AFRL-RY-HS-TR-2010-0028

Cognitively Inspired Neural Network for Situation Recognition

Roman Ilin and Leonid Perlovsky

AFRL/RYHE
80 Scott Drive
Hanscom AFB MA 01731-2909

14 January 2010

Technical Report

AIR FORCE RESEARCH LABORATORY

Sensors Directorate

Electromagnetics Technology Division

80 Scott Drive

Hanscom AFB MA 01731-2909

NOTICE AND SIGNATURE PAGE

Distribution: Unlimited, Statement A

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE US GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY RELATE TO THEM.

THIS TECHNICAL REPORT WAS CLEARED FOR PUBLIC RELEASE BY THE ELECTRONICS SYSTEMS CENTER PUBLIC AFFAIRS OFFICE FOR THE AIR FORCE RESEARCH LABORATORY ELECTROMAGNETICS TECHNOLOGY DIVISION AND IS AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONALS. COPIES MAY BE OBTAINED FROM THE DEFENSE TECHNICAL INFROMATION CENTER (DTIC) (http://www.dtic.mil)

AFRL-RY-HS-TR-2010-0028 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

Leonid Perlovsky

LEONID I. PERLOVSKY
Principal Research Physicist
Electromagnetic Scattering Branch

Bertus Weijers

BERTUS WEIJERS
Branch Chief
Electromagnetic Scattering Branch

Robert V. McGahan

ROBERT V. McGAHAN
Technical Communication Advisor
Electromagnetics Technology Division

This report is published in the interest of scientific and technical information exchange and does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
14-01-2010	Interim Report	12/15/08- 12/15/09

4. TITLE AND SUBTITLE

Cognitively Inspired Neural Network for Recognition of Situations

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)

Roman Ilin and Leonid Perlovsky

5d. PROJECT NUMBER
2311

5e. TASK NUMBER
HE

5f. WORK UNIT NUMBER
03

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)

AFRL/RYHE
80 Scott Drive
Hanscom AFB MA 01731-2909

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Electromagnetics Technology Division SC-437490
Sensors Directorate
Air Force Research Laboratory
80 Scott Drive
Hanscom AFB MA 01731-2909

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL-RY-HS

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RY-HS-TR-2009-0028

12. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution A. Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Technical Report; Clearance number: ESC/PA : 66ABW-2010-0854

14. ABSTRACT:

We developed a cognitively inspired mathematical learning framework called Neural Modeling Fields (NMF). We applied it to learning and to recognition of situation composed of objects. NMF can successfully overcome the combinatorial complexity of associating subsets of objects with situations and provides fast and reliable convergence. We discuss the implications of the results of this current work for building multi-layered intelligent systems.

15. SUBJECT TERMS: Neural Modeling Fields, Maximum Likelihood Estimation, Parametric Models, Neural Networks, Situational Awareness

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Leonid Perlovsky
Unclassified	Unclassified	Unclassified	UU	42	19b. TELEPHONE NUMBER *(include area code)* N/A

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The field of Artificial Neural Networks has dramatically expanded over the past decades (Bishop, 1996; Haykin, 1999; Perlovsky, 2001). Neural networks have been established as powerful tools in the areas of pattern recognition, function approximation, and control, to name just a few. This latest expansion is a result of the advances in the development of efficient learning algorithms for feed-forward and recurrent architectures. Despite the successes, the neural networks, along with the other computing paradigms, run into serious limitations as the size of the data increases. Going beyond the neural networks paradigm, modeling complex systems with methods of artificial intelligence, pattern recognition, or modeling processes in the mind encountered computational complexity in many applications. The fundamental principles of artificial intelligence and learning were summarized in Perlovsky 2001; Cherkassky and Mulier 2007; Mitchell 1997.

Consider a simple object perception that involves signals from sensory organs and internal mind's representations (memories) of objects. During perception, the mind associates subsets of sensor signals corresponding to objects with representations of specific objects. This produces object recognition and activates brain signals that lead to mental and behavioral responses, and contributes to understanding.

Mathematical modeling of the very first recognition step in this seemingly simple association-recognition-understanding process has encountered a number of difficulties over the decades. These difficulties were first identified in pattern recognition and classification research in the 1960s and were named "the curse of dimensionality" (Bellman 1961). It seemed that learning algorithms and neural networks could learn solutions to any problem 'on their own', if they were provided with a sufficient number of training examples. The following thirty years of developing learning algorithms led to the conclusion that the required number of training examples was often combinatorially large. Self-learning pattern recognition and neural network approaches encountered a combinatorial complexity (CC) of

learning requirements. Various ways of overcoming CC in neural networks include techniques like pruning, regularization, and weight sharing. For examples of such approaches see (LeCun et al, 1990; Ilin et al 2008).

Rule systems were proposed in the 1970's to overcome the CC of learning (Minsky, 1975; Winston 1984). A leading idea was that rules would capture the required knowledge and eliminate the need for learning. However in the presence of variability, the number of rules grew, and the rules became contingent on other rules. This caused combinations of rules to be considered and these rule systems encountered a CC of rules.

Model systems were proposed in the 1980's to combine the advantages of a priori knowledge and learning. Model systems used models that depended on adaptive parameters. The knowledge was encapsulated in the models, while unknown aspects of particular situations were to be learned by fitting the model parameters. However, fitting the models to data required selecting data subsets corresponding to various models. The number of subsets, however, is combinatorially large. A general popular algorithm for fitting models to the data, multiple hypotheses testing (Singer at el, 1974) is known to face CC of computations. Unfortunately, model-based approaches encountered computational CC (Perlovsky at el, 1998b).

Computational difficulties have been summarized under the notion of CC in (Perlovsky, 1998a). In general, CC refers to multiple combinations of various elements in a complex system. For example, recognition of a scene often requires concurrent recognition of multiple elements that could be encountered in various combinations. CC is prohibitive because the number of combinations can be very large. For example, consider 100 elements (not too large a number). The total number of subsets of a

set with 100 elements is 2^{100}. This exceeds the number of all elementary events in life of the Universe and no computer could ever be able to compute that many combinations.

The following research relates CC to formal logic, the basis of various algorithms and neural networks (Perlovsky 2001). Formal logic is based on the "law of excluded middle," where every statement is either true or false. Therefore, algorithms based on formal logic have to evaluate every combination of data and internal representations as a separate logical statement and a large number of these combinations will cause combinatorial complexity. It turns out that all popular algorithms and neural networks rely on logic. Rule systems are based on logic in a straightforward way. Model systems are based on logic in the matching process, which consists in testing logical hypotheses of the type: "this model corresponds to that subset of data." Learning algorithms, such as pattern recognition and neural networks, use logic in the training process, consisting of logical statements "this is a chair" (and therefore of combinations of logical statements: "this is a chair and that is a cat"). Fuzzy logic encountered a difficulty that related to the degree of fuzziness that is set by using formal logic. Complex systems require different degrees of fuzziness in various subsystems and at various steps of the system operations, and searching for the appropriate degrees of fuzziness among combinations of steps and subsystems again would lead to CC. Statistical Learning Theory is a powerful learning paradigm developed by Vapnik (Cherkassky & Mulier 2007; Perlovsky 2001). But it also could not avoid CC when combining learning with knowledge. In fact, all algorithms and neural networks used logic in some way. Combinatorial complexity of algorithms based on logic is related to Gödel's theorem: it is a manifestation of the incompleteness of logic in finite systems. For a complete discussion please see (Perlovsky, 1996).

The mathematical foundations of this paper are Neural Modeling Fields (NMF) and Dynamic Logic (DL). NMF combine the structure of logic with the dynamics of a connectionist paradigm to achieve a goal of modeling systems without CC. Models encapsulate the prior knowledge that exists before learning begins. These models interact with sensor data. While models generate top-down signals, sensor organs generate bottom-up signals. NMF avoids CC by using DL (Perlovsky, 2001, 2006). DL is described as process-logic. It is a process "from vague to crisp," from vague representations, models, decisions, to crisp ones. Because the senses always interact with more than one object (we usually see many objects at the same time), DL must solve the data association problem and the recognition problem. As mentioned above, the models are initially vague and are associated with the entire input data set. In the DL process associations between the models and data become crisper, allowing to converge to correspond more closely to patterns in the data. The vagueness in the data association matches the uncertainty of the models. At the end of the process correct associations are formed and the models provide a close fit to the data. The process is guided by the goal of maximizing the similarity between the data and the models.

The DL process can be described as an interaction between the bottom-up signals coming from sensors and the top-down signals coming from models. The perception and cognition result from matching the top-down and bottom-up signals. The meeting point is the convergence of the abstract concept into a concrete perception. From the neuro-physiological point of view, the bottom-up signals flow from sensor neural activations, for example from the retina to the visual cortex. Top-down signals flow from activated models/representations "down" to the visual cortex (Grossberg, 1982; Schacter, 1987; Kosslyn, 1994). In a recent study (Bar et al, 2006), it has been demonstrated that the object recognition by human subjects occurring in the temporal cortex is facilitated by the top-bottom signals originating in the orbitofrontal cortex. The initial top-down signals (coming from the models/representations) are vague confirming the Dynamic Logic mechanism of "vague to crisp" process (Perlovsky, 2009d).

It is interesting that logic was invented by Aristotle, but he himself did not seem to consider logic to be the basic mechanism of the mind (Aristotle IV BCE; Perlovsky 2006c; 2009e). We consider Aristotle to be the originator of the idea of matching vague concepts with concrete percepts. The Aristotelian theory of mind postulates the existence of a priori Forms that are abstract concepts existing in the mind. We perceive concrete ideas by imagining Forms in the mind. According to Aristotle, Forms in the mind do not obey logic. They become logical by experiencing real matter. Aristotle emphasized that the initial states of Forms, Forms-as-Potentialities, are not logical (i.e. they are vague). But their final forms, forms-as-actualities, are attained from the result of learning, and are logical. This Aristotelian description corresponds to the DL process.

The NMF-DL approach provides a mathematical description of the Aristotelian cognitive process, and provides an algorithm for the perception of multiple patterns in the environment. The DL theory goes on to postulate that this algorithm is a universal mechanism of the mind (Perlovsky, 2001; Perlovsky, 2006b; Perlovsky and Kozma, 2007). The mind is considered as a layered system with the models of each layer sending signals to the layer above. In a simplified description, bottom layers in the mind hierarchy recognize objects in the outside world. Higher layers contain abstract models and can recognize more abstract concepts and situations.

Several journal articles and books have demonstrated how the NMF-DL is utilized in perceptual tasks such as object recognition (Deming, 1998; Deming et al, 2007; Kozma et al, 2007; Linnehan et al, 2007; Perlovsky et al, 2007). In this contribution we demonstrate how the NMF-DL approach can operate on a more abstract level for recognizing situations involving multiple objects.

This introduction is followed by five more sections: 2. Neural Modeling Fields (NMF); 3. NMF-DL (Dynamic Logic) for learning situations; 4. Simulation Results; 5. Computational Complexity; 6. Discussion and Conclusions.

2. Neural Modeling Fields (NMF)

Neural Modeling Fields (NMF) together with Dynamic Logic (DL) provides a generic framework for learning from data. It is developed in (Perlovsky, 1987; Perlovsky & McManus, 1991; Perlovsky, 2001) as a model framework based on the known mind mechanisms. NMF-DL finds the best match between the internal models and the inputs while avoiding the computational complexity of data association. The mathematical formulation of NMF is given in this section. We will use bold letters to denote vector quantities.

The main components of the NMF framework are the input data and the parametric models. We denote the input data by $\boldsymbol{x}$, $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$. The data is a set of stimuli that are coming onto the retina and therefore represent bottom-up signals. Although, sensor data are continuously coming into the mind, for simplicity we will use a fixed number of input signals, N. We denote the set of models by $\boldsymbol{M}$, $\boldsymbol{M} = (\boldsymbol{M}_1, \ldots, \boldsymbol{M}_H)$. Here H is the total number of concept models. Each model depends on its parameters $\boldsymbol{S}_h$: $\boldsymbol{M}_h = \boldsymbol{M}_h(\boldsymbol{S}_h)$.

Model M_h predicts the value of $\boldsymbol{x}_n$ based on the current model parameters $\boldsymbol{S}_h$. We introduce a measure of partial similarity,

$$l(\boldsymbol{x}_n | \boldsymbol{M}_h(\boldsymbol{S}_h)) \quad (1),$$

between a given input data element $\boldsymbol{x}_n$ and a given model $\boldsymbol{M}_h$. For simplicity we will also denote partial similarity as $l(n|h)$. It is a function of the data and the model parameters. It provides the measure of similarity between the predicted and the true values of $\boldsymbol{x}_n$. The specific form of $l(n|k)$ will be considered later. The similarity is maximized when the model's prediction is exact and it vanishes when the model's prediction is far from the true value. We use notations similar to standard statistical

description, and partial similarity corresponds to conditional probability density functions (PDFs), under certain conditions.

A data input, x_n, can be associated with any of the H models. Using a probabilistic formulation, the similarity between a given data element x_n and all the models is given by a sum over all of the H models.

$$l(n) = \sum_{h=1}^{H} l(n|h) \qquad (2)$$

If one of the models predicts the data very well, its similarity will dominate in the equation. If none of the models predicts the data element x_n with high accuracy then the total similarity will be small.

The total similarity between all the data and all the models is defined as the product of the individual similarities,

$$L(\boldsymbol{x}|\mathbf{M}) = \prod_{n=1}^{N}\left[\sum_{h=1}^{H} l(n|h)\right] \qquad (3).$$

The product of all data elements corresponds to the requirement that all data must be processed. Therefore if even a single data element is predicted poorly the whole similarity could easily be severely affected. Expression (3) is a shorthand mathematical formulation of NMF. A concrete implementation requires the specification of the partial similarities l(n|h) and the models that make these similarities. Detailed discussions of mathematical expressions for similarities and models for a number of applications can be found in (Perlovsky 2001; 2006; Deming 1998; Deming at el 2007, Deming and Perlovsky 2007). In section 3 we discuss specific expressions for learning and situation recognition. We emphasize once more that maximizing expression (3) using brute force lead to CC. The efficient use of NMF requires combining it with DL; the referenced publication discuss in detail how DL is combined with NMF for many applications.

Equation (3) is similar to a standard probabilistic formulation. In a standard probabilistic formulation, a statistical parameter p(h) is used in front of l(n|h); We should note that in a probabilistic formulation, a product assumes independence. However, in our case, the product in (3) does not imply that data x_n are independent. In our case, expression (2) is a PDF of error between the data x_n and the model prediction, when using the multiple hypothesis assumption. Using a probabilistic interpretation, these errors can be assumed to be independent.

We reiterate that under certain conditions equation (3) is the total likelihood. Finding the parameters of models by maximizing the likelihood provides the maximum likelihood (ML) estimation representing the best possible estimate under certain conditions (Kay , 1993). But the beneficial properties of the ML estimation, however, are of secondary concern in this paper. The likelihood expression (3) contains combinatorially many items. Standard estimation algorithms, such as (Singer at el 1974) maximize these items one by one, and then select the largest. This leads to CC of all state of the art algorithms, and has caused this problem to be unsolvable for decades. Maximization of (3) with respect to model parameters $\mathbf{S}_h$, h=1,..., H, can be attempted by gradient ascent. Gradient ascent is a non-combinatorial solution, and its complexity is linear. The difficulty of this approach is that similarity (3) is a highly nonlinear function with a combinatorial number of local maxima. Our main contribution in this paper is to solve this "unsolvable" problem without CC. The reason DL can solve it without CC is that in the process "from vague to crisp," the local maxima are ironed out in the initial stages of the problem. The solution converges to a crisp one, and local maxima may cause problems only in a later stage of the DL process, when a solution is close to the global maximum, and the local maxima has been avoided. This property of DL has been demonstrated and discussed in details in dozens of problems and in hundreds of publications (see references in the text).

It is convenient to introduce a log similarity function, which replaces the product with the summation. The introduction of this logarithm does not change the nature of similarity measure since log is a monotonically increasing function.

$$LL(\boldsymbol{x}|\mathbf{M}) = \log L(\boldsymbol{x}|\mathbf{M}) = \sum_{\mathrm{n}=1}^{\mathrm{N}} \log\left[\sum_{\mathrm{h}=1}^{\mathrm{H}} l(n|h)\right] \qquad (4)$$

The gradient ascent is given by

$$d\boldsymbol{S}_h = dt\, \frac{\partial LL(\boldsymbol{x}|\mathbf{M})}{\partial \boldsymbol{S}_h} \qquad (5)$$

Here we use the symbol for partial derivative with respect to a vector of model parameters S_h. This notation follows (Perlovsky, 2001, 2006a, 2006b, 2006c) and we comment that the gradient symbol could be used instead.

Consider the expression for the derivative of the total similarity with respect to the parameters of one of the models with index h,

$$\frac{\partial LL(\boldsymbol{x}|\mathbf{M})}{\partial \boldsymbol{S}_h} = \sum_{n=1}^{N} \frac{\partial}{\partial \boldsymbol{S}_h}\left(\log\left[\sum_{\mathrm{h}_2=1}^{\mathrm{H}} l(n|h_2)\right]\right) = \sum_{\mathrm{n}=1}^{\mathrm{N}} \frac{l(n|h)}{\sum_{\mathrm{h}_2=1}^{\mathrm{H}} l(n|h_2)} \frac{\partial \log l(n|h)}{\partial \boldsymbol{S}_h} \qquad (6)$$

We introduced the subscript h_2 for the internal summation, and reserve h for the subscript of the model we are differentiating with respect to. The last expression is obtained using the fact that the function $l(n|h)$ depends only on $\boldsymbol{S}_h$ and by using the identity $\frac{d\, log\, y}{dy} = \frac{1}{y}$.

We introduce a set of association functions defined by

$$f(h|n) = \frac{l(n|h)}{\sum_{\mathrm{h}_2=1}^{\mathrm{H}} l(n|h_2)} \qquad (7)$$

These functions define associations between data n and model h and they are convenient because they belong to an interval [0,1]. They are defined similar to a posteriori Bayes probabilities, and under certain

conditions they converge to a posteriori Bayes probabilities. The gradient ascent is re-written using the functions f(h|n) in the following equation

$$\frac{\mathrm{d}\boldsymbol{S}_k}{\mathrm{dt}} = \sum_{\mathrm{n=1}}^{\mathrm{N}} f(h|n) \frac{\partial \log l(n|h)}{\partial \boldsymbol{S}_h} \qquad (8)$$

At this point we define an algorithm to maximize the total similarity. The algorithm consists of performing the steps of the gradient ascent which involve iterative evaluation of (7) and (8) until some convergence criteria are satisfied.

Consider an alternative way of maximizing the similarity. Instead of performing the gradient ascent we set the gradient $\frac{\partial LL(\boldsymbol{x},\mathbf{M})}{\partial \boldsymbol{S}_h}$ to zero and solve the resulting equation for the unknown parameters $\boldsymbol{S}_h$.

$$\sum_{\mathrm{n=1}}^{\mathrm{N}} f(k|n) \frac{\partial \log l(n|h)}{\partial \boldsymbol{S}_h} = 0 \;\rightarrow \boldsymbol{S}_h \qquad (9)$$

Consider the following iterative process

1. *Initialize parameters*	$\boldsymbol{S}_h{}^I = \boldsymbol{S}_k{}^0$
2. *Compute associations*	$f(h\|n) = \frac{l(n\|h)}{\sum_{\mathrm{h_2=1}}^{\mathrm{H}} l(n\|h_2)}, given\ \boldsymbol{S}_k{}^I$
3. *Estimate parameters*	$\sum_{\mathrm{n=1}}^{\mathrm{N}} f(h\|n) \frac{\partial \log l(n\|h)}{\partial \boldsymbol{S}_h} = 0 \rightarrow \boldsymbol{S}_h{}^{I+1}, given\ f(h\|n)$
4. *Set and repeat 2 and 3*	$\boldsymbol{S}_h{}^I = \boldsymbol{S}_h{}^{I+1}$

(10)

This procedure has been shown to increase the total similarity after each step and therefore converges (Perlovsky, 2001). This approach converges faster than the gradient ascent, therefore we use this procedure.

The NMF-DL is a biologically motivated approach. It uses well known interactions between bottom-up and top-down signals, that are also used in several types of neural networks (Carpenter and Grossberg, 1998, Kosslyn, 1994, Schacter, 1987, Bar et al, 2006). As previously mentioned, Bar et al, 2006 have

demonstrated that the DL process "from vague to crisp" is actually the mechanism used in the brain. This property of vague states of initial representations, that gradually become crisp, is a unique property of NMF-DL, which eliminates CC.

The cognitive foundation of the NMF is aided by visualizing the framework as an artificial neural network as shown in Fig. 1. The elements of this network are not individual neurons but become populations of neurons that represent internal models. The input layer sends the bottom-up signals to the middle layer consisting of numerical weights and similarity measures, and the model layer sends the top-down signals to the middle layer. Both top-down and bottom-up signals are necessary to compute the similarities and the association weights of the middle layer, to provide feedback to the model layer forming a recurrent loop. The iterations between the weights layer and the models layer converge to the best match between the input data and the models.

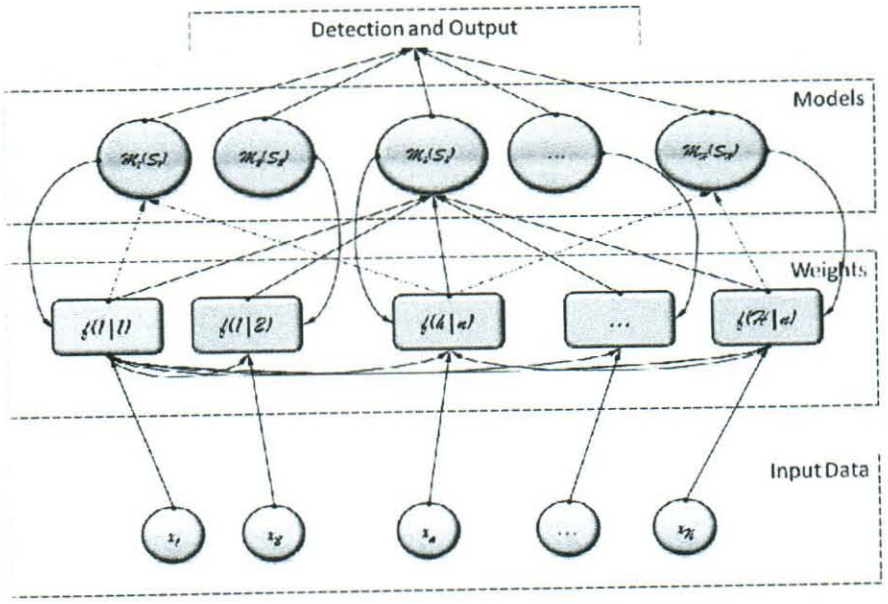

Figure 1

Neural interpretation of the NMF system; the three functional parts of the system are Input Data, Models, and Weights. The weights f(h|n) are computed by interactions between the inputs and weights layer and the

competitive-cooperative interactions within the weights layer . The model parameters S_h depend on the weights forming a recurrent loop.

Figure 1 illustrates how the NMF framework can be interpreted and implemented as a recurrent neural network with weight update rules given by equations (7) and (8) or (10). Moreover, the evaluation of association weights (7) can be cast in a differential form as well making the equations suitable for neural network interpretation (Perlovsky, 2001).

3. NMF-DL FOR LEARNING SITUATIONS

In this section the general formulation of NMF is extended to the case of learning and recognizing scenes from objects. We will refer to the set of objects that are essential for creating a meaning of the observed scene as a situation. For example, the presence of paved roads and tall buildings in the scene means that the observer is looking at an urban landscape. The presence of tables, chairs, plates, forks and spoons is enough to create a restaurant situation. The essential objects are intermixed with other objects that do not relate to the essential objects. These objects play the role of noise by making situation recognition difficult.

We denote D_x to be the total number of objects that exist in the world. This is a large but finite number. An observer can perceive N_p objects in the scene. These N_p objects are a much smaller number in comparison to D_x. Each situation is characterized by the presence of N_s objects, where N_s is smaller than N_p. The sets of objects that constitute different situations may overlap, with some objects being essential to more than one situation. We assume that each object is encountered in the scene only one time. This simplification is not essential since we can consider sets of similar objects as a new object type. For example, "book" is an object type and "books" is another object type referring to more than

one book. If necessary, a new object type – "lots of books" - can be introduced to refer to a large collection of books and with such an object it may be essential to distinguish between situations like "library" and "office".

Perception of objects can be represented as a binary vector $\mathbf{x_n} = (x_1 \ldots x_i \ldots x_{Dx})$. If the value of x_i is one the object i is present in the situation and if x_i is zero, the corresponding object is not present. Since D_x is a large number, $\mathbf{x_n}$ is a large binary vector with most of its element equal to zero.

We introduce a situation model as a vector of probabilities $\boldsymbol{p}_h = (p_{h_1} \ldots p_{h_i} \ldots p_{h_{Dx}})$. Here p_{h_i} is the probability of object i being part of the situation k. This situation model contains D_x unknown parameters. Estimating these parameters constitutes learning.

The similarity between vector $\mathbf{x_n}$ and model $\mathbf{p_h}$ representing a situation h is then given by the following formula (Duda et al, 2000).

$$l(n|h) = prob(\boldsymbol{x}_n|h) = \prod_{i=1}^{D_x} p_{h_i}^{x_{n_i}} (1 - p_{h_i})^{(1-x_{n_i})} \qquad (11)$$

This equation is a distribution of independent objects. Should objects appearing in a context of a situation be considered independent, so that there is a dependence between objects and consequently a "context" emerges in the result of the learning situations? Or should we assume that the objects in context are correlated, in other words, are these infants born with genetically specified contexts? It is obvious to us that an assumption of independence seems more reasonable. However, we do not have to solve this problem here. It is sufficient to consider the independence assumption in (17) as a model, and to demonstrate that even with this model, dependence-contexts appear in the result of learning situations. (Of course, any correlation between objects would make a problem easier to solve, even with the model (11)). We use the formula for the probability of binary vector $\mathbf{x_n}$ as the measure of similarity between this binary vector and its model $\mathbf{p_h}$. This expression vanishes when $p_{hi} = 0$ or $p_{hi} = 1$. In order to

avoid numerical instabilities in this implementation we impose limits on p_{hi} that will always keep it above zero and below one.

We would like to add the following for future research. In this paper $\mathbf{x}_n$ is a binary vector and we separated object recognition and situation recognition for simplicity. However, in actual brain operations, objects and situations are processed in parallel. Situation learning and recognition are ongoing processes and not the one-time job as we model it in this paper. In addition, before the situation learning begins, object recognition is not finished, and describing object identities with binary variables may not be adequate. Continuous variables $x_{n,i} \in [0,1]$ could be more appropriate, with a PDF of a similar functional form, when properly normalized. Emotional interactions can also be modeled (Barrett and Bar, 2009). Conceptual-emotion model of top-down and bottom-up interactions among layers in a hierarchical system, adequate for the brain modeling, is an additional topic for future research.

We substitute the similarity measure given by equation (11) into equation (4) to obtain the total similarity for our case. By taking the partial derivative of $l(n|h)$ and substituting it into (9) it can be shown to yield the following formula.

$$\frac{\partial LL}{\partial p_{h_i}} = \sum_{n=1}^{N} f(h|n) \frac{x_{n_i} - p_{h_i}}{p_{h_i}(1 - p_{h_i})} \qquad (12)$$

Setting this expression to zero we obtain the following expression for p_{hi}.

$$p_{h_i} = \frac{\sum_{n=1}^{N} f(h|n) x_{n_i}}{\sum_{n=1}^{N} f(h|n)} \qquad (13)$$

This expression is used in the parameter estimation step in equation (10). The association step remains the same.

4. Simulation Results

As we previously discussed, the mind has multiple levels that range from simple features and objects at the "bottom" to situations and abstract concepts at "higher' levels. Here we consider a single level of situation recognition. A situation is characterized by a set of objects. The objects are recognized at lower levels. In a real mind, multiple levels operate in parallel, but here we consider the level of situations separately. Our approach is applicable to higher levels as well, but we limit this paper to a single level by considering only the types of objects that are normally present in a given situation. In such a formulation the problem remains difficult due to the large number of possible situations and due to the presence of random objects that introduce strong "noise" into the data.

The problem of learning situations is complicated, because the learning system is exposed to various situations in a random order and without explicit teaching (most situations are unlabeled). In every situation most observed objects are irrelevant to this situation (clutter), and only a small number of objects are uniquely specific for this situation. In addition, most of observations are "clutter", and they do not relate to anything worth learning in important "situations," but contain only irrelevant objects. These difficulties are specific manifestations of CC for situational learning and recognition. For these reasons, learning situations for decades have remained a long-standing unsolved problem.

To summarize, simulation examples in this section all pertain to the following problem: an intelligent agent (a child) can recognize D_x objects in the world. He/She observes N_p objects at a time, called a sample, or an observation, or a realization of a situation. Observations are repeated many times. There are H_s different situations, that we call important, and each is characterized by N_s objects essential for this situation (always repeated in this situation) and another N_p-N_s "clutter" objects that are selected randomly for each observation (sample). In addition there are many "non-important" clutter situations,

that are only characterized by N_p clutter objects (that are selected randomly for every observation of this situation). No supervision is provided. The problem is (1) to learn these important situations, (2) to learn the essential objects that characterize each of them, and (3) to separate clutter situations from important situations. In this work we use synthetic data, so that the results can be evaluated with respect to the truth.

Each observation results in the recognition of N_p objects. This is represented as a vector $\mathbf{X}_n$ of binary variables with each component indicating the presence or absence of the object with corresponding index. Note that the identities of the objects, emerging in the hierarchical brain system, are not discussed in this paper, so we simply use object indices varying from 1 to D_x.

We initialize the DL process with H-1 situational models and one clutter model corresponding to random collections of objects, to give a total of H models. The clutter model is initialized with each object probability, p_{hi}, equal to 0.5. Similarly to meanings of objects, the meanings of situations emerge in the brain hierarchy. We do not consider this process here and use situation indices to identify them. We refer to objects and situations by using corresponding indices: i for an object, h for a situation, and n for a sample or a situation observation.

Table 1

Situation Index	Indices of Essential Objects
1	2 16 17 53 59
2	17 19 22 68 88
3	5 24 42 65 96
4	19 22 35 68 94
5	43 51 53 65 71
6	6 25 49 63 87
7	13 19 50 60 87
8	23 47 52 53 97
9	19 61 71 84 87
10	9 17 43 49 57

In the first example we set the total number of objects equal to 100. The number of objects in each data sample N_p is 12. The number of situation-essential objects that repeat in each instance of the same situation N_s is set to 5. There are 5 essential objects that distinguish a particular situation. However the observer sees them along with $N_p - N_s$ other objects that are irrelevant to this situation, that are randomly selected clutter objects. The total number of different situations that the learning system is exposed to is H_s=10.

To generate data we randomly selected 10 groups (H_s=10) of 5 (N_s =5) objects and fixed them as essential for this situation. Table 1 shows the indices of essential objects in each of the 10 situations. For each situation we add N_p - N_s =7 randomly selected objects. We also generated 10 more groups of 12 (N_p =12) randomly selected objects (clutter) to model random-clutter perceptions that do not correspond to important situations in the sense that they are not characterized by permanent essential objects. We generated 25 data samples for each situation resulting in the total of 500 data samples.

The input data is visualized in Figure 2. Here the horizontal axis corresponds to the index of the sample (the observation of one situation) and the vertical axis corresponds to the index of the object. The bright white pixels indicate the presence of objects in situations. The samples in Figure 2 are sorted by the situation index (horizontally) by grouping the samples of the same situation together. This makes samples from the same situation located next to each other on the plot, the bright spots form horizontal lines correspond to repeated essential objects and appear as bright lines. Note that the object indices of the lines that correspond to the object indices in Table 1. The horizontal length of each bright line is 25 pixels corresponding to 25 samples of each situation. The first 10 situations are important situations, characterized by repeated essential objects. Clutter situations, without bright lines (all objects are random clutter) follow on the right.

Figure 2 contains the solution of the problem, all situations and essential objects are clearly visible. In real life, however, situations do not come "sorted" together. Situations are observed as they appear, without any order, and it is not clear what is clutter and what is an important situation, and how do we know what objects are essential and should be identified. Figure 3 shows this real-life case, with the same data in Figure 2 where the sample index (horizontal axes) has been randomly permuted. This corresponds to the random order of the situations that are observed. The horizontal lines have disappeared. And the problem becomes difficult to solve. If we use a simple-minded sorting by looking at all possible rearrangements, until horizontal lines for essential objects become obviously visible, we would have to evaluate all permutations of the horizontal index, N! =500!.

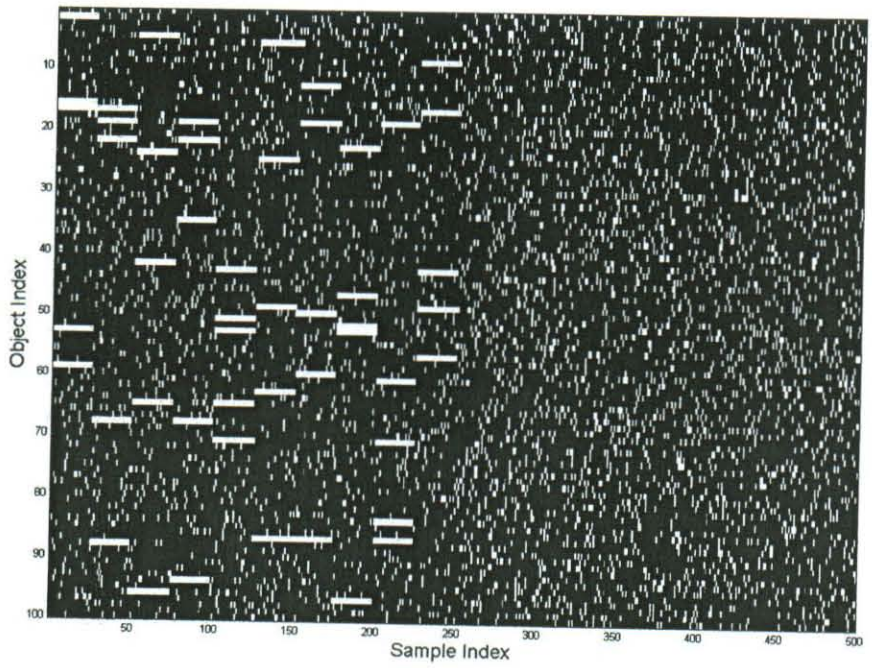

Figure 2

Visualization of the binary data input for the experiments with N_p=12. The object index is shown along the vertical axes and the sample index along the horizontal axes. For each column in this image, bright squares represent presence of an object in a sample and dark squares represent the absence of the object. Each column contains 12 bright squares. The data is sorted by situation making repeated objects visible as 25 pixel horizontal lines.

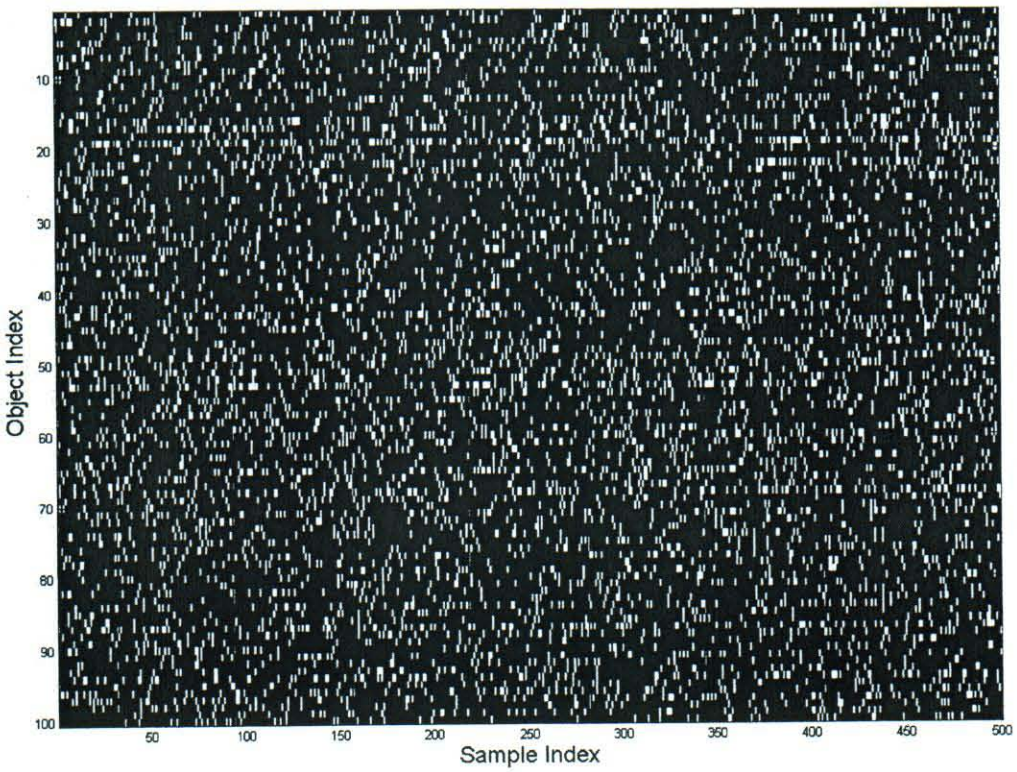

Figure 3

Visualization of the randomized binary data input for the experiments with N_p=12. The object index is shown along the vertical axes and the sample index along the horizontal axes. For each column in this image, bright squares represent presence of an object in a sample and dark squares represent the absence of the object. Each column contains 12 bright squares. The samples are presented in random order making visual identification of repeated objects impossible.

The algorithm given by equation (10) when the parameter estimation step is given by equation (19) is applied to the data. The parameters of each model are $S_h = \{p_{h_i}, i = 1, \dots, D_x\}$. To apply the algorithm we need to initialize 20 important situational models (an arbitrary assumption given that the true number of important models is 10 with values of p_{h_i} drawn from a uniform distribution with limits 0.3 and 0.7). These initial values correspond to the initial vague state of DL causing all objects to have a significant chance to belong to each situation. All initializations should be different, because models initiated with the same parameters will change in exactly same way. The 21st model, random clutter, has

all of its components equal to 0.5. Note that we start with a number of models that is greater than the number of true situations. In reality the number of situations that we need to learn is not known in advance and the algorithm can be modified to add or delete models as necessary to maximize the similarity.

Figure 4 illustrates the operation of the algorithm. Each image corresponds to one of the iterations of equation (10). We ran a total of 10 iterations. The figure displays the first 3 iterations and the last iteration. The horizontal axis corresponds to the model index varying from 1 to 20. The random noise model is not displayed since it does not change between the iterations. The vertical axis corresponds to the object index as in Figures 2 and 3. The brightness of the pixels corresponds to the values p_{h_i} with bright white corresponding to 1 and black corresponding to 0.

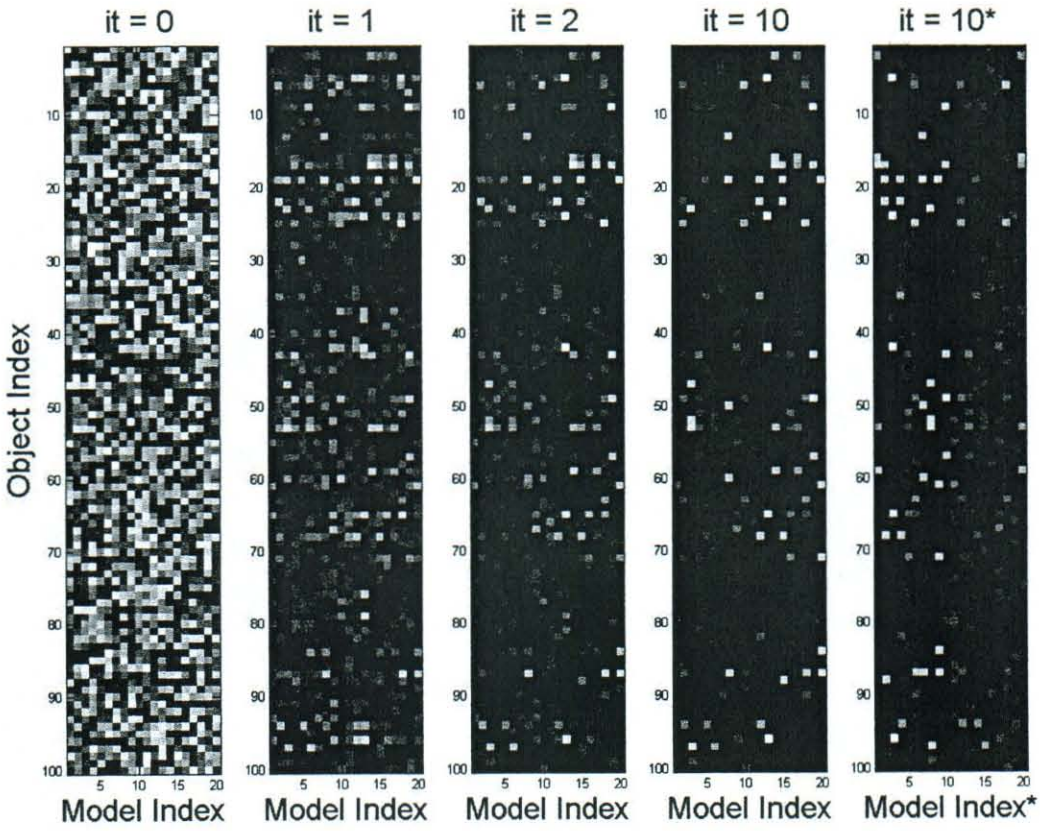

Figure 4

Iterations of the learning algorithm with random initialization mode and N_p=12. Each image displays the values of probabilities for each of the 20 models. As the iterations progress the probabilities of the essential object increase and the probabilities of the other objects decrease. *The last image repeats the 10^{th} iteration with the models rearranged so that the first 10 models match real situations. The right hand side of the image contains no bright spots, and correspond to random noise models.

The initial state of all models assigns all objects to all situational models with significant probabilities and there are many bright spots. These are the initial vague models. After several iterations the probabilities of the essential objects become bright and the probabilities of the random objects become gray or black. There are 10 out of 20 models that exhibit bright pixels, and the other 10 models exhibit a more or less uniform gray color. The last panel in Figure 4 rearranges the models to place the 10 brightest models on the left hand side. These models correspond to the 10 most important situations. A direct check confirms that the indices of the brightest pixels correspond to the indices in Table 1. We

emphasize once more that the DL iterative process that progresses from vague to crisp (starting from its initial vague state) **avoids local maxima** that plagued the previous state of the art algorithms.

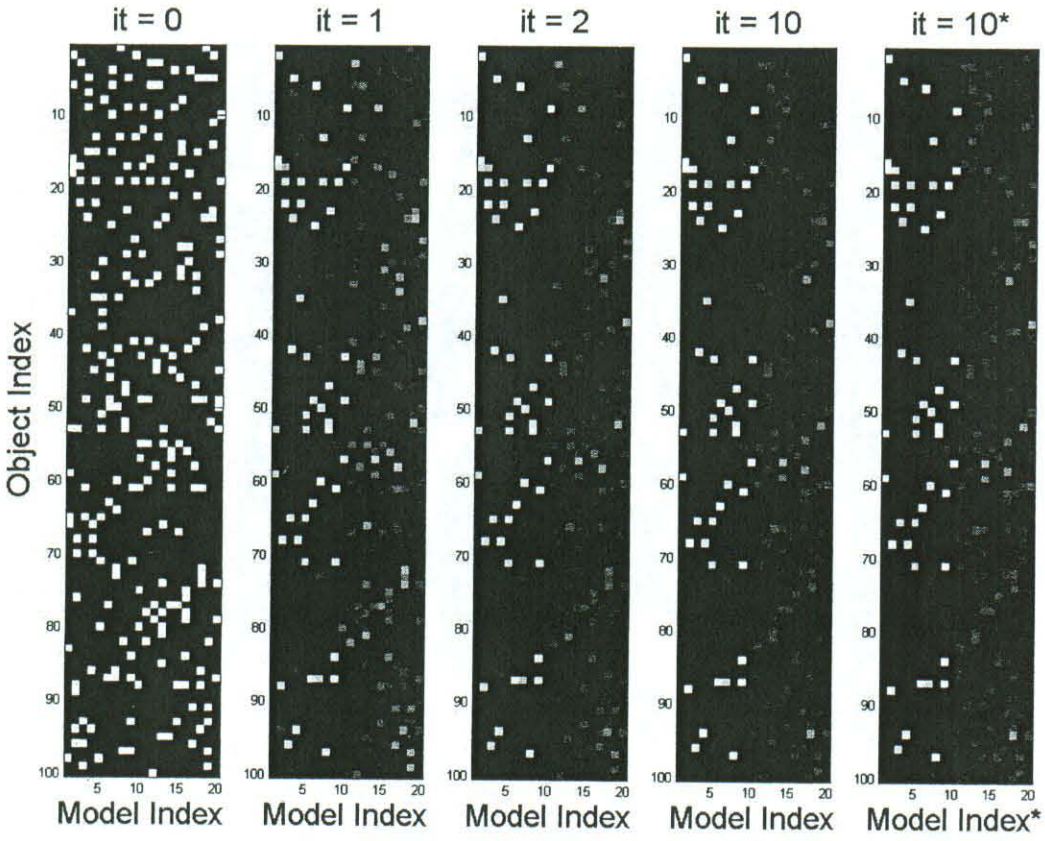

Figure 5

Iterations of the learning algorithm with partially supervised initialization and N_p=12. Each image displays the values of probabilities for each of the 20 models. As the iterations progress the essential object probabilities increase and the other object probabilities decrease. *The last image repeats the 10^{th} iteration with the models rearranged so that the first 10 models give the best match to real situations. The right hand side of the image contains no bright spots, and correspond to random noise models. Note that unlike Figure 4, rearranging the models did not change the image in this case, since model identities are "predefined" by initialization.

The initialization of setting the models to vague states with random initial probabilities corresponds to unsupervised learning. In reality humans are told about the situation in at least some instances. A child coming to a supermarket for the first time may be told that this is a supermarket. However the next time she is in a different supermarket she may not be told about it. One way of modeling this type of

learning is by using one of the situation samples for model initialization. We call this the partially supervised initialization mode. In this mode the initial probabilities of the objects that are present in the selected sample are set to high values, usually between 0.7 and 0.8. The other object probabilities are set to low values close to 0.1.

We have conducted four experiments with the parameters given in Table 2. In all of them, the algorithm was stopped after 10 iterations. Experiment 3 was described above and illustrated in Figures 2, 3 and 4.

Table 2

	Experiment 1	Experiment 2	Experiment 3	Experiment 4
N	500	500	500	500
D_x	100	100	100	100
N_p	10	10	12	12
N_s	5	5	5	5
H_s	10	10	10	10
H	21	21	21	21
D_s	100	100	100	100
Initialization	Random	Partially supervised	Random	Partially Supervised

Figure 5 illustrates the iterations in experiment 4 when using partial supervision. The first subplot (it=0) illustrates the initialization. The initial models already contain high probability values for the objects essential to corresponding situations and some random objects. As the iterations progress the probabilities of non-essential objects vanish but the essential objects maintain high probability values. After 10 iterations the first 10 models converge to the true situations. The rest of the models no longer contain any bright spots corresponding to high probabilities. Rearrangement of the models does not change the picture since in this case the initialization has predetermined which model will converge to which situation. The results obtained for experiments 1 and 2 are similar to those of experiments 3 and 4. In our case the partially supervised mode resulted in extremely fast learning and the brightness of the pixels changes very little after the second iteration.

We compute the pairwise Euclidean distance between the final 20 models and the true situations. Then we select the best match for each of the true situations. This procedure identifies which of the models correspond to a true situation and we also use it to evaluate the error for each iteration. Figure 6 shows the changes in the sum squared errors and the total similarity during the operation of NMF algorithm in experiments 1 and 3. The sum squared error is computed based on the top 10 matches between the models and the true situations described above. The total similarity is estimated using equation (4). As expected, the supervised case results in better performance since the initial conditions are closer to the solution.

Figure 7 shows the sum squared errors and the total similarity for experiments 3 and 4. The plots here are very similar to Figure 6. The only difference between the two input sets is the number of extra non-essential objects in each situation, which is interpreted as the amount of noise in the data. The difference in the final error between the supervised and unsupervised cases is larger with increased noise levels. The algorithm achieved the solution in both cases.

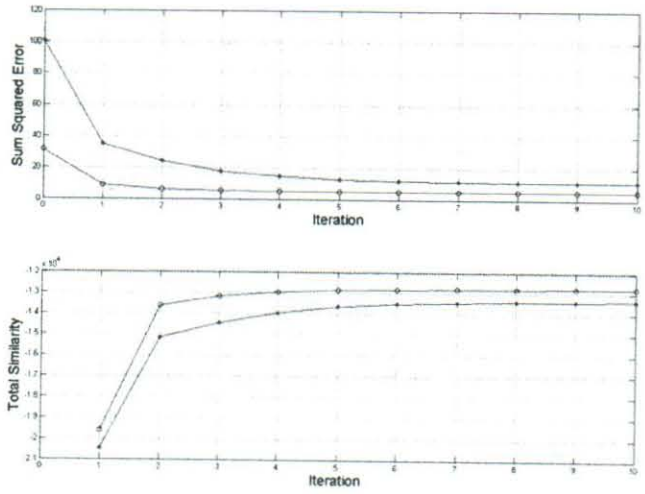

Figure 6
The sum squared error of the top 10 models (top) and the total similarity (bottom) for the case of 10 objects. Circles show the supervised case and stars show the unsupervised case.

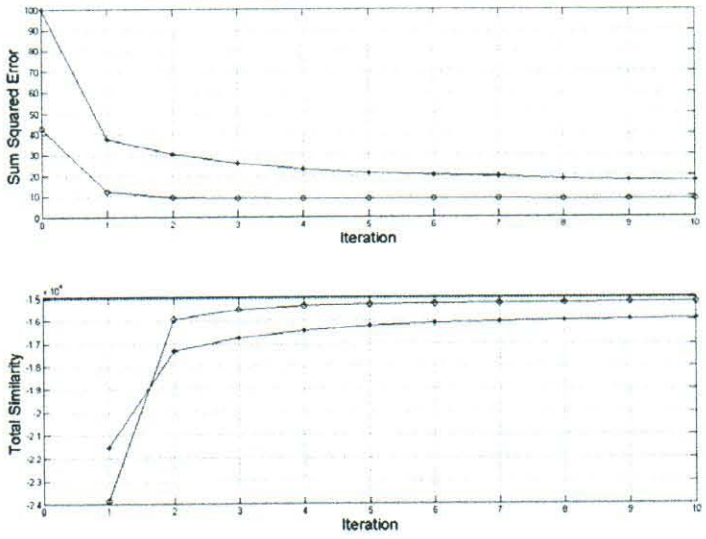

Figure 7

The sum squared error of the top 10 models (top) and the total similarity (bottom) for the case of 12 objects. Circles show the supervised case and stars show the unsupervised case.

The high similarities and low errors that occur after only a few iterations in Figures 6 and 7 correspond to the last images in Figures 4 and 5. The first 10 models in Figures 4 and 5 have the bright spots in exactly the same locations as the horizontal lines in the original data given in Figure 2.

5. Computational Complexity

The computational complexity of the algorithm given by equation (10) can be estimated in terms of the number of data inputs N , the number of models H, the dimensionality of the data D_x and the number of model parameters which we will denote by D_s. D_s is the length of vector $\boldsymbol{S_h}$. The complexity is given in as

$$C_{NMF} = C_{max} \cdot O(NHD_xD_s) \quad (14)$$

Equation (14) is obtained by considering the algorithm in equation (10). The computation of f(h|n) requires H evaluations of the similarity for each of the data inputs yielding an order of NH evaluations. We assume that the cost of each similarity evaluation is proportional to the size of the vector x_n, which is D_x. The parameter estimation step requires N evaluations of the derivative of log-similarity with respect to D_s parameters for each of the H models. We again assume that the cost of evaluation of the derivative is proportional to D_x. Therefore the total cost of one iteration is proportional to the product of the four numbers. The iteration is repeated until it converges. The number of iterations usually is not large and it is accounted for in the constant C_{max}. On the other hand, the problem of finding the best match between N data inputs and H models requires in general an exponential number of steps given as

$$C_{exp} = O(NH)H^N \quad (15)$$

This is the number of evaluations of the total similarity that has to be performed with all possible models and data assignments. Each evaluation requires an order of NH operations and the number of assignments is exponential.

The computational complexity of the problem of matching N = 500 to H=21 models is combinatorial and is given by equation (15) evaluating to $O(21^{500})$, which is a huge number. However the NMF converged in 10 iterations with the cost of each iteration given by $O(500*21*100*100)=O(10^8)$.

6. Discussion and Conclusions

This paper has outlined steps toward the learning and recognition of situations, a problem that has remained unsolved for decades due to combinatorial complexity of existing algorithms. This problem is closely related to another unsolved problem, situational awareness, which is defined as "the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" (Endsley, 1995). This ability is essential for a variety of military and civilian applications. This solution for situational learning opens a possibility for solving situational awareness. In Figure 8 we illustrate an architecture for solving this more complex problem.

Figure 8 is comparable with the classical data fusion process where a similar hierarchy is described as part of the military threat assessment framework (Hall and Llinas, 2001). In the presence of multiple objects and clutter, fusion remained an unsolved problem until an NMF-DL solution was developed (Deming & Perlovsky, 2007). In Figure 8 the bottom layer is concerned with recognizing objects and their movement over time. The next layer recognizes the situation formed by the presence of various objects. Observing the change of situations over time can allow the system to learn to predict the future developments in the environment. Then the decision making layer alters the state of the system in response to the perceived and predicted situation. There are feedback loops connecting all the layers necessary for the time domain processing.

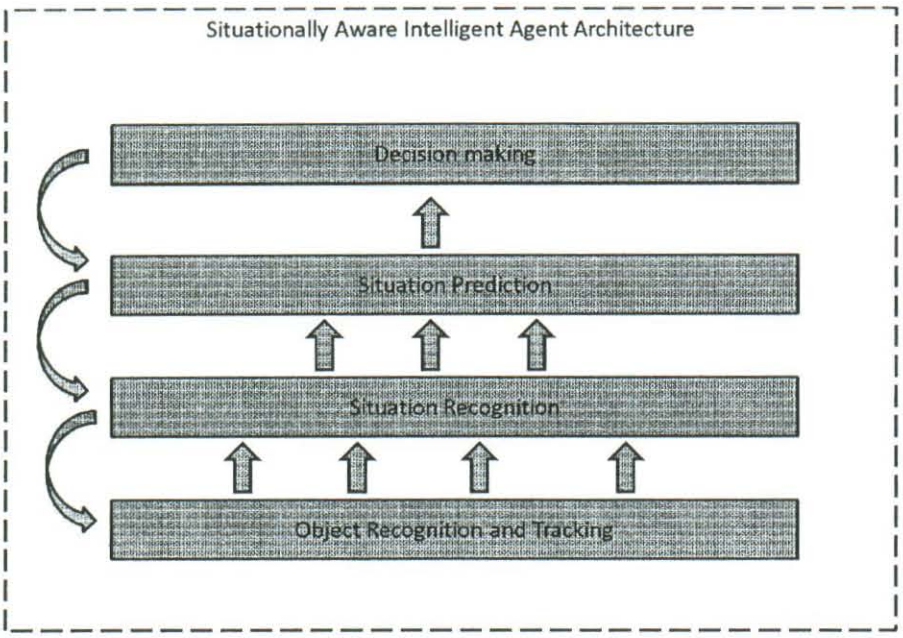

Figure 8

The layered architecture for an intelligent agent with each layer implementing the NMF network shown in Figure 1.

The strength of the NMF-DL approach is that every level of this hierarchy can be implemented using the same computational framework. The previous work mentioned in the introduction was concerned with the object recognition and tracking layer. There the input data correspond to sequences of sensor images and the models correspond to shapes and trajectories of objects. This work illustrates how the same computational framework is employed in the second layer.

The agent often receives clues from the environment that help it learn very fast. We show how such clues can be seamlessly incorporated into the framework as part of model initialization. This is illustrated by comparing the unsupervised and the partially supervised modes of operation. In our experiments NMF successfully solves the problem in either mode. However the partially supervised

mode resulted in faster learning and a better final result. We expect that in significantly more complex cases some form of partial supervision may become necessary.

In real life multiple situations are often perceived without clear breaks. In parallel, a language stream is perceived using words to label situations. Usually it is not obvious which label-word refers to which situation. The case of partial supervision is sometimes modeled as cross-situational learning (Fontanari et al 2009). In the future we will address situational learning in parallel with proper associations among word-labels and situations.

Our future work will involve the necessary steps to expand and merge the current applications of NMF into a single system shown in Figure 8. The issues of integration between layers need to be solved and the effect of feedback between the layers needs to be investigated.

This single layer study needs to be expanded to a larger data set size by increasing the number of situations and objects. As our limited experiments have demonstrated, the performance of the algorithm depends not only on the data input size but also on the relative sizes of the total set of objects, essential objects and noise objects in the data.

Let us now outline our future research directions leading to modeling the capabilities of the human mind. This research will include relations between objects. In the fully developed approach, relations are no different mathematically from objects when using the following method. Relations and markers, indicating which relations and objects are involved, can be included among other objects in the current method. Another direction toward modeling the mind would be to extend the developed approach to a multi-layer hierarchical system. At each level in a hierarchical system the output to the next higher level is a set of signals produced by models identified, learned, or recognized at the given level. The more general and abstract higher-level concept-models at the next level are learned as combinations of the lower-level concept-models in the same way as we demonstrated learning of situations from objects. In

this way the hierarchical cognition of the mind can be modeled. Our approach could include language. Language and syntax are learned from surrounding language similarly to how concepts and relations are learned from perceptual signals. Joint evolution of interacting language and cognition would be modeled following Perlovsky 1997;2004;2005;2006d; 2009a; Fontanari and Perlovski 2007; 2008a;2008b. Next, this system would model emotions, following Tikhanoff at el, 2006; Perlovsky 2002;2006; 2007a;2007b;2007c;2008;2009b;2009c; Levine & Perlovsky 2008. By considering multi-agent interaction of systems consisting of such conceptual-emotional intelligent agents communicating using language, we would be able to model human cultures (Perlovsky 2009b;c).

ACKNOWLEDGMENT

R.I. would like to express thanks to Dr. Ross Deming and Dr. Rob Linnehan for insightful discussions. The authors would like to thank R. Brockett, R. Deming, D. Levine, R. Kozma, and B. Weijers for discussions, help, and advice, and to AFOSR for supporting part of this research under the Lab. Task 05SNO2COR, PM Dr. Jon Sjogren, and Lab. Task 05SNO0COR, PM Dr. Doug Cochran, as well as the anonymous reviewers for numerous suggestions, which helped to significantly improve this work.

REFERENCES

Aristotle. (IV BCE / 1995) *"Rhetoric to Alexander."* In The Complete Works of Aristotle, Ed. J.Barnes, Princeton, NJ: Princeton Univ. Press, IV BCE/1995.

Barrett, L. F. & Bar, M. (2009). *See it with feeling: Affective predictions* in the human brain Royal Society Phil Trans B, 364, 1325-1334.

Bar, M., Kassam, K. S., Ghuman, A. S., Boshyan, J., Schmid, A. M., Dale, A. M., Hämäläinen, M. S., Marinkovic, K., Schacter, D. L., Rosen, B. R. & Halgren, E. (2006). *Top-down facilitation of visual recognition*. Proceedings of the National Academy of Sciences USA, 103, 449-54.

Bellman R. E. (1961) *Adaptive Control Processes*. Princeton, NJ: Princeton University Press.

Bishop, C. (1996). *Neural networks for Pattern Recognition*, Oxford University Press, USA, 1st edition.

Carpenter, G. A. and Grossberg, S. (1987). *A massively parallel architecture for a self-organizing neural pattern recognition machine*, Computer Vision, Graphics, and Image Processing, 37, 54-115.

Cherkassky V. and Mulier P. (2007), *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, Inc., Hoboken, New Jersey

Deming, R.W., (1998). *Automatic buried mine detection using the maximum likelihood adaptive neural system (MLANS).* Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings of the 1998 IEEE International Symposium on 14-17 Sept. 1998 Page(s):428 – 433

Deming, R.; Schindler, J.; Perlovsky, L. (2007), *Concurrent Tracking and Detection of Slowly Moving Targets using Dynamic Logic*. Integration of Knowledge Intensive Multi-Agent Systems, 2007. KIMAS 2007. International Conference on April 30 2007-May 3 2007 Page(s):7 – 11

Deming, R.W. and Perlovsky, L.I. (2007). *Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors*, Information Fusion, 8, pp.316-330.

Duda R., Hart P., and Stork D. (2000), *Pattern Classification*. Wiley-Interscience, 2nd edition

Endsley, M. R. (1995). *Toward a theory of situation awareness in dynamic systems*. Human Factors 37(1), 32-64.

Fontanari, J.F. and Perlovsky, L.I. (2007). *Evolving Compositionality in Evolutionary Language Games*. IEEE Transactions on Evolutionary Computations, 11(6), pp. 758-769;

Fontanari, J.F. and Perlovsky, L.I. (2008a). *How language can help discrimination in the Neural Modeling Fields framework*. Neural Networks, 21(2-3), pp. 250–256

Fontanari, J.F. and Perlovsky, L.I. (2008b). *A game theoretical approach to the evolution of structured communication codes*, Theory in Biosciences, 127, pp.205-214.

Fontanari J.F., Tikhanoff V., Cangelosi A., Perlovsky L. (2009). *A cross-situational algorithm for learning a lexicon using neural modeling fields.* IJCNN-09 International Joint Conference on Neural Networks, Atlanta, June 2009.

Grossberg S. (1982). *Studies of Mind and Brain*. Dordrecht, Holland: D. Reidel Publishing Co.

Hall, D. L. and Llinas J. (2001). *Handbook of multisensory data fusion*, CRC press LLC

Haykin, S (1999). *Neural Networks: a comprehensive foundation*. Pearson Education, Inc.

Ilin, R.; Kozma, R.; Werbos, P.J. (2008). *Beyond Feedforward Models Trained by Backpropagation: A Practical Training Tool for a More Efficient Universal Approximator.* , IEEE Transactions on Neural Networks, Volume 19, Issue 6, June 2008 Page(s):929 – 937

Kay, Steven M. (1993). *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall.

Kosslyn S. M. (1994). *Image and Brain*. Cambridge, MA: MIT Press.

Kozma, R.; Deming, R.W.; Perlovsky, L.I. (2007). *Estimation of Propagating Phase Transients in EEG Data - Application of Dynamic Logic Neural Modeling Approach Neural Networks.* International Joint Conference on Neural Networks 12-17 Aug. 2007 Page(s):2341 – 2345

Minsky, M.L. (1975). *A Framework for Representing Knowlege*. In The Psychology of Computer Vision, ed. P. H. Whinston, McGraw-Hill Book, New York.

Mitchell, T.M, (1997), *Machine Learning*, WCB/McGraw-Hill

LeCun Y., Matan O., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D. and Baird H. S.(1990), Handwritten Zip Code Recognition with Multilayer Networks, in IAPR (Eds), Proc. of the International Conference on Pattern Recognition, II:35-40, IEEE, Atlantic City, invited paper

Levine, D.S., Perlovsky, L.I. (2008). *Neuroscientific Insights on Biblical Myths: Simplifying Heuristics versus Careful Thinking: Scientific Analysis of Millennial Spiritual Issues*. Zygon, Journal of Science and Religion, 43(4), 797-821.

Linnehan, R.; Schindler, J.; Brady, D.; Kozma, R.; Deming, R.; Perlovsky, L. (2007), *Dynamic Logic Applied to SAR Data for Parameter Estimation Behind Walls*, Radar Conference, 2007 IEEE 17-20 April 2007 Page(s):850 – 855

Perlovsky L. I. (1987). *Multiple Sensor Fusion and Neural Networks*. DARPA Neural Network Study, MIT/Lincoln Laboratory, Lexington, MA.

Perlovsky L. I. (1996), *Gödel Theorem and Semiotics*. In Proc. Conference on Intelligent Systems and Semiotics '96. Gaithersburg, MD, vol.2, pp. 14-18.

Perlovsky, L.I. (1998a). *Conundrum of Combinatorial Complexity*. IEEE Trans. PAMI, 20(6) pp. 666-670.

Perlovsky, L.I., Webb, V.H., Bradley, S.R. & Hansen, C.A. (1998b). *Improved ROTHR Detection and Tracking Using MLANS* . AGU Radio Science, 33(4), pp.1034-44.

Perlovsky, L. I. & McManus, M. M. (1991). *Maximum Likelihood Neural Networks for Sensor Fusion and Adaptive Classification*. Neural Networks, 4(1), 89-102.

Perlovsky, L.I. (1997). *Physical Concepts of Intellect*. Proceedings of Russian Academy of Sciences, **354**(3), pp. 320-323.

Perlovsky, L.I. (2001). *Neural Networks and Intellect: using model based concepts*. New York: Oxford University Press.

Perlovsky, L.I. (2002). *Physical Theory of Information Processing in the Mind: concepts and emotions*. SEED On Line Journal, 2002 2(2), pp. 36-54.

Perlovsky, L.I. (2004). *Integrating Language and Cognition*. IEEE Connections, Feature Article, 2(2), pp. 8-12.

Perlovsky, L.I. (2005). *Evolving Agents: Communication and Cognition*. Chapter in Autonomous Intelligent Systems: Agents and Data Mining. Eds: V. Gorodetsky, J. Liu, V.A. Skormin. Springer-Verlag GmbH, pp.37-49

Perlovsky, L. I. (2006a). *Toward physics of the mind: concepts, emotions, consciousness, and symbols*. Physics of Life Reviews 3, 23-55.

Perlovsky L.I. (2006b), *Modeling Field Theory of Higher Cognitive Functions*. In Artificial Cognition Systems, Eds. A. Loula, R. Gudwin and J. Queiroz. Hershey, PA: Idea Group, pp. 64-105.

Perlovsky, L.I. (2006c). *Fuzzy Dynamic Logic*. New Math. and Natural Computation, 2(1), pp.43-55.

Perlovsky, L.I. (2006d). *Symbols: Integrated Cognition and Language*. Chapter in Semiotics and Intelligent Systems Development. Eds. R. Gudwin, J. Queiroz. Idea Group, Hershey, PA, pp.121-151.

Perlovsky, L.I. (2007a). *Evolution of Languages, Consciousness, and Cultures*. IEEE Computational Intelligence Magazine, 2(3), pp.25-39

Perlovsky, L.I. (2007b). Cognitive high level information fusion, Information Sciences, 177, pp. 2099-2118.

Perlovsky, L.I. (2007c). *Neural Dynamic Logic of Consciousness: the Knowledge Instinct*. Chapter in Neurodynamics of Higher-Level Cognition and Consciousness, Eds. Perlovsky, L.I., Kozma, R. Springer Verlag, Heidelberg, Germany.

Perlovsky, L.I., Deming, R.W. (2007). *Neural Networks for Improved Tracking*, IEEE Transactions on Neural Networks , Volume 18, Issue 6, Nov. 2007 Page(s):1854 – 1857

Perlovsky L. I., Kozma R. (2007) Eds. *Neurodynamics of Higher-Level Cognition and Consciousness*. Heidelberg, Germany: Springer-Verlag.

Perlovsky, L.I. (2008). *Music and Consciousness*, Leonardo, Journal of Arts, Sciences and Technology, 41(4), pp.420-421.

Perlovsky, L.I. (2009a). *Language and Cognition*. Neural Networks, 22(3), 247-257.

Perlovsky, L.I. (2009b). *Language and Emotions: Emotional Sapir-Whorf Hypothesis*. Neural Networks, 22(5-6), 518-526.

Perlovsky, L.I. (2009c). *Origins and role of musical emotions in evolution of the mind and consciousness*. Phys. Life Rev. Submitted

Perlovsky, L.I. (2009d). *'Vague-to-Crisp' Neural Mechanism of Perception*. IEEE Trans. Neural Networks, 20(8), 1363-1367.

Perlovsky, L.I. (2009e). *Neural Mechanisms of the Mind, Aristotle, Zadeh, & fMRI*, IEEE Trans. Neural Networks, submitted.

Singer, R.A., Sea, R.G. and Housewright, R.B. (1974). *Derivation and Evaluation of Improved Tracking Filters for Use in Dense Multitarget Environments*, IEEE Transactions on Information Theory, IT-20, pp. 423-432

Schacter D. L. (1987). *Implicit memory: History and current status*. Journal of Experimental Psychology: Learning, Memory, and Cognition, vol. 13, pp. 501-518.

Tikhanoff. V., Fontanari, J. F., Cangelosi, A. & Perlovsky, L. I. (2006). *Language and cognition integration through modeling field theory: category formation for symbol grounding*. In Book Series in Computer Science, v. 4131, Heidelberg: Springer.

Winston, P.H. (1984). *Artificial Intelligence*. 2nd edition, Addison-Wesley. Reading, MA.