JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 25, 1737-1751 (2009)

Blood Cell Identification Using Emotional Neural Networks

ADNAN KHASHMAN
Intelligent Systems Research Group (ISRG)
Faculty of Engineering
Near East University
Lefkosa, Mersin 10, Turkey

The idea of machines having emotions sounds like science fiction, however, few decades ago the idea of machines with intelligence seemed also like fiction, but today we are developing intelligent machines with successful applications. We have always overlooked the emotional factors during machine learning and decision making; however, it is quite conceivable to artificially model certain emotions in machine learning. This paper presents an emotional neural network (EmNN) that is based on the emotional back propagation (EmBP) learning algorithm. The EmNN has emotional weights and two emotional parameters; anxiety and confidence, which are updated during learning. The performances of the EmNN and a conventional BP-based neural network, using two topologies for each network, will be compared when applied to a blood cell type identification problem. Experimental results show that the additional emotional parameters and weights improved the identification rate as well as the classification time.

Keywords: neural networks, back propagation, emotion modeling, anxiety and confidence, object identification, blood cells

1. INTRODUCTION

Recent definitions of emotion have either emphasized the external stimuli that trigger emotion, or the internal responses involved in the emotional state, when in fact emotion includes both of those things and much more [1]. The effective role of emotions on cognitive processing, learning and decision making in animals and humans has been emphasized by several researchers [2-8]. Emotions play an important role in human decision-making process, and thus they should be embedded within the reasoning process when we try to model human reactions [9]. Although computers do not have physiologies like humans, information signals and regulatory signals travel within them; there will be functions in an intelligent complex adaptive system, that have to respond to unpredictable, complex information that play the role that emotions play in people [10]. Such computers will have the same emotional functionality, but not the same emotional mechanisms as human emotions. We may think of machine emotions as machine intelligence; we do not expect machines to "feel" the way we feel, but we could simulate machine emotions just as we do machine intelligence [9].

There have been examples of research works that attempted to incorporate emotions in machines in one way or another [8, 9, 11-22]. It was concluded from these works that if emotions such as anxiety, fear, and stress are included in systems that aim to simulate the human behaviour in certain circumstances (*e.g.*, human-computer interfaces, education, entertainment, *etc.*), the system will be more user-friendly and its responses will be

Received January 23, 2008; revised September 23 & December 9, 2008; accepted December 24, 2008.
Communicated by Pau-Choo Chung.

more similar to human behaviour. Other recent research works suggested the use of emotional components within neural models and control systems. For example, Abu Maria and Abu Zitar [23] proposed and implemented a regular and an emotional agent architecture which is supposed to resemble some of the human behaviours. They noticed that artificial emotions can be used in different ways to influence decision-making. Gobbini and Haxby [24] proposed a model for distributed neural systems that participate in the recognition of familiar faces, highlighting that this spatially distributed process involves not only visual areas but also areas that primarily have cognitive and social functions such as person knowledge and emotional responses. Coutinho and Cangelosi [6] suggested the use of modelling techniques to tack into the emotion/cognition paradigm, and presented two possible frameworks that could account for their investigation, one of which explored the emergence of emotion mechanisms. Most of these previous attempts on incorporating emotions in to machine learning have shown successful results, and provided a positive trend to developing machines with emotions, albeit simulated.

The novelty of our work in this paper is that the emotional neural network (EmNN) model has emotional neurons, two emotional parameters (anxiety and confidence), and emotional weights. The emotional factors are updated during learning, and the final emotional weights are used together with the network's conventional weights to make decisions. The incorporation of the simulated emotionality within a neural network structure aims at further improving the network's learning and decision making abilities. The presented EmNN model is based on the Emotional Back Propagation (EmBP) learning algorithm [25]. The additional emotional coefficients (anxiety and confidence) are updated each iteration or epoch during the learning process, and their values are used to update the emotional weights associated with the emotional neurons.

There are two emotional neurons feeding the hidden and output layers in the EmNN. These emotional neurons differ from the normal neurons in that they are non-processing neurons which receive global average values of input images, rather than segments or pixel values from that image, also their associated emotional weights are updated using the two emotional coefficients, rather than the conventional learning and momentum rates. In practice, what the two emotional parameters mean is that when the emotional neural network is trained, and as the epochs progress, one term (anxiety level) tells the system to pay less and less attention to the derivative of the error of the training pattern using all nodes as the input average value of the samples of the training pattern, while the other term (confidence level) tells the system to pay more and more attention to the previous change it made to the weights, which is some sort of an increasing inertia term to modify the level of change from one pattern to the next as the training epochs progress.

In order to assess the performance of the emotional neural network, four neural networks (two emotional and two conventional BP-based) will be implemented to a identify blood cell types. The difference between the four networks comes from the pattern averaging method which provides the input data for training a neural network. The performances of the two emotional networks and the two conventional networks will be compared and conclusions will be accordingly drawn.

2. EMOTIONAL NEURAL NETWORK ALGORITHM

In this section, the learning algorithm of the emotional neural network (EmNN) will be described. EmNN is based on the emotional back propagation (EmBP) learning algorithm [25], which is a modified version of the conventional back propagation (BP) learning algorithm. BP has been popularly used ever since its introduction by Rumelhart *et al.* in 1986 [26], due to its implementation simplicity and quick training, specially, when sufficient training database is available. The EmBP algorithm is described next according to the flow of information within the EmNN, which consists of three layers: input layer with (i) neurons, hidden layer with (h) neurons, and output layer with (j) neurons. Fig. 1 shows the EmNN configuration during feed forward calculations.

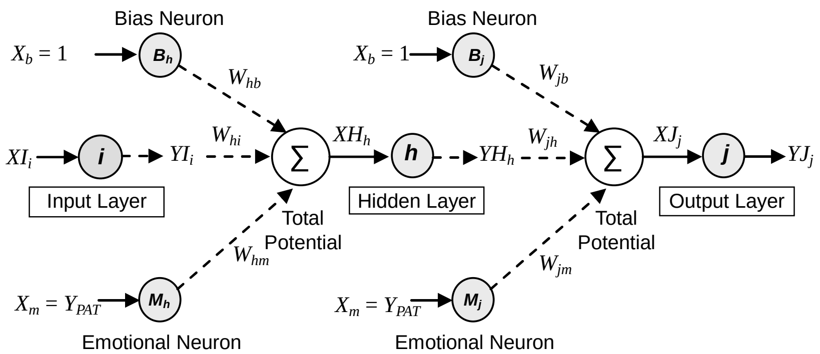


Fig. 1. Emotional neural network (EmNN) configuration during feed forward calculations.

2.1 Feed Forward Calculations

These computations are required in both training and generalization of the EmNN. Firstly, the output of each input layer non-processing neuron is defined as:

$$YI_i = XI_i, \tag{1}$$

where, XI_i and YI_i are, respectively, input and output values of neuron i in the input layer. The two additional emotional neurons are also non-processing neurons, and thus similarly to input layer neurons, their output values are equal to their input values. Secondly, the hidden layer neurons, which are processing neurons; therefore, the sigmoid activation function is used to activate each neuron in this layer. Here it is assumed that there is one hidden layer, however the same process can be applied for more than one hidden layer. The output of each hidden layer neuron is defined in Eq. (2) where, XH_h and YH_h are input and output of neuron h in hidden layer, respectively.

$$YH_h = \left(\frac{1}{1+\exp(-XH_h)}\right). \tag{2}$$

The input to a hidden layer neuron XH_h is calculated using the total potential (TP_h) of all input values coming into that neuron. The total potential is the sum of multiplications of input values and their associated weights. There are three different types of inputs, and consequently three types of total potential. The first total potential (TP_{hc}) is the conventional total potential which is obtained using the output values of the previous layer (input layer in this case) and the conventional weight matrix. The second total potential (TP_{hb}) is obtained using the hidden layer bias neuron and its associated weights. The third total potential (TP_{hm}) is obtained using the hidden layer emotional neuron and its associated emotional weights. Therefore, the input to a hidden neuron is defined as:

$$XH_h = TP_{hc} + TP_{hb} + TP_{hm}. \tag{3}$$

TP_{hc}, TP_{hb}, and TP_{hm} are, respectively, defined as follows:

$$TP_{hc} = \sum_{i=1}^{r} W_{hi} \cdot YI_i,\ TP_{hb} = W_{hb} \cdot X_b,\ \text{and } TP_{hm} = W_{hm} \cdot X_m, \tag{4}$$

where, W_{hi} is the conventional weight given by hidden neuron h to input neuron i, and YI_i is the output of input neuron i. r is the total number of input neurons. W_{hb} is the bias weight given by hidden neuron h to the hidden layer bias neuron B_h, and X_b is the input value to the bias neuron which is set to 1. W_{hm} is the emotional weight given by hidden neuron h to the hidden layer emotional neuron M_h, and X_m is the input value to the emotional neuron. X_m represents the input pattern average value, which models the input visual stimuli that affects the emotionality of the neural network [27]. Therefore, the global average of an input pattern is fed into the emotional neuron and the resulting total potential is used to calculate the output of the hidden layer. X_m is calculated as:

$$X_m = Y_{PAT} = \sum_{x=1,\,y=1}^{x_{\max},\,y_{\max}} P(x, y) \Big/ x_{\max} \cdot y_{\max}\ , \tag{5}$$

where Y_{PAT} is the global input pattern average value for an input image $P(x, y)$, and $x_{\max}$ and $y_{\max}$ are the highest number of pixels in the x and y directions of image $P(x, y)$ respectively. Thirdly, the output layer neurons which are also processing neurons, and thus similarly to the hidden layer, the sigmoid activation function is used to activate each neuron in this layer. The output of each output layer neuron is in Eq. (6) where, XJ_j and YJ_j are input and output values of neuron j in output layer, respectively.

$$YJ_j = \left(\frac{1}{1 + \exp(-XJ_j)} \right) \tag{6}$$

The input to an output layer neuron XJ_j is also calculated using the total potential (TP_j) of all input values coming into that neuron from the previous hidden layer, bias neuron, and emotional neuron. Therefore, the input to an output neuron is defined as:

$$XJ_j = TP_{jc} + TP_{jb} + TP_{jm}. \tag{7}$$

TP_{jc}, TP_{jb}, and TP_{jm} are, respectively, defined as:

$$TP_{jc} = \sum_{h=1}^{l} W_{jh} \cdot YH_h,\ TP_{jb} = W_{jb} \cdot X_b,\ \text{and } TP_{jm} = W_{jm} \cdot X_m, \tag{8}$$

where, W_{jh} is the weight given by output neuron j to hidden neuron h, and YH_h is the output of hidden neuron h. l is the total number of hidden neurons. W_{jb} is the weight given by output neuron j to the output layer bias neuron B_j, and X_b is the input value to the bias neuron which is set to 1. W_{jm} is the weight given by output neuron j to the output layer emotional neuron M_j, and X_m is the input value to the emotional neuron. X_m is calculated as in Eq. (5).

2.2 The Emotional Parameters

Here the added emotional parameters are used together with the existing learning coefficient (η) and momentum rate (α) in order to adjust the neural network weights, based on error minimization. The emotional parameters are the *Anxiety coefficient* (μ) and the *Confidence coefficient* (k). We can think of how these two parameters behavior within the EmNN as follows: when we learn a new task, the anxiety level is high at the beginning and the confidence level is low. After time, practice and getting positive feedback, the anxiety level decreases while the confidence level increases. Once learning is achieved, we tend to be less anxious and more confident doing a task that we have already learnt. The EmNN incorporates both emotional coefficients during the learning process resulting in additional emotional weights, which are used after training in the generalization process. Both coefficients have normalized values between “0” and “1”.

The anxiety level is dependent on the input pattern average values and the error signal of each epoch. The average input values are always positive, since the pixel values are of gray level images (0 to 255) which are normalized to values between (0 to 1). On the other hand, the error signal may provide a negative feedback if learning is unstable. In this case the learning of the emotional network (similarly to conventional networks) will be both unreliable and unstable. Therefore, during the training of the emotional network, three important parameters are adjusted until a stable learning (smooth conversion to error) is found. The three adjustable parameters are the learning rate, momentum rate and number of hidden neurons. Notice that the anxiety and confidence coefficients are not adjustable by the human user, but rather change during learning depending on the input patterns and the feedback error. In summary, both input pattern average values and the error signal are both positive during the training of the emotional neural network, however the anxiety level should decrease as learning progresses, since the error value decreases until the network converges.

The increase of the value of the confidence coefficient as learning progresses does not imply the possibility of overtraining, which is usually a potential problem with the conventional BP-based neural network. The primary reason of overtraining is when the amount of training data is too small and the amount of network parameters is too large [28]. To avoid this problem, suggestions were made to use a test set in order to detect overtraining during training [29], for example, rotating (parts of) the training set and the test set. In order to avoid overtraining the EmNN, the training/testing data has been enriched using rotated images of the blood cells, while keeping the network parameters as small as possible, using different input image segmentation methods to maintain the input layer size small. The anxiety coefficient (μ_i) value is defined as:

$$\mu_i = Y_{AvPAT} + E_i, \tag{9}$$

where Y_{AvPAT} is the average value of all presented patterns to the EmNN in each iteration; and is defined as:

$$Y_{AvPAT} = \sum_{p=1}^{N} Y_{PAT\_p} \Big/ N\,, \tag{10}$$

where p is pattern index, N is the total number of presented patterns within one epoch, Y_{PAT_p} is the global average value of pattern p as defined in Eq. (5). The error feedback E_i in iteration i, is defined as:

$$E_i = \sum_{p=1}^{N} \sum_{j=1}^{N_j} (T_{pj} - YJ_{pj})^2 \Big/ N \cdot N_j\,, \tag{11}$$

where j is output neuron index from first to the last neuron N_j, and N is total number of presented patterns p. YJ_j and T_j are the actual and target output value for neuron j, respectively. The confidence coefficient (k) value is defined as:

$$k = \mu_0 - \mu_i, \tag{12}$$

where (μ_0) is the anxiety coefficient value at the end of the first iteration (a constant value representing the highest anxiety value at the start of new training) and (μ_i) is the anxiety coefficient value at the end of subsequent iteration i (a variable value that is updated each iteration). Anxiety (μ) and confidence (k) coefficients are dependent on each other; when training starts the initial values of (μ) and (k) are set to 1 and 0, respectively; these values are updated during each iteration using firstly Eq. (9) to update anxiety value (μ), and then Eq. (12) to update confidence value (k).

2.3 Error Back Propagation Calculations

These computations are performed only during the training of the EmNN until conversion to a required error level is achieved. Rumelhart *et al.* [26] defined an error term that depends on the difference between the output value an output neuron is supposed to have, called the target value T_{pj}, and the value it actually has as a result of the feed forward calculations, YJ_{pj}. The error term represents a measure of how well a network is training on a particular training set. Eq. (13) presents the definition for the error E_p. The subscript p denotes what the value is for a given pattern.

$$E_p = \sum_{j=1}^{N_j} (T_{pj} - YJ_{pj})^2. \tag{13}$$

The aim of the training process is to minimize this error over all training patterns. From Eq. (6), it can be seen that the output of a neuron in the output layer is a function of its input, or

$$YJ_j = f(XJ_j). \tag{14}$$

The first derivative of this function, $f'(XJ_j)$, is an important element in error back propagation. For output layer neurons, a quantity called the error signal is represented by Δ_j which is defined as:

$$\Delta_j = f'(XJ_j) \cdot (T_j - YJ_j) = YJ_j \cdot (1 - YJ_j) \cdot (T_j - YJ_j), \tag{15}$$

where YJ_j is the output value of neuron j in the output layer, and T_j is the target output value at neuron j. This error value is propagated back and appropriate weight adjustments are performed. This is done by accumulating the Δ's for each neuron for the entire training set, adding them, and propagating back the error based on the grand total Δ. This is called batch (epoch) training. Assuming that the neural network has three layers (input, hidden and output layers), the following subsections describe EmNN weight updating procedure using the error signal.

2.3.1 Hidden-to-output weight adjustment

The new updated conventional weights $W_{jh}(new)$ (Eq. (16)), and bias weights $W_{jb}(new)$ (Eq. (17)) are calculated as:

$$W_{jh}(new) = W_{jh}(old) + \eta \cdot \Delta_j \cdot YH_h + \alpha \cdot [\delta W_{jh}(old)], \tag{16}$$

$$W_{jb}(new) = W_{jb}(old) + \eta \cdot \Delta_j + \alpha \cdot [\delta W_{jb}(old)], \tag{17}$$

where η is the learning rate, α is the momentum rate, $W_{jh}(old)$ is the previous conventional weights, YH_h is the output value of neuron h in the hidden layer, and $\delta W_{jh}(old)$ is the previous change in conventional weights. $W_{jb}(old)$ is the previous bias weights, and $\delta W_{jb}(old)$ is the previous change in bias weights.

The EmNN has emotional neurons, and therefore, an extra set of weights must be accounted for. The new updated emotional weights $W_{jm}(new)$ are calculated as:

$$W_{jm}(new) = W_{jm}(old) + \mu \cdot \Delta_j \cdot Y_{AvPAT} + k \cdot [\delta W_{jm}(old)], \tag{18}$$

where $W_{jm}(old)$ is the previous emotional weights, μ is the anxiety coefficient, k is the confidence coefficient, $\delta W_{jm}(old)$ is the previous change in emotional weight, and Y_{AvPAT} is the average value of all presented patterns to the EmNN in each iteration, as defined in Eq. (10).

2.3.2 Input-to-hidden weight adjustment

The error term for an output layer is defined in Eq. (15). For the hidden layer an error signal definition was provided by Rumelhart *et al*. [26] as follows:

$$\Delta_h = f'(XH_h) \cdot \sum_{j=1}^{N_j} W_{jh} \cdot \Delta_j = YH_h \cdot (1 - YH_h) \cdot \sum_{j=1}^{N_j} W_{jh} \cdot \Delta_j, \tag{19}$$

where YH_h is the output value of neuron h in the hidden layer, and N_j is the total number of neurons in the output layer.

The weight adjustments for the conventional, bias, and emotional connections feeding the hidden layer from the input layer are now calculated in a similar manner to those feeding the output layer. The new updated conventional weights $W_{hi}(new)$, bias weights $W_{hb}(new)$, and emotional weights $W_{hm}(new)$ are calculated as in Eqs. (20)-(22), respectively.

$$W_{hi}(new) = W_{hi}(old) + \eta \cdot \Delta_h \cdot YI_i + \alpha \cdot [\delta W_{hi}(old)], \tag{20}$$

$$W_{hb}(new) = W_{hb}(old) + \eta \cdot \Delta_h + \alpha \cdot [\delta W_{hb}(old)], \tag{21}$$

$$W_{hm}(new) = W_{hm}(old) + \mu \cdot \Delta_h \cdot Y_{AvPAT} + k \cdot [\delta W_{hm}(old)], \tag{22}$$

where $W_{hi}(old)$ is the previous conventional weights, η is the learning rate, α is the momentum rate, YI_i is the output value of neuron i in the input layer, and $\delta W_{hi}(old)$ is the previous change in conventional weights. $W_{hb}(old)$ is the previous bias weights, and $\delta W_{hb}(old)$ is the previous change in bias weight. η is the learning rate, α is the momentum rate. $W_{hm}(old)$ is the previous emotional weights, μ is the anxiety coefficient, k is the confidence coefficient, $\delta W_{hm}(old)$ is the previous change in emotional weights, and Y_{AvPAT} is the is the average value of all presented patterns.

3. BLOOD CELL IDENTIFICATION

In this section, two designs of the emotional neural network (EmNN), as well as two designs on the conventional BP-based neural network (BPNN), will be implemented to identify the different types of blood cells. There are three major cellular constituents of the blood: Erythrocytes or red cells, Leukocytes or white blood cells, and Platelets. The three blood cell types can be differentiated from each other by their different sizes and different morphological features such as the presence or absence of a nucleus in the cells and the shape of the nucleus [30].

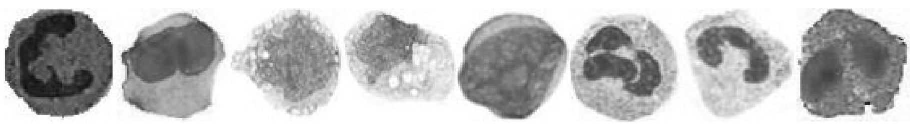
The process of automatic blood cell classification involves: *acquisition*, *detection*, *feature extraction*, and *classification*. During acquisition, the blood smear is magnified to a suitable scale under the microscope, and then transformed into a digital image using a modern charge-coupled device (CCD) camera. In detection, cell segmentation yields a number of single-cell images, and each single-cell image is segmented into three regions: Cell nucleus, cytoplasm, and background. During feature extraction each segmented cell is analyzed to form a feature vector from color, shape, and texture features. Finally, in classification, each blood cell is labeled by the classifier according to its feature vectors [31]. Recent research works have suggested different methods for blood cell image segmentation, which is part of the cell detection process [32-35]. However, detection process and segmentation are not considered in this application which focuses on the feature extraction and classification processes using single-cell images for the development and implementation. Recent works on feature extraction and classification of blood cells aimed at identifying the three major cell types or the identification and counting of white cells [31, 36, 37]. However, despite the successful results that were demonstrated by these methods, they only targeted the identification of white blood cells. The classification of the three blood cell types using conventional neural networks was also addressed in previous works [30, 38-40]. The following subsections describe the implementation method to the problem of blood cell identification.

3.1 Blood Cell Images

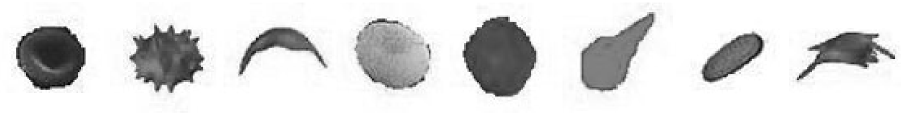
In general, databases used in developing blood cell classification systems rely on microscopic cell images. The variety of information in these images makes cell identification difficult for machines due to the different sizes, shapes and colors. However, these differences are not considered as an obstacle in this work and are left to the neural network to learn using its input data as obtained by pattern averaging [40].

The basic idea of the proposed identification system is to simulate a human expert identifying blood cells once he/she had a look at their images. The task is not trivial for a non-expert human, and certainly difficult for a machine that usually analyses a blood cell image as a vector with numbers. The purpose of image rotations was to, firstly make the identification task more challenging for a machine, and secondly to enlarge the database for training and testing the neural networks. The natural variety in shape and size of blood cells (even within one group, say red or white) makes it unnecessary to consider deformable blood cells in this particular application.

The implementation uses 360 single-cell images (see examples in Fig. 2), which were manually segmented from blood smear images obtained from Franklin Institute database [41], representing 90 different blood cells (30 Red, 30 White, and 30 Platelets) obtained by rotating each cell by 90^o. The rotations (at 0^o, 90^o, 180^o and 270^o) are aimed at testing the identification system's rotational invariance capability. The obtained original color images were then converted from RGB to gray level, and resized to (70×70) pixels, in preparation for the feature extraction phase.



(a) White blood cells.



(b) Red blood cells.

(c) Platelets blood cells.

Fig. 2. Blood cell examples of varying shapes and sizes [41].

3.2 Feature Extraction

Pattern averaging is used to extract the feature vectors from the cell images. Averaging is a simple but efficient method that creates "fuzzy" patterns as compared to multiple "crisp" patterns, which provides the neural network with meaningful learning while reducing computational expense [40]. Additionally, pattern averaging overcomes the problem of varying pixel values within the segmented blocks as a result of rotation or varying scale, thus, providing a rotation and scale invariant system.

Feature extraction in this work uses average values of non-overlap segmented image blocks. Each averaged block value is fed into a neuron in the network's input layer. Thus, the choice of block size affects the total number of blocks representing an input image, and consequently the number of input layer neurons. We use two block sizes in this work: (5×5 and 7×7) pixels. If by coincidence, images of two different cells have the same global gray level distribution (histogram), then their global average value will be identical, however their local blocks average values will be different, and thus the two cells can be separated and identified by the emotional network due to the use of local block average values in this phase. Considering the size of the input images (70×70) pixels, two block sizes have been used and average values representing the image were obtained, thus resulting in 196 average values (using 5×5 pixel blocks) and 100 average values (using 7×7 pixel blocks). These block sizes were chosen because they provide sufficient representation of input images for meaningful learning while maintaining the size of the neural network as small as possible, in order to reduce computational expenses. The obtained averaged values will be used as the input to the network for both training and testing.

3.3 Neural Network Classification

During this phase two EmNNs and two BPNNs will be trained to identify the blood cells. The difference between the two EmNNs (and similarly the two BPNN) is the size of the input layer. The two emotional networks are EmNN1 (196 input neurons) and EmNN2 (100 input neurons), whereas the two conventional networks are BPNN1 (196 input neurons) and BPNN2 (100 input neurons). Each of the four networks has also one hidden layer with six neurons, which was determined after many experiments involving the adjustment of the number of hidden neurons from one neuron to 100 neurons, and output layer with three neurons corresponding to the three blood cell types. Training the four networks uses 60 non-rotated blood cell images (20 Red, 20 White, 20 Platelets). The remaining 300 blood cell images are not exposed to the neural networks during training, and are used to test/generalize the trained networks. During the learning phase, the number of hidden layer neurons, learning coefficient, and momentum rate were adjusted during various experiments in order to achieve the required minimum error value of 0.003 which was considered as sufficient for this application.

4. EXPERIMENTAL RESULTS

The experiments that have been carried out, aimed at testing and demonstrating the performances of emotional neural networks in comparison to conventional networks. The performance evaluation criteria was firstly based a network converging to the required error value of 0.003. The least number of iterations required to converge to this error value by any of the networks is marked, and then the other three networks are trained for the marked number of iterations. Training and testing the neural networks was implemented using the following system configuration: 2.8 GHz PC with 512 MB of RAM using Windows XP operating system, C-language code and Borland C^{++} compiler.

The emotional neural network EmNN1 required the least number of iterations (5396) to converge with error value 0.003. Thus all networks are trained for 5396 iterations and their final parameters, training and identification run times, as well as correct identification rates are recorded as shown in Table 1.

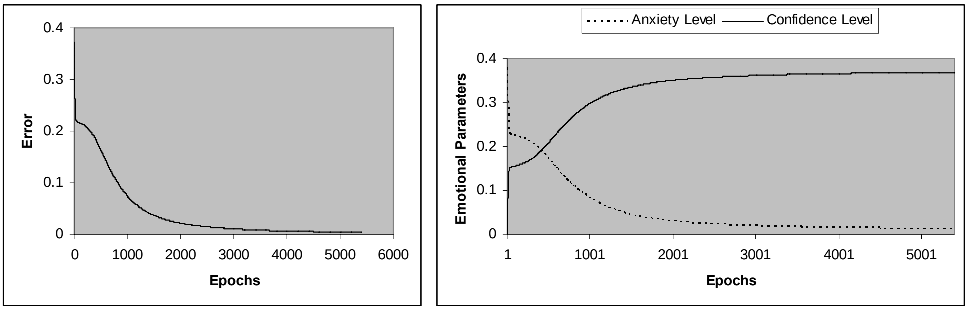
Table 1. Final simulation parameter values and correct identification rates using 300 testing set blood cell images of size (70 × 70) pixels.

Neural Networks	Emotional		Conventional	
	EmNN1	EmNN2	BPNN1	BPNN2
Averaged input pattern block size (pixel)	5 × 5	7 × 7	5 × 5	7 × 7
Input neurons	196	100	196	100
Hidden neurons	6	6	6	6
Output neurons	3	3	3	3
Learning coefficient (η)	0.0015	0.0015	0.0015	0.0015
Momentum rate (α)	0.43	0.43	0.43	0.43
Required minimum error (e)	0.003	0.003	0.003	0.003
Obtained error	0.003	0.004	0.146	0.212
Iterations	5396	5396	5396	5396
Anxiety coefficient (μ)	0.011410	0.012415	–	–
Confidence coefficient (k)	0.360583	0.368322	–	–
Training time (seconds)	115.72	91.36	93.03	76.56
Identification time (seconds)	3.1×10^{-4}	2.07×10^{-4}	3.63×10^{-4}	3.67×10^{-4}
Correct identification rate	96.33 %	97 %	66.33 %	66.33 %

When evaluating the performances of the neural networks, there are three measures to consider: firstly, the training time which is how long it takes the network to converge using the required minimum error or number of iterations criteria. Secondly, the generalization or run time (identification time) which is how long it takes a trained neural network to perform one forward pass yielding an identification output. Finally, the correct identification rate which is the most important measure, as it indicates how robust and successful the network learning was.

Both emotional networks outperform the conventional networks when considering the identification rate and time measures. Training time, however, was less using conventional networks, which is anticipated, as the emotional networks have the extra emotional weight sets. Amongst the two emotional networks, EmNN2 is chosen as the best performer due to its highest correct identification rate (97%) of 300 blood cell images that were not presented to the networks before, as well as its fastest identification time of 2.07×10^{-4} seconds. Fig. 3 shows the learning curves of EmNN2 during training.

The general outcome of these experiments reveals that the addition of the two emotional parameters has significantly improved the learning and performance of the emo-



(a) Error curve. (b) Emotional parameters curves.

Fig. 3. Emotional neural network EmNN2 learning curves (a) error minimization, (b) anxiety and confidence levels during learning.

tional neural networks in comparison to the conventional BP-based networks. This improvement is due to the way the two emotional parameters affect weight updating as the epochs progress, where one term (anxiety) tells the system to pay less and less attention to the derivative of the error of the training pattern using all nodes as the input average value of the samples of the training pattern, while the other term (confidence level) tells the system to pay more and more attention to the previous change it made to the weights, which is some sort of an increasing inertia term to modify the level of change from one pattern to the next as the training epochs progress.

5. CONCLUSIONS

This paper presented an emotional neural network based on supervised learning using the emotional back propagation learning algorithm. The paper contributes also to a growing effort to investigate modeling human emotional responses within artificial intelligent systems.

The emotional neural network has two emotional parameters simulating (anxiety and confidence) levels during the learning process, as well as two emotional neurons and their associated emotional weights. The emotional parameters and weights are updated in each epoch, using the average values of input images as visual stimuli, in addition to the error signals of each epoch. Anxiety is high when learning a new task, but is gradually reduced as the learner becomes familiar with the input patterns, and as the feedback error is minimized. Confidence, which is dependent on anxiety level, is low at start, but it increases as anxiety level decreases.

Different feature extraction methods were used to obtain local average values of non-overlap segmented blocks of the input blood cell images. The use of averaging aims at reducing computational expense while providing the neural networks with meaningful input data. Two feature extraction methods; based on different block sizes, lead to different neural network designs. Therefore, two emotional neural networks, as well as two conventional BP-based neural networks were implemented to the problem of blood cell identification, and their performances were compared. The identification of red, white and platelets blood cells by machines is not an easy task, because of the irregular shapes, different sizes and colors of the cells.

The performance evaluation and comparison of the four investigated neural networks show that emotional neural networks marginally outperform conventional neural networks. One of the two emotional networks (EmNN2) was chosen as the best performer due to its fast identification time (2.07×10^{-4} seconds) and high correct identification rate (97%). The concept of modeling human emotions in machines or robots is not impossible, as long as we do not expect the machines to "feel" the way we humans do.

Further work will include testing the presented system using deformable cases of blood cells, and exploring different application areas where emotional neural networks could be efficiently used.

REFERENCES

1. D. S. Levine, "Neural network modeling of emotion," *Physics of Life Reviews*, Vol. 4, 2007, pp. 37-63.
2. D. I. Lewinm, "Why is that computer laughing?" *IEEE Intelligent Systems*, Vol. 16, 2001, pp. 79-81.
3. J. G. Taylor, K. Scherer, and R. Cowie, "Emotion and brain: Understanding emotions and modelling their recognition," *Neural Networks*, Vol. 18, 2005, pp. 313-316.
4. W. Gnadt and S. Grossberg, "SOVEREIGN: An autonomous neural system for incrementally learning planned action sequences to navigate towards a rewarded goal," *Neural Networks*, Vol. 21, 2008, pp. 699-758.
5. L. Canamero, "Emotion understanding from the perspective of autonomous robots research," *Neural Networks*, Vol. 18, 2005, pp. 445-455.
6. E. Coutinho and A. Cangelosi, "Emotion and embodiment in cognitive agents: From instincts to music," in *Proceedings of the IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems*, 2007, pp. 133-138.
7. S. Grossberg and J. W. L. Merrill, "A neural-network model of adaptively timed reinforcement learning and hippocampal dynamics," *Cognitive Brain Research*, Vol. 1, 1992, pp. 3-38.
8. N. Fragopanagos and J. G. Taylor, "Modelling the interaction of attention and emotion," *Neurocomputing*, Vol. 69, 2006, pp. 1977-1983.
9. J. Martinez-Miranda and A. Aldea, "Emotions in human and artificial intelligence," *Computers in Human Behavior*, Vol. 21, 2005, pp. 323-341.
10. R. W. Picard, *Affective Computing*, MIT Press, Cambridge, MA, 1997.
11. S. Grossberg and W. E. Gutowski, "Neural dynamics of decision making under risk: Affective balance and cognitiveemotional interactions," *Psychological Review*, Vol. 94, 1987, pp. 300-318.
12. J. Bates, "The role of emotion in believable agents," *Communications of the ACM*, Vol. 37, 1994, pp. 122-125.
13. H. Ushida, Y. Hirayama, and H. Nakajima, "Emotion model for life-like agent and its evaluation," in *Proceedings of the 15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, 1998, pp. 62-69.
14. M. S. El-Nasr, J. Yen, and T. R. Ioerger, "FLAME-fuzzy logic adaptive model of emotions," *Autonomous Agents and Multi-Agent Systems*, Vol. 3, 2000, pp. 219-257.
15. J. Gratch, "Modelling the interplay between emotion and decision making," in *Pro-*

ceedings of the 9th Conference on Computer Generated Forces and Behavioral Representation, 2000, pp. 1-11.
16. M. Poel, R. D. Akker, A. Nijholt, and A. J. van Kesteren, "Learning emotions in virtual environments," in *Proceedings of the 16th European Meeting on Cybernetics and System Research*, Vol. 2, 2002, pp. 751-756.
17. K. Doya, "Meta-learning and neuro-modulation," *Neural Networks*, Vol. 15, 2002, pp. 495-506.
18. J. G. Taylor and N. Fragopanagos, "Modelling the interaction of attention and emotion," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2005, pp. 1663-1668.
19. W. F. Clocksin, "Memory and emotion in the cognitive architecture," *Visions of Mind*, 2004, pp. 122-139.
20. E. Coutinho, E. R. Miranda, and A. Cangelosi, "Towards a model for embodied emotions," in *Proceedings of the Portugese Conference on Artificial Intelligence*, 2005, pp. 54-63.
21. D. Canamero, "Designing emotions for activity selection in autonomous agents," *Emotions in Humans and Artifacts*, 2003, pp. 115-148.
22. S. Grossberg and D. Seidman, "Neural dynamics of autistic behaviors: Cognitive, emotional, and timing substrates," *Psychological Review*, Vol. 113, 2006, pp. 483-525.
23. K. A. Maria and R. A. Zitar, "Emotional agents: A modeling and an application," *Information and Software Technology*, Vol. 49, 2007, pp. 695-716.
24. M. I. Gobbini and J. V. Haxby, "Neural systems for recognition of familiar faces," *Neuropsychologia*, Vol. 45, 2007, pp. 32-41.
25. A. Khashman, "A modified back propagation learning algorithm with added emotional coefficients," *IEEE Transactions on Neural Networks*, Vol. 19, 2008, pp. 1896-1909.
26. D. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parallel Distributed Processing*, 1986, pp. 318-362.
27. T. Baumgartner, M. Esslen, and L. Jancke, "From emotion perception to emotion experience: Emotions evoked by pictures and classical music," *International Journal of Psychophysiology*, Vol. 60, 2006, pp. 34-43.
28. D. Huang and X. Zhao, "Research on several problems in BP neural network application," in *Proceedings of International Conference on Wireless Communications*, *Networking and Mobile Computing*, 2007, pp. 6713-6716.
29. R. P. W. Duin, "Learned from neural networks," in *Proceedings of the 6th Annual Conference of the Advanced School for Computing and Imaging*, 2000, pp. 1-30.
30. W. Lin, J. Xiao, and E. Micheli-Tzanakou, "A computational intelligence system for cell classification," in *Proceedings of IEEE International Conference on Information Technology applications to Biomedicine*, 1998, pp. 105-109.
31. W. Shitong and W. Min, "A new detection algorithm (NDA) based on fuzzy cellular neural networks for white blood cell detection," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 10, 2006, pp. 5-10.
32. B. F. Buxton, H. Abdallahi, D. Fernandez-Reyes, and W. Jarra, "Development of an extension of the Otsu algorithm for multidimensional image segmentation of thin-film blood slides," in *Proceedings of IEEE International Conference on Computing:*

Theory and Applications, 2007, pp. 552-562.
33. F. Scotti, "Robust segmentation and measurements techniques of white cells in blood microscope images," in *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, 2006, pp. 43-48.
34. W. Wang, H. Song, and Q. Zhao, "A modified watersheds image segmentation algorithm for blood cell," in *Proceedings of IEEE International Conference on Communications, Circuits and Systems*, Vol. 1, 2006, pp. 450-454.
35. F. Yi, Z. Chongxun, P. Chen, and L. Li, "White blood cell image segmentation using on-line trained neural network," in *Proceedings of IEEE Conference on Engineering in Medicine and Biology*, 2005, pp. 6476-6479.
36. N. Theera-Umpon and S. Dhompongsa, "Morphological granulometric features of nucleus in automatic bone marrow white blood cell classification," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 11, 2007, pp. 353-359.
37. S. Mircic and N. Jorgovanovic, "Application of neural network for automatic classification of leukocytes," in *Proceedings of IEEE 8th Seminar on Neural Network Applications in Electrical Engineering*, 2006, pp. 141-144.
38. T. Markiewicz, S. Osowski, B. Marianska, and L. Moszczynski, "Automatic recognition of the blood cells of myelogenous leukemia using SVM," in *Proceedings of International Joint Conference on Neural Networks*, 2005, pp. 2496-2501.
39. Q. Zheng, B. K. Milthorpe, and A. S. Jones, "Direct neural network application for automated cell recognition," *Cytometry*, Vol. 57, 2004, pp. 1-9.
40. A. Khashman, "IBCIS: Intelligent blood cell identification system," *Progress in Natural Science*, Vol. 18, 2008, pp. 1309-1314.
41. The Franklin Institute, "Resources for science learning," http://www.fi.edu/learn/heart/blood/blood.html, 2008.

Adnan Khashman received his Ph.D. and M.S. degrees in Electronic Engineering from University of Nottingham, England, UK, in 1992 and 1997, respectively, and his B.S. degree in Electronic and Communication Engineering from University of Birmingham, England, UK, in 1991. During 1998-2001 he was an Assistant Professor and the Chairman of Computer Engineering Department, Near East University, Lefkosa, Turkey. Since 2001 he is An Associate Professor and Chairman of Electrical and Electronic Engineering Department at the same university. From 2007 till 2008 he was also the Vice-Dean of Engineering Faculty. He is the founder (in 2001) and Head of the Intelligent Systems Research Group (ISRG) at the same university. As of 2009 he is a full Professor at the Engineering Faculty. His current research interests include neural networks and their engineering applications, intelligent systems and pattern recognition.