Codification of information with Hebbian ensemble structures in neural networks

Ernst Kussul, Tatiana Baidyk, Enrique Cabello Pardos, Cristina Conde Vilda

Abstract**—**An ensemble of neurons can correspond to a property, a characteristic, or any information identity. The ensembles can be intersected and can have common neurons that can be used in future as an evaluation level of association between the information identities. On the base of this type of information presentation it is possible to create a new model of associative memory. So, it is very important to investigate the problem of information capacity of neural network (NN) with neural ensembles.

Keywords**—** Associative memory, information capacity, neural network, neural ensemble.

I. Introduction

THE neural network with ensembles' structures may be used for semi supervised training. This type of training allows minimizing the work in preparing the data for training process. For example, in the task of information presentation in knowledge systems and pattern recognition we can use neural networks with ensembles structures.

At the UNAM, National Autonomous University of Mexico, we work in the areas of image recognition and neural computing. In recent years was done preliminary work on the research of neural networks with structures of ensembles. To generate the ensemble we use random permutations of the code [1]. This procedure allows us to store not only information about the parameters, but also about their numeric values.

The first neural network models used one neuron at the output layer to represent one information unit [2]. Many theories suggest neural ensembles (a set of neurons instead of one) represent the information in the brain [3] that gives the explanation of the associative memories properties. For example, the Hopfield neural network [4], [5] stores a number of neural ensembles. The concept of neural ensemble was taken from the biology and was proposed by Donald Hebb [3] in 1949. According with his theory every feature is coded with a set of neurons. Between neurons, there are connections. The rule of their activation is the following: "what is activated together, is connected together". Every ensemble can represent one "idea" or "concept".

If one ensemble is activated, due to the common neurons, another ensemble can be excited. The activity propagation between ensembles corresponds to associations between the concepts. The auto associative memories have larger storage capacities than hetero associative memories [6].

These studies showed the importance of information encoding methods for this type of neural networks. It is necessary to develop methods to obtain large information capacity of neural networks.

Our main task is to develop and study these methods in new tasks.

The aim of the research is to generate new knowledge about the area of neural networks with ensembles structure for patterns recognition. The generation of this knowledge will allow us to apply the semi supervised training to pattern recognition task.

In our previous works we used the name "Ensemble Neural Network" to denote the artificial neural network that contains Hebbian neural ensembles, i.e. subsets of the neurons that have connections between each other with synaptic weights much higher than average value of synaptic weights in the neural network.

At present the term "Ensemble Neural Network" frequently is used for a group of different neural networks that collaborates to solve common task (for example, pattern recognition). For this reason we will call our networks "Hebbian Ensemble Neural Networks". This type of neural networks was proposed by D.O.Hebb in [3]. D.O.Hebb supposed that each concept may be presented in neural network not by single neuron (as output neuron that corresponds to recognized class in neural classifier) but by a subset of neurons that forms neural ensemble. When a part of Hebbian neural ensemble is excited, the excitation spreads on the whole ensemble. Though Hebbian ensemble consists of many neurons it behaves as a single information unit. Hebbian ensembles can have intersections. So each neuron can belong to many ensembles. This peculiarity generates the whole network excitation problem when initially only one ensemble is excited. This problem can be solved by simulation of attention mechanisms [7], [8].

Many authors (e.g., [9] - [12]) analyzed the properties of Hebbian ensembles using mathematical methods or computer simulations, but not so much attention was paid to practical

E. Kussul is with the CCADET, UNAM , Circuito Exterior s/n, Ciudad Universitaria, 04510, Mexico city, D.F., Mexico (phone: 52 55 56228602; fax: 52 56 55500654; e-mail: ekussul@ unam.mx).

T. Baidyk is with the CCADET, UNAM, 04510, Mexico city, D.F., Mexico. (ekussul@unam.mx).

E. Cabello Pardos is with the University Rey Juan Carlos, Madrid, Spain.

C. Conde Vilda is with the University Rey Juan Carlos, Madrid, Spain.

This work was supported partially by projects PAPIIT IN110510-3, PAPIIT IN119610.

applications of this paradigm. In this article we propose to apply Hebbian ensemble neural network for semisupervised learning in the image recognition tasks.

Before description of our investigation we want to demonstrate one example of neural ensembles application.

The idea that an image is presented by not just one neuron but by an ensemble is fruitful. Any concept may have different meanings. Its content may vary depending on the context. Only the central core of the concept whose activity may dominate in the system as a whole can be almost unchangeable. The possible presentation of an image or concept with one neuron deprives this concept of its features and characteristics. Neuron ensemble makes it possible to present a concept or image description with all features and characteristics. These features can be influenced by the context of the situation where the concept is used. For example, we create the model of the concept "building" (Fig.1). We can observe the building from different positions. A perceived object (building) consists of a number of perceptual elements. We can see many windows or a door.

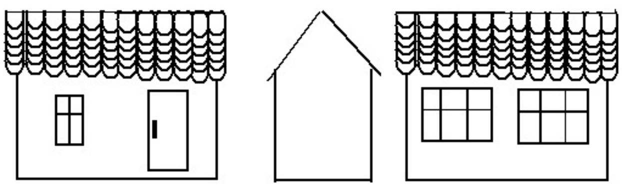

Fig.1. Model of the concept "building"

However from different positions we can see the walls and the roof of this building. In an ensemble that represents the model of the concept "building," a set of neurons corresponds to the walls, other neurons correspond to windows, and others correspond to the white color of the walls, and so on. The more frequently perceived features of the building form the core of the ensemble, and rare features create a fringe of the ensemble (Fig.2).

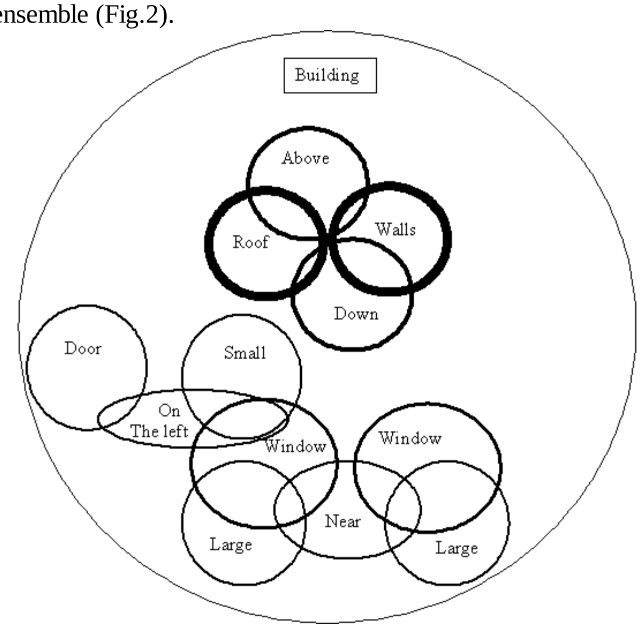

Fig.2. Neural presentation of a model "building"

Due to the fringe of the ensemble, different concepts may have a large number of associations with other concepts. The notion "Fringe" was introduced by Hebb to explain how associations are provided. Different circumstances lead to varying fringe activity. If it is day, the white colour of the building will be observed, and in the model the neuron set that corresponds to colour will be excited. "Core" is the most connected part of the ensemble. In our example, the core will be neurons that correspond to walls and windows. The conceptual activity that can be aroused with limited stimulation must have its organized core, but it may also have a fringe content, or meaning, that varies with the circumstances of arousal.

There are different approaches to neural network with ensemble presentation. There were developed different paradigms of ensemble neural networks. Among them the most popularity has Hopfield neural network [4], [5].

Hopfield described his neural networks in 1982 [4]. The structure of neural network has the neural processing elements. The output of every processing element is the inputs of other neural processing elements. The transfer function of every processing element is:

$$X_i^{t+1} = \begin{cases} 1, & if \quad \sum_{j=1}^{n} w_{ij} X_j^t > T_i, \\ X_j^t, & if \quad \sum_{j=1}^{n} w_{ij} X_j^t = T_i, \\ -1, & if \quad \sum_{j=1}^{n} w_{ij} X_j^t < T_i, \end{cases} \tag{1}$$

for $i = 1,\dots, n$,

where w_{ij} is the weight of the input with the restrictions $w_{ij} = w_{ji}$ and $w_{ii} = 0$, T_i is the threshold.

The behavior of the Hopfield network is organized in such a way to minimize the energy function. No matter what is its initial state, the Hopfield network always converges to a stable state in a finite number of processing element update steps.

There are many versions of the Hopfield network which are used in different applications. For example, the Hopfield network is a base of a hybrid Hopfield network-simulated annealing algorithm used for frequency assignment in satellite communications systems [13]. They use a fast digital Hopfield neural network which manages the problem constraints hybridized with a simulated annealing which improves the quality of the solutions obtained. Another example is Hopfield neural network application for general predictive control [14]. In this case the Hopfield neural network is used to solve quadratic optimizing problem. Existing predictive control method is very complex and time consuming. With this proposition of Hopfield neural network application and development of neural network chip, the method has a promising future in industry. The Hopfield neural network is used in information retrieval system too [15]. In recent years, with the rapid development of Internet and easy access to a large amount of information on it, information retrieval has become more indispensable for picking out useful data from the massive resources. With the heuristic function of Hopfield network this model is used in query expansion, and therefore can solve the problems of information overload and word mismatch to some extent.

So, neural network with ensembles can be used to resolve many practical tasks.

II. Information Coding

Many different methods of input information coding for the neural networks are proposed and developed [16], [17], [18] which is connected with the great variety of the neural networks.

In this paragraph we consider the so-called stochastic or local connected coding, and also shift coding [19]. These methods of coding relate to the distributed coding with the low level of activity (sparse coding) [20], [21]. This means that in the neural network any coded object is represented not with one neuron but with many neurons. Let us term these neurons as neuron ensemble. The low level of activity means that the number of neurons in the neuron ensemble is much less than the total number of neurons in the neural network.

The special feature of local connected coding consists in the fact that the different numerical values of the parameters are represented by different binary stochastic vectors. For this purpose all the numerical values are presented as integer numbers in the specified range.

$$X = \{x_1, x_2, \dots, x_q\}\ , \tag{2}$$

where

$$x_i - x_{i-1} = 1.$$

Let the minimal integer is X_{min} and the maximum integer is X_{max} and we have a parameter y that is limited by the values Y_{min} and Y_{max}. Here y, Y_{min} and Y_{max} can be real numbers. To present them as the integer number in the range (X_{min}, X_{max}) it is necessary to calculate the value x^*:

$$x^* = X_{min} + ((y - Y_{min}) / (Y_{max} - Y_{min})) * (X_{max} - X_{min}), \tag{3}$$

and then round x^* up to the nearest integer value x. The value x will correspond to the numerical value of the parameter in the range (X_{min}, X_{max}).

Each feature must be represented in the form of the n-dimensional binary vector which has m components equal "1". Here n is the quantity of the neurons in the neural network, m is the quantity of neurons in a neuron ensemble, moreover $m<<n$. Such binary vectors we call input masks. To create the input mask we set all the components of binary code vector to "0" and then randomly select m components and change their values to "1". In order to feed mask to the input of the neural network it is necessary to excite the neurons of this network whose numbers coincide with the numbers of the positions of unit elements in the binary vector-mask. After this the network will have active neurons that correspond to values "1" in the code vector.

Randomly generated input mask is the binary code vector:

$$V^1 = \{v_1^{\ 1}, v_2^{\ 1}, \dots, v_n^{\ 1}\}. \tag{4}$$

This code vector corresponds to the value of parameter y equal to x_1. To obtain input mask for parameter value x_2, we do the following operations:

1. Randomly select k elements from m components of code vector that had value "1" and change each value to "0" ($k << m$).
2. Randomly select k elements from (n - m) components of code vector that had value "0" and change each value to "1".

After these operations we obtain the input mask for the value x_2. Repeating this procedure, we obtain the input masks for all parameter values.

Let us consider the function $I(V^i, V^j)$. This function equals to the number of components r that fit the condition:

$$(v_r^{\ i} = 1)\ \&\&\ (v_r^{\ j} = 1) \tag{5}$$

Locally connected coding has the following property: if we create the code vectors for two different parameter values, the function I will be large if the difference between parameter values is small, and it will be smaller when the difference will increase. This property is very important to create the ensembles that form the core from code vectors of the similar objects.

The shift coding was the next step in information coding. In the shift coding just as in the local connected coding the neural code of the feature name is the random binary vector. This code is stored in the memory of computer and is extracted each time when it is necessary to do coding or decoding of the feature. Let us term this vector the feature mask. The position of feature in the image is coded by the appropriate shift of the feature mask.

Next chapters are dedicated to our neural network presentation and the ensemble formation in neural network.

III. Neural Networks with Ensembles

As it was mentioned, a neural network is used as an autoassociative memory [22]. It consists of N neurons with full connectivity, in other words, each neuron is connected with all neurons, including itself (in contrast with Hopfield neural network). The example of neural network is presented in Fig.3.

The response E of the neural network to the input vector X is calculated by the equation:

$$E = S(X^T W) \tag{6}$$

where X^T is the transpose input vector X, whose elements x_i =

$[0, 1]$ for $i = 1, 2, \ldots, N$, E is the output vector, whose elements $e_i = [0, 1]$ for $i = 1, 2, \ldots, N$, and the function $S()$ is the threshold function of the neurons:

$$S(y_i) = \begin{cases} 1, & if\ y_i \geq \Theta_i \\ 0, & if\ y_i < \Theta_i \end{cases}, \qquad (7)$$

where Θ_i is the threshold value for neuron y_i.

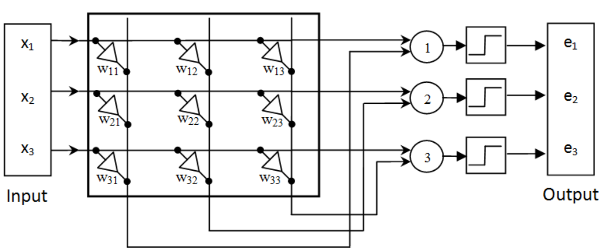

Fig.3. Three-neuron network

IV. Ensemble Creation

The neurons of one ensemble are randomly chosen. The size of ensemble is M, which specifies how many neurons will be in one ensemble. We also define the number of ensembles K to be memorized by the neural network. The ensemble is presented as a vector (these two terms will be used as equals) $X = X_1, X_2, \ldots, X_K$, whose internal product $X_j \cdot X_j = M$. The internal product is used and understood as the measurement of similarity between vectors.

Now the interconnection weights among the ensemble neurons are formed using the Hebbian training rule:

$$(w_{ij})_{t+1} = (w_{ij})_t + 1, if\,(x_i = 1 \wedge x_j = 1) \qquad (8)$$

where $(w_{ij})_t$ is the synaptic weight between neurons x_i and x_j at time t, $(w_{ij})_{t+1}$ is the synaptic weight at time $(t+1)$ and $\wedge$ denotes the logical disjunction.

The training process can be represented using the external product of each ensemble with itself:

$$W = \sum_{k=1}^{K} (W + X_k X_k^T) \qquad (9)$$

where W is the matrix of dimension (N x N) initialized with zeros.

V. Information Capacity

To investigate the information capacity of the neural network, the next procedure is implemented.

It is added noise to every ensemble from set X in such way that the half of elements in each vector changes, but without altering the ensembles size M.

Let us consider the example for a neural network size $N = 10$ with ensemble size $M = 4$, the vector to represent the ensemble is:

$X_1 = [1, 0, 0, 0, 1, 0, 1, 0, 1. 0]$,

the vector X_1 is changed with noise into the vector X_1*:

X_1*$= [0, 1, 0, 0, 1, 1, 0, 0, 1, 0]$.

In this process $M/2$ active neurons from vector X_1 were randomly changed and M/2 inactive neurons changed their value too, in order to keep constant the ensemble size M.

Once the original ensembles were altered with noise, we have the set $X^* = X_1^*, X_2^*, \ldots, X_K^*$.

Every vector is presented at the input of the neural network and the activity if neural network is recalculated R times (R is the number of cycles) in order to recover the original ensemble. This procedure is shown in Fig.4.

In Fig.4 the scheme of Hebbian ensemble neural network is presented. The scheme of the Hebbian ensemble neural network consists of input neurons, synaptic weight matrix and output neurons (Fig.4).

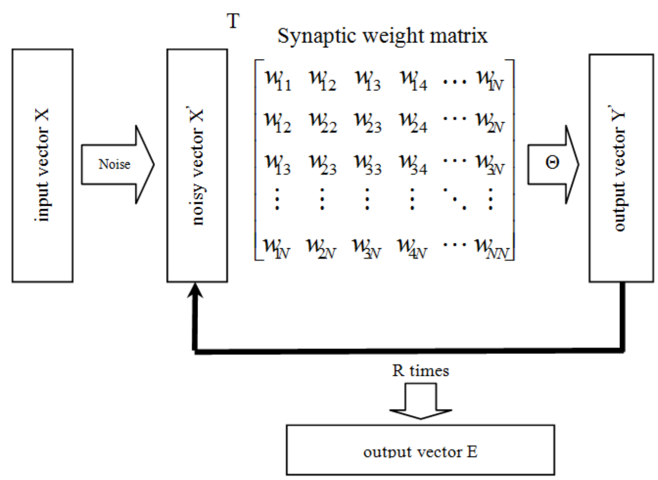

Fig.4. Diagram with R cycles

Input neurons are latches that store the value of input code vector. Output neurons sum up the signals of input neurons multiplied with synaptic weights and perform threshold function to produce binary output vector.

$$y_i = f\left(\sum_{j=1}^{N} w_{ji} \cdot x_j\right), \qquad (10)$$

where
$$f()=1, \text{ if } \sum_{j=1}^{N} w_{ji} \cdot x_j \geq \theta,$$
$$f()=0, \text{ if } \sum_{j=1}^{N} w_{ji} \cdot x_j < \theta,$$

where x_j is the binary output value of the j-th input neuron, y_i is the binary output value of the i-th, θ is the output neuron threshold. The threshold θ is the same for all output neurons. Its value is calculated by special algorithm in each cycle of neural network recalculation. The threshold is selected to achieve the number of active neurons close to the constant value m that is given for the neuron network as a parameter. As a rule the threshold value is larger for larger synaptic weights between ensemble neurons. The threshold value can be used to estimate "how strong" ensemble is excited at the moment.

In our investigation R=5 like optimal experimental value, where the output vector Y^* is fed back as the new input X. The value Θ shown in Fig.4 and mentioned in equations (7) and (10) is chosen in each calculation like the value that allows the activation of not more than M neurons in vector Y^*, because it is desirable to fulfill the condition $(E_i \text{x } E_i) < M$.

After R cycles we have the set of output vectors $E^*=E_1^*$, E_2^*, …, E_K^*. Each output vector E_k is compared with the original ensemble by internal product. If $(E_i \text{ x } X_i) \geq 0.9M$, then it is said that the ensemble has been recognized successfully, otherwise an error occurred, the ensemble was not recognized.

The objective of this methodology is to store the maximal number of ensembles K in the neural networks of size N, but allowing the recovering of at least 90% of each original ensemble of size M even though it was changed with noise (50%).

To estimate the storage capacity we wrote special program. With this program we evaluated the maximum number of correctly restored ensembles for different ensemble sizes. The maximum number of ensembles we obtained in cases when the number of retrieval errors less than 1%.

VI. Results of Neural Network Simulation

We obtained the following results for experiments with size of neural network N=28 000 neurons and ensemble size M=64. The number of ensembles that can be generated and then recuperated from noise was near 190,000. This information capacity demonstrates that the number of ensembles is more than number of neurons in the neural network.

For the size of neural network N=40 000 neurons we obtained the following results. In the experiments with ensemble size M=64 we obtained 410,000 ensembles.

The number of ensembles in the neural network depends on the size of ensemble. G.Palm and A.Knoblauch [22], [23] made theoretical estimations of optimal size of the neural ensembles and obtained the asymptotical value:

$$m = \log N / \log 2 . \qquad (11)$$

For our first case when N=28 000, we obtain m=14.8. The closest integer is 15. We did the experiments with ensemble size M=15. In this case for M=15 we obtained the number of ensembles of 12,000. At the same time for M=64 the number of ensembles was 190,000. G. Palm did not analyze the presence of noise in the input code. When we have noise it is necessary to increase the size of ensemble to obtain optimal storage capacity.

The next step in the investigation of information capacity will be investigation of the ensembles that have correlation of their codes. This type of ensembles appears when we start to codify real world objects, for example triplets of the letters that are used in handwritten text recognition.

VII. Ensemble Neural Application in FRAV2D

In this article we describe the application of Hebbian ensemble neural network for the problem of object detection in the image. We use FRAV2D image database that was collected and developed in Rey Juan Carlos University, Madrid, Spain [24], [25]. The FRAV2D image database contains 1696 images: 16 different images for each person, in total the database contain 106 people's images. 16 different images correspond to expressions of different emotional states, different head inclination to the left, to the right, and so on. One example of the FRAV2D image database is presented in Fig.5.

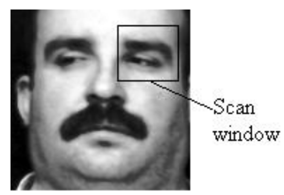

Fig.5. Scan window in image from FRAV2D

We suppose that Hebbian ensemble neural network will be capable to find the objects as eyes, nose, and mouth without preliminary marking them on the images.

Preliminary processing of the images includes correction of mean brightness and filtration with high spatial frequencies filters.

After preliminary processing it is possible to start the program of Hebbian ensembles formation. For this purpose we use relatively small window (Fig. 5), that scans the image and produces the binary code of the image part that appears in the window. The size of the window should be selected as follows: the internal space of the window has to include the whole object or the major part of the object to be found. If we expect to recognize the objects of different sizes, various windows should be selected, and the image should be sequentially scanned several times.

VIII. Image Coding

The scheme of image part coding in the window is shown in Fig.6. It works as follows: preprocessed image is scanned by the window and for each position of the window feature extractor FE finds the specific features in the window.

Different types of feature extractors can be used. In this work we use feature extractor from PCNC (Permutation Coding Neural Classifier [26]). To use this feature extractor we consider the scan window as a whole image and create smaller window that is moved within scan image to provide the information for encoder ENC (Fig.6). Encoder produces binary code that is used for Hebbian ensembles formation.

The Hebbian ensembles can be used to find the similarity of different objects. If scan window finds two images that produce code vectors with many equal components, the synaptic weights between neurons that correspond to these components grow up. The Hebbian ensemble corresponding to similar images starts to appear. If similar images are found many times, corresponding synaptic weights achieve values much larger than the mean values of synaptic weights in the matrix. It means that the Hebbian ensemble is formed for the group of similar images.

At the first stage of semisupervised learning the scan window looks through all images of the training database and the synaptic weight matrix is corrected in each position of the scan window. For this purpose the following equation is used:

$$w_{ij}(t+1) = w_{ij}(t) + 1 \text{ , if } ((v_i = 1) \ \& \ (v_j = 1)), \qquad (12)$$

where $w_{ij}(t+1)$ is the value of synaptic matrix component after correction, $w_{ij}(t+1)$ is the value of synaptic matrix component before correction, v_i is the i-th component of the binary code vector that appears at the output of the encoder ENC (Fig.6), v_j is the j-th component of the code vector.

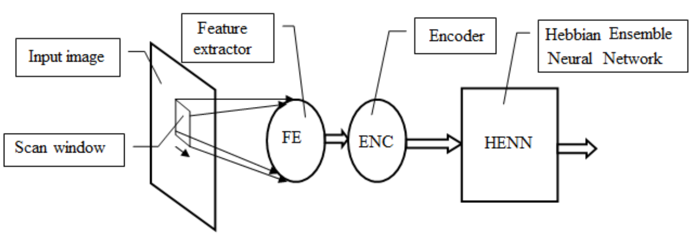

Fig.6. Image coding

When all images of the training database are scanned we have the Hebbian ensembles that correspond to different objects that were found in the images. Now it is necessary to give the names to the ensembles using the names of corresponding objects. For this purpose it is necessary to prepare the code vectors of the names. For each class of the objects that can be found in the image we prepare a random binary vector that will serve as the name of this class.

When all the names are created the second stage of semisupervised learning is started. The training database is scanned secondly. In each position of the scan window after coding of input information Hebbian ensemble neural network is recalculated according to the equation (10) and the threshold value θ is selected to obtain approximately m active neurons at the output. If the value θ is low, no Hebbian ensemble corresponds to the window position. If the value θ is high, the supervisor (operator) looks at the image and selects the name of the object that appears in the scan window. If one of the created names is good for the scan window image, the binary code vector of the name is added to the code vector of excited ensemble, and correction of synaptic weights is made according to the equation (12). The code of the name contains less than m active neurons, but these active neurons repeatedly appear in each example of the ensemble that corresponds to the objects with the same name.

Hebbian ensembles have relatively complex internal structure. As a rule a part of the ensemble neurons has very strong synaptic connections between them. This part is termed “nucleus”. The neurons of the nucleus are excited always when initially was excited any part of the ensemble. Other parts of the ensemble belong to the “fringe”. Real image determines what part of the fringe will be excited in each case. In our system the name of the object corresponds to the “nucleus” of the ensemble.

When we apply this network to the test database, the neurons corresponding to the names of the objects will appear when the system will “see” the object in scan window, i.e. the system will recognize the objects.

IX. Discussion

The Hebbian ensemble neural network permits us to work with unmarked image databases for semisupervised learning. Semisupervised learning starts from the Hebbian ensemble formation. These ensembles include feature combinations that frequently appear in corresponding objects.

After ensemble formation the operator gives the object names to the corresponding ensembles. This process demands much of work at the initial stage. Little by little the system begins to recognize the names of the objects, and operator has to correct it only when the system makes the error. For large image databases this process will demand less work than the mark up process for supervised learning.

This coding technique with Hebbian ensembles can be used with different types of neural classifiers [27]-[30].

X. Conclusion

The neural networks with ensembles are congruent with the biology and the representation of the information, which explain and illustrate diverse neurological phenomena and can be used for artificial intelligent systems development. It was shown in this article that ensemble neural network has a large capability to store independent ensembles. For correlated data from real world, the capacity will decreases, but this fact needed further investigation.

There are some limitations for the ensemble sizes to code more precisely the information we want to memorize. This fact has a direct impact in the size of the neural network and in the maximal number of ensembles that can be stored.

It is interesting to investigate deeper the relationships between parameters of the neural networks, the number of ensembles and the correct decoding. It must be increased the database provided by the simulator. In future we plan to investigate the ensemble neural network application in new situations that were not presented in the training process.

The structure of ensemble neural network is proposed in this article. The evaluation of storage capacity of ensemble neural network shows that the number of neural ensembles will be larger that the number of neurons. In our experiments we worked with independent Hebbian neural ensembles.

The Hebbian ensemble neural network is proposed for semisupervised learning. The learning process begins from automatic formation of the neural ensembles that correspond to the images of different objects. At the second stage of semisupervised learning operator gives the names of the objects to the neural ensembles that were formed previously. This type of learning will be used for very large image databases that are difficult to mark up.

ACKNOWLEDGMENT

We thank DGAPA, UNAM for half year sabbatical grant.

REFERENCES

[1] E. Kussul, T. Baydyk, D. Wunsch, Image recognition systems with permutative coding, *International Joint Conference on Neural Networks*, Montreal, July 31 2005, pp. 1788-1793.

[2] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, *Psychological Review*, Vol. 65, No. 6, 1958.

[3] D. Hebb, *The Organization of Behavior,* New York: Wiley, 1949.

[4] J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Acad. Sci. USA*, vol 79, 1982, pp. 2554-2558.

[5] J. Hopfield, Neurons with grated response have collective computational properties like those of two-state neurons, *Proc. Nat. Acad. Sci*., 81, 1984, pp. 3088-3092.

[6] J. A. Anderson, *An Introduction to Neural Networks*, Massachusetts Institute of Technology Press 1995.

[7] Milner, P., *The Autonomous Brain: A Neural Theory of Attention and Learning*. Lawrence Erlbaum Associates, Inc. Publishers, 1999.

[8] Breitenberg, V., Cell ensembles in the cerebral cortex. *Lect. Notes Biomath*, 21: 171-178, 1978.

[9] Rachkovskij, D., Kussul, E., "Binding and normalization of binary sparse distributed representations by context-depending thinning", *Neural Computation*, 13: 411-452, 2001.

[10] Goltsev, A., *Neural Networks with the Assembly Oranization*, Naukova Dumka, 2005.

[11] Kussul, E., *Associative neuron structures*, Naukova Dumka, 1992.

[12] Kussul, E., and Baidyk, T., "Structure of Neural Ensemble", *In the RNNS/IEEE Symposium on Neuroinformatics and Neurocomputers,* Rostov-on-Don, Russia, 1: 423–434, 1992.

[13] S. Salcedo-Sanz, R. Santiago-Mozos, C. Bousoño-Calzón, A Hybrid Hopfield Network-Simulated Annealing Approach for Frequency Assignment in Satellite Communications Systems, *IEEE Transactions on Systems, Man, and Cybernetics*. Part B, Vol. 34, N 2, April 2004, pp.1108-1116.

[14] Pu Han, Peng Guo, Dong-Feng Wang, The Research of GPS Based on Hopfield Network and its Application in Unit Load System, *Proceedings of the Second Conference on Machine Learning and Cybernetics*, Xi'an, 2-5 November 2003, pp.1226- 1230.

[15] Xiaowei Sheng, Minghu Jiang, A Information Retrieval System Based on Automatic Query Expansion and Hopfield Network, *Proceedings of the IEEE International Neural Networks and Signal Processing*, Nanjing, China, December 14-17, 2003, pp.1624-1627.

[16] T. Kohonen, *Self-organization and associative memory*, Berlin, Springer, 1984, 255 p.

[17] L.D. Jackel, R.E. Howard, J.S. Denker et al. Buildings a Hierarchy with Neural Networks: an Example – Image Vector Quantization, *Appl. Optics*, 1987, 26, N 23, pp.5031–5034.

[18] K. Nakano, Associatron – a Model of Associative Memory, *IEEE Transactions a Systems, Man, and Cybernetics*, SMC–2, N 3 July 1972, pp. 380–388.

[19] E. Kussul, T. Baidyk, D. Wunsch, *Neural Networks and Micro Mechanics*, Springer-Verlag, 2010, pp.210.

[20] A. Lansner, O. Ekeberg, Reliability and Speed of Recall in an Associative Network, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI–7, N4, 1985, pp. 490–498.

[21] M. Tsodyks, Associative Memory in Asymmetric Diluted Network with Low Level of Activity, *Europhys. Lett*. 7 (3), 1988, pp. 203–208.

[22] G. Palm and F.T. Sommer, Information capacity in recurrent McCulloch-Pitts networks with sparsely coded memory states, *Network* 3, 1992, pp.177-186.

[23] A. Knoblauch, Neural associative memory for brain modeling and information retrieval, *Information Processing Letters*, Vol.95, Issue 6, 2005, pp.537-544.

[24] Universidad Rey Juan Carlos, Madrid, Spain, http://frav.escet.urjc.es.

[25] C. Conde Vilda, "Verificación facial multimodal: 2D y 3D", Tesis Doctoral, Universidad Rey Juan Carlos, Madrid, Spain, 2006, pp.219.

[26] Kussul, E., Baidyk, T., Wunsch, D., Makeyev, O., Martín, A., "Permutation coding technique for image recognition systems", *IEEE Transactions on Neural Networks*, 17(6): 1566-1579, 2006.

[27] Baydyk T., Kussul E., Hernández Acosta M., "New Application of LIRA Neural Network," in Proc. *16th WSEAS Intern. Conf. on Circuits*, Kos Island, Greece, 14-17 July 2012, pp.115-119.

[28] T. Baidyk, E.Kussul, "Neural Network Based Vision System for Micro Workpieces Manufacturing," *WSEAS Transactions on Systems*, Issue 2, Volume 3, 2004, pp.483-488.

[29] T. Baidyk, E. Kussul, O. Makeyev, "Texture Recognition with Random Subspace Neural Classifier," *WSEAS Transactions on Circuits and Sysytems*, Issue 4, Volume 4, April 2005, pp.319-325.

[30] T. Baidyk, E. Kussul, O. Makeyev, A. Vega, Limited receptive area neural classifier based image recognition in micromechanics and agriculture, *International Journal of Applied Mathematics and Informatics*, Issue 3, Vol. 2, 2008, pp.96-103.

Ernst Kussul (Ph.D.'67). From 1967 till 1998 he worked in the Institute of Cybernetics of Ukrainian Academy of Sciences. In 1982 he received the D. Sci. degree in artificial neural networks. From 1988 to 1998 he was Head of the Department of Neural Networks in the International Research and Training Center of UNESCO/IIP of Information Technologies and Systems. Since 1998 he is Researcher level "C" in UNAM, Mexico.

He is author and coauthor more than 200 publications, 4 books, 11 patents. His present research interests are micromechanics, mechatronics, neural networks, and pattern recognition.

Dr. Kussul is a member of IEEE. He is a member of the Mexican Academy of Sciences and National System of Scientists.

Tatiana Baidyk (Ph.D'83). From 1977 to 1995 she was with the Institute of Cybernetics of the Ukrainian Academy of Sciences. In 1994 T. Baidyk received D.Sci. in the area of neural networks. From 1995 to 1999 she was coordinator of the International Program of Education and Science in Ukraine, International Renaissance Foundation. From 1998 she was with the International Research and Training Center of UNESCO/IIP of Information Technologies and Systems as a leading researcher. From 2001 till now she has been Researcher in UNAM, Mexico.

She has near 200 publications, 3 books, 3 patents. Baidyk´s present research interests are neural networks, pattern recognition, control systems, and industrial applications.

Dr. Baidyk is a member of the Mexican Academy of Sciences and National System of Scientists.

Enrique Cabello Pardos (Ph.D.'2004) is Full Professor of University Rey Juan Carlos (URJC). He is principle investigator of FRAV, URJC. He obtained his bachelor's degree in Physics Science from University of Salamanca, Spain. He obtained Ph.D. in Informatics from Polytechnic Institute of Madrid, Spain in 2004.

His research interests are biometrics, face recognition in 2D, normalization techniques, registration algorithms for recognition, neural networks, image processing for road safety, artificial vision for security.

Dr. Cabello is reviewer for ACM (Association for Computing Machinery).

Cristina Conde Vilda (Ph.D.'2006) is Assistant Professor of FRAV, University Rey Juan Carlos (URJC). She obtained her bachelor's degree in Physics Science from University Complutense, Madrid, Spain. She obtained her Ph.D. in Informatics from URJC in 2006.

Her research interests are image coding based on events, face recognition in 3D, normalization techniques, registration algorithms for recognition, image processing for road safety, artificial vision for security.