AUTOMATIC IDENTIFICATION OF HISTOLOGICAL FEATURES IN HISTOLOGICAL IMAGES USING ARTIFICIAL NEURAL NETWORKS

M. Palfy*, M. Zorman* and J. L. Sánchez **

* Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, SI-2000 Maribor
** Department of Fundamental and Applied Physics, Electronics and Systems, University of La Laguna, Avenida Francisco Sanchez, S/N 38205, La Laguna, Spain

miroslav.palfy@uni-mb.si, milan.zorman@uni-mb.si, jose@cyc.ull.es

Abstract: A software-based system for identification of histological features in histological images is presented. Artificial neural networks, known as a well established data mining technique, were used for image analysis. Reliability was tested on histological images depicting a case of follicular lymphoma. The results acquired are compared and commented upon.

Introduction

Histological images are normally interpreted by a skilled pathologist in which case the possibility of a human error and time consumption play an important role.

It is the aim of this paper to describe a possible software solution for automatic identification of histological features, thus providing a tool for faster and easier interpretation of histological images. It focuses on a key element of the identification process: distinguishing particular parts of an image from each other. Identification is demonstrated on histological images depicting a case of follicular lymphoma. For this purpose image pixels were classified into groups corresponding to particular image features (follicles). Classification was done by the means of artificial neural networks, computational structures frequently used for data-mining purposes.

Methods

Data mining techniques provide a variety of different approaches for image analysis [1]. Artificial neural networks (ANN) are one of them. They are increasingly used in problem domains involving classification and are capable of finding shared features in a set of seemingly unrelated data. It was for this reason they were used as a tool for resolving our classification problem.

Artificial neural networks are abstract computational models of the human brain. The human brain has an estimated 10^{11} tiny units called neurons. These neurons are interconnected with an estimated 10^{15} links. Similar to the brain, an ANN is composed of artificial neurons (or processing units) and interconnections. There are many different types of ANNs, differing in topology, type of interconnections and learning technique used in training process [3].

Although many neural-network models have been proposed for solving a wide array of different tasks, the multilayer feedforward network (also known as multilayer perceptron) with a backpropagation-learning mechanism, is the most widely used model in terms of practical applications. Its efficiency and capability to quickly learn from given data convinced us to choose this model. Figure 1 depicts an example of multilayer neural network in the form of a graph. It is clear from this picture how individual artificial neurons are arranged in interconnected layers. For our particular problem we have chosen a fully connected network where each neuron in any layer of the network is connected to all the nodes (neurons) in the previous layer.

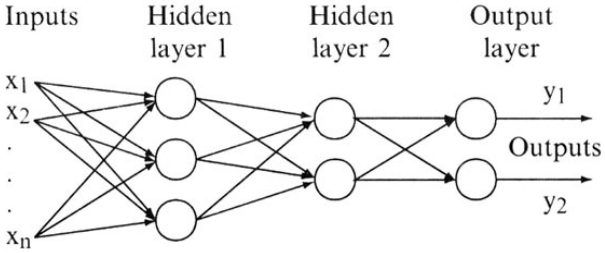

Figure 1: Fully connected multilayer perceptron with two hidden layers

Multilayer feedforward perceptron consists of a set of inputs that constitute the input layer of the network, one or more hidden layers of computational nodes, and finally an output layer of computational nodes. Nodes belonging to the same layer are not interconnected; all connections are directed from the input to the output layer so the processing is in a forward direction on a layer-by-layer basis. A vector of normalized numerical values is applied to the input nodes (input layer) of the network and its effects propagate through the network as each artificial neuron produces an output as a function of its inputs. This process roughly simulates thinking processes taking place in the human brain. Finally, a set of outputs is produced as the actual response of the network.

ISSN: 1727-1983

An artificial neuron is an information-processing unit that is fundamental to the operation of an ANN. The block diagram (Figure 2), which is a model of an artificial neuron shows that it consists of three basic elements:

1. A set of weighted inputs (outputs from the previous layer) and an externally applied bias.
2. An adder for summing the weighted input signals.
3. An activation function for limiting the amplitude of the neuron's output.

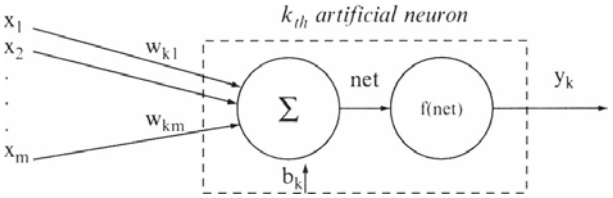

Figure 2: Model of an artificial neuron

In mathematical terms, an artificial neuron is an abstract model of a natural neuron where each input x_i is multiplied by the corresponding weight w_{ki} where k is the index of a given neuron in an ANN. The weights simulate the biological synaptic strengths in a natural neuron. The weighted sum of products $x_i w_{ki}$, for i = 1, …, m and bias b (used for increasing or lowering the net input of the activation function) is usually denoted as *net* in the ANN literature:

$$net_k = x_1 w_{k1} + x_2 w_{k2} + ... + x_m w_{km} + b_k \quad (1)$$

Many different forms of artificial neuron's activation function are commonly used. We have selected a log-sigmoid function:

$$f(net_k) = 1/(1 + e^{-net}) \quad (2)$$

When using ANNs for solving complex classification problems, the network's output is usually converted to binary value, which unambiguously defines the output class. By applying input element's numerical values to input nodes of an ANN the generated output represents the element's class.

Before the actual classification can take place an ANN must go through a thorough learning process. The ability to learn from its environment based on real-life examples and to improve its performance by doing so is of primary significance for an ANN. The network learns about its environment through an interactive process of adjustments applied to its connection weights. Ideally, the network becomes more knowledgeable about its environment after each iteration in the learning process. In our case learning is done by training the network in a supervised manner with a highly popular algorithm known as the *error backpropagation algorithm*. Basically, error backpropagation learning consists of two phases performed through the different layers of the network: a forward pass and a backward pass. In the forward pass, a training sample (input data vector) is applied to the input nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward phase, the synaptic weights of the network are all fixed. During the backward phase, on the other hand, the weights are all adjusted. Specifically, the actual response of the network is subtracted from a desired (target) response, which is a part of the training sample, to produce an error signal. If this error signal is outside the predefined tolerance bounds, it is than propagated backward through the network, against the direction of synaptic connections. The synaptic weights are adjusted to make the actual response of the network closer to the desired response. Detailed description of the backpropagation algorithm is omitted here for brevity but is well documented in literature [2]. It should be mentioned though that *generalized delta rule* was used for adjusting the weights. It includes a *momentum* constant and so allows for faster learning while avoiding the danger of instability:

$$\Delta w_{ji}(n) = \eta \cdot \delta_j(n) \cdot x_i(n) + \alpha \cdot \Delta w_{ji}(n-1) \quad (3)$$

Δw_{ji} : Correction of the weight factor for connection i of neuron j

n : Iteration (training sample index)

η : Learning rate parameter (positive constant)

δ_j : Local gradient for neuron j

x_i : Input value of connection i

α : Momentum (positive constant)

With each training sample the network accumulates more knowledge and is able to classify input elements with greater reliability. In our case input elements were pixels of histological images, defined by their numerical values as described in the following section.

ISSN: 1727-1983

Test Case: Follicular lymphoma

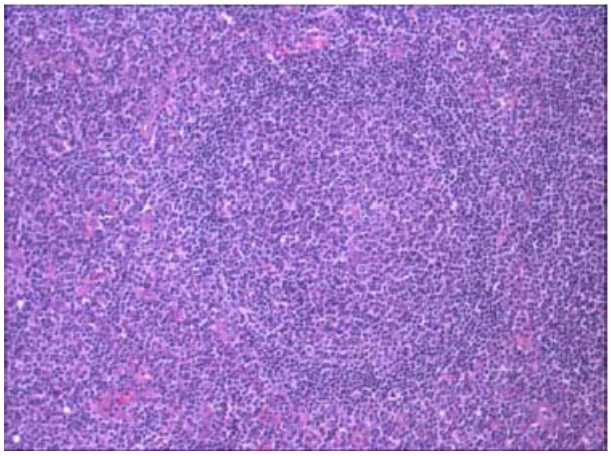

Figure 3: Detail from a histological image with visible malignant follicle

We put the efficacy of identifying histological features by using artificial neural networks to the test. Using a histological image of the cancerous lymph tissue (Figure 3) as a test example we tried to identify malignant B-cell lymphocytes characteristic of follicular lymphoma. The size and shape of these cells (follicles) can help diagnose the type (if any) and stage of follicular lymphoma.

Before trying to use an ANN for classifying individual pixels of the histological image it was necessary to complete the learning process. For that purpose the pixels were pre-classified into two classes, depending whether they were a part of follicle border or not. Each pixel was represented with nine numerical values (pixel colour components, number of similar contiguous pixels in different directions, etc.) acquired from the original image and after applying image filters (Figure 4).

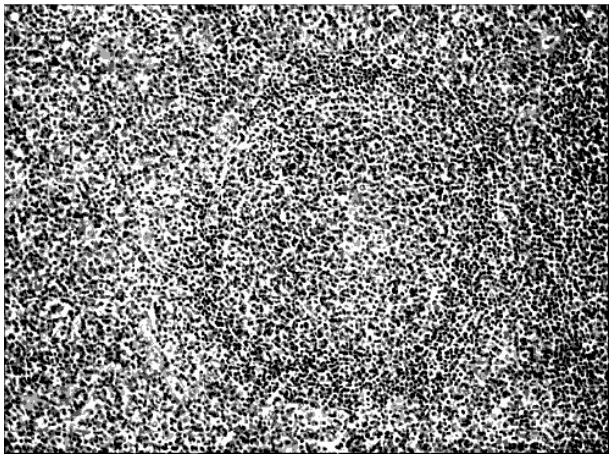

Figure 4: Greyscale image after increasing b/w contrast

These numerical values were normalized and applied to the input nodes of a user-defined network. The learning process consisted of a selected number of training samples (pixels) from the original image while other pixels were later classified with the trained network. The acquired results were compared with expected outputs which provided us with an estimation of ANN's reliability.

Results

We carried out 33 different training/classification processes (Table 1), each time using a different feedforward neural network with two hidden layers. Tests differed in training set size, learning process parameters, number of neurons in hidden layers and set of applied inputs. For the reference test we chose a neural network with the following properties:

- Number of neurons in hidden layers: 9, 5
- Use externally applied bias: yes
- Inputs applied to input layer: all
- Training set size: 3000 elements (pixels)
- Number of passes through training set: 100
- Learning rate: 0.1
- Momentum: 0.6
- Output tolerance: 0.3

Table 1: Classification efficiency

	Altered value	**New value**	**Result (%)**
1	- (reference test)	-	86.4%
2	training set size	1000	81.5%
3	training set size	2000	85.8%
4	training set size	4000	89.0%
5	number of passes	1	82.1%
6	number of passes	10	84.3%
7	number of passes	1000	90.9%
8	number of passes	10000	93.4%
9	set of input values	without *cpd* in direction 0°	85.9%
10	set of input values	without *cpd* in direction 90°	86.9%
11	set of input values	without *cpd* in direction 45°	85.2%
12	set of input values	without *cpd* in direction -45°	85.0%
13	set of input values	without all *cpd* values	64.5%
14	set of input values	without *hsfcm*	82.9%
15	set of input values	without *piv*	69.9%
16	set of input values	without *rpc*	87.6%
17	set of input values	without *gpc*	85.5%
18	set of input values	without *bpc*	86.5%
19	set of input values	without *rpc*, *gpc* and *bpc*	86.9%
20	use bias	no	85.9%
21	learning rate	0.3	88.4%
22	learning rate	0.5	87.9%
23	learning rate	0.7	86.7%
24	learning rate	0.9	88.5%
25	momentum	0.2	85.1%
26	momentum	0.4	84.2%

ISSN: 1727-1983

27	momentum	0.8	87.6%
28	tolerance	0.1	87.6%
29	tolerance	0.2	86.8%
30	tolerance	0.4	85.0%
31	number of neurons in hidden layers	4, 2	86.8%
32	number of neurons in hidden layers	18, 10	85.7%
33	number of neurons in hidden layers	90, 50	79.4%

Explanation of abbreviations:
cpd: number of similar contiguous pixels in one of the four directions (0°, 90°, 45°, -45°)
hsfcm: high spatial frequencies content measurement [4]
piv: pixel image value after convolution with Gaussian filter
rpc: red colour component
gpc: green colour component
bpc: blue colour component

Classification results were very good as classification accuracy of most tests exceeded 85%. Training set size as well as number of passes through training set had the most significant influence on results for the highest accuracy (93.4% correctly classified pixels) was achieved with 30,000,000 training samples (10,000 passes through a set of 3,000 pixels). For comparison, the reference test consisted of "only" 300,000 training samples (100 passes through a set of 3,000 pixels) and achieved an accuracy of 86.4%. When reducing the number of passes and the training set size to 1 and 1,000 respectively, accuracy dropped to 82.1% and 81.5%.

Amongst the inputs considered during classification, number of similar contiguous pixels in different directions proved to be of most importance. When omitting only one of those inputs (out of four) classification accuracy was only slightly reduced but without all of them the accuracy plummeted to 64.5%. Other important input was the pixel image value after convolution with Gaussian filter. Without it the network was able to successfully classify only 69.9% of all pixels. Somewhat surprisingly, the three colour components of image pixels appear to contribute very little or nothing to network's abilities as results deviated very slightly (85.5% - 87.6%).

Changing the parameters of the learning process (momentum, learning rate, tolerance) within reasonable boundaries had little effect on end results. Increasing the learning rate and momentum did somewhat improve the accuracy (up to 88.5%) while increasing the tolerance had a negative effect. Setting the values of those parameters outside recommended boundaries would lead to much more drastic decrease in performance.

Altering the topology of network by increasing the number of neurons in hidden layers to 90 and 50 (instead of 9 and 5) reduced the classification accuracy to 79.4% showing that more is not always better.

Discussion

As expected, the extent of training process directly influences the quality of classification. If provided with enough dissimilar training samples an ANN is capable of reliably solving otherwise complex classification problems. General rules on which an ANN bases its decisions are usually hard to extract but we can draw some conclusions:

- The number of similar contiguous pixels in different directions provides valuable information about groups of similar pixels and so helps identifying follicle borders.
- Applying filters to original images provides additional information about pixel relations not easily obtained from original images.
- Pixel colours by themselves are not enough.
- Increasing the size of neural network doesn't necessarily lead to better results. The network may end up memorizing the training data and losing the ability to generalize between similar patterns. Such a phenomenon is referred to as *overfitting*.

Conclusions

As seen from test results, artificial neural networks can be effectively used for identifying histological features. It is reasonable to believe comparable results could be achieved with wide variety of histological images of different types if proper learning process is assured. Once trained, an ANN can be used over and over again and even improve its performance through additional training. Another positive finding from this study is the fact that colour of pixels doesn't play a crucial role when trying to detect histological structures. This can allow for use of histological images from different sources and taken under different conditions.

Taking previous conclusions into consideration, the approach of using artificial neural networks as key elements in identification of histological features could be used to full extent by making it the foundation for building an intelligent system for automated diagnosis [4]. Such system wouldn't be able to replace a skilled pathologist; however it could help making a correct diagnosis after analyzing histological images. It could even be expanded with functionality to suggest possible treatments.

References

[1] D. HAND, H. MANNILA, P. SMYTH (2001): 'A Systematic Overview of Data Mining Algorithms', in 'Principles of Data Mining', MIT Press, Cambridge, pp. 87-101

[2] M. KANTARDZIC (2003): 'Artificial Neural Networks', in 'Data Mining: Concepts, Models, Methods, and Algorithms', John Wiley & Sons, New York

ISSN: 1727-1983

[3] M. ZORMAN, V. PODGORELEC, M. LENIČ, P. POVALEJ, P. KOKOL, A. TAPAJNER (2003): 'Nevronske mreže', in 'Inteligentni sistemi in profesionalni vsakdan', Center za interdisciplinarne in multidisciplinarne raziskave in študije Univerze v Mariboru, Maribor, pp. 75-85

[4] J. I. ESTÉVEZ, J. F. SIGUT, J. L. SÁNCHEZ, J. D. PIÑEIRO, S. ALAYÓN, R. L. MARICHAL, J. M. TORRES (2005) 'Intelligent System for the Automated Diagnosis of Histological Images', Wseas Transactions on Information Science & Applications, 2 (1), pp. 14-19