Available online at www.sciencedirect.com
SCIENCE DIRECT®
Applied Soft Computing xxx (2004) xxx–xxx
Applied Soft Computing
www.elsevier.com/locate/asoc

# Neural network-based colonoscopic diagnosis using on-line learning and differential evolution

George D. Magoulas [a,*], Vassilis P. Plagianakos [b,1], Michael N. Vrahatis [b,1]

[a] *School of Computer Science and Information Systems, Birkbeck College, University of London, Malet Street, London WC1E 7HX, UK*
[b] *Department of Mathematics and University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece*

Received 3 September 2002; received in revised form 12 March 2003; accepted 14 January 2004

**Abstract**

In this paper, on-line training of neural networks is investigated in the context of computer-assisted colonoscopic diagnosis. A memory-based adaptation of the learning rate for the on-line back-propagation (BP) is proposed and used to seed an on-line evolution process that applies a differential evolution (DE) strategy to (re-) adapt the neural network to modified environmental conditions. Our approach looks at on-line training from the perspective of tracking the changing location of an approximate solution of a pattern-based, and thus, dynamically changing, error function. The proposed hybrid strategy is compared with other standard training methods that have traditionally been used for training neural networks off-line. Results in interpreting colonoscopy images and frames of video sequences are promising and suggest that networks trained with this strategy detect malignant regions of interest with accuracy.


*Keywords:* Minimally invasive imaging procedures; Back-propagation networks; Medical image interpretation; On-line learning; Differential evolution strategies; Artificial evolution

## 1. Introduction

In medical practice, endoscopic diagnosis and other minimally invasive imaging procedures, such as computed tomography, ultrasonography, confocal microscopy, computed radiography, or magnetic resonance imaging, are now permitting visualisation of previously inaccessible regions of the body. Their objective is to increase the expert's ability in identifying malignant regions and decrease the need for intervention while maintaining the ability for accurate diagnosis. Furthermore, it may be possible to examine a larger area, studying living tissue in vivo—possibly at a distance [5]—and, thus, minimise the shortcomings of biopsies, such as limited number of tissue samples, delay in diagnosis and discomfort for the patient.

In this paper, we focus on neural network-assisted diagnosis of colonoscopy images. Colonoscopy is a minimally invasive technique for the production of images of the colon: a narrow pipe like structure, an endoscope, is passed into the patient's body. Video endoscopes have small cameras in their tips, when passed into a body, what the camera observes is displayed on a television monitor (see Fig. 1 for frame samples of a video sequence). The physician controls

* Corresponding author. Tel.: +44-20-7631-6717; fax: +44-20-7631-6727.
*E-mail addresses:* gmagoulas@dcs.bbk.ac.uk (G.D. Magoulas), vpp@math.upatras.gr (V.P. Plagianakos), vrahatis@math.upatras.gr (M.N. Vrahatis).
[1] Tel.: +30-61-997374, fax: ++30-61-992965.

1568-4946/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2004.01.005

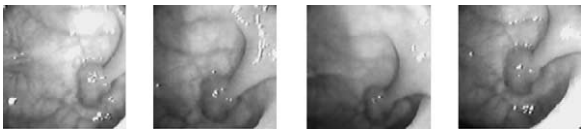

Fig. 1. Frames of a video sequence showing a polypoid tumor of the colon.

the endoscope's direction using wheels and buttons and the whole procedure is carried out under variable perceptual conditions (shadings, shadows, lighting condition variations, reflections, etc.).

Neural network-based methodologies present some interesting qualities, such as learning from experience, generalisation, and are able to handle uncertainty and ambiguity in distorted or noisy images to some extent. Thus, these methods provide human experts with significant assistance in medical diagnosis [8,10,12,13,22].

The use of neural networks for the detection of malignant regions in colonoscopy video sequences encounters several problems: the time varying nature of the process; changes in the perceptual direction of the physician; variations in the diffused light conditions. In most of these cases, off-line learning or knowledge-based approaches are not able to represent all possible variations of the environment. On-line training and retraining allow the network to update its weights during operation by taking into account both the already stored knowledge and the knowledge extracted from the current data, and are proposed as alternatives to batch learning-based approaches. Of course, the main challenge when dealing with adaptive techniques for learning, such as on-line training and retraining, is to balance the information related to recently acquired data with the information already embodied in the network [3,6,26].

Thus, in this paper we explore on-line training and retraining of neural networks with the aim to detect malignant regions in colonoscopy images though a formulation of the problem that is based on the idea of tracking the moving "optimum" of a dynamically changing pattern-based error measure. This approach coincides with the way adaptation on the evolutionary time scale is considered [28], and allows us to explore and expand further research on the tracking performance of evolution strategies (ESs) and genetic algorithms (GAs) [2,28,34]. Hence, the reader should keep in mind that in this paper we do not seek global minimisers of the error function, but we are interested in developing an on-line evolution strategy that will converge to an approximation of the optimum solution (the interesting topic of finding global minimisers in neural networks training is described elsewhere [23]).

The paper is organised as follows. Section 2 explains how textural variations of the tissue are modelled in our approach. Section 3 discusses existing learning approaches, while Section 4 describes the proposed on-line evolution strategy. In Section 5 experimental results are presented and findings are discussed. Lastly, conclusions are drawn in Section 6.

## 2. Tissue classification for endoscopic diagnosis

In endoscopic diagnosis, the medical expert, based on a distributed percept of local changes, interprets the physical surface properties of the tissue—such as roughness or smoothness, regularity, and shape—to detect abnormalities. It is important to note, however, the vast difficulties in physical attributes of the organs. For example, in colonoscopy, no two colons are alike. Even within the same colon, one section may have very different characteristics from another. Adjacent regions of the colon lining showing different properties are distinguished on the basis of the textural variations of their tissue (pit patterns) [21]. These difficulties introduce severe limitations in the use of computer-assisted endoscopic diagnosis [13,22]. Given a medical image, the "true" features associated with the physical surface properties of the tissue are not exactly known to the image-interpretation system developer. Usually, one or more feature-extraction models [15,16] are used to provide values for each feature's parameters. The findings are then used to infer the correct interpretation. On this same task of interpretation on the basis of local changes of the properties of the tissue under examination, the performance of human perception is considered outstanding. Furthermore, medical experts have the ability to either add or remove components from an image to give meaning to what they see. Medical experts can also adapt to changes to the extent that even a distorted image can be recognised.

In computerised systems, the classification of image regions is usually quite sophisticated and involves

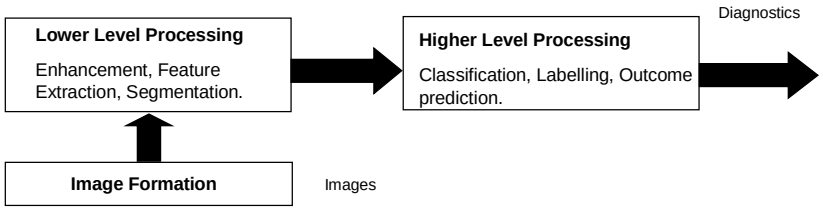


Fig. 2. Model for diagnostic system that uses medical images.

multiple levels of processing. In general, a model with three stages is employed as shown in Fig. 2 (adapted from [15]).

The lower-level processing takes image pixels as input and performs various tasks such as image enhancement, feature extraction and image segmentation. The higher-level processing takes the output from the lower-level processing as input and generates output related to medical diagnostics. Tasks accomplished in the higher-level processing include classification of features, detection of specific lesions and diagnosis for various abnormalities.

An important stage of the implementation is the feature-extraction process (see Fig. 3). In our experiments the method of co-occurrence matrices was used for feature extraction. Co-occurrence matrices, [9], represent the spatial distribution and the dependence of the grey levels within a local area. Each $p(i, j)$ entry of the matrices, represents the probability of going from one pixel with a grey level ($i$) to another with a grey level ($j$) under a predefined distance and angle. From these matrices several sets of statistical measures, or feature vectors, are computed to build different texture models. In our implementation, the colonoscopy image was separated into windows of size $16 \times 16$ pixels with 8 pixels overlap. Then the co-occurrence matrices algorithm was used to gather information from the pixels of an image window. Four angles, namely $0^\circ$, $45^\circ$, $90^\circ$, $135^\circ$, were considered as well as a predefined distance of one pixel in the formation of the co-occurrence matrices. Therefore, four co-occurrence matrices using the following four statistical measures were formed (see [12] for details):

Energy-angular second moment :

$$f_1 = \sum_i \sum_j p(i, j), \tag{1}$$

Correlation :

$$f_2 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i \times j)\, p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}, \tag{2}$$

Inverse difference moment :

$$f_3 = \sum_i \sum_j \frac{1}{1 + (i - j)}\, p(i, j), \tag{3}$$

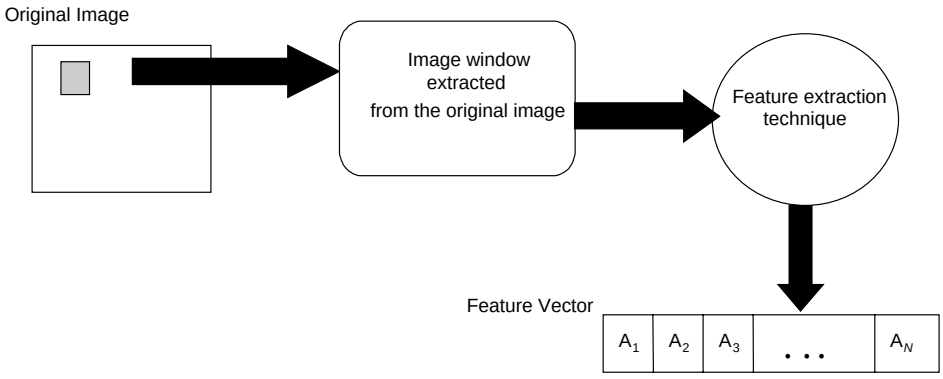


Fig. 3. Stages in feature extraction.

$$\text{Entropy}: \quad f_4 = -\sum_i \sum_j p(i,j)\log(p(i,j)), \qquad (4)$$

where $N_g$ is the number of grey levels; $\mu_x$, $\mu_y$ are the marginal mean values of $x$ (along the horizontal pixel axis) and $y$ (along the vertical pixel axis), respectively, and $\sigma_x$, $\sigma_y$ are the corresponding standard deviations. Thus, a set of 16 features describing spatial distribution in each window is obtained and used to formulate inputs for high level processing.

## 3. Batch learning of multilayer perceptrons

The most popular neural network model is the so-called multi-layer perceptron (MLP). In an MLP, whose $l$th layer contains $N_l$ nodes ($l = 1, \ldots, M$), artificial neurons operate according to the following equations:

$$\text{net}_j^l = \sum_{i=1}^{N_l} w_{ij}^{l-1,l} y_i^{l-1} \qquad (5)$$

$$y_j^l = f(\text{net}_j^l) \qquad (6)$$

where $\text{net}_j^l$ is, for the $j$th neuron in the $l$th layer ($j = 1, \ldots, N_l$), the sum of its weighted inputs. The weights for connections from the $i$th neuron at the ($l-1$) layer to the $j$th neuron at the $l$th layer are denoted by; is the output of the $j$th neuron that belongs to the $l$th layer, and the logistic function is the $j$th's neuron non-linear activation function.

Training an MLP to recognise abnormalities in image regions is typically realised by adopting an error correction strategy that adjusts the network weights through minimisation of learning error:

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{N_M}(y_{j,p}^M - t_{j,p})^2 = \frac{1}{2}\sum_{p=1}^{P} E_p \qquad (7)$$

where $(y_{j,p}^M - t_{j,p})^2$ is the squared difference between the actual output value at the $j$th output layer neuron, for an input sample $p$, and the target output value; $p$ is an index over input–output patterns.

A variety of approaches adapted from the theory of unconstrained optimisation have been applied to the minimisation of function $E$. For example, let us consider the class of batch learning algorithms that adjust the weights according to the iterative scheme:

$$w^{k+1} = w^k + \eta\varphi^k \qquad k = 0, 1, 2, \ldots \qquad (8)$$

Note that in (8) the weights of the MLP are expressed in a simplified form using vector notation. Thus, $w^k$ defines the current weight vector; $\varphi^k$, a correction term; and $\eta$ is the learning rate at the $k$th iteration. Various choices of the correction term $\varphi^k$ give rise to distinct batch learning algorithms, which are usually classified as *first-order* or *second-order* algorithms depending on the derivative-related information they use to generate the correction term. Thus, first-order algorithms are based on the first derivative of the learning error with respect to the weights, while second-order algorithms on the second derivative (see [4] for a review on first-order and second-order training algorithms).

A broad class of batch-type first-order algorithms, which are considered much simpler to implement than second-order methods, uses the correction term $\varphi^k = -\nabla E(w^k)$. The term $\nabla E(w^k)$ defines the gradient vector of the MLP and is obtained by means of back-propagation (BP) of the error through the layers of the network. The most popular algorithm of this class, called batch back-propagation applies the steepest descent method with a constant, heuristically chosen, learning rate $\eta$ that usually takes values in the interval (0, 1) [16]. Values in this interval are considered small enough to ensure the convergence of the BP training algorithm and consequently the success of learning [4]. However, it is well known that this practice tends to be inefficient [4,16] and the use of adaptive learning rate strategies is suggested in order to accelerate the learning process (see [4,18] for reviews on adaptive learning rate algorithms).

With regards to second-order training algorithms, non-linear conjugate gradient methods, such as the Fletcher-Reeves or the Polak-Ribiere methods [20], or variable metric methods, such as the Broyden–Fletcher–Goldfarb–Shanno method [4], or even modification of Newton's method [7,17] have been proposed in the literature. These methods exploit derivative calculations and subminimisation procedures (e.g. the non-linear conjugate gradient methods) and/or approximations of various matrices (e.g. the Hessian matrix for the variable metric or

quasi-Newton methods) to accelerate the learning process.

## 4. On-line evolution strategy

On-line training in neural networks is related to updating the network parameters after the presentation of each training example, which may be sampled with or without repetition. On-line training may be the appropriate choice for learning a task either because of the very large (or even redundant) training set, or because of the slowly time-varying nature of the task. Although batch training seems faster for small-size training sets and networks, on-line training is probably more efficient for large training sets and networks. It helps to escape local minima and provides a more natural approach to learning in non-stationary environments. On-line methods seem to be more robust than batch methods as errors, omissions or redundant data in the training set can be corrected or ejected during the training phase. Additionally, training data can often be generated easily and in great quantities when the system is in operation, whereas they are usually scarce and precious before. Lastly, on-line training is necessary in order to learn and track time varying functions and continuously (re-) adapt in a changing environment.

Despite the abundance of methods for learning from examples, there are only few that can be used effectively for on-line learning. For example, the classic batch training algorithms cannot straightforwardly handle nonstationary data. Even when some of them are used in on-line training there exists the problem of "catastrophic interference", in which training on new examples interferes excessively with previously learned examples leading to saturation and slow convergence [3,33]. Below we present an on-line BP-seeded *differential evolution* (DE) strategy for on-line neural network training. Firstly, we briefly present the on-line BP learning stage of the proposed strategy. Then we proceed by describing the on-line DE stage. Note that the description below focuses on the problem of adapting the weights on-line, assuming that on-line evolution is always activated, and does not require the input and desired output data to be known a priori. Our experiments, reported in the next section, were also conducted under the same assumptions to test the robustness of our approach. Note, however, that in practice, whenever the changes of the environment are not considered significant and the performance is satisfactory, the weights and structure of the network should remain constant [6].

### 4.1. On-line back-propagation learning

On-line BP learning strategies are usually based on the use of stochastic gradient descent due to the inherent efficiency of this method in time-varying environments [1,29,30,32,33]. On-line learning has been analysed within the framework of statistics and it has been shown that it is asymptotically as effective as batch (also called *off-line*) learning. However, sensitivity to learning parameters is a common drawback of these schemes [27]. Advanced optimisation methods, such as conjugate gradient, variable metric, simulated annealing etc., cannot be used in this context, as they rely on a fixed error surface and need information from the whole training set [27].

In [19], a variant of the on-line BP has been proposed. The method can be considered as a *meta-learning algorithm* in the sense that it learns the learning rate parameters of an underlying base learning system (i.e. of the stochastic gradient descent). To this end, the new variant uses a memory-based learning rate adaptation schedule that exploits gradient related information from the current as well as the two previous pattern presentations:

$$\eta^{k+1} = \eta^k + \gamma_1 \langle \nabla E_{p-1}(w^{k-1}), \nabla E_p(w^k) \rangle + \gamma_2 \langle \nabla E_{p-2}(w^{k-2}), \nabla E_{p-1}(w^{k-1}) \rangle \quad (9)$$

At the start of the learning procedure, $k = 0$, the learning rate is set to a small positive value; e.g. the initial learning rate was set to 0.001 in our experiments. Then, the weights are updated on-line, for each pattern $p$, following the iterative scheme:

$$w^{k+1} = w^k - \eta^k \nabla E_p(w^k) \quad (10)$$

In (9) $\langle .,. \rangle$ stands for the usual inner product in $\boldsymbol{R}^n$; $E_p$, the *pattern*-based error measure; $\nabla E_p$, the corresponding gradient vector; $\eta$, the learning rate; and $\gamma_1$, $\gamma_2$ are the meta-learning rates ($\gamma_1 < \gamma_2 < 1$). Meta-learning rates are also in use by other on-line learning schemes, such as in [1,29,30,32], and can take various forms depending on the method. Previous

experiments with the new variant have shown that the scheme of Eq. (9) seems to provide additional stabilisation in the calculated values of the learning rate, and helps the stochastic gradient descent to exhibit fast convergence and high success rate [19]. In addition, the method is characterised by low storage requirements and inexpensive computations, as it only uses already calculated information from the current, as well as the previous iteration. The idea of considering the gradient of the previous iterations in a learning rate adaptation scheme has also been proposed in the context of off-line learning. Particularly, Jacobs in the *delta-bar-delta* algorithm, [11], measures the running average of the current, $\nabla E(w^k)$, and past partial derivatives in order to check whether the current gradient has the same sign as the average gradient. Then the algorithm either increases the learning rate by adding a positive constant to the current value, or decreases it by multiplying the current value with a positive, smaller than one, constant. Finally, the weights are updated using a variant of Eq. (8), as the delta-bar-delta algorithm needs information from the whole training set (i.e. it performs batch learning).

The on-line iterative scheme of (9)–(10) was shown to provide increased speed and higher possibility of good performance in different classes of problems when compared against the classic on-line BP and other meta-learning rate algorithms (see [19,24] for details and comparisons). The role of on-line BP in the context of computer-assisted colonoscopic diagnosis is to initialise the population of the DE strategy with an initial approximation of the solution, as will be described below.

### 4.2. Differential evolution strategy

Evolution strategies are adaptive stochastic search methods that mimic the metaphor of natural biological evolution. The main differences between ESs and genetic algorithms lie in that the self-adaptation of the mutation operator is a key feature of the ESs, and in that GAs prefer smaller mutation probability (rate) [2,28]. Here we use the Differential evolution strategies, which have been designed as stochastic parallel direct search methods that can handle non-differentiable, non-linear, and multimodal objective functions efficiently, and require few easily chosen control parameters [31]. Experimental results have shown that DE strategies have good convergence properties and outperform other evolutionary algorithms and annealing methods [31]. To apply DE strategies to neural network training we start with a specific number (*NP*) of *n*-dimensional weight vectors, as initial population, and evolve them over time; *NP* is fixed throughout the training process and the weight population is initialised by perturbing the approximate solution provided by the on-line BP (see Relations (9)–(10)). Thus, the on-line BP seeds the DE, so the initial population might be generated by adding normally distributed random deviations to the nominal solution.

Let us now describe the proposed version of DE strategy that is used in the on-line evolution strategy. The weight vectors evolve randomly with each pattern presentation (iteration) through the relation

$$v_i^{k+1} = w_i^k + \mu(w_{\text{best}}^k - w_i^k + w_{r1} - w_{r2}),$$
$$i = 1, \ldots, NP, \tag{11}$$

where $w_{\text{best}}^k$ is the best population member of the previous iteration, $\mu > 0$ is a real parameter (mutation constant) which regulates the contribution of the difference between weight vectors, and $w_{r_1}, w_{r_2}$ are weight vectors randomly chosen from the population with $r_1, r_2 \in \{1, 2, \ldots, i-1, i+1, \ldots, NP\}$, i.e. $r_1, r_2$ are random integers mutually different from the running index *i*. Aiming at increasing the diversity of the weight vectors further, a crossover-type operation is introduced in Relation (12). Thus, the so-called *trial* vector, $u_i^{k+1}$, $i = 1, \ldots, NP$, is generated.

$$u_{i,j}^{k+1} = \begin{cases} v_{i,j}^{k+1} & \text{if } r_j \le \rho \quad \text{or} \quad j = \text{rand}_i \\ w_{i,j}^k & \text{if } r_j > \rho \quad \text{and} \quad j \ne \text{rand}_i \end{cases} \tag{12}$$

This operation works as follows: the *mutant* weight vectors, $v_i^{k+1}, i = 1, \ldots NP$, are mixed with the "*target*" vectors, $w_i^k$. Specifically, we randomly choose a real number $r$ in the interval [0,1] for each component $j$, $j = 1, 2, \ldots, n$, of $v_i^{k+1}$. This number is compared with $\rho \in [0, 1]$ (crossover constant), and if $r \le \rho$ then the $j$th component of the trial vector $u_i^{k+1}$ gets the value of the $j$th component of the mutant vector, $v_i^{k+1}$; otherwise, it gets the value of the $j$th component of the target vector, $w_i^k$. In (12), $\text{rand}_i$ is a randomly selected index that is used to ensure the trial vector has at least one component from the

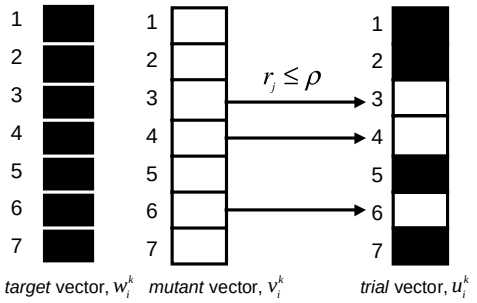


Fig. 4. Illustration of the crossover operation.

mutant vector. An application example of this operation is shown in Fig. 4 for a seven-dimensional weight vector.

The trial vector is accepted for the next iteration if and only if it reduces the value of the pattern-based error measure $E_p$; otherwise the old value, $w_i^k$, is retained. This last operation, called *selection*, ensures that the fitness starts steadily decreasing at some iteration, and is described in Relation (13).

$$w_i^{k+1} = \begin{cases} u_i^{k+1} & \text{if } E_p(u_i^{k+1}) < E_p(w_i^k) \\ w_i^{k+1} & \text{if } E_p(u_i^{k+1}) \geq E_p(w_i^k) \end{cases} \quad (13)$$

The combined action of mutation and crossover operation is responsible for much of the effectiveness of DE search, and allows DE strategies to act as parallel, noise-tolerant hill-climbing algorithms, which efficiently search the whole space for solutions [31].

## 5. Experiments and results

In our experiments, the colonoscopy images and video frames were separated into windows of size $16 \times 16$ pixels with overlap of 8 pixels. Then the co-occurrence matrices algorithm was applied to gather information regarding pixel neighbourhoods of randomly selected image windows, as described in Section 2. The procedure results in 16-dimensional feature vectors, which are very noisy as no pre-filtering or segmentation techniques is applied, and are used in the experiments described below. The learning parameters of the on-line evolution strategy have been set following the recommendations of [24,31]: $\gamma_1 = 0.05$, $\gamma_2 = 0.95$, $\mu = 0.5$, $\rho = 0.9$. Lastly, $NP = 100$. In the first set of experiments, 1000 MLPs with varying number of hidden nodes (from 8 to 21) were trained using two batch learning algorithms, the adaptive learning rate back-propagation (ABP) proposed by Vogl et al. [35], and the Levenberg–Marquardt (LM) method [7], as typical examples of first- and second-order training algorithms, respectively. The MLPs were trained using 10 normal/10 abnormal samples from image windows that were randomly extracted from Images (a) and (b) (see Fig. 5) and tested with different tissue samples taken from the two images. Note that the malignant regions in these images belong to two different types: Image (a) is a low grade cancer, while Image (b) is a moderately differentiated carcinoma [14]. The performance of the trained MLPs has been tested on a set of 80 texture samples (40 normal and 40 malignant) randomly selected from the two images and different from the training set. Only a small sample out of 1000 trained MLPs of the different architectures exhibited classification success of 90% or higher. Detailed results are shown in Fig. 6. More specifically, only 150 MLPs with 8 hidden nodes, out of the 1000 trained, exhibited the desired classification success (see Fig. 6, left part). Note also the significant difference in the number of the MLPs with acceptable classification success among the ABP and the LM trained ones. The LM algorithm also reveals a higher average percentage of classification success, as shown on the right part of Fig. 6. In fact MLPs with 11 hidden nodes exhibit the highest average in classification success (96.75%). Thus, 11 hidden node MLPs were used in the second set of experiments.

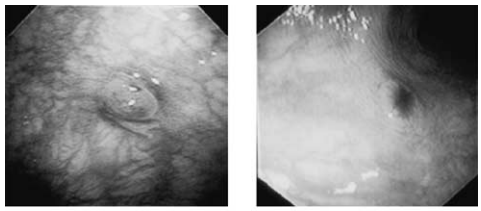


Fig. 5. Colonoscopy images used in the experiments.

In the second set of experiments, 1000 MLPs of 16-11-2 architecture were trained off-line to detect malignant regions in a frame of colonoscopy video sequence using a training set of 150 normal/150 abnormal patterns. Three batch-learning methods were

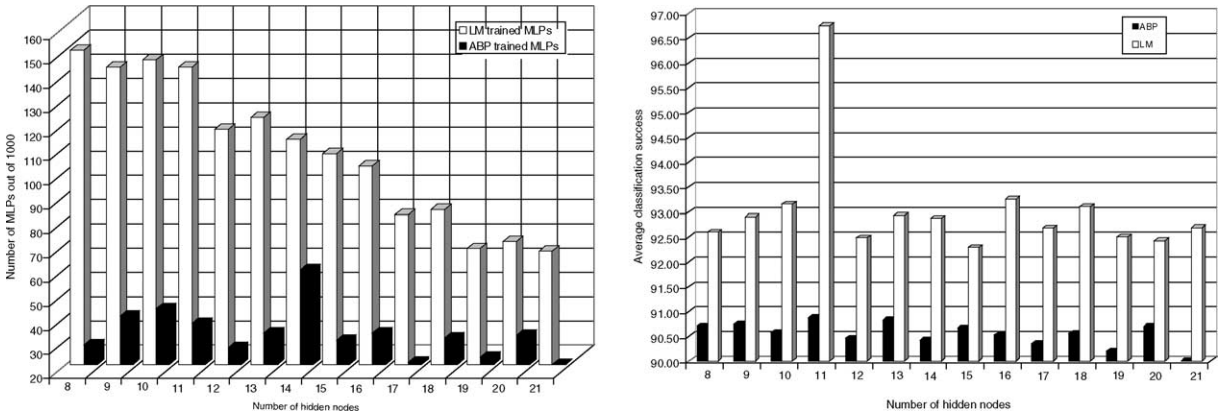


Fig. 6. Number of MLPs (out of 1000) with classification success greater than 89% (left), and average classification success (right) for these MLPs. Results are for the images of Fig. 5.

comparatively evaluated in this round of experiments: the Levenberg–Marquardt that exhibited good performance in the previous round, the scaled conjugate gradient (SCG) method, [20], that is considered according to the literature as a good alternative to the use of second-order methods [16], and the Rprop algorithm, [25], which is a first-order method that applies heuristics to adapt a different learning rate for each weight of the network and combines successfully effectiveness with low computational requirements [16]. The percentage of classification success in testing (test set included 3969 patterns, i.e. the whole region covered in the video frame) for the 1000 trained networks is shown in Fig. 7. One can observe that it is not easy to locate weights that will allow the networks to detect malignant regions with a success of over 90%. For example, in Fig. 7, only two networks out of the 1000 trained with the Rprop algorithm achieved recognition success from 90 to 100%. For the SCG method the corresponding number is 3 out of 1000, while for the LM method this number is slightly higher, as 6 out of the 1000 networks exhibited classification success between 90 and 100%. The best result for each training method is: 92% for the Rprop, 92.4% for the LM and 92.6% for the SCG. Rprop needs on average more epochs to converge than the SCG and LM

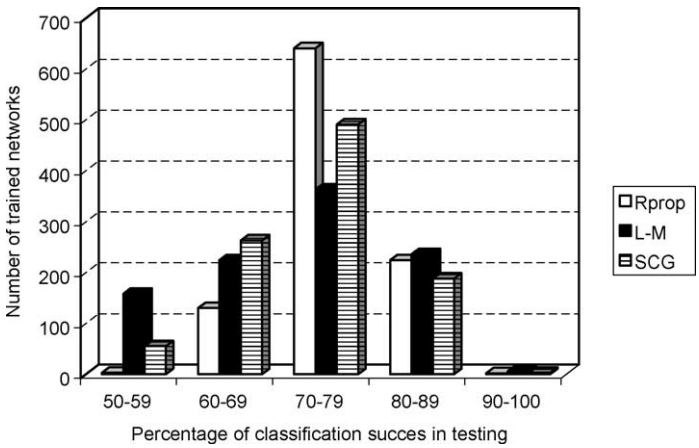


Fig. 7. Generalisation results for three batch-training algorithms.

Table 1
Best classification success for two first-order batch-learning methods

| Method | Frame 1 (%) | Frame 2 (%) | Frame 3 (%) | Frame 4 (%) |
|---|---|---|---|---|
| Rprop | 92 | 91 | 92 | 93 |
| ABP | 81 | 85 | 83 | 81 |

methods but does not require heavy matrix computations or subminimisations. As a consequence, it was observed that the average time for training with Rprop was shorter than the corresponding time of SCG or LM. Thus, we decided to keep Rprop and run experiments with data from other video frames of the same video sequence. The best results are summarised in Table 1.

From the results of Table 1, it is clear that Rprop exhibits the best overall performance compared with ABP. Note that the results of Table 1 have been achieved by training off-line frame-dedicated MLPs with 11 hidden nodes using 300 patterns randomly chosen from each frame and testing using data of the same frame.

In the third set of experiments, the Rprop algorithm was compared with the classic on-line BP using data from another frame of the same video sequence. Three hundred patterns were used for training and 3969 for testing (i.e. the whole tissue region covered in the frame). The capability of the trained network (16-11-2 MLPs were used) with the best performance in assigning appropriate characterisations (normal/abnormal) to frame regions is shown in Table 2.

The Rprop reveals, in general, a higher percentage of success than the on-line BP. The reader should, of course, keep in mind that Rprop minimises a batch error measure, i.e. it uses the true gradient of the error function as it exploits information from all the training patterns. The on-line BP, on the other hand, minimises a pattern-based error measure and works with an instantaneous approximation of the true gradient because information from only one pattern is used at each iteration. Therefore, on-line BP can be used for (re-) adapting to modified environment conditions, while Rprop requires all information about input-output patterns to be known a priori and, thus, fails to work when all the relevant features of the environment are not explicitly defined in advance. However, the results of the experiments made clear that the classic on-line BP needs further improvement in order to train networks to detect malignant regions with accuracy comparable to batch training methods.

Table 2
Best performance in terms of generalisation for Rprop and on-line BP

| Method | Abnormal (%) | Normal (%) | Mean (%) |
|---|---|---|---|
| Rprop | 83 | 96 | 93 |
| On-line BP | 73 | 93 | 88 |

In the fourth set of experiments, 16-11-2 MLPs have been trained on-line to detect malignant regions in a set of four frames from the same video sequence. The frames used in the two previous experiments were included in the set. The networks have been trained on-line, following the iterative scheme (9) for adapting the learning rate, to recognise patterns from the first frame. Then on-line learning with differential evolution occurred as data from the second frame appeared at the input. The on-line evolution learning strategy continuously adapts the network as patterns from other frames are presented in random order at the input. In total, 1200 patterns from the four frames of the video sequence were presented to the network during the training phase. The network was then tested using 15876 patterns from the four frames (4000 patterns approximately cover the whole tissue region of a frame and include normal as well as malignant areas). The average capability of the trained networks in assigning appropriate characterisations to explored colon lining regions is presented in Table 3.

The on-line BP seeded DE scheme provides generalisation results close to the best results obtained by the batch training methods, as reported in the previous experiments. For example, the best SCG-trained dedicated network in the second experiment (trained off-line and tested using data from Frame 1) had 92.6% success, and the best Rprop-trained dedicated network

Table 3
Normal/abnormal detection accuracy

| Method | Frame 1 (%) | Frame 2 (%) | Frame 3 (%) | Frame 4 (%) |
|---|---|---|---|---|
| On-line BP | 83 | 84 | 77 | 88 |
| On-line BP seeded DE | 93 | 92 | 84 | 90 |

in the third experiment (trained off-line and tested using data from Frame 2) had 93% success.

## 6. Conclusions and future work

In this paper a new scheme for neural network-based colonoscopic diagnosis was introduced. The proposed on-line evolution strategy can be considered as a hybrid algorithm. It uses an on-line back-propagation strategy with adaptive learning rate to seed the initial population of the on-line Differential Evolution strategy. In our experiments, neural networks trained with the proposed on-line evolution strategy exhibited satisfactory performance under changing environmental conditions, as data from different frames were presented to the network.

In the reported experiments no emphasis was put in fine-tuning the heuristic parameters of our scheme; classic values found in the relevant literature of differential evolution strategy were used instead. In future work we will fully investigate the properties, study the effect of the heuristic parameters and evaluate the full potential of the hybrid learning strategy in colonoscopic diagnosis by means of extensive testing on long video sequences and interpretation of complex tissue regions.

## Acknowledgements

The authors gratefully acknowledge the contribution of Dr. S. Karkanis (Technological Educational Institute of Lamia, Greece) and D. Iakovidis (University of Athens, Greece) in the acquisition of video sequences and extraction of features.

## References

[1] L.B. Almeida, T. Langlois, J.D. Amaral, A. Plankhov, Parameter adaptation in stochastic optimisation, in: D. Saad (Ed.), Proceedings of the On-line Learning in Neural Networks, Cambridge University Press, 1998, pp. 111–134.

[2] P. Angeline, Tracking extrema in dynamic environments, in: Proceedings of the Sixth Annual Conference on Evolutionary Programming VI, Springer, 1997, pp. 335–345.

[3] D. Anguita, Smart adaptive systems: state of the art and future directions of research, in: Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems-EUNITE 2001, Tenerife, Spain, 2001, pp. 1–4.

[4] R. Battiti, First- and second-order methods for learning: between steepest descent and Newton's method, Neural Comput. 4 (1992) 141–166.

[5] P.M. Delaney, G.D. Papworth, R.G. King, Fibre optic confocal imaging (FOCI) for in vivo subsurface microscopy of the colon, in: V.R. Preedy, R.R. Watson (Eds.), Methods in Disease: Investigating the Gastrointestinal Tract, Greenwich Medical Media, London, UK, 1998.

[6] A.D. Doulamis, N.D. Doulamis, S.D. Kollias, On-line retrainable neural networks: improving the performance of neural networks in image analysis problems, IEEE Trans. Neural Networks 11 (2000) 137–155.

[7] M. Hagan, M. Menhaj, Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Networks 5 (1994) 989–993.

[8] R. Hanka, T.P. Harte, A.K. Dixon, D.J. Lomas, P.D. Britton, Neural networks in the interpretation of contrast-enhanced magnetic resonance images of the breast, Healthcare Computing 1996, Harrogate, UK, 1996, pp. 275–283.

[9] R.M. Haralick, Statistical and structural approaches to texture, IEEE Proc. 67 (1979) 786–804.

[10] P.R. Innocent, M. Barnes, R. John, Application of the fuzzy ART/MAP and MinMax/MAP neural network models to radiographic image classification, Artif. Intell. Med. 11 (1997) 241–263.

[11] R. Jacobs, Increased rates of convergence through learning rate adaptation, Neural Networks 1 (4) (1988) 295–307.

[12] S. Karkanis, G.D. Magoulas, N. Theofanous, Image recognition and neuronal networks: intelligent systems for the improvement of imaging information, Minim. Invasive Ther. Allied Technol. 9 (2000) 225–230.

[13] S.A. Karkanis, G.D. Magoulas, D.K. Iakovidis, D.A. Karras, D.E. Maroulis, Evaluation of textural feature extraction schemes for neural network-based interpretation of regions in medical images, in: Proceedings of IEEE International Conference on Image Processing, Thessaloniki, Greece, 7–10 October 2001.

[14] S.E. Kudo, H. Kashida, T. Tamura, E. Kogure, Y. Imai, H. Yamano, A.R. Hart, Colonoscopic diagnosis and management of nonpolypoid early colorectal cancer, World J. Surg. 24 (9) (2000) 1081–1090.

[15] C.T. Leondes, Image processing and pattern recognition, Neural Network Systems Techniques and Applications, Series 5, Academic Press, 1998.

[16] C.G. Looney, Pattern Recognition using Neural Networks, Oxford University Press, Oxford, UK, 1997.

[17] G.D. Magoulas, M.N. Vrahatis, T.N. Grapsa, G.S. Androulakis, Neural network supervised training based on a dimension reducing method, in: S.W. Ellacot, J.C. Mason, I.J. Anderson (Eds.), Mathematics of Neural Networks: Models, Algorithms and Applications, Kluwer, 1997, pp. 245–249.

[18] G.D. Magoulas, M.N. Vrahatis, G.S. Androulakis, Improving the convergence of the backpropagation algorithm using learning rate adaptation methods, Neural Comput. 11 (1999) 1769–1796.

[19] G.D. Magoulas, V.P. Plagianakos, M.N. Vrahatis, Adaptive stepsize algorithms for on-line training of neural networks, Nonlin. Anal.: Theor. Methods Appl. 47 (2001) 3425–3430.
[20] M. Möller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 525–533.
[21] S. Nagata, S. Tanaka, K. Haruma, M. Yoshihara, K. Sumii, G. Kajiyama, F. Shimamoto, Pit pattern diagnosis of early colorectal carcinoma by magnifying colonoscopy: clinical and histological implications, Int. J. Oncol. 16 (2000) 927–934.
[22] S.J. Phee, W.S. Ng, I.M. Chen, F. Seow-Choen, B.L. Davies, Automation of colonoscopy part II: visual-control aspects, IEEE Eng. Med. Biol. 17 (3) (1998) 81–88.
[23] V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, Supervised training using global search methods, in: N. Hadjisavvas, P. Pardalos (Eds.), Advances in Convex Analysis and Global Optimisation, Noncovex Optimization and its Applications, vol. 54, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 421–432.
[24] V.P. Plagianakos, G.D. Magoulas, M.N. Vrahatis, Learning rate adaptation in stochastic gradient descent, in: N. Hadjisavvas, P. Pardalos (Eds.), Advances in Convex Analysis and Global Optimisation, Noncovex Optimization and its Applications, vol. 54, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 433–444.
[25] M. Riedmiller, H. Braun, A direct adaptive method for faster back-propagation learning: the Rprop algorithm, in: Proceedings of IEEE International Conference on Neural Networks, San Francisco, 1993, pp. 586–591.
[26] N.M. Roehl, C.E. Pedreira, An online learning approach: a methodology for time varying applications, Neural Comput. Appl. 10 (2001) 101–107.
[27] D. Saad, On-line Learning in Neural Networks, Cambridge University Press, 1998.
[28] R. Salomon, P. Eggenberger, Adaptation on the evolutionary time scale: a working hypothesis and basic experiments, in: Proceedings of the Third European Conference on Artificial Evolution (AE'97), Nimes, France, Lecture Notes in Computer Science, vol. 1363, Springer, 1998.
[29] N.N. Schraudolph, Online local gain adaptation for multi-layer perceptrons, Technical Report, IDSIA-09-98, IDSIA, Lugano, Switzerland, 1998.
[30] N.N. Schraudolph, Local gain adaptation in stochastic gradient descend, Technical Report, IDSIA-09-99, IDSIA, Lugano, Switzerland, 1999.
[31] R. Storn, K. Price, Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.
[32] R.S. Sutton, Adapting bias by gradient descent: an incremental version of delta-bar-delta, in: Proceedings of the 10th National Conference on Artificial Intelligence, MIT Press, 1992, pp. 171–176.
[33] R.S. Sutton, S.D. Whitehead, Online learning with random representations, in: Proceedings of the 10th International Conference on Machine Learning, Morgan Kaufmann, 1993, pp. 314–321.
[34] F. Vavak, T.C. Fogarty, A comparative study of steady state and generational genetic algorithms, Evolutionary Computing: AISB Workshop, Lecture Notes in Computer Science, vol. 1143, Springer, 1996.
[35] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L. Alkon, Accelerating the convergence of the back-propagation method, Biol. Cybern. 59 (1988) 257–263.