GENERALIZATION BY COMPUTATION THROUGH MEMORY

Petro Gopych

Abstract: Usually, generalization is considered as a function of learning from a set of examples. In present work on the basis of recent neural network assembly memory model (NNAMM), a biologically plausible 'grandmother' model for vision, where each separate memory unit itself can generalize, has been proposed. For such a generalization by computation through memory, analytical formulae and numerical procedure are found to calculate exactly the perfectly learned memory unit's generalization ability. The model's memory has complex hierarchical structure, can be learned from one example by a one-step process, and may be considered as a semi-representational one. A simple binary neural network for bell-shaped tuning is described.

Keywords: generalization, 'grandmother' model for vision, neural network assembly memory model, one-step learning, learning from one example, neuron receptive field, bell-shaped tuning, semi-representation.

ACM Classification Keywords: Memory structures (B.3), associative memories; reliability, testing, and fault-tolerance (B.8.1); learning (I.2.6), connectionism and neural nets; vision and scene understanding (I.2.10), representations, data structures, and transforms; image representation (I.4.10), hierarchical.

1. Introduction

We know from our everyday experience that even under difficult observation conditions, the recognition of complex visual objects occurs in practice immediately, in an on-line regime. The ability to recognize visual objects regardless of the side of view, their illumination, occlusion, or particular distortion is called generalization ability; up to present its brain mechanisms remain unclear [1].

In real life, any two successive images, although they correspond to the same particular object, cannot coincide literally, point-by-point. As a result the amount of all possible images of all possible objects to be recognized is extremely large and, consequently, they the all cannot be stored in human memory even of very large but limited capacity. To overcome this difficult problem, it is supposed that it is enough to remember labels of only some typical images (examples) and to learn the common memory/generalization system to predict to a huge amount of unknown images, not storing in memory. Such a statement of the problem implies that for a given object each its particular image can continuously be transformed, possibly not too sharp, into any other its image through an infinite continuous series of its intermediate images.

The classic learning theory [2] gives a formal definition of generalization and rules to ensure it. For the generalization purposes, the leaning theory provides the best possible functional relationship between an input image, x, and its label, y, by learning from a set of n examples, x_i, y_i. This problem is similar to the problem of fitting a continuous smooth function of some arguments to measurement data, x_i, y_i, or, in other words, the ability of estimating correctly the values of this function in points where data are not available (i.e. it is assumed implicitly that sets of unknown images and their labels are continuous).

Within this approach, for a given training set (x_i, y_i; i = 1, 2, …, n), the empirical risk minimization (ERM) learning algorithm can find the estimated interpolating function f which minimizes empirical error — the quantity defining through a loss function the quality of fitting f to the training set of examples. To provide the good generalization, f should also guarantee the minimization of predictive error — the quantity defining through the same loss function the quality of fitting f to new samples — in such a way that the difference between empirical and predictive errors is zero in probability as $n \to \infty$. It may be possible if f, chosen from a given functional hypothesis space, is simple enough. In general case (for finite sets of examples and complex hypothesis spaces) by using the classic ERM learning, the solution needed it is not always possible to find [2,3]. For this reason a new paradigm of learning

was proposed which provides 'conditions for generalization in terms of precise stability property of the learning process: when training set is perturbed by deleting one example, the learned hypothesis does not change much' [3]. This stability criterion means that if after deleting any *i*th training sample (example) from any large training set of samples (examples) almost always the learned interpolating function *f* changes in small, then it generalizes well. Formally it is demanded a cross-validation leave-one-out stability with stability of empirical and expected errors: for any *i*, for sets of training samples *S* and S^i (S^i is the set *S* with the deleted item *i*), supremums of differences between corresponding loss functions, corresponding empirical errors, and corresponding expected errors equal zero in probability as $n \to \infty$. Such stability ensures that good (predictive) generalization functions may be found by not only the ERM process but also other learning algorithms [3] and, consequently, this method of generalization is suitable (see ref. 3 and references therein) for solving those practical problems where classic ERM learning [2] does not work. But for both cases (minimization of empirical error within a given hypothesis space or stabilization of the learning process), the important challenge of the finiteness of training sets remains unsolved because all above results are valid only asymptotically ($n \to \infty$), i.e. for a rather large amount of training examples.

The approach based on learning from a set of examples is not the only possible. Indeed, it is naturally to assume that in human visual system the real world is actually represented as a set/series of 'frames,' discrete and only perceived continuously (as in a movie). If it is, then the amount of information needed to be maintained reduces crucially and for this reason memory system, serving vision and dealing with a finite set of discrete images, may computationally become simpler. This work follows such an alternative approach.

2. Generalization by Interpolating among Examples

Within the classic learning theory [2], generalization by interpolating among examples supports a popular neural network (NN) architecture that combines the activity of some hidden broadly tuned 'units' (local NN circuits), each of which is learned to respond to one of the training examples optimally and to a variety of other images at sub-maximal firing rates. This idea is consistent with the fact that bell-shaped tuning is common among neurons in visual cortex and that in infero-temporal cortex, IT, there exist neurons tuned to different complex objects or their parts.

Mathematically, using the method of regularization, the learning from examples may be formulated as measurement data approximation by a smooth function, $f(x) = \sum w_i k(x,x_i)$, which minimizes the empirical error (error of training); here $f(x)$ is a weighted sum (weights w_i) of basis functions, $k(x,x_i)$, depending on a new (unknown) image, x. For example, function $k(x,x_i)$ may be a radial Gaussian centered on x_i, representing the *i*th neuron's receptive field, and responding optimally to (memorizing) x_i (that is so called radial basic function approach, RBF). The width of $k(x,x_i)$ defines also the unit's selectivity as a memory device: for broadly tuned k, its selectivity is poor but a linear combination of such functions provides a good generalization ability; for sharply tuned k (e.g., a delta function or very narrow Gaussian), its selectivity is perfect but such a $k(x,x_i)$ cannot be used for generalization. In $f(x)$, functions $k(x,x_i)$ may be learned from their inputs, x_i, in a passive regime (without the feedback) while weights, w_i, depend also on outputs, y_i, and demand more complicate iterative learning from examples, x_i, y_i. That is, the learning process splits into two parts: learning the basis functions (memory units and, simultaneously, neuron receptive fields) and learning the weights of the whole network (learning to generalize using already learned memory units). The algorithm described can implement a feedforward NN with one hidden layer containing as many units as training examples; parameters w_i are interpreted as synaptic weights between corresponding units and the output, $f(x)$ [1].

In this case [1] the ability to generalize is traditionally [2,3] grounded on the use of many training examples and is paid by the poor selectivity of all memory units (a large value of the regularization parameter), the assumption of low biological plausibility.

3. 'Grandmother' Model for Vision

In the classic 'grandmother' theory for vision, an image recognition happens when the combination of all its features precisely coincides with such a combination associated with particular grandmother neuron, i.e. in this case between the input image and different memory records a direct literally comparison is needed. The lack of generalization is the basic problem of such a model. To solve it, the model was essentially extended: it is supposed that 'generalization emerges from linear combinations of neurons tuned to an optimal stimulus' [1] (see also Section 2). We propose another extension solving the generalization problem under assumption that each memory unit itself can generalize.

As Figure 1 demonstrates, in our model all sensory-specific stages of input visual data processing coincide completely with those that Poggio & Bizzi [1] discussed and, consequently, in this part both models are biologically equally plausible. In particular, in the model proposed AIT neurons (open circles), tuned to respond to complex visual images, are also used although in present work the architecture and operation of local NN circuits, employed for tuning, are quite different (see Section 5). But the main distinction between our model (Figure 1) and Poggio & Bizzi model (Figure 2 in [1]) consists in the structure of their sensory-independent parts: in Figure 1, it is grounded on the neural network assembly memory model, NNAMM, discussed in Section 4 [4].

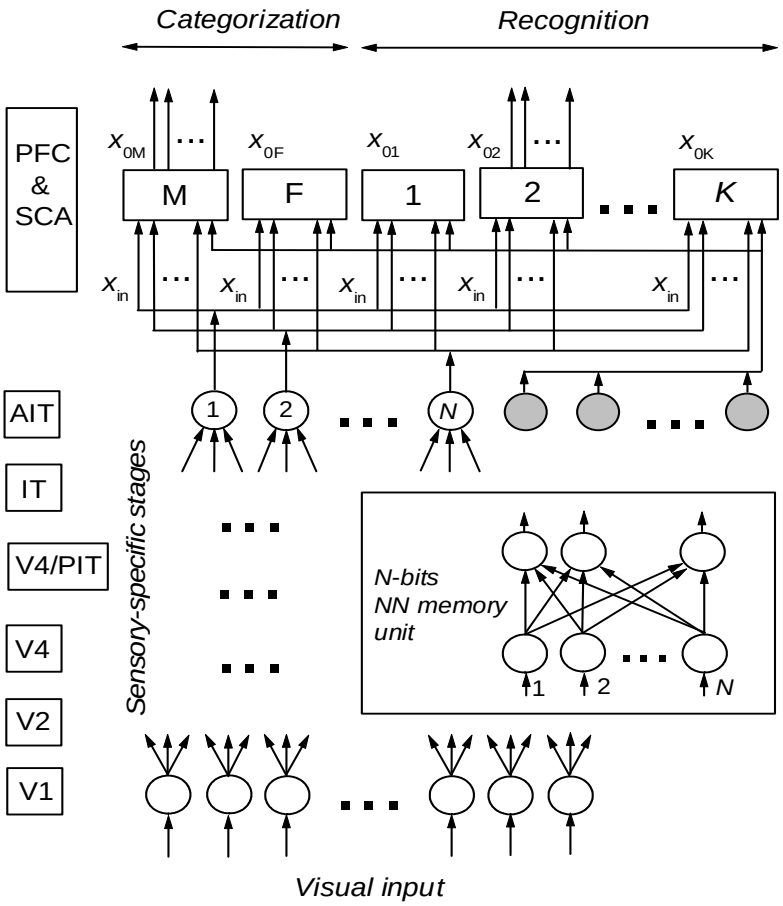

Figure 1. An oversimplified scheme of a 'grandmother' model for vision based on the NNAMM [4]. At the bottom, in V1, cells have small receptive fields and respond preferably to oriented bars; along the ventral visual stream they increase gradually their receptive fields and complexity of their preferable images and at the top, in AIT, neurons respond optimally already to rather complex objects. AIT neurons 1,…,*N* (open circles) could code the image of current interest, e.g. a face, as a binary (±1) feature vector x_{in}; other similar neurons (filled circles) can

code (respond optimally to) other complex objects. Boxes M and F correspond to assembly memory units, AMUs (Figure 2), storing reference codes (representations) of the 'ideal' (reference) male, x_{0M}, and female, x_{0F}, faces; boxes 1,…,K denote AMUs storing the codes (representations) $x_{01},\ldots,x_{0K}$ which represent known (previously encountered) faces 1,...,K, regardless of their categorization. The case is presented where a current face feature code x_{in}, extracted from the current visual input, is recognized as the face number 2 and categorized as a male face (x_{in} initiates the correct retrieval of memory traces x_{0M} and x_{02} designated as output arrows from boxes M and 2, respectively). In the insertion, a feedforward NN, related to particular AMU_i and storing the code (representation) x_{0i}, is shown (see also box 2 in Figure 2; AIT neurons 1,...,N may be equivalent to exit-layer neurons of such an NN). If x_{in} does not correspond to one of the codes (representations) $x_{01},\ldots,x_{0K}$ but is recognized as x_{0M} or x_{0F} then it can be remembered in the (K + 1)th empty AMU, AMU_{K+1}, which is not shown. V1, primary visual cortex; V2 and V4, extrastriate visual areas; IT, infero-temporal cortex; AIT, anterior IT; PIT, posterior IT; PFC, prefrontal cortex; SCA, subcortical areas (e.g., as it is shown in Section 4.2, hippocampus).

We suppose that visual memory is constructed as a set of the NNAMM's assembly memory units, AMUs (Figure 2 in Section 4.2), interconnected between each other and storing only one memory trace per one AMU. Memory traces are N-dimensional binary (±1) vectors represented particular images (e.g., known faces, $x_{01},\ldots,x_{0K}$) or categories of such images (e.g., male, x_{0M}, and female, x_{0F}, faces). Tuned AIT neurons 1,...,N (open circles) convey the code x_{in}, extracted from the current visual input at sensory-specific stages of data processing and representing the current face, to all AMUs devoted to vision. Similar codes of other images, available in the current visual input, are also extracted and other tuned neurons (filled circles) convey them to all AMUs devoted to vision. But by means of a spatio-temporal synchrony mechanism and anatomically in part, the AMUs shown select only the code of their interest, x_{in}; other similar codes may be the codes of interest for other AMUs, which are not shown.

Even if the analyzed visual scene is stable, the current (at the moment t_0) visual input may slightly be changed, for example, due to a saccadic eye movement. In such a case, at the next moment, t_1, the hierarchy of tuned local NN circuits, constituting the pathways of sensory-specific stages of initial visual signal processing (see Figure 1 and Section 5), produces, most probably, binary feature vector $x_{in}(t_1)$ which is equal to previous one, $x_{in}(t_0)$. As $x_{in}(t_0) = x_{in}(t_1)$, at sensory-independent but memory-specific stage of data processing, $x_{in}(t_1)$ initiates the recall/retrieval of memory patterns, the same as $x_{in}(t_0)$ initiates, e.g., x_{0M} and x_{02} (see Figure 1). That is, in numerous slightly (even continually) changed visual inputs, it takes place the recall/recognition of the same image of interest (e.g., a face) of the same category (e.g., male faces) whose binary representations in visual memory are vectors x_{0M} and x_{02}, respectively.

If in two successive visual scenes the difference between images of interest is not very small and not very large simultaneously then visual pathways mentioned at moments t_0 and t_1 may produce binary feature vectors $x_{in}(t_0)$ and $x_{in}(t_1)$ which are different but related to the same finite set of them characterized by the same value of the damage degree, d, or intensity of cue, q (see Section 4.1). In such a case, in spite of the fact that $x_{in}(t_0) \neq x_{in}(t_1)$, at sensory-independent but memory-specific stage of visual data processing, $x_{in}(t_1)$ initiates the recall/retrieval of memory patterns x_{0M} and x_{02}, the same as $x_{in}(t_0)$ initiates, with the same probabilities defined by Equation 5 or 6. That is, in numerous visual inputs containing rather changed or damaged images of interest, it also takes place their equally successful categorization and recall/recognition.

If the difference between images of interest in successive visual scenes is large then sensory-specific visual pathways may produce feature vectors $x_{in}(t_0)$ and $x_{in}(t_1)$ which are related to different sets of them characterized by different values of d (or q). In such a case, $x_{in}(t_1)$ also initiates successful recall/retrieval of patterns x_{0M} and x_{02}, the same as $x_{in}(t_0)$ initiates, but already with other probabilities. That is, even in numerous visual inputs containing essentially changed or damaged images of interest, their successful categorization and recall/recognition takes also place.

Consequently, the model for vision proposed provides successful categorization and recall/recognition of numerous changed, in particular essentially changed, versions of the same visual image employing its single

binary representation, x_0, stored in visual memory. In other words, it implements the idea of generalization in its conventional form (Section 1) but in a new way — by generalization through a single NNAMM memory unit, AMU, storing only *one* binary representation, x_0, of all possible versions of the image of interest, which may differ from each other in small as well as in large.

As one can see, a learned AMU itself ensures generalization (recall/generalization) of only its binary inputs, x_{in}, (Section 4.1) with the probability may be calculated exactly (Equation 5 or 6). To find the probability of generalization (recall/recognition) of any initial half-tone visual image completely, the probability of producing these binary feature vectors, x_{in}, is also required. For solving the latter problem, we should specify beforehand the architecture of sensory-specific visual pathways (Figure 1) as a hierarchy of tuned local NN circuits, extracting step-by-step from the initial image its more and more general features/properties (see also Section 5). When this hierarchical architecture (specific, in general, for each category of images of interest) will completely be constructed, its performance may be found as performance of a device built in a known manner from building blocks with known properties. Hence, the content of particular visual memory is *jointly* defined by the content of corresponding AMU (a rather short binary vector x_0) *and* complete hierarchical architecture of tuned local NN circuits, which perform a sensory-specific visual data processing and extract from complex initial input the feature vector x_{in} that, in turn, initiates the retrieval of x_0. Very early (in the retina) stages of this many-stage process play a special role because here the binarization of initial half-tone images is carried out and the quality of binarization exerts an essential influence on the quality of the final representation of images in the entire visual system. As it was empirically demonstrated [5], the binarization required may be performed optimally, without the loss of information essential for the following binary data processing according to optimal binary algorithms described in Section 4.1.

Within the model proposed, representation of an image in visual system may be considered as a complex dynamic process consisting of three successive stages: i) binarization (in the retina) of an initial half-tone image; ii) allocating essential features of the image binarized and production of its rather short binary representation, x_0 (in a hierarchical architecture of local functionally similar tuned NN circuits which constitute visual data processing pathways); iii) storing x_0 (in visual memory) and its multi-purpose use for planning and maintaining different possible mental and behavioral operations. Owing to this three-level structure of data processing and due to the data graduate compression, the code (representation) x_0 stored in visual memory can along not specify completely its corresponding visual (perceptual) input and the same x_0, but in memory devoted to another modality, could in general code (represent) a quite different object or idea, for example, the odour of a perfume (if x_0 is stored in olfactory memory). For the same reason, each visual (perceptual) memory (each AMU) should intimately be related to corresponding final stages of their sensory-specific pathways, strictly anatomically defined. Consequently, according to the model, in the brain should exist areas preferably devoted and responded to different specific memories and to specific categories of these memories. This theoretical prediction is completely consistent with the available anatomical findings demonstrating that is actually the fact. For example, the fusiform face area (a part of fusiform gyrus located in brain temporal lobe) is devoted to face perception in humans [6,7]. Brain damages to or near to the fusiform face area lead to specific mental disorder — prosopagnosia, an inability to perceive faces while all other mental properties remain intact [8]. Some persons suffered of prosopagnosia retain, in spite of that, the ability of face categorization (e.g., they differ males from females or olds from youngs) and can correctly identify faces of familiar persons unconsciously (e.g., their galvanic skin response rises when they hear the correct name). These neuropsychology findings are also consistent with the model proposed which predicts, in particular, that brain areas devoted to face recognition and face categorization should anatomically be segregated in part, that face perception is a many-stage process in a hierarchical brain structure (visual pathways in Figure 1) with anatomically segregated levels (areas) and damages to higher levels (areas) of visual pathways do not hinder the normal functioning of their lower levels (areas).

As it has been pointed out, an initial half-tone visual input can be binarized optimally, without the loss of information essential for the further processing of obtained binary data [5]. Perfectly learned local NN units (Section 5) and AMUs (Section 4.2) processing this data also operate over their binary inputs optimally (in the sense of pattern recognition quality, Section 4.1). Consequently, if the chain a binarization device (retina)—feature-extractive hierarchical architecture of tuned local NN units (sensory-specific visual pathways)—AMUs (visual memory, as in Figure 1) is constructed (hard-wired) in an optimal manner (that is the problem of animal evolution or an engineer who builds a machine, data processing system or device) then its operation performance may also be optimal. In sum: within the model proposed, for each specific category of images, the entire visual data processing system/algorithm may surely be optimal but the architecture, needed to implement this theoretical possibility as an algorithm or device, is not specified so far completely.

For the construction of optimal data processing architecture providing generalization through memory of visual images of different categories, only its building blocks (learned AMUs and tuned local NN units) having optimal operation performance are now available. But that is enough to conclude that such a future system/algorithm cannot solve the inverse problem: reconstruction of the initial visual input when its binary representation (x_0 stored in an AMU), properties of the retina, tuned local NN units, AMU and their connections are known. The reason is in the irreversibility of these all components constituting jointly the hierarchical architecture required. For example, a learned two-layer NN, the heart of all AMUs and tuned local NN units, is served by a finite set of its input binary vectors x_{in} and has only the single output, x_0, providing the solution and specified strictly by the additional learned 'grandmother' neuron (Section 4.1). From these follows directly the convergence of learned AMUs and tuned local NN circuits (by definition, all their inputs, x_{in}, lead to the single solution, x_0, stored in corresponding NN and its learned 'grandmother' neuron) and, simultaneously, their irreversibility (by definition, their the given solution, x_0, cannot exactly specify that one of many particular NN inputs, x_{in}, which has earlier initiated the recall/retrieval of x_0). It is clear that the reversible processing system (computer algorithm) required to solve an inverse problem cannot be built from irreversible components.

There exists two opposite viewpoints of the nature of human memory. On the one hand, traditionally (e.g., [1]), it is supposed that objects, actions, etc are stored in memory as their representations, i.e., as coded messages may be used, if necessary, as instructions governing the learned mental or physical behavior. On the other hand, it was introduced the so called nonrepresentational memory, an ability of dynamic system 'to repeat or suppress a mental or physical act' or an 'ordered sequence of brain activities … that, in time, leads to a particular neuron output.' 'In this view, a memory is dynamically *generated* from the activity of selected subsets of circuits' [9]. Within our BSDT/NNAMM approach, a memory is defined as consisting of two closely related parts: the representational code x_0, stored in an AMU related to particular visual (perceptual) memory, and the stream of neuron activity, dynamically generated in visual pathways and directed from the retina to the AMU mentioned. That is, our memory model has some properties of representational as well as nonrepresentational memories and may be qualified as a *semi-representational* one.

In contrast to Section 2, the NNAMM's memory unit itself provides perfect memory trace selectivity as well as generalization through memory. Because each AMU contains a 'grandmother' neuron (for details see Sections 4 and 5), we can consider the model for vision introduced as a 'grandmother' one.

4. NNAMM as a Memory Model Used

P.M.Gopych has proposed a ternary/binary data coding and demonstrated [10] that corresponding NN decoding algorithm (inspired by J.J.Hopfield [11]) is simultaneously the retrieval mechanism for an NN memory. As NNs used for data decoding and memory storing/retrieval are the same (see insertion in Figure 1), they have also common data-decoding/memory-retrieval performance (Section 4.3). Later this data coding/decoding approach was developed into the binary signal detection theory (BSDT) [12] and neural network assembly memory model (NNAMM) [4] closely interrelated in their roots and providing the best quality performance. The price paid for the

NNAMM optimality is the fact that it places each memory trace in its own AMU (an estimation of human memory capacity, though it is possibly too optimistic — 10^{8432} bits [13], supports this assumption).

4.1 Formal Background

Let us denote a vector with components x^i (i = 1,…,N), whose magnitudes are ±1, as x. It can carry N bits of information and its dimension N is the size of a local receptive field for the NN/convolutional feature discrimination algorithm [5] or the size of an NN memory unit discussed below. If x represents information stored or that should be stored in the NN then we term it reference vector x_0. If the signs of all components of x are randomly chosen with uniform probability, ½, then that is random vector x_r or binary noise. We define also a damaged reference vector $x(d)$ with components

$$x_i(d) = \begin{cases} x_0^i, if\, u_i = 0, \\ x_r^i, if\, u_i = 1 \end{cases} \qquad d = \sum u_i / N, \quad i = 1,\ldots,N, \tag{1}$$

where marks u_i take magnitudes 0 or 1 and may randomly be chosen with uniform probability, ½; d is a damage degree of x_0. If the number of marks u_i = 1 is m then the fraction of noise components in $x(d)$ is $d = m/N$; $0 \le d \le 1$, $x(0) = x_0$ and $x(1) = x_r$. The fraction of intact components of x_0 in $x(d)$, $q = 1 - d$, is *intensity of cue* or cue index; $0 \le q \le 1$, $q + d = 1$, d and q are proper fractions. For a given $d = m/N$, the number of different vectors $x(d)$ is $2^m C^N_m$, $C^N_m = N!/(N - m)!/m!$; for d ranged $0 \le d \le 1$, complete finite set of all vectors $x(d)$ consists of $\sum 2^m C^N_m = 3^N$ elements (m = 0,1,…,N).

For decoding the data coded as described, we use a two-layer NN with N McCalloch-Pitts model neurons in its entrance and exit layers; these neurons are linked as in the insertion of Figure 1, 'all-entrance-layer-neurons-to-all-exit-layer-neurons.'

For a learned NN, its synapse matrix elements, w_{ij}, are

$$w_{ij} = \xi x_0^i x_0^j \tag{2}$$

where $\xi > 0$ is a parameter (below ξ = 1); x^i_0 and x^j_0 are the ith and the jth components of x_0, respectively. Hence, the matrix w is defined by vector x_0 and Equation 2 unambiguously. We refer to w as the perfectly learned NN and it is of crucial importance that it remembers only *one* pattern x_0 (the available possibility of storing other memories in the same NN is intentionally disregarded). It is also assumed that the NN's input vector x_{in} is decoded (reference or state vector x_0 is extracted) successfully if the learned NN transforms an x_{in} into the output vector $x_{out} = x_0$ (an additional *'grandmother'* neuron checks this fact; see also Sections 3, 4.2, 5, and ref. 14).

The transformation algorithm is the following. For the jth exit-layer neuron, its input signal, h_j, is

$$h_j = \sum w_{ij} v_i, \quad i = 1,\ldots,N \tag{3}$$

where v_i is an output signal of the ith entrance-layer neuron. The jth exit-layer neuron's output, x^j_{out}, is calculated by a rectangular response function with the neuron's triggering threshold $\theta \ge 0$ (for the case $\theta < 0$, see ref. 14):

$$x_{out}^j = \begin{cases} +1, & if \quad h_j > \theta \\ -1, & if \quad h_j \le \theta \end{cases} \tag{4}$$

where for $h_j = \theta$ the value $v_j = -1$ is arbitrary assigned.

Since entrance-layer neurons of the NN used play only the role of input fan-outs, which convey their inputs to all exit-layer neurons, in Equation 3 $v_i = x^i_{in}$. Of this fact and Equations 3 and 4 for the jth exit layer neuron we have: $h_j = \sum w_{ij} x^i_{in} = x^j_0 \sum x^i_0 x^i_{in} = x^j_0 Q$ where $Q = \sum x^i_0 x^i_{in}$ is a convolution of x_0 and x_{in}. The substitution of $h_j = x^j_0 Q$ into

Equation 4 gives that $x_{out} = x_0$ and an input vector x_{in} is decoded (reference vector x_0 is extracted) successfully if $Q > \theta$. Since for each x_{in} exists such a vector $x(d)$ that $x_{in} = x(d)$, inequality $Q > \theta$ can also be written as a function of $d = m/N$: $Q(d) = \sum x^i_0 x_i(d) > \theta$ (i = 1,2,…,N) where θ is the threshold of Q and, simultaneously, the neuron's triggering threshold. Hence, for perfectly learned intact NNs, NN and convolutional decoding algorithms are functionally equivalent.

Since $Q > \theta$ and $D = (N - Q)/2$, where D is Hamming distance between x_0 and specific $x(d)$, the inequality $D < (N - \theta)/2$ is also valid and NN, convolutional, and Hamming distance decoding algorithms mentioned are equivalent. As Hamming decoding algorithm is the best (optimal) in the sense of statistical pattern recognition quality (i.e., there is no other algorithm outperforming it), NN and convolutional algorithms are also optimal (the best). Moreover, similar decoding algorithms based on locally damaged NNs may also be optimal [4,15] (see example in Table 1 of Section 6).

4.2 AMU's Architecture

We saw that a two-layer NN (as in the insertion of Figure 1) can be used for optimal one-trace memory storing/retrieval. But for such an NN, one separate randomly chosen vector $x_{in} = x(d)$ can initiate successful retrieval only randomly. Thus, to implement the model's possibilities completely, the retrieval should be initiated by a *series* of different vectors x_{in} and it happens when one of the next x_{in} leads suddenly to the emergence of the output $x_{out} = x_0$. For this reason the minimal architecture, needed to provide optimal memory trace retrieval from the learned NN (box 2), should be as in Figure 2. Because the retrieval is initiated by vectors $x(d)$, which constitute complete finite set of binary representations of those images (or 'frames') that were mentioned in the last paragraph of Section 1, such an architecture provides also the optimal generalization by computation through memory. The internal loop, 1-2-3-4-1, ensures the generation of different (e.g., random) vectors $x_{in} = x(d)$ with a given value of d while the external loop, 1-2-3-4-5-6-1, maintains the internal one.

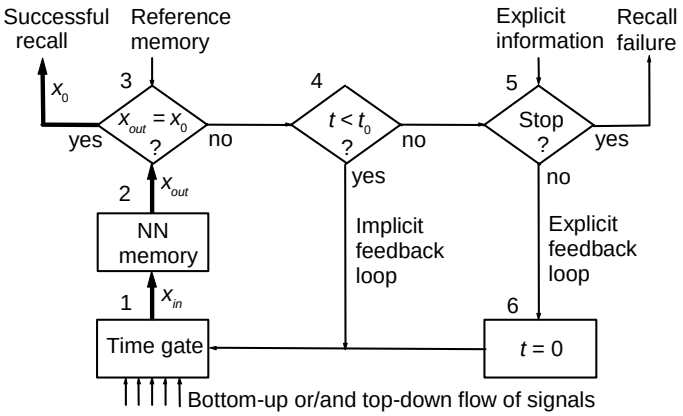

Figure 2. The flow chart (the architecture) of an assembly memory unit, AMU, and its short-distance environment adopted from [4]. The structure of the NN memory unit (box 2) specifies the insertion in Figure 1. Pathways and connections are shown in thick and thin arrows, respectively.

Within the NNAMM, the whole memory is a very large set of interconnected AMUs of rather small capacity (N ~ 100 or less), organized hierarchically. An AMU (Figure 2) consists of boxes 1, 2 and 6, diamonds 3, 4 and 5; their internal and external pathways and connections are designed to propagate synchronized groups of signals [vectors $x(d)$] and asynchronous control information, respectively. AMUs implement directly the BSDT for solving the problem of optimal generalization and memory storing/retrieval.

Box 1 (a kind of N-channel time gate) transforms initial ternary (0,±1) sparsely coded very-high-dimensional vectors into binary (±1) densely coded and rather low-dimensional ones. Here from the flood of asynchronous

input spikes, a synchronized pattern of signals in the form of N-dimensional feature vector $x_{in} = x(d)$ is extracted by a dynamical spatiotemporal synchrony mechanism. Box 2 is an NN learned according to Equation 2 (or Equation 7 from Section 4.4) where each input, x_{in}, is transformed into its corresponding output, x_{out}. Diamond 3 (a kind of comparator or familiarity/novelty detector) performs the comparison of x_{out}, just now emerged, with the reference vector (trace) x_0 from *reference memory* (RM, see below). If $x_{out} = x_0$, then the retrieval is successful and it is finished. In the opposite case, if current time of retrieval, t, is less than its given maximal value, t_0, (this fact is checked in diamond 4) then the loop 1-2-3-4-1 is activated, retrieval starts again from box 1, and so forth. If t_0, a parameter of time dependent neurons, was found as insufficient to retrieve x_0 then diamond 5 examines whether an external reason exists to continue retrieval. If it is, then the loop 1-2-3-4-5-6-1 is activated, the count of time begins anew (box 6), and internal cycle 1-2-3-4-1 is repeated again with a given frequency f, or time period $1/f$, while $t < t_0$.

The trace x_0 is held simultaneously in a particular NN memory (box 2) and in its auxiliary reference memory (RM) that may be interpreted as a *tag* of corresponding NN memory or as a *card* in a long-term memory catalog. An RM performs two interconnected functions: verification of current memory retrieval results (diamond 3 serves as a comparator) and validation of the fact that a particular memory record actually exists in the long-term memory store (diamond 3 serves as a familiarity/novelty detector). Thus, specific RM is a part of memory about memory or *'metamemory,'* in other words. In contrast to the NN memory, which is a kind of computer register and is conventionally associated with a real biological network, particular RM is a kind of *slot* devoted to the comparison of a current vector x_{out} with the reference pattern x_0 and may be associated with a coincidence integrate-and-fire *'grandmother'* neuron (cf. Sections 3, 4.1, 5, and ref. 14).

All elements of the internal feedback (reentry) loop, 1-2-3-4-1, run routinely in an automatic regime and for this reason they may be interpreted as related to an *implicit* (unconscious) memory. Consequently, under the NNAMM, all operations at synaptic and NN memory levels are unconscious. External feedback (reentry) loop, 1-2-3-4-5-6-1, is activated in an unpredictable manner because it relies on external (environmental and, consequently, unpredictable) information and in this way provides unlimited diversity of possible memory retrieval modes. For this reason, an AMU can be viewed as a particular *explicit* (conscious) memory unit. An external information used in diamond 5 can be thought of as an explicit or conscious one.

Recent evidences demonstrate that learning induces molecular changes in neocortex and hippocampus; this finding, along with based on it physiological theory assuming that any long-term memory record is stored in parallel in the neocortex and hippocampus [16], supports the NNAMM's idea of storing simultaneously each memory trace in an NN (a counterpart to a neocortex network) and in a 'grandmother' neuron (probably, a cell in hippocampal structures). This point of view is also consistent with the content of ref. 17,18 where a hippocampal comparator or familiarity/novelty detector is considered. For some other arguments in favor of the NNAMM's biological plausibility see ref. 4.

4.3 AMU's Basic Performance

The best data-decoding/memory-retrieval algorithms considered have common quality performance function, $P(d,\theta)$, the probability of correct decoding/retrieval or generalization, conditioned under the presence or absence of x_0 in the data analyzed, as a function of d and θ ($d = 1 - q$, all notations are as in Section 4.1).

The finiteness of the set of vectors $x(d)$ makes possible to find $P(d,\theta)$ by multiple computations [10]:

$$P(d,\theta) = n(d,\theta)/n(d) \qquad (5)$$

where $n(d)$ is a given number of different inputs with a given value of d, $x_{in} = x(d)$; $n(d,\theta)$ is the number of those $x(d)$ which are leading (under condition that for their decoding the NN algorithm with triggering threshold θ is applied) to the NN's response $x_{out} = x_0$. For small N, $P(d,\theta)$ can be calculated exactly because the number of items in the complete set of $x(d)$, $n(d) = 2^m C^N_m$, is small and they the all can be taken into account. For large N,

$P(d,\theta)$ can be estimated by multiple computations approximately but, using a sufficiently large set of randomly chosen inputs $x(d)$, with any given accuracy.

For intact perfectly learned NNs, convolutional (Hamming) version of the BSDT/NNAMM formalism allows to derive an expression for $P(d,\theta)$ analytically [15]:

$$P(d,\theta) = \sum_{k=0}^{k\max} C_k^m / 2^m, \quad k\max_0 = \begin{cases} (N-\theta-1)/2, & if \quad N \quad is \quad odd \\ (N-\theta)/2-1, & if \quad N \quad is \quad even. \end{cases} \tag{6}$$

Here if $k\max \le m$ then $k\max = m$ else $k\max = k\max_0$, C^m_k denotes a binomial coefficient.

Since θ (triggering threshold) and F (false-alarm probability), d (damage degree) and q (intensity of cue) are related (see details in ref. 12), functions $P(d,\theta)$ can, for example, be written as ROCs [receiver operating characteristic curves, $P_q(F)$], or BMPs [basic memory performance curves, $P_F(q)$] [4].

4.4 AMU's Learning

Equation 2 defines perfect one-step learning from one example because in this case for the NN considered its input, x_0, and its output, x_0 (the label, 'teacher,' or 'supervisor'), are exactly known. But often unsupervised learning is also needed.

Let us use the traditional delta learning rule in the form

$$w_{ij}^{(n+1)} = w_{ij}^{(n)} + \eta\, v_j^{(n)} h_i^{(n)} \tag{7}$$

where n and $\eta > 0$ are an iteration number and a learning parameter, respectively; $v_j = x^i_{\text{in}}$; $h_i = \sum w_{ik} v_k$, $k = 1,\ldots,N$. Here the training set consists of only one sample, $x_{\text{in}} = x_0$, and, consequently, Equation 7 describes learning from one example (such an iteration process does not feedback the NN's output x_{out} to the NN's input; the current value of w_{ij} is estimated using its previous value, the values of η and components of x_0).

If η is small ($\eta < 1$) then the learning rate achieved is low and asymptotic values of w_{ij} are not reached. This case has no essential practical significance. If η is large ($\eta > 100$) then the iteration process leads to a fast, one-trial, without the 'catastrophic forgetting' learning because already the first iteration gives the result which is close to the asymptote and next iterations do not lead to the essential advance.

Let us consider the NN with $N = 40$, continuous w_{ij}, v_j, h_i, x^i_{in}, x^i_{out} and all initial values of w_{ij} chosen randomly with uniform probability from the range [–1,1]. If the initial learning pattern is $x_{\text{in}} = x_0$ then after each next iteration an NN with the next version of its weight matrix w_{ij} provides the emergence of the next version of x_{out} (the next approximation of x_0). For example, for $\eta = 400$ already the first iteration gives the approximation's quality estimation $\sum |x^i_{\text{out}} - x^i_0| < 10^{-30}$ ($i = 1,\ldots,N$) which is more than enough for practical purposes.

Simultaneously with the NN itself, its specific reference memory (RM) should also be learned (for example, by direct recording the components of x_0 into RM).

5 Neuron RFs and NNs for Tuning

Figure 3 illustrates the process of visual data processing using the NN described in Section 4.1 and AMU described in Section 4.2. A binarization algorithm (e.g., [5]) transforms vector y, a half-tone image, into spinlike vector x_{in} without loss of information important for the following feature discrimination procedure (e.g., if $y_i > bd_i$ then $x^i_{\text{in}} = 1$ else $x^i_{\text{in}} = -1$). It is supposed that binarization of components of y or h means spike generation; h may be interpreted as a simplified 1D profile of a 'grandmother' neuron's receptive field (RF) which results in the process of internal weighted network computations (Equation 3, see also ref.14). Profiles of such RFs can be as it is typical for on-cells (panels *a*, *c*, *d*) or for off-cells (panel *b*) and, as panels *a* and *b* demonstrate, noise $x_{\text{in}} = x_r$ can initiate the reverse of the RF polarity (these predictions are consistent with current physiological results [19]). At a given level of data processing, the set of outputs of 'grandmothers' of different NNs (the top raw in Figure 3)

reduces the redundancy of initial data and can constitute an x_{in} for NNs at the next level of data processing hierarchy; in particular, in AIT, an x_{in} could already represent a face (Section 3).

From the above consideration and example (Figure 3) follows that within the theory for vision proposed for image recognition at any level of data processing hierarchy, the NNs of the same structure are used [Equations 2-4, Figures 1 (the insertion) and 3]. These NNs perform a given normalization of a current input (Equation 3) and thresholding the result (Equation 4). The main distinction between them consists in the fact that they (and their 'grandmothers') are learned ('tuned') to recognize different binary patterns x_0 which, depending on the context, can code simple elements/features of a visual scene (e.g., oriented bars in V1) as well its rather complex objects or their parts (e.g., human faces in AIT). Tuning the NNs considered to recognize optimally (to respond preferably to) x_0 is extremely simple because that is a one-step learning from one example, according to Equation 2 or 7. As NNs can generalize, they can recognize patterns $x(d)$, which are x_0 damaged by noise, and the more the d the smaller $P(d,\theta)$ is [d is a damage degree of x_0, $0 \le d \le 1$, $x(0) = x_0$; $P(d,\theta)$ is the probability of recognizing x_0 in $x(d)$; how at $\theta = 0$ $P(d,\theta)$ depends on d, one can see from examples in Figure 3 of ref. 4]. Thanks to this property of functions $P(d,\theta)$, the NN's tuning is 'bell-shaped' (at $\theta = 0$, Figure 3 of ref. 4 displays examples of some possible bell-shaped profiles of tuning). Hence, the NN introduced may be interpreted as an universal circuit underlying bell-shaped tuning in different visual brain areas (here, it is important to note that because within our 'grandmother' theory for vision each NN discussed is connected to its 'grandmother' neuron, the terms 'tuning the NN' and 'tuning the neuron' are synonymous).

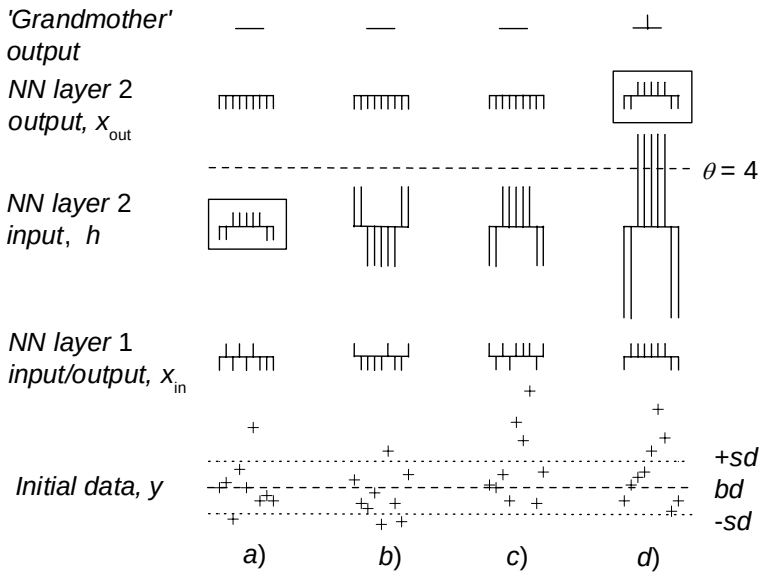

Figure 3. Computer simulated samples of initial visual data, y (e.g., an electric output of light-sensitive retina cells), and their processing results, x_{in}, h, x_{out}, in four N-channel data processing windows (crosses are values of y in each channel). In panels a and b, y is a fixed background, bd, damaged by Poisson-like noise, $bd = 100$; in panels c and d, y is a Gaussian peak on the background, bd, their sum is damaged by Poisson-like noise, the peak's amplitude is $a = 20$, its full width at half maximum is $fwhm = 5$. Vectors y, x_0 (boxed), x_{in}, h (1D profile of an RF), and x_{out} are N-dimensional ones, $N = 9$; positive and negative components of x_0, x_{in}, h, x_{out} correspond to upward and downward bars, respectively; intact NN and its 'grandmother' hold $x_0 = (-1, -1, 1, 1, 1, 1, 1, -1, -1)$, a kernel for the convolutional decoding/retrieval algorithm (the neuron's triggering threshold is $\theta = 4$); peak is identified in panel d; $sd = bd^{1/2}$, standard deviation of bd.

It is clear that the employment of NNs discussed for solving the problem of generalization may be considered as a kind of alternative to radial basic function approach (RBF), mentioned in Section 2.

6 Generalization by Computation through Memory Performance

Since $P(q,\theta)$ defines (Equation 5) the fraction of vectors $x_{in} \neq x_0$ leading, along with x_0, to successful retrieval of the trace x_0 from the learned NN (the insertion in Figure 1 and box 2 in Figure 2), the probability of memory retrieval, $P(q,\theta)$, and generalization ability by computation through memory, $g(q,\theta)$, are numerically equal, $g(q,\theta) = P(q,\theta)$.

Table 1

Generalization ability, $g(q,\theta) = n(q,\theta)/n(q)$, for an AMU storing the trace $x_0 = (-1, -1, 1, 1, 1, 1, 1, -1, -1)$[1].

q	Intact NN, $g(q,6)$,%[2]	Damaged NN, $g(q,0)$,%[3]	q	Intact NN, $g(q,6)$,%	Damaged NN, $g(q,0)$,%
1	2	3	4	5	6
0/9	10/512 = 1.953	10/512 = 1.953	5/9	5/16 = 31.250	630/2016 = 31.250
1/9	9/256 = 3.516	81/2304 = 3.516	6/9	4/8 = 50.000	336/672 = 50.000
2/9	8/128 = 6.250	288/4608 = 6.250	7/9	3/4 = 75.000	108/144 = 75.000
3/9	7/64 = 10.938	588/5376 = 10.938	8/9	2/2 = 100.000	18/18 = 100.000
4/9	6/32 = 18.750	756/4032 = 18.750	9/9	1/1 = 100.000	1/1 = 100.000

[1] q, intensity of cue ($q = 1 - d = 1 - m/N$, $0 \leq m \leq N$, $N = 9$; $q = 0$, free recall; $0 < q < 1$, cued recall; $q = 1$, recognition); θ, the neuron's triggering threshold; for definitions of $n(q,\theta)$ and $n(q)$, see Section 4.3.

[2] Values of $g(q,6)$ were calculated by Equations 5 and 6, results are equal.

[3] Values of $g(q,0)$ were calculated by Equation 5; 30 disrupted interneuron connections (entrance-layer neuron, exit-layer neuron) are as follows: (2,1), (4,1), (5,1), (6,1), (8,1), (3,2), (5,2), (7,2), (1,3), (4,3), (5,3), (2,4), (4,4), (2,5), (3,5), (7,5), (9,5), (3,6), (7,6), (8,6), (9,6), (1,7), (2,7), (4,7), (8,7), (1,8), (5,8), (3,9), (6,9), (7,9); this set of disrupted connections was chosen to illustrate the fact that similar to intact NNs, damaged NNs can also provide the best decoding/retrieval/generalization performance (in columns 2 and 3, 5 and 6, generalization abilities coincide completely).

In Table 1, generalization abilities for two AMUs, containing an intact and a damaged NN, are compared. In columns 2, 3, 5, and 6, values of $g(q,\theta)$ provide optimal (the best in the sense of pattern recall/recognition quality) generalization abilities; $g(0,6) = g(0,0) \sim 1\%$ was, for example, chosen as that is typical for professionals [5].

Usually, generalization is considered as a function of the relative size $\alpha = n/N$ of the training set of n examples and the learning strategy. For very large networks ($N \rightarrow \infty$) and $\alpha >> 1$, the error of generalization decreases as $\sim \alpha^{-1}$ [20]; for small networks (for learning from few examples), the problem of generalization remains unsolved in theory [1]. The approach proposed in this work gives a solution of this problem because it makes possible learning even from one example (Section 4.4).

7 Conclusion

The first solution of the problem of generalization through memory has been proposed and illustrated by an original 'grandmother' theory for vision, here introduced using the recent neural network assembly memory model, NNAMM [4]. For the NNAMM's intact NN memory unit, analytical formulae and a numerical procedure are found to calculate exactly optimal values of generalization as a function of the cue index, q, and the neuron's triggering threshold, θ; for two specific NNs their generalization abilities are numerically calculated (it is important that in all calculations simple binary/digital mathematics is only used). It has been demonstrated that the approach proposed provides generalization for the case of learning even from one example and that binary NNs discussed can also be interpreted as universal circuits underlying bell-shaped tuning of neurons in different visual brain areas.

Acknowledgments

I am grateful to the Health InterNetwork Access to Research Initiative (HINARI) for free on-line access to current full-text research journals, to an anonymous reviewer for useful comments, and to my family and my friends for their help and support.

Bibliography

1. T.Poggio and E.Bizzi. Generalization in vision and motor control. *Nature*, 2004, 431(7010), 768-774.
2. V.N.Vapnik. *Statistical learning theory*. Wiley, New York, 1998.
3. T.Poggio, R.Rifkin, S.Mukherjee and P.Niyogi. General conditions for predictivity learning. *Nature*, 2004, 428(6981), 419-422.
4. P.M.Gopych. A neural network assembly memory model based on an optimal binary signal detection theory. *Problemy Programmirovaniya* (*Programming Problems*, Kyiv, Ukraine), 2004, no. 2-3, 473-479; see also http://arXiv.org/abs/cs.AI/0309036.
5. P.M.Gopych. Identification of peaks in line spectra using the algorithm imitating the neural network operation. *Instruments and Experimental Techniques*, 1998*,* 41(3), 341-346.
6. N.Kanwisher, J.McDermott and M.M.Chun. The fusiform face area: A module in human extrastriate cortex specialized for face perception. *The Journal of Neurosciences*, 1997, 17(11), 4302-4311.
7. E.Tong, K.Nakayama, M.Moskovitch, M.Weinrib and N.Kanwisher. Response properties of human fusiform face area. *Cognitive Neuropsychology*, 2000, 17(1), 257-280.
8. Y.Wada and T.Yamamoto. Selective impairment of face recognition due to a haematoma restricted to the right fusiform and lateral occipital region. *Journal Neurology, Neurosurgery and Psychiatry*, 2001, 71(2), 254-257.
9. G.Edelman and G.Tononi. *A universe of consciousness: How matter becomes imagination*. Basic Books, New York, 2000.
10. P.M.Gopych. Determination of memory performance. *JINR Rapid Communications*, 1999*,* 4[96]-99, 61-68 (in Russian).
11. J.J.Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings National Academy of Sciences USA*, 1982, 79(8), 2554-2558.
12. P.M.Gopych. Sensitivity and bias within the binary signal detection theory, BSDT. *Int. Journal Information Theories & Applications*, 2004*,* 11(4), 318-328.
13. Yingxu Wang, D.Liu and Ying Wang. Discovering the capacity of human memory. *Brain and Mind*, 2003, 4(2), 189-198.
14. P.M.Gopych. Neural network computations with negative triggering thresholds. In *ICANN 2005, Lecture Notes in Computer Sciences 3696*, W. Duch et al. editors, pages 223-228. Springer-Verlag, Berlin-Heidelberg, 2005.
15. P.M.Gopych. ROC curves within the framework of neural network assembly memory model: Some analytic results. *Int. Journal Information Theories & Applications*, 2003*,* 10(2), 189-197.
16. P.K.Dash, A.E.Hebert and J.D.Runyan. A unified theory for systems and cellular memory consolidation. *Brain Research Reviews*, 2004, 45(1), 30-37.
17. J.Gray. The content of consciousness: A neuropsychological conjecture. *Behavioral Brain Sciences*, 1995, 18(4), 659-722.
18. D.C.Dennett. Overworking the hippocampus. *Behavioral Brain Sciences*, 1995, 18(4), 677-678.
19. G.C.DeAngelis, I.Ohzava and R.D.Freeman. Receptive-field dynamics in the central visual pathways, *Trends in Neurosciences*, 1995, 18(10), 451-458.
20. M.Opper. Statistical mechanics of generalization. In *The Handbook of Brain Theory and Neural Networks,* Michael A. Arbib editor, pages 922-925. The MIT Press, Cambridge, Massachusetts, 1995.

Author's Information

Petro Mykhaylovych Gopych – V.N.Karazin Kharkiv National University; Svoboda Sq., 4, Kharkiv, 61077, Ukraine; e-mail: pmg@kharkov.com.