — instead of solving the problem. The only way to define what constitutes a visual analogy may be a description of the connections of the visual cortex of the human brain. Any attempt to describe it by literal and formal-logical methods may lead to something less manageable. But this means that *the connections of the brain might be the simplest description of the functions it can perform.* Von Neumann remarks: "In fact, results in modern logic indicate that phenomena like this have to be expected when we deal with really complicated entities."

But the brain consists of about 10^{12} neurons and 10^{16} connections. How long will it take to produce a description it? Furthermore, the structure of the brain only partly defines visual analogy. The data flow, i.e., the processing of the data, must also be described. Such a description might be finite, but it is obviously *transcomputational.* It cannot be expressed by using all the atoms in the universe. "Obviously, there is on this problem no more profit in the McCulloch-Pitts result." Von Neumann concludes his discussion of the theorem with the remarks: "It may be, however, that in the process of understanding the central nervous system, logic will have to undergo a pseudomorphosis to neurology to a much greater extent than the reverse. One of the relevant things we can do at this moment with respect to the theory of the central nervous system is to point out the directions in which the real problem does not lie."

7 Learning from nature

In the previous section I showed some general limitations of artificial automata. Nevertheless, each system, whether natural or artificial, can and should *improve its capabilities*. Learning is a dominant feature of living beings. Therefore, my research group at the GMD concentrates on learning methods used in nature. Currently we model two different natural learning methods. One method models collective learning of populations similar to *Darwinian evolution*, the other method models individual learning done by organisms equipped with a *brain*. Learning and adaptation is one of the most important features of nature. Therefore it seems that *learning from nature* is a good strategy. This was already advocated by John von Neumann [1948]. He wrote: "Some of the regularities which we observe in the organization of natural organisms may be instructive in our thinking and planning of artificial automata. Conversely, a good deal of our experiences with our artificial automata can be to some extent projected on our interpretations of natural organisms."

In learning by simulating evolution, we distinguish between two models. One model is based on *natural evolution* without any central control Mühlenbein [1991], the other model is based on *artificial selection* as carried out by human breeders. The second model, the *breeder genetic algorithm* has been used successfully for large-scale optimization problems. The theory of this algorithm is based on the equation for the response to selection, which is also used by breeders. An overview can be found in Mühlenbein and Schlierkamp-Voosen [1993, 1994].

The second learning method models learning in individuals. The emphasis is on real-world applications. The learning method is surprisingly similar to learning by evolution. It can be called the *Darwinian model of individual learning.* The model is based on the philosophy of Popper [1972]. From the data seen so far, the learning system generates hypotheses explaining the data. The hypotheses are used to predict the outcome for new data. All hypotheses are preliminary, they are evaluated according to how well they explain the data. A fundamental problem is the fact that *hypotheses cannot be assigned a probability of being true.* This was most clearly stated by Popper [1972]. Hypotheses cannot be ranked according to a probability, two hypotheses can only be compared according to their likelihood of explaining the data. This is a general formulation of the classical *induction problem.*

The induction problem can easily be shown. Let the unknown function generating the data be a Boolean function of input size n. If $2^n - 1$ inputs are given, two hypotheses are left which explain all the data. Each hypothesis will correctly predict the output with a probability of only 0.5 for the very last input. If a smaller input set is given, the probability of correct prediction goes to zero rapidly.

Quantum mechanics is faced with a surprisingly similar problem. Not every "history" in quantum mechanics can be assigned a probability of being true. In order to derive an understandable calculus, Gell-Mann [1992] proposes a decoherence functional. It is a complex functional on any pair of histories in the set of alternative histories. Decoherence is also critical to molecular electronics. Two quantum systems that have interacted in the past (which is necessary to process information) and evolve coherently in time cannot be separated again. In order to assure the independent preparation and measurement of the subsystems, it is necessary to include dissipation which destroys the coherence between the two subsystems. To make my point clear: I am not saying, that researchers in quantum mechanics are working on the induction problem for learning systems in general. But within their smaller domain of research, they seem to have similar methodological problems as a designer of a learning system.

Our current approach to learning in an open-world problem is based on the idea of *reflection*. The learning system continuously observes and assesses its own behavior. It tries at every step to be aware of *what it knows and what it*

does not know. A system meeting this claim is able to learn incrementally, and, furthermore, to actively explore its environment. Our first implementation is a hand-eye robot which consists of two arms and sensors. Design principles and some applications can be found in Beyer and Smieja [1995] and Smieja [1995].

8 A quantum neural computer

So far, all computers have been designed based on rigid all-or-none concepts. I have shown the limitations of this approach. Currently, more flexible concepts are emulated by software on the otherwise rigid hardware. Intelligence has been taken by many scientists to emerge from the complexity of the interconnections between the neurons of the brain. I have argued in this paper that it seems to be impossible to model this interconnection scheme on a computer. It is transcomputational. It seems therefore fruitless to build an intelligent system by a "copy-the-brain" approch.

The most promising way is the "learning-from-examples" approach. Unfortunately this approach suffers from the induction problem, which has been discussed most vividly by Popper [1972]. I hope, that this paper has shown, that molecular electronics and computer scientists should work together for two reasons. The first one, the conventional one, is just to increase the speed of the computation and leave the computational model as it is. The second one, the theoretical one, is to investigate new models of computation, based on quantum mechanics. A promising approach is the "many histories" view of Gell-Mann.

Literature

Beyer, U. and Śmieja, F.J.: *Learning from examples, agent teams and the concept of reflection*, International Journal of Pattern Recognition and Artificial Intelligence, to be published (1995).

Bremermann, H.J.:*Optimizing through Evolution and Recombination.* In: Self-Organizing Systems, M.C. Yovits (ed.), Spartan Books, Washington, pp. 93-106 (1962).

Gell-Mann,M. and Hartle, J.B.: *Quantum Mechanics in the Light of Quantum Cosmology.* In: Complexity, Entropy and the Physics of Information; W.H. Zurek (ed); pp. 425-458; Addison-Wesley, New York (1992).

MacKay, D.J.C.: *Bayesian Methods of Adaptive Models.* Ph.D Thesis California Institute of Technology, Pasadena, (1992).

Jaynes, E.T.:*Probability in Quantum Theory.* In: Complexity, Entropy and the Physics of Information; W.H. Zurek (ed); pp. 381-404; Addison-Wesley, New York (1992).

Klir, G.J. and Folger, T.A.: *Fuzzy Sets, Uncertainty, and Information.*, Prentice Hall, London. (1988).

McCulloch, W.S. and Pitts, W.: *A logical calculus of the ideas immanent in nervous activity.*, Bull. of Mathematical Biophysics 9:127-147 (1943).

Mühlenbein, H.: *Evolution in Time and Space: The Parallel Genetic Algorithm.*, In. Foundations of Genetic Algorithms; G. Rawlins (ed.); pp.316-337, Morgan Kaufmann, San Mateo, (1991).

Mühlenbein, H. and Schlierkamp-Voosen, D.: *Predictive Models for the Breeder Genetic Algorithm: Continuous Parameter Optimization*, Evolutionary Computation 1;1:26, (1993).

Mühlenbein, H. and Schlierkamp-Voosen, D.: *The science of breeding and its application to the breeder genetic algorithm*, Evolutionary Computation 1;335-360, (1994).

Paaß, G. and Kindermann, J.: *Bayesian Query Construction for Neural network Models.*, In: Advances in Neural Information Processing Systems 7; G. Tesauro, G., D. S. Touretzky, T. K. Leen (eds.), MIT Press, (1995).

Popper, K.R.: *Objective Knowledge*, Clarendon Press, Oxford, (1972).

Rissanen, J.: *Complexity of models.*, In: Complexity, Entropy and the Physics of Information; W.H. Zurek (ed); pp. 117-126; Addison-Wesley, New York (1992).

Śmieja, F.J.: *The Pandemonium system of reflective agents*, IEEE Transactions on Neural Networks, (to be published) (1995).

Traub, J.F. and Wozniakowski, H.: *A General Theory of Optimal Algorithms.*, Academic Press, New York, (1980).
Von Neumann, J.: *On the Logical and Mathematical Theory of Automata*; in Collected Works of John von Neumann V, pp:288-328, Pergamon Press, London, (1965).