Using Associative Memories as a Mean for Image Accumulating

Alexander B. Bekiarski[1]

Abstract – **Image accumulating is a method for collecting the similar pictures form a chosen class. The direct storing of each image is possible but not effectively, because this cause a great amount of occupied memory. The goal of this article is to use the associative memories as a means for image accumulating. The ability of the associative memories for storing images with similar features using association is the main reason to achieving memory space decreasing with the proposed method of image accumulating. The method is examined with a simulation of the associative memory as a neural network.**

Keywords – **Image Accumulating, Associative Memories, Neural Networks.**

I. Introduction

The image accumulating is investigated since many years. The reasons for this permanent scientific interest are different. In some of the applications, for example the communications [1], the image accumulating give the advantages in speed of transmission, in medicine [2] - increase the diagnostic precision, in robotic [3] – the recognition etc.

The image accumulating effectiveness is related to the method of image presentation in moment of the image collecting in memory or from the number of calculations for this method.

In this article it is choose to use the associative memories as a mean for image accumulating. This give some important advantages for memory size decreasing and using accumulated images in some basic image applications as features extraction, recognition, image database, fast image searching and retrieving etc.

A common way to represent an artificial model of an associative memory is a neural network [4, 5]. There are some types of neural networks, which can be used as an associative memory presentation. These types can be analyzed and compared for him advantages and effectiveness to image accumulating. A very important condition to achieve this goal is to pose some conditions of this analysis and comparison.

In this article it is used the simulation of the image accumulating in associative memory represented as a neural network. The simulation is made with real images, which are preliminary stored in one of the standard image file formats like BMP, TIFF etc. The results from the simulation are presented, analyzed, compared and followed by the comments and conclusion.

[1]Alexanded B. Bekiarski is with the Faculty of Communications, of Technical University – Sofia, 8 Kliment Ohridski St., 1756 Sofia, Bulgaria, E-mail: aabbv@tu-sofia.bg

II. Associative Memories Representation

The artificial associative memory representation takes the idea from the human ability to perceive, collect and “remembering” images in brain not only from the information of the plain images, but in the cases when eyes see only a representative part of the images or an image with some distortions. These parts or distorted images serve as a “cue” for the plain or original images. This human ability is investigated carefully and it is proved, that it is related of the way the image information is entered, processed and stored in the brain. The way of storing the images in the conventional electronic memories differ mainly from this of the human. This is because the conventional memories work like one, two or multi dimensional linear arrays. The places (or addresses) of the stored images are not related with any features of the images, they are arranged only in size or in time when they are entered and stored. This is a very non effective, but easy method, which is commonly used if there is not so important to manipulate the stored images in the sense of feature searching or retrieving, clustering of the similar images, classification, recognition etc. The implementation of the human model of collecting images in an associative manner is the base of the artificial associative memory representation.

The other fundamental idea of artificial associative memory representation is taken from the physical phenomenon and systems. In some physical systems, for example a bowl in which a ball bearing is allowed to roll freely, it is possible to describe its operations in terms of the network's `energy'. Suppose the ball go from a point somewhere up the side of the bowl with, possibly, a push to one side as well. The ball will roll back and forth and around the bowl until it comes to rest at the bottom.

The physical description of what has happened may be couched in terms of the energy of the system. The ball initially has some potential energy. That is work was done to push it up the side of the bowl to get it there and it now has the potential to gain speed and acquire energy. When the ball is released, the potential energy is released and the ball rolls around the bowl (it gains kinetic energy). Eventually the ball comes to rest where its energy (potential and kinetic) is zero. (The kinetic energy gets converted to heat via friction with the bowl and the air). The main point is that the ball comes to rest in the same place every time and this place is determined by the energy minimum of the system (ball + bowl). The resting state is said to be stable because the system remains there after it has been reached.

There is another way of thinking of this process which ties in with ideas about memory. Suppose that the ball comes to rest in the same place each time because it `remembers' where the bottom of the bowl is. The location of the bottom of the bowl can be described as a vector $\mathbf{x_0}$, which represents the stored pattern. Thus, the ball position or "state" at any time is given by the three coordinates or the position vector $\mathbf{x}$. By writing the ball's vector as the sum of vector $\mathbf{x_0}$ and a displacement $\mathbf{\Delta x}$ thus,

$$\mathbf{x} = \mathbf{x_0} + \mathbf{\Delta x} \tag{1}$$

It is possible to think of the ball's initial position as representing the partial knowledge or cue for recall, since it approximates to the memory $\mathbf{x_0}$ .If it is now used a corrugated surface instead of a single depression (the bowl) it may store many `memories' (vectors):

$$\{\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_3}, \mathbf{x_4}, \ldots\ldots\ldots\ldots \} \tag{2}$$

If now the ball is started somewhere on this surface, it will eventually come to rest at the local depression which is closest to its initial starting point. That is it evokes the stored pattern which is closest to its initial partial pattern or cue. Once again, this is an energy minimum for the system.

There are therefore two complementary ways of looking at what is happening. One is to say that the system falls into an energy minimum; the other is that it stores a set of patterns and recalls that which is closest to its initial state. If it is build a network which behaves like this, it is completely described by the following key elements:

- a state vector

$$\mathbf{v} = (v_1, v_2, \ldots\ldots\ldots, v_n); \tag{3}$$

- a set of stable states

$$\mathbf{v_1}, \mathbf{v_2}, \ldots\ldots\ldots\ldots \mathbf{v_m}, \tag{4}$$

which correspond to the stored patterns and, in, the corrugated surface example, were the bottoms of the depressions in the surface;

- the system evolves in time from any arbitrary starting state $\mathbf{v}$ to one of the stable states, and this may be described as the system decreasing its energy E. This corresponds to the process of memory recall.

The 'energy' principles described here can be used as a base to image accumulating designing an associative memory, which is responding to the mentioned above conditions. There are many possibilities to realize the associative memory working for image accumulating. In this article it is proposed to use a neural network for this purpose. The type of neural network can be chosen from a lot of existing neural network types and each of these types must be analyzed, examined etc.

III. Associative Memories with Neural Networks

A very often representation of an associative memory is a neural network. There are some types of neural networks, which can be used as an associative memory presentation and can be used for image accumulating. After a precision analysis of these types it is decided to use in this work one of the most applicable neural networks type - the Hopfield Neural Network. The architecture, as it is presented in Matlab for simulation of this network is shown in Fig.1.

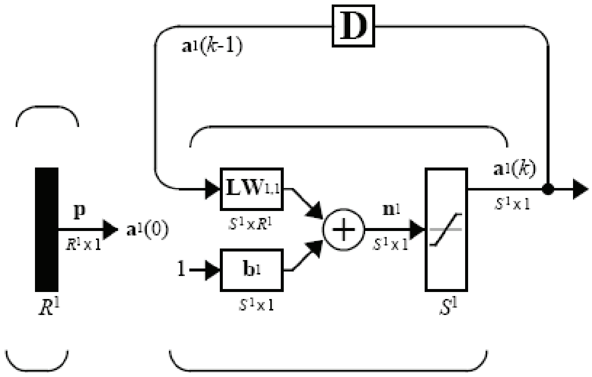

Fig. 1. Figure example

The input vector $\mathbf{p}$ of the Hopfield neural network have $R^1 x1$ dimension. For the image accumulating purpose it is necessary to calculate the dimension for the input vector as:

$$R = N_x + N_y, \tag{5}$$

where

N_x, N_y are image dimensions in horizontal and vertical direction.

The other Hopfield parameters, shown in Fig.1, are described precisely in Matlab Neural Network Toolbox [5], but in this article it is important to discuss the number of neurons in the network - N_{num}. This number can be determined for a concrete Hopfield neural network application as an associative memory for image accumulating, but it must satisfy the following condition:

$$N_{num} \geq N_{icl}, \tag{5}$$

where

N_{icl} is the number of image classes chosen for the desired image accumulating example.

The Hopfield neural network can be consider also as a tree. Their nodes are the states of the network. They can be thinking as a net of images for accumulating. Every node is connected to every other node (but *not* to itself) and the connection strengths or weights are symmetric in that the weight from node *i* to node *j* is the same as that from node *j* to node *i*.

This means that:

$$w_{ij} = w_{ji}\text{, and } w_{ii} = 0 \text{ for all } i, j\,. \tag{6}$$

Notice that the flow of information in this net is not in a single direction as it has been in the nets dealt with so far. It is possible for information to flow from a node back to itself via other nodes. That is, there is feedback in the network and so they are known as feedback or *recurrent* nets as opposed to *feedforward* nets which were the subject of the Backpropagation algorithm. This mean that the transitions for a node forward and backward is a change for an image to another image.

The state of the net at any time is given by the vector of the nodes outputs:

$$(x_1, x_2, x_3, ...)\,. \tag{7}$$

Suppose, that the net now start in some initial state or image and choose a node at random and let it update its output or `fire'. That is, it evaluates its activation in the normal way and outputs a `1' if this is greater than or equal to zero and a `0' otherwise. The net now finds itself either in the same state or image as it started in, or in a new state, that mean new image, which is at Hamming distance one from the old. Then it is chosen a new node or image at random to fire and continue in this way over many steps. For each state, it may evaluate the next state given each of the nodes fires. This means to activate a new image for accumulating. An example of state diagram for using Hopfield neural network as associative memory for image accumulating is given in Table 1.

TABLE I
STATE DIAGRAM FOR IMAGE ACCUMULATING

Current state in time t	New state in time $t+1$
$x_1(t)$	$x_1(t+1)$
$x_2(t)$	$x_2(t+1)$
$x_3(t)$	$x_3(t+1)$
......	......
......	......

The other way to describe state transition in tree representation of image accumulating is as a *state transition diagram*.

States or images are represented by the circles with their associated state number. Directed arcs represent possible transitions between states or images and the number alongside each arc is the probability that each transition will take place. The states or images have been arranged in such a way that transitions tend to take place down the diagram; this will be shown to reflect the way the system decreases its energy and represent the principle of the association. The important thing to notice at this stage is that, no matter where start in the diagram, the net will eventually find itself in one of the stable states or image. These reenter themselves with probability 1. That is they are stable states or accumulated images - once the net finds itself in one of these it stays there. These state vectors or images are the `memories' stored by the net, it mean stored or accumulated images.

The dynamics of the net accumulating images are described perfectly by the state transition table or diagram. However, greater insight may be derived if we can express this in terms of an energy function and, using this formulation, it is possible to show that stable states or images will always be reached in such a net. Consider two nodes i, j, representing images in the net which are connected by a positive weight and where *j* is currently outputting a `0' while *i* is outputting a `1'.

If for the image *j* were given the chance to update or fire, the contribution to its activation from image *i* is positive and this may well serve to bring *j*'s activation above threshold and make it output a `1'. A similar situation would prevail if the initial output states of the two nodes, representing images, had been reversed since the connection is symmetric. If, on the other hand, both units are `on' they are reinforcing each other's current output. The weight may therefore be thought of as fixing a constraint between *i* and *j* images that tends to make them both take on the value `1'. A negative weight would tend to enforce opposite outputs. One way of viewing these image accumulating networks is therefore as *constraint satisfaction* nets.

This idea may be captured quantitatively in the form of a suitable energy function for the neural network for image accumulating. Define:

$$e_{i,j} = -w_{i,j} x_i x_j \tag{8}$$

If the weight is positive then the last entry is negative and is the lowest value in the table. If is regarded as the `energy' of the image pair *ij* then the lowest energy occurs when both units are on which is consistent with the arguments above. If the weight is negative, the image state is the highest energy state and is not favoured. The energy of the neural network is found by summing over all pairs of nodes or presented images:

$$E = \sum_{pairs} e_{i,j} = -\sum_{pairs} w_{i,j} x_i x_j \tag{9}$$

This may be written:

$$E = -\frac{1}{2} \sum_{i,j} w_{i,j} x_i x_j \tag{10}$$

Since the sum includes each pair twice as $w_{i,j} x_i x_j$ and $w_{j,i} x_j x_i$ and $w_{i,j} = w_{j,i}$, $w_{i,i} = 0$.

It is now instructive to see what the change in energy of the neural network is when a node fires or an image is activated. Suppose node k is chosen to be updated with a new image. The energy E of the neural network by singling out the terms involving this node can be written as:

$$E = -\frac{1}{2}\sum_{\substack{i\neq k\\ j\neq k}} w_{i,j}x_i x_j - \frac{1}{2}\sum_i w_{k,i}x_k x_i - -\frac{1}{2}\sum_i w_{i,k}x_i x_k \tag{11}$$

Now, because $w_{i,k} = w_{k,i}$, the last two sums may be combined

$$E = -\frac{1}{2}\sum_{\substack{i\neq k\\ j\neq k}} w_{i,j}x_i x_j - \sum_i w_{k,i}x_k x_i \tag{12}$$

For ease of notation, denote the first sum by S and take the x_k outside the sum since it is constant throughout, then the energy E of the neural network is:

$$E = S - x_k \sum_i w_{k,i}x_k x_i, \tag{13}$$

but the sum here is just the activation of the kth node so that:

$$E = S - x_k a^k \tag{14}$$

Let the energy E of the neural network after k has updated be $E^{'}$ and the new output be $x_k^{'}$. Then:

$$E^{'} = S - x_k^{'} a^k \tag{15}$$

Denote the change in energy $E^{'} - E$ by ΔE and the change in output $x_k^{'} - x_k$ by Δx_k , then subtracting (14) from (15) gives:

$$\Delta E = -\Delta x_k a^k \tag{16}$$

There are now two cases to consider:

- $a^k \geq 0$. Then the output goes from `0' to `1' or stays at `1'. In either case $\Delta x_k \geq 0$. Therefore $\Delta x_k a^k \geq 0$ and so, $\Delta E \leq 0$;
- $a^k \langle 0$. Then the output goes from `1' to `0' or stays at `0'. In either case $\Delta x_k \leq 0$. Therefore $\Delta x_k a^k \geq 0$ and so, $\Delta E \leq 0$;

Thus, for any node or image accumulating being updated we always have $\Delta E \leq 0$ and so the energy of the neural network decreases or stays the same. But the energy of the associative memory is bounded below by a value obtained by putting all the $x_i x_j = 1$ in (10). Thus energy of the neural network E must reach some fixed value and then stay the same. Once this has occurred, it is possible for further changes in the network's state to take place since $\Delta E = 0$ is still allowed. However, for this to be the case $\Delta x_k \neq 0$ and $\Delta E = 0$ it must have $a^k = 0$. This implies the change must be of the form $0 \to 1$. There can be at most N (of course there may be none) of these changes, where N is the number of nodes or images to accumulating in the neural network. After this there can be no more change to the net's state and a stable state has been reached. This means that the neural network and respectively the associative memory is in the stable state – the accumulated image.

IV. Conclusion

The above described equations are used to simulating the work of the proposed associative memory as a mean for image accumulating. The preliminary results are good and show, that the proposed method is applicable in some cases when the goal is to collect images as a database or in some preferred features used as an association for accumulating or retrieving images.

References

[1] R. H. Reeve, P. Tim-Kui Tam, “The Image-Accumulation Technique as a Variable in Multiple-Image Communication”, Annual Meeting of the Association for Educational Communications and Technology, pp. 341-350, Philadelphia, Pa., March 22-26, 1991.

[2] K. Numata, M. Morimoto, T. Isozaki, "Use of accumulation images obtained by arterial-phase contrast-enhanced harmonic grayscale ultrasonography to evaluate tumor vessels in focal nodular hyperplasia and other hepatic tumors", Springer Journal of Medical Ultrasonics, Conference Proceedings, pp. 3-10, Volume 34, Number 1, March, 2007.

[3] M. E. Nilsback, “A Cue-Integration Scheme for Object Recognition Using Discriminative Accumulation”, Thesis, Department of Numerical Analysis and Computer Science, Stockholm, Sweden, 2004.

[4] T. Kohonen, “Self-Organizing Maps”, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 1995, 1997, 2001, Third Extended Edition.

[5] H. Demut, M. Bea, “Neural Network Toolbox”, The MathWork, 2006.