External and internal complexity of complex adaptive systems

Jürgen Jost*†

December 16, 2003

Abstract

We introduce concepts of external and internal complexity to analyze the relation between an adaptive system and its environment. We apply this theoretical framework to the construction of models in a cognitive system and the selection between hypotheses through selective observations performed on a data set in a recurrent process and propose a corresponding neural network architecture.

1 Introduction: Complex adaptive systems

A complex adaptive system is situated in an environment. That environment is always more complex than the system itself, and therefore, it can never be completely predictable for the system, but the system depends on regularities of the environment for maintaining its energy supply needed to support its internal structures and processes. Thus, the input that the system can receive or extract from its environment has regularities as well as aspects that appear random to the system. Only the regularities are useful for the system because by its very nature, a system will itself be defined by regularities that it constructs from its input and that are maintained through and expressed by internal processes. So the system needs external regularities that it can translate into these internal ones while random input at best is useless and at worst detrimental for the system. It depends on the system itself and its internal model of the external environment, however, which part of the potential input is meaningful and regular and which part is devoid of meaning and structure, and random. In that situation, adaptation consists in increasing the former at the expense of the latter, under the capacity constraints imposed by the system's internal structure. This means that the system on one hand will try to extract as many regularities as possible from the environment and on the other hand internally represent those as efficiently as possible in order to make optimal use of its capacity. We shall introduce the notions of external complexity and internal complexity in order to be able to investigate these two complementary aspects conceptually and quantitatively. Our main thesis will be that complex adaptive systems

*Max Planck Institute for Mathematics in the Sciences, Inselstr.22-26, 04103 Leipzig, Germany, jost@mis.mpg.de

†Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA, jost@santafe.edu

try to increase their external complexity and to reduce their internal complexity. Each of these two processes will operate on its own time scale(s), but they are also intricately linked and mutually dependent upon each other. So, for example the internal complexity will only be reduced under the assumption of fixed input, fixed external complexity, to represent that given input more efficiently, but when the system wants to handle additional, new input, to increase its external complexity, it may then also first increase its internal complexity and thereby create the potential for a subsequent reduction of the internal complexity on another time scale perhaps. The increase of internal complexity can for example occur through the creation of redundancy, e.g. duplication of some internal units or structures. Upon this redundancy, a process of differentiation or specialization can operate, through random mechanisms or internal selection, so that the system will become able to handle more diverse input and thereby increase its external complexity. Once this happened, the system can then again try to represent this newly acquired input more efficiently and thus decrease its internal complexity. Conversely, for the decrease of internal complexity, the system can also discard some of its input as irrelevant and meaningless for its purposes and thus decrease the external complexity. Again, the decrease of external complexity required for the selection of the most relevant input is a subsidiary process, and the primary goal remains to increase external complexity and to decrease internal complexity.

While ultimately, the system depends on its environment, what is relevant for the system is only what is reflected in its internal model. Therefore, for example, external complexity is not evaluated as the amount of raw data gathered by the system, but rather by what can be processed in the internal model. More precisely, the system does not represent some invariant external reality, but rather constructs its own model according to which it operates and that is only modulated by external input (see [2] for a discussion of this issue in the context of neurosemantics). What is counted as input is decided by the internal model and not by the environment. In particular, complexity, even what is called external complexity, thus ultimately becomes an internal criterion.

At least this is what we can learn from the neurosciences and the theory of cognition. A similar duality has also been proposed in evolutionary biology. Gould[5] contrasts form and function. The latter refers to the adaptation to an external environment that could possibly be lethal for the organism. The former represents the internal laws of structure and development that predetermine those adaptation possibilities that the system possesses. An evolved organism is not a completely flexible conglomeration of functional adaptations to various external requirements and challenges, but rather an intrinsically constrained and therefore rather stable form that had evolved some features that by chance proved adaptable to new external circumstances, perhaps by acquiring new functions, different from those for which it originally evolved.

The environment is dangerous, and the organism has to adapt by generating appropriate functions (here, on an evolutionary time scale). Its potential for adaptations, however, is determined essentially by its internal form. It is essential to capture this balance conceptually.

2 External and internal complexity

External complexity measures the amount of input, information, energy obtained from the environment that the system is capable of handling, processing. Of course, for our purposes, it is important that this can be measured as an entropy – and therefore, terms like “energy” need some qualification when employed in this context. In this sense, external complexity is data complexity.

Internal complexity measures the complexity of the representation of this input by the system. In this sense, internal complexity is model complexity.
The aim of the system then is to handle as much input, as many data as possible with as simple a model as possible. In fact, the simpler the model for representing some given data set is, the more capacity is free to process additional input and to increase its external complexity. The more input can be handled, the more the system can grow and develop and enlarge its capacity and the higher the system's potential for reducing internal complexity is. Thus, the system will try to increase, to maximize its external complexity, and to reduce, to minimize its internal complexity.
These aims may seem conflicting, but such a conflict is avoided when these two processes operate on different time scales. So, given the internal model that the system employs for organizing the input, it tries to increase the complexity of that input. Given the input, conversely, it tries to simplify its model representing that input and thus to decrease complexity. Of course, what is for example the input depends on the time scale under investigation. On a short time scale, the input just consists of individual signals, whereas on a longer time scale, it is given by a probability distribution for the signals drawn from a certain signal space. A successful model cannot just be based on single input signals, but its aim is to account for the regularities in sequences of input signals, and so, it cannot be fully adapted on the time scale on which the individual signals are received as input. Nevertheless, the two processes of the increase of external complexity and the decrease of internal complexity are not independent of each other, and so, their respective time scales become linked. Also, what is considered and accepted as input in the sense of a meaningful signal, instead of being discarded as meaningless noise, in turn depends on the internal model of the system. In the present essay, however, we do not explore the constructive act of the creation of meaning by the system, and we refer to [9] instead.
Before we are able to give formal definitions, we need to clarify one further aspect. There is no such thing as raw data for a system. What is a datum for a system, as opposed to something which is simply ignored and does not exist for the system, and also what constitutes a distinction between different inputs, as opposed to inputs that are lumped together and considered as identical by the system, depends on an internal model θ. In that situation, the system does not just attempt to acquire inputs that increase the complexity but are meaningless within the model, but on one hand also adapts the model, while on the other hand it also selects the inputs so that the model can get simplified. Thus, a mechanism for decreasing internal complexity is necessary for balancing the increase of external complexity through taking in additional input.

If a system differentiates into subsystems, then the roles of external complexity and internal complexity get reversed for the subsystems. Namely, the encompassing system becomes part of the environment for the subsystems, to the degree that they become autonomous and independent from the rest of the system, and conversely, the subsystems then can become part of the inner environment for the original system. Thus what is internal to the whole system may represent external input for the subsystems, and on the basis of their internal processes and operations, the subsystems produce output that can then be treated by the rest of the system as external input. Of course, the subsystems will never become completely autonomous and independent of the rest of the system, but only partially so. Our concepts of external complexity and internal complexity then provide formal tools to analyze this relationship.

We now proceed to formal definitions of our complexity concepts based on the

entropy concepts of statistical mechanics and information theory. Given a model θ, the system can model data as $X(\theta)$, with $X = (X_1, ..., X_k)$, and we assume that $X(\theta)$ introduces an internal probability distribution $P(X(\theta))$ so that an entropy can be computed in (1) below. Our hypothesis then is that the system will try to maximize the **external complexity**

$$-\sum_{i=1}^{k} P(X_i(\theta)) \log_2 P(X_i(\theta)). \tag{1}$$

The purpose of the probability distribution $P(X(\theta))$ is simply to quantify the information value of the data $X(\theta)$. In principle, this quantification is also possible through other means, for example through the length of the representation of the data in the internal code of the system. If we assume optimal coding, however, which is a consequence of the minimization of internal complexity, then the length of the representation of a datum $X_i(\theta)$ behaves like $\log_2 P(X_i(\theta))$ (a code is good if frequent inputs are represented by short code words, see [17]).
How external complexity can be increased then depends on the time scale involved. The system can try to increase the amount of information $X(\theta)$ that is meaningful within the given model θ on a short time scale, or it can adapt the model θ on a longer time scale so as to be able to process more input as meaningful. When the input is given, however, for example when the system has gathered input on a time scale when the distribution of input patterns Ξ (we use a different letter now to denote the inputs because we are now considering patterns on a different time scale) becomes stationary, then the model should be improved to handle that input as efficiently as possible, i.e. to decrease the internal complexity which we now define as follows:

$$= -\sum_{i=1}^{k} P(\Xi_i|\theta) \log_2 P(\Xi_i|\theta) \ - \log_2 P(\theta) \tag{2}$$

wrt θ. Thus, as in Rissanen's minimum description length principle [17],[1] the variational problem is

$$\min_{\theta}(-\log_2 P(\Xi|\theta) - \log_2 P(\theta)). \tag{3}$$

(We are leaving out the notation for the expectation value, i.e. the summation over the different input patterns, for the first term here, as the emphasis is on the model and not on the data.) The expression to be minimized now consists of two terms, the first measuring how efficiently the data are encoded by the model, and the second one how complicated the model is. Of course, the probability $P(\theta)$ assigned to a model depends on the internal structure of the system, and in principle that internal structure then also becomes subject to optimization, in the sense that frequently used or otherwise important models get higher probabilities than obscure ones. In computer science, this term simply corresponds to the length of the program required to encode the model, and so ultimately it depends on the binary alphabet as the ultimate code. Obviously, other systems may employ other basic codes that have to be considered as fixed on the time scale under consideration, like the neural code as represented by spiking patterns of neurons in brains. This does not affect our general scheme that does not depend on the nature of the code employed.
Below, however, the variational principle (3) will undergo an important modification when we shall introduce the concept of an observation in addition to datum and model.

[1] Essential differences between that principle and our approach will appear below.

3 Learning

We shall now analyze the preceding concepts in a more specific setting, namely the one of learning.

Learning is the transformation of correlations into associations. These associations serve to predict and anticipate future events.

There exist different criteria for the evaluation of a learning process. In statistical learning theory as developed by Vapnik and Chervonenkis ([19, 20]), the natural criterion is the expected prediction error on future data of a model based on partial and incomplete information. The task is to construct a probability distribution drawn from an a-priori specified class for representing the distribution underlying the random data received. In this theory, it is shown that this error depends both on the accuracy of the model on the training set as well as on its simplicity. If the model produces a high error on the training data, then it is plausible that one should also expect a high error on future data drawn from the same distribution. If the model is too complicated, one encounters the risk of over-fitting the training data and to thus incorporate spurious or putative regularities into the model that will not be born out by future data. Therefore, the model should be drawn from a model class of bounded complexity, the precise complexity measure in this context being the Vapnik-Chervonenkis dimension. In this theory, one may then also adapt this complexity as the size of the data set grows. In its simplest form, statistical learning theory thus finds a representation with smallest error in a class with given complexity constraint and on this basis estimates the expected error on future data drawn from the same distribution by the error on the test or training set plus an over-fitting error that is controlled by the complexity (the Vapnik-Chervonenkis dimension). In particular, in the framework of statistical learning theory, one can understand the issue of over-fitting vs leaving out some regularities. This is not the same, because over-fitting is caused by including into the model what is noise in the data. This leads to putative regularities. Leaving out regularities in our sense means that one does not recognize that the model can be simplified. Over-fitting also makes the model more complicated than necessary, but for the reason that too much attention is paid to details that turn out to be irrelevant. Either can be avoided by imposing constraints on the model complexity.

Our question is what amount of data compression is allowed by the regularities present in the data set. If the internal complexity is chosen too small the model does not have enough capacity to represent all the important aspects of the data set, i.e. is not sufficient to recreate the data to the desired degree of accuracy. If the internal complexity is too large, on the other hand, then the model is not forced to represent the data efficiently, i.e., can leave out some of the regularities and forego some possibilities for compression of the data set. A special case of this problem is solved by the approach of Computational Mechanics as developed by Crutchfield, Shalizi and coworkers (see in particular [18]) that first identifies the class of all representations with maximal prediction power and then chooses the simplest one among those. However, in our setting the problem is what are the regularities in the data that on one hand allow a compression and on the other hand are both valid and sufficient for future predictions, as opposed to what is noise and random effects. As will become clear below what is noise is not solely a property of the data set, but also depends on the model adopted or constructed by the system.

The problem addressed by statistical learning theory is to construct a probability distribution on the basis of a training set that is a random and incomplete representation of an underlying input space. Other theories rather treat the problem of the representation of a given data set without regard to future data, i.e. in a setting where only the data received are of interest. In Rissanen's principle of Minimum Description Length ([17]), the criterion is the efficiency of the representation of a

given data set. This again has two components, one being the complexity of the model that represents the data, or the length of the program needed to describe the model, the other one being the length of the representation of the data by the model. So, a good model is both simple itself and able to represent the data efficiently. In this theory, one may derive an estimate for the expected error on future data drawn from the distribution for which the model has been developed that is qualitatively similar to the one from statistical learning theory. A practical difficulty can be the guess for a good model[2] class. In statistical learning theory, this is not an issue because the model class is typically assumed given. In the approach of Gell-Mann and Lloyd ([4]), the description of the data set is split into the regular part, represented by an ensemble of which the data distribution is a typical member, and the random part that is not captured by the properties of this ensemble. For an efficient representation, the sum of the corresponding complexity measures should not exceed the complexity of the data set by itself. In this manner, the latter represents an effective lower bound for the efficiency of the representation.
All these approaches punish overly complicated models, either indirectly by causing an over-fitting error, or directly by including a term measuring the model complexity into the functional to be minimized.
This also fits well into our framework. The internal complexity should not be too large, or preferably even minimized under appropriate constraints on the adequacy of the representation of the data. This internal complexity then represents the model complexity. The other term is related to the external complexity, because that is enlarged if the data are represented accurately, i.e., if the error on the training set is controlled. This is related to Jaynes' principle of maximizing the ignorance ([8]). Jaynes argued that a model representing data should have the largest possible entropy under the constraint that all observations made on the data be reproduced. The advantage of this principle is that this eliminates any putative regularities in the model not supported by the data. In particular, this avoids the problem of overfitting. As argued by Gell-Mann and Lloyd, this principle needs to be constrained so as not to eliminate the essential regularities in the data set as represented by the ensemble in which they are embedded, and to avoid having an overly complex representation through ignoring established regularities in the data. In any case, this principle expands the random part at the expense of the regular part, and Gell-Mann and Lloyd then assure that one still stays on the optimal line given by the intrinsic complexity of the data set. Thus, Jaynes' principle maximizes the external complexity, but it needs to be constrained so as not to overshoot its purpose by ignoring possibilities for data compression, i.e. for decreasing internal complexity which then frees capacities for handling additional data and thereby increase the external complexity more efficiently in a new direction.
This also leads to still another criterion for the evaluation of models representing data drawn from some distribution, namely the information gain made possible both by the choice of the model, and by the selection of the data to be gathered or observed. The latter is a new aspect that has not yet been addressed in the above discussion. This leads us to a more dynamic view of the development of models and already touches the fundamental issue of the creation of meaning of data with respect to an internal model.

In order to clarify these issues further, and to demonstrate the conceptual aspects of our framework, we now make the fundamental distinction between **model, observation, and datum**. Here, the middle term, observation, refers to the extraction

[2]We employ the term "model" here as used in the mathematical theory of parametric statistics, that is as an element of some given set of probability distributions – constituting the class –, determined inside that set by the values of finitely many parameters. This is more restricted than the sense in which the word "model" is employed elsewhere in this article.

of the value of a particular quantity or function (in the mathematical sense), like brightness, color, temperature, relative frequency of a particular item,..., from a given datum or data pool. In fact, from the perspective of neurobiology, even these quantities are already indirect constructions on the basis of more basic sensory perceptions, like recordings of photons on the retina. What constitutes an observation, or, better, what the system can observe, depends on its internal structure and its general model of the environment. The result of the observation, however, is determined by the data at hand. This is trivial, but the important point is that the system never has direct access to the data, but has to construct a model of the environment solely on the basis of the values of its observations. This brings us already beyond the framework underlying the theories just described as we now need to consider feedback loops. In fact, our theory needs two feedback loops (see [9] for more details). The internal feedback loop selects the observations on a given data set on the basis of the internal model and its complexity. A prominent neurobiological example is the feedback between the visual cortex and the intermediate relais in the LGN for the processing of visual information (see in particular [16]). The outer feedback loop selects the data subjected to the model's observations. An example is the senso-motoric loop.

In the light of the above discussion, Jaynes' principle then says that for given data, or, more precisely, given the observations made on the data set, the maximum entropy representation should be chosen - with the modification of Gell-Mann and Lloyd, of course, so as not to loose the essential regularities observed in the data. In this way, as much entropy as possible is assigned to the data and as little as possible to the model itself (in the simplest situation, this maximum entropy model then simply is a Gibbs distribution as familiar in statistical mechanics). There is an important point here, already emphasized above and to be addressed again below, namely that the probability distribution whose entropy is to be maximized here is not one that lives on the data, but one that lives inside the system, on the class of models that the system is capable of forming, for example giving the probability of various stored patterns to match the input data. In particular, it will only reflect those aspects of the data that can exhibit some regularities for the system.

In contrast, if the model, or, more precisely, the method for determining it on the basis of the observations made, is given, for example a Gibbs distribution, and if observations can be made on a given data set, then these observations should be selected so as to minimize the resulting entropy of the model, with the purpose of minimizing the uncertainty left about the data. In particular, the observations should be made independent of each other, since an observation whose results can already be predicted from the results of other observations made does not decrease the entropy or reduce the uncertainty (see e.g. [11] for a formal treatment). The inner feedback loop thus tries to reduce complexity. A neural network implementation of this principle is presently under investigation.

Finally, if the data set itself can be varied, i.e. if the system can choose its input, then again the complexity should be maximized, because now we are dealing with external complexity. Thus, data should be chosen for maximal information gain. Jaynes' principle appears here once more, but now as the principle to maximize surprise when exploring the environment. It should be stressed, however, that, as before, the principle cannot be applied unconditionally since the system does not have direct access to the external data, but only through the observations it can make on the basis of its internal model. This should be a fundamental design principle for autonomous robots, but here we cannot explore this issue any further.

An input is meaningful if it leads to some information gain in an internal model. This is a different way to explain the preceding principle of the maximization of external complexity. If, in contrast, the input does not affect the model it is useless for the system and ignored. This important topic will be further explored in [9].

The preceding analysis does not yet apply to distributed systems consisting of individual units that do not have full access to the information contained in the system and that therefore need to optimize their own information flow based on complexity measures when participating in the information processing tasks of the system as a whole. Here, one has to distinguish carefully between the perspective of the system for which the units constitute part of its (internal) environment (following [15]) and the perspective of these units or other subsystems for which the encompassing system is partly external. The system theoretic aspects of this situation are explored in [9]. For an analysis in the context of neural networks, we can for example refer to [14, 1, 10].

4 Complexity of data analysis and internal models, or how to design a neural network

In this section, we shall exemplify the general scheme developed above in a more concrete situation. We shall describe how the capacity of a – very simple – system that analyzes data and builds internal models of them increases through a sequence of steps that increase, reduce or shift complexity to another scale. In order to have a more concrete example available, we shall describe the corresponding steps for the design of a neural network. (The sequence of steps given below does not represent a temporal order, but only some attempt at a conceptual dissection of aspects that are usually intertwined.)

Preparation: *Adapt to some signal space, or specify the task.*

The adaptation here refers to systems that are products of an evolution in some environment, like organisms as representatives of biological species. Such organisms have sensory organs that receive signals from a certain input space, and the corresponding range of signals that the system is capable of receiving is the result of an adaptation. The task specification, in contrast, is the result of a system design, for example for a neural network. That is not all, however, as the system should be adapted or designed so that on one hand, it covers the full range of the signal space, and on the other hand that the signals are received without unnecessary redundancy. In the simplest case, this means that each signal that falls within the range to be covered is received by precisely one sensor. Thus, the receptive fields of the sensors should cover the signal space without overlaps. That latter property can be achieved through the mechanism of **global competition** between the sensors. This means that whichever sensor responds most strongly to some signal (because its receptive field is positioned best for that particular signal) should be able to suppress the responses of all other sensors so that it becomes the one that responds exclusively to that particular signal. This guarantees the useful mathematical property that $\sum_i p_i = 1$ where p_i is defined as the probability that a signal x drawn from the signal probability distribution $p(x)$ falls into the receptive field of the sensor i.

Step 1: *Place receptive fields so as to cover a signal space efficiently.*

We consider a system that is trying to reconstruct the underlying probability distribution $p(x)$ of signals coming from some input space. This is a task addressed by several neural network architectures, like the Kohonen algorithm [12]. The probability distribution p is unknown to the system that only receives random signals following that distribution. The system covers the input space by the receptive fields of its sensors, and on the basis of each signal received, it can perform some adaptation (for example through a stochastic gradient descent). If it simply tries to increase the external complexity of its coverage of the signal space, it will shrink

each receptive field when stimulated by a signal; namely that external complexity should be computed as $-\sum_i p_i \log p_i$ where the index i refers to a sensor and p_i denotes the probability that the receptive field of i is stimulated. It is important to realize at this point that the external complexity is not simply a property or function of the input distribution, but that it also depends on how the system structures the input space, here through the positions and sizes of its receptive fields. Of course, in the situation analyzed here, p_i is unknown to the system. Nevertheless, on average, shrinking receptive fields that receive signals will increase that complexity because that will lead to a more even distribution of the sizes p_i of the receptive fields.

Step 2: *Predict future signals.*

In addition to its attempts at extracting as much information as possible from the signal space through a good covering by receptive fields, the system can also construct some internal estimate q for p that assigns some "subjective" probability q_i to each sensor for receiving a signal. That, in contrast, should be increased whenever that sensor receives a signal so as to decrease the corresponding internal complexity. This is rather obvious, of course, since a sensor that is frequently stimulated should be expected to receive many signals in the future as well. In that manner, the uncertainty about future signals is reduced through an adaptation of the internal model, here represented by q. This analysis can be carried somewhat further (for example by considering the Kullback-Leibler distance $\sum_i p_i \log \frac{p_i}{q_i}$ between p and q), but the preceding should be sufficient for understanding the principle behind this example.

In any case, prediction can only be successful in the presence of **regularities** in the input. We shall need to return to the point how the system can identify and exploit regularities in its input. At this moment, we only observe that the preceding is meaningful when while the input signals are allowed to occur randomly the underlying probability distribution that governs this randomness in turn is not random itself, but remains invariant over the course of the input analysis performed by the system. Of course, one can conceive and analyze situations where that probability distribution is changing as well, but in the end, all such changes must exhibit some regularities, follow some rules. Otherwise, no input analysis is possible.

Step 3: *Improve the arrangement of the receptive fields.*

So far, the preceding would be rather useless as a neural network design. While the sizes of the receptive fields are determined by our entropy principle, their shapes are still arbitrary. Moreover, the locations of the receptive fields do not reflect any relationships between the sensors, that is, sensors that are close to each other in the system need not have similar receptive fields. The first issue could be solved in a certain sense automatically, through a self-organization process based on internal constraints. For example, if the receptive fields compete with each other for coverage of the signal space and try to expand around some center, then as a result of such a competition, we should obtain a rather regular tesselation of the signal space by the receptive fields. This will work the better the fewer degrees of freedom are available for each such field. In other words, strict internal constraints lead to a covering of the input space that is efficient in the sense that the shapes of the receptive fields are simple and efficient for grouping the signals. This is somewhat analogous to one of the main results of statistical learning theory [19]. Concerning the other issue, namely that it is desirable that sensors that are for example neighbors inside the system should also form neighboring receptive fields so that the topology of the input is reflected in the relative positions of the sensors inside the system, this must be stipulated by some additional rule. The Kohonen algorithm [12] represents a good example; here the rule is that whenever a sensor adapts its receptive field on the basis of some signal received, its neighboring sensors also change their fields in the same direction (but typically by a smaller amount). The general principle

here is **local cooperation**. The global competition in the signal space and the local cooperation inside the system do not get into conflict with each other as they operate on different scales.

Step 4: *Integrate the information from different sensors.*

We proceed to the next step where the system increases the external complexity by reducing the complexity of the individual sensors and shifting complexity to a higher level that coordinates and integrates the results of those sensors. Namely, as discussed, so far each sensor has been operating individually, analyzing all aspects or dimensions of the input, and carving out its own receptive field, typically in competition with the other sensors to prevent their corresponding receptive fields from overlapping, thereby avoiding unnecessary redundancy[3]. Now a much more efficient representation of the inputs can be achieved if each sensor specializes on one specific dimension of the signal space. Thus, a sensor does not record anymore all the features of a signal, that is, its position in signal space, but only one **feature** of it, that is, in a geometric terminology, one coordinate value of that position. The position of the signal in the input space then is not anymore determined by an individual sensor, namely the one into whose receptive field that signal falls, but by the combination of the values of several sensors that record complementary features of that signal. This, of course, is a well-known and trivial point, namely that a combinatorial code is vastly superior to one that requires an individual symbol for each item to be encoded. Nevertheless, for implementing that, the system needs some kind of integration mechanism that assigns complementary features to its sensors, instead of letting them directly compete with each other for carving out their own specific receptive fields. It also needs to reconstruct the signal position from the values of the individual coordinates or features. Thus, the system has to develop some complexity at some higher level while at the lower level of the individual elements, the sensors, the complexity is reduced. If the system is large enough, or, more precisely, if it possesses sufficiently many elements, then the overall (internal) complexity should decrease when a more complex coordination or integration mechanism is introduced that permits in turn a simplification of those elements. At the same time, the external complexity is increased.

Step 5: *Patterns as specific signal combinations.*

Through a combinatorial code, the system can now also evaluate a collection of signals simultaneously, or compare the signals in some temporal sequence. Specific signal combinations that either occur particularly frequently, or at least sufficiently well represent certain similarity classes of signal combinations, or that, alternatively, are given to a neural network as training patterns, can then be encoded as internal patterns. The system can then assign new input to one of these stored patterns, based on similarity criteria. The Hopfield network [7] implements this task through the gradient descent of some energy function whose minima correspond to those stored patterns.

However, the performance of such neural networks is rather limited, and in any case, the preceding is somewhat misleading. That one already sees from the purely qualitative principle of **information as a difference that makes a difference**, before any attempts at a quantification. For the analysis of a visual image under varying lighting conditions it is useless to simply record the grey or color levels at individual receptors. The only meaningful aspects are brightness or color differences in a visual scene, between different receptors. In fact, this is what is recorded by the sensory apparatus of animals. This then leads to an automatic correction for uniform background properties. And, of course, this is the basis of all edge and motion detection and the like in our visual system, namely the analysis of a visual

[3]Of course, in order to guarantee the robust functioning of the system, often, some such overlap and redundancy should be helpful, but this is not the issue presently considered.

scene on the basis of spatial or temporal differences between recordings from receptors. Obviously, for going beyond the identification of the most simple structures in a visual scenes, this needs to be iterated. In a neural network (for example [13]), this is then easily implemented by introducing several internal layers each of which detects differences in the output received from the preceding one. While, as we shall see below, this is still far too simple, at this point we are naturally led to

Step 6: *Evaluate higher order properties instead of individual signals.*

The next step that again reduces complexity at some internal level consists in not analyzing individual input signals anymore, but evaluating the average feature values of some collection of them. This reduction of internal complexity in turn enables the system to increase its external complexity by becoming able to analyze input signals that occur simultaneously and to detect patterns in such signal sets. In visual input spaces, such a collection of signals then is an **image**, and the evaluation of a specific feature on such an image is an **observation**.[4] In technical terms, the features under consideration could be certain Fourier modes. These observations, that is, the features to be evaluated, should then be chosen so as to reduce the uncertainty about the image. We can think here of some probability distribution on the space of images. For maximizing external complexity, that distribution is chosen as the one of maximal entropy under the constraints given by the observed feature values, a Gibbs distribution in mathematical terminology. Now for reducing the uncertainty, for decreasing the internal complexity, the features should be chosen in such a manner as to decrease the entropy of that Gibbs distribution, as in [21]. In words, the features selected, that is the observations performed, should be as informative as possible about the image and narrow down the possibilities for the structure of the image – note that the system does not have access to the full image, i.e., to all its details, but has to guess or reconstruct it solely on the basis of the observations performed. Of course, if the system were able to perform arbitrarily many such observations, the uncertainty could be reduced as much as desired, but the number and the types of such observations are constrained by the system's internal structure. This is a good example of the general principle that the system needs to build up a complex structure, here the collection of features it can evaluate, on some large time scale in order to be able to reduce complexity on a short time scale, here the uncertainty about the images constituted by the signals received.

General principle: *Construct a model of the input space.*

This aspect is the most important one underlying all the steps analyzed here. Whatever information the system obtains and extracts from the signals received from the input space, it always has to operate with only partial knowledge. When the sensors arrange their receptive fields as described above, the only information the system then gets from a single signal is in which receptive field it falls. The more precise position within that receptive field is accessible to the system only when it can perform additional, more detailed observations, but the system we consider here is assumed to operate with the precision limit given by the receptive fields of the sensors. Of course, one can then stipulate or construct a system that is capable of more precise observations, but again that system then will have its own limitations. When the system then constructs, or is given, some prototypical patterns of signal combinations, as analyzed above, it will then classify the inputs on the basis of these patterns. These patterns then constitute the internal model (or, to be consistent with the terminology employed above, the model class) within which the system analyzes the input. Thus, it will no longer **predict** the input itself, or **reconstruct** an image on the basis of incomplete information, as in previous steps, but rather **recognize** the input signals as belonging to one of the stored patterns. This leads

[4] Note, however, that the term "observation" has been used in a more general sense above.

to

Step 7: *Relate the input to the internal patterns or categories.*

This means that when input is received, the system does not compute a probability on the space of external images, but on the internal space of patterns. In other words, the system is not interested in the details of the input by themselves, but rather tries to assess to which of its patterns or categories that input belongs. Thus, the relevant mathematical object is a probability distribution on the space of patterns. The entropy of that distribution should be maximized subject to the constraints coming from the observations performed on the input. Thus, constrained by the knowledge obtained from the observations, the entropy for the identity of the pattern corresponding to that input is maximized. When the patterns are faces – face recognition is a standard task for neural networks – the observations are about certain features of the visual input from an actual face, in particular relative sizes and positions of certain parts, and these observations of course make some of the stored patterns more likely to correspond to the input than others. Again, the observations should be selected so as to have maximal distinctive power between different ones of the more probable patterns, that is, they should reduce the uncertainty as much as possible. As explained in the previous section, this requires an **internal feedback loop** where depending on the present state of the probability distribution on the patterns, that is on the internal model, the observations to be performed on the input are chosen among those that the system is capable of.

An important aspect here is that our system, for example a neural network, needs to operate on two distinct time scales, a slow one for constructing the model class, or learning the training patterns, and a fast one for relating the input received on-line to those patterns. Of course, when the system is learning, and even if it is trained through supervised learning, these patterns reflect, or are chosen to reflect, some conspicuous or important aspects of the input. Typically, those aspects can be correlations in longer sequences of inputs, and the system then learns by transforming those into internal associations. A good example is the Hebb-type learning rule [6] commonly employed in neural networks where synapses are strengthened according to the correlation between the activities of the pre- and the postsynaptic neuron. In that way, the network will become able to form an association between those activities, in the sense that an input in the presynaptic neuron can trigger, or at least facilitate, the firing of the postsynaptic one. What is underlying this is another

General principle: *Detect regularities and construct invariants.*

These regularities may occur inside or between images. They allow a compression of the description and internal representation of images or collections of images, as analyzed for example in [4]. Following that reference, those regularities should then constitute the model as opposed to the non-regular aspects of an input that can only be treated as random and are subject to the entropy maximization principle. It is insightful, however, to bring in a complementary aspect. We consider once more the example of face recognition. If the system has stored a particular face as a pattern it then needs to identify visual images as instances of that pattern even when they are rotated or translated in space (so that the object is seen from a perhaps completely different direction or the individual parts are recorded by different sensors than in the original image that may have led to the pattern) or when viewed under different lighting conditions (so that not only brightness, but also the distribution of lit and shaded parts varies considerably, even dramatically so for automatic image recognition programs). In mathematical terminology, following [3], this means that the input differs from the stored pattern by a certain transformation. The system then has two possibilities: Either divide out these transformations and only store the resulting quotient as an abstract invariant pattern and do the same with all images received. Or subject any image received as input to suitable

transformations so as to match the transformed image with the stored pattern. In either case, the system has to implement the corresponding class of transformations, either implicitly, for example through automatic correction for overall brightness as discussed above, or explicitly, for example by internally (mentally) rotating images that are perceived as upside-down. In [3], it has been argued that the aspects of a pattern that are invariant under a suitable class of transformations constitute, or better, lead to a "gestalt". The cognitive system or neural network should then not recognize a particular pattern that corresponds to an external image in a one-to-one manner, but rather only that abstract gestalt. So, in conclusion, we are not simply distinguishing the regular and the random aspects of some input (relative to an internal model), but we demand a specific operation that divides out those aspects that are not essential for the gestalt. In that manner, the two aspects get linked more tightly inasmuch as that operation as an application of our transformations that suppresses the inessential aspects in turn essentially defines the gestalt.

5 Example: Pattern recognition in a neural network

This section is based on a project carried out with Holger Arnold. Here, I describe the principles according to which a neural network can recognize patterns on the basis of the selective evaluation of input features via an internal feedback loop. A detailed presentation of the actual neural network will be given elsewhere.
The main focus of this example will be on Steps 6 and 7 above, but on the background of the other ones. We assume that the network or system has stored or identified a collection of patterns labelled by $i = 1, ..., n$. These patterns might correspond to faces, visual shapes or other geometric objects,...; for thinking about this example, it is probably useful to think about patterns to be recognized in visual scenes. Also, on its input, the system can evaluate certain features $\alpha = 1, ..., m$, like edges, corners, or, better, features of a somewhat higher level, like specific distributions of input pixels on some small subregion of the retina, or relative distances between certain conspicuous points in the scene. It is important for understanding the purpose of the network that we assume to be in a situation where the network is not capable of evaluating all the possible features simultaneously in its input, simply because there are typically far too many possibilities. Rather, the idea is that the network will selectively perform observations, that is, evaluate those features that have the highest potential for discriminating between those patterns that are probable candidates on the basis of the observations already performed. Thus, the basic design principle is a feedback loop between observations that affect the probability distribution in the space of patterns and the selection of further observations on the basis of that probability distribution.
We first need to implement the relationship between patterns and features. This can be done on the basis of supervised learning as is standard in neural networks. So, the observed values x^α of the features induce activations y^i of the patterns:

$$y^i = f(\sum_\alpha w_{i\alpha} x^\alpha) \tag{4}$$

where f might be a sigmoid function $f(s) = \frac{1}{1+e^{-\kappa s}}$ where for our purpose a rather large value of the parameter κ might be best so as to get a sharp threshold later on. Namely, we call a pattern i activated if $y^i > \theta$ where θ is some threshold that we can tune to our convenience, perhaps again by supervised learning. The $w_{i\alpha}$

are weights that can likewise be learned through supervised Hebbian learning. The essential point is that they should be positive, and perhaps large, if feature α occurs in pattern i, and 0 or negative if not.
Conversely, each pattern i then makes a prediction $\xi^\alpha(i)$ about the values of the features occurring in the input; this is nothing but the value or the strength with which feature α occurs in pattern i. Again, this will be the result of supervised learning or direct implementation. The point is that an activated pattern predicts the values of certain observations that could be, but have not yet been performed on the input, and these predictions can thus be checked against observations, in order to confirm or refute the hypothesis that a pattern i is the one present in the input. We now need to make this more precise.
The preceding two operations, namely the activation of patterns on the basis of observations and the prediction of the outcomes of further observations by the activated patterns are fast. In certain cases, a feature observed may alredy determine a unique pattern as the single one whose activation is above threshold. In that case, no feedback is necessary, and the task is solved. In most cases, however, an observed feature will activate several patterns, and the decision between those patterns then has to be achieved on the basis of selected further observations, and this is the task the network has to solve. For that purpose, we now need to compute a probability distribution that assigns probabilities to the activated patterns on the basis of the observations performed so far. According to our general reasoning with entropy maximization, this should be a Gibbs distribution. Thus, the probability of pattern i is given by

$$p(i) = \frac{1}{Z} e^{-\sum_\alpha \lambda_\alpha \xi^\alpha(i)} \tag{5}$$

where the sum extends over all observations α performed so far, with the partition function $Z = \sum_i e^{-\sum_\alpha \lambda_\alpha \xi^\alpha(i)}$ and multipliers λ_α determined on the basis of the observations α performed,

$$E_p(\xi^\alpha) := \sum_i \xi^\alpha(i) \frac{1}{Z} e^{-\sum_\alpha \lambda_\alpha \xi^\alpha(i)} = x^\alpha \tag{6}$$

where x^α is the observed value of feature α.[5] One should note that those multipliers do not always exist so that a slight modification might be required which, however, will not affect the general scheme. For example, if the observations only admit binary values, namely tell whether a feature is present or not, then our Gibbs distribution will simply assign uniform probabilities to all those patterns that match the observation and discard the others (formally, this simply means that the corresponding multiplier is infinite, and we need to look at a higher order expansion). Also, there is a problem if the vector $(x^1, ..., x^M)$ of observed feature values falls outside the convex hull of the vectors predicted by the competing hypotheses $i = 1, ..., n$. This, of course, simply means that no combination of the hypotheses can recover the observed features, and clearly, in such a situation, the system needs to admit or generate new hypotheses. Having discussed this point, we now return to the case where our multipliers exist. The entropy of our Gibbs distribution encodes the uncertainty about which pattern is the correct one on the basis of the observations performed so far. The system now needs to select that observation to be performed next that reduces that uncertainty as much as possible.[6] Roughly, the

[5] We might also stipulate a different rule for the determination of the λ_α, depending on the precise circumstances in which the system is operating. This will not substantially affect the general scheme.

[6] This step is formally similar to the one in [21], but the conceptual aspects are different because the system needs to determine here what observations to perform on the basis of its current internal hypotheses.

idea should be to find a new observation $\star$ with maximal discriminative power, i.e., with maximal

$$\sum_i p_\star(i)|\xi^\star(i) - \bar{x}^\star|^2 \ ; \tag{7}$$

here $p_\star$ is the new probability distribution taking into account feature $\star$, and $\bar{x}^\star$ is the expected value of that feature. That value could be either computed from the probability distribution itself, that is as $\sum_i p_\star(i)\xi^\star(i)$ or be taken as long-term average given before the actual observation of the feature is performed, i.e., before its value is measured. Thus, we stipulate that this expectation value is formed on the basis of the long-term observations of the system, that is, on the basis of all the previous pattern recognition operations carried out by the system in the past. Thus, this value depends on the experience of the system and can be implemented into the system through a process of unsupervised learning. In the context of the analysis of visual scenes, this represents the statistics of natural scenes as learned by the system either in its individual ontogenesis or through an evolutionary process. In any case, this represents the longest time scale involved in our example.
We now need to address the question about how to identify that feature $\star$, or, in a more weaker form as we are argueing that it is impractical for the system to check all possible features, to identify some feature that leads to a reasonably large decay of the uncertainty. Doing so, we shall also see that the above heuristic idea needs a modification, in order to take the fact into account that in general the features are not independent of each other. More precisely, we need to take the correlations of our new feature with the ones already observed into account.
The following approximation argument is useful: We wish to change the probability distribution p given in (5) to

$$p_\star(i) = \frac{1}{Z} e^{-\sum_\alpha \lambda_\alpha \xi^\alpha(i) - \lambda_\star \xi^\star(i)} \tag{8}$$

with a coefficient $\lambda_\star$ to be determined analogously to (6). Instead of doing that, however, we rather study the infinitesimal effect on the entropy of $p_\star$ near $\lambda_\star = 0$. That entropy is given as

$$\begin{aligned} & H(\lambda, \lambda_\star) = -\sum_i p_\star(i) \log p_\star(i) \qquad (9) \\ = \ & \log Z(\lambda, \lambda_\star) \\ & + \sum_i (\lambda_\star \xi^\star(i) \frac{1}{Z} e^{-\sum_\alpha \lambda_\alpha \xi^\alpha(i) - \lambda_\star \xi^\star(i)} + \sum_\alpha \lambda_\alpha \xi^\alpha(i) \frac{1}{Z} e^{-\sum_\alpha \lambda_\alpha \xi^\alpha(i) - \lambda_\star \xi^\star(i)}) \end{aligned}$$

with the partition function

$$Z(\lambda, \lambda_\star) = \sum_i e^{-\sum_\alpha \lambda_\alpha \xi^\alpha(i) - \lambda_\star \xi^\star(i)}. \tag{10}$$

Since the entropy is maximal for $\lambda_\star = 0$ under the constraints (6), the first derivative wrt $\lambda_\star$ vanishes at 0, and the leading term will be given by the second derivative. Moreover, when we compute derivatives, on account of (6), for the last term in (9),

we only need to compute derivatives of the coefficient λ_α. We now compute

$$\begin{aligned}
& \frac{\partial^2}{\partial\lambda_\star^2}H(\lambda,\lambda_\star) \\
= \quad & \frac{-1}{Z(\lambda,\lambda_\star)}\sum_i \xi^\star(i)(\frac{\partial\lambda_\alpha}{\partial\lambda_\star}\xi^\alpha(i)+\xi^\star(i))e^{-\sum_\alpha \lambda_\alpha\xi^\alpha(i)-\lambda_\star\xi^\star(i)} \\
+ \quad & \frac{1}{Z(\lambda,\lambda_\star)^2}(\sum_i(\frac{\partial\lambda_\alpha}{\partial\lambda_\star}\xi^\alpha(i)+\xi^\star(i))e^{-\sum_\alpha \lambda_\alpha\xi^\alpha(i)-\lambda_\star\xi^\star(i)})(\sum_i \xi^\star(i)e^{-\sum_\alpha \lambda_\alpha\xi^\alpha(i)-\lambda_\star\xi^\star(i)}) \\
= \quad & E_p(\xi^\star)E_p(\xi^\star-\frac{\partial\lambda}{\partial\lambda_\star}\xi)-E_p(\xi^\star(\xi^\star-\frac{\partial\lambda}{\partial\lambda_\star}\xi)) \qquad (11)
\end{aligned}$$

that is, the (negative of the) variance w.r.t. our probability distribution p of the new feature $\xi^\star$ corrected for its dependence on the old ones. That dependence is easily computed from (6). Namely, if we write those constraints in the form

$$F^\alpha := \sum_i \xi^\alpha(i)\frac{1}{Z}e^{-\sum_\alpha \lambda_\alpha\xi^\alpha(i)} = x^\alpha \qquad (12)$$

we have

$$\frac{\partial\lambda}{\partial\lambda_\star} = -(\frac{\partial F}{\partial\lambda})^{-1}\frac{\partial F}{\partial\lambda_\star} \qquad (13)$$

where $\frac{\partial F}{\partial\lambda}$ stands for the matrix with entries $\frac{\partial F^\alpha}{\partial\lambda_\beta}$ and $\frac{\partial F}{\partial\lambda_\star}$ for the vector $\frac{\partial F^\alpha}{\partial\lambda_\star}$. Those entries are computed from (6) as

$$\frac{\partial F^\alpha}{\partial\lambda_\beta} = -\sum_i \xi^\alpha\xi^\beta e^{-\sum_\alpha \lambda_\alpha\xi^\alpha(i)-\lambda_\star\xi^\star(i)} = -E_p(\xi^\alpha\xi^\beta) \qquad (14)$$

$$\frac{\partial F^\alpha}{\partial\lambda_\star} = -\sum_i \xi^\alpha\xi^\star e^{-\sum_\alpha \lambda_\alpha\xi^\alpha(i)-\lambda_\star\xi^\star(i)} = -E_p(\xi^\alpha\xi^\star). \qquad (15)$$

Since we are at a maximum, the second derivatives of H have to be non-positive, and this checks from (11), as they are the negatives of variances. In particular, H is locally decreased most if we select a new feature for which that variance is maximal. The important point is of course that this variance of the feature to be evaluated depends on the patterns currently under consideration because it is computed w.r.t. the probability distribution p that is determined by those patterns. In particular, we are not attempting to maximize the absolute information gain resulting from evaluating a new feature, but rather seek a feature that can discriminate best between the currently active hypotheses. Thus, the variance occurring here is not the one for the probability distribution of the feature, but rather the one for the prediction of the feature by the hypotheses.
In practice, the network will not be able to check all possible features for identifying the one with maximal variance, but rather a collection of patterns must activate some subset of features among which then the maximization can be carried. Again, that relationship must either be based on the hypotheses presently active, that is compute some $g(\sum_i v_{\alpha i}\xi^\alpha(i))$ for some bounded increasing function g (a sigmoid, for example) and some coefficients $v_{\alpha i}$ determined by Hebbian learning, and then select the class of those features for which the value of that expression is above some threshold, or alternatively on long-term experience, that is on unsupervised learning as already explained above.

Acknowledgement: I am grateful to Olaf Breidbach for discussions and joint intellectual enterprises that addressed many of the topics treated in this paper and profoundly shaped my approach presented here. In addition, I should like to thank him for his insightful comments on the present article.

References

[1] N.Ay, Information geometry on complexity and stochastic interaction, Preprint MPIMIS, Leipzig, 2001

[2] O.Breidbach, K.Holthausen, J.Jost, Interne Repräsentationen – Über die "Welt"generierungseigenschaften des Nervengewebes. Prolegomena zu einer Neurosemantik, in: A.Ziemke, O.Breidbach (eds.), Repräsentationismus – Was sonst?, Vieweg, Braunschweig, Wiesbaden, 1996

[3] O.Breidbach, J.Jost, Zum Begriff der Gestalt, in: Jahrbuch für Geschichte und Theorie der Biologie 10, Berlin VWB, 2003

[4] M.Gell-Mann, S.Lloyd, Information measures, effective complexity, and total information, Complexity 2, 44-52(1996)

[5] S.J.Gould, The structure of evolutionary theory, Harvard Univ.Press, 2002

[6] D.Hebb, The organization of behavior, Wiley, New York, 1949

[7] J.Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sc. USA 79, 2554-2558 (1982).

[8] E.Jaynes, Information theory and statistical mechanics, Phys.Rev.106, 620-630 (1957)

[9] J.Jost, Complex systems and cognitive structures, Monograph, to appear

[10] J.Jost, Neural networks, Monograph, to appear

[11] J.Jost, Information geometry, Lecture Notes, Leipzig, 2002

[12] T.Kohonen, Self-organizing maps, Springer, 1995

[13] R.Linsker, From basic network principles to neural architecture, Proc.Nat.Acad.Sciences USA 83, 7508-7512, 8390-8394, 8779-8783 (1986)

[14] R.Linsker, Self-organization in a perceptual network, Computer 21(3), 105-117 (1988)

[15] N.Luhmann, Soziale Systeme, Suhrkamp, Frankfurt/M., 1984, ^{7}1999

[16] D.Mumford, On the computational architecture of the neocortex. I: The rôle of the thalamo-cortical loop, Biol.Cybern. 65, 135-145 (1991)

[17] J.Rissanen, Stochastic complexity in statistical inquiry, World Scientific, Singapore, 1989

[18] C.Shalizi, J.Crutchfield, Computational mechanics: Pattern and prediction, structure and simplicity, J.Stat.Phys.104, 819-881 (2001)

[19] V.N.Vapnik, The nature of statistical learning theory, Springer, New York, 1995

[20] V.N.Vapnik, Statistical learning theory, J.Wiley, New York, 1998

[21] S.C.Zhu, Y.N.Wu, and D.Mumford, Minimax entropy principle and its application to texture modeling, Neural Comp.9, 1627-1660 (1997)