Multiscale cancer modeling: In the line of fast simulation and chemotherapy

Article *in* Mathematical and Computer Modelling · April 2009

DOI: 10.1016/j.mcm.2008.08.011 · Source: DBLP

4 authors:

Elham Bavafaye Haghighi
University of Tehran
1 PUBLICATION **3** CITATIONS
SEE PROFILE

M.J. Yazdanpanah
University of Tehran
177 PUBLICATIONS **926** CITATIONS
SEE PROFILE

Bita Kalaghchi
Tehran University of Medical Sciences
24 PUBLICATIONS **90** CITATIONS
SEE PROFILE

Hamid Soltanian-Zadeh
96 PUBLICATIONS **421** CITATIONS
SEE PROFILE

Mathematical and Computer Modelling 49 (2009) 1449–1464

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Multiscale cancer modeling: In the line of fast simulation and chemotherapy

E. Bavafaye-Haghighi [a,*], M.J. Yazdanpanah [a,*], B. Kalaghchi [b], H. Soltanian-Zadeh [a]

[a] Control & Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

[b] Cancer Research Center, Cancer Institute, University of Tehran, Tehran, Iran

ARTICLE INFO

Article history:
Received 21 March 2008
Received in revised form 6 June 2008
Accepted 12 August 2008

Keywords:
Multiscale cancer modeling
Nested hierarchical SOM
Gompertz model
Chemotherapy
Emergent

ABSTRACT

Although Multiscale Cancer Modeling has a realistic view in the process of tumor growth, its numerical algorithm is time consuming. Therefore, it is problematic to run and to find the best treatment plan for chemotherapy, even in case of a small size of tissue. Using an artificial neural network, this paper simulates the multiscale cancer model faster than its numerical algorithm. In order to find the best treatment plan, it suggests applying a simpler avascular model called Gompertz. By using these proposed methods, multiscale cancer modeling may be extendable to chemotherapy for a realistic size of tissue.

In order to simulate multiscale model, a hierarchical neural network called Nested Hierarchical Self Organizing Map (NHSOM) is used. The basis of the NHSOM is an enhanced version of SOM, with an adaptive vigilance parameter. Corresponding parameter and the overall bottom-up design guarantee the quality of clustering, and the embedded top-down architecture reduces computational complexity.

Although by applying NHSOM, the process of simulation runs faster compared with that of the numerical algorithm, it is not possible to check a simple search space. As a result, a set containing the best treatment plans of a simpler model (Gompertz) is used. Additionally, it is assumed in this paper, that the distribution of drug in vessels has a linear relation with the blood flow rate. The technical advantage of this assumption is that by using a simple linear relation, a given diffusion of a drug dosage may be scaled to the desired one.

By extracting a proper feature vector from the multiscale model and using NHSOM, applying the scaled-best treatment plans of Gompertz model is done for a small size of tissue. In addition, simulating the effect of stress reduction on normal tissue after chemotherapy is another advantage of using NHSOM, which is a kind of "emergent".

1. Introduction

1.1. Multiscale cancer modeling

Mathematical and computer modeling is a way to improve treatment plans for some of diseases [1–4]. In this way, cancer modeling and simulation have a long way to go, while the evolution of a tumor is a complex process which is normally affected by many factors on different scales [4–30]. Several mathematical models are introduced to simulate tumor growth and chemotherapy [4–19]. In addition, there are some related topics, such as drug resistance [20] or angiogenesis, and other

* Corresponding address: Faculty of Engineering, Campus#2, University of Tehran, Tehran, P.O. Box 14395-515, Iran. Tel.: +98 21 8208 4925; fax: +98 21 8877 8690.

E-mail addresses: e.bavafa@ece.ut.ac.ir (E. Bavafaye-Haghighi), yazdan@ut.ac.ir (M.J. Yazdanpanah), kalaghchi@tums.ac.ir (B. Kalaghchi), hszadeh@ut.ac.ir (H. Soltanian-Zadeh).

0895-7177/$ – see front matter
doi:10.1016/j.mcm.2008.08.011

interactions between underlying cells and the corresponding tissue [21–27] in this regard. By referring to these research works, cancer models are divided into vascular and avascular [4–12]. Usually, by considering the factors affecting tumor growth, these models may be evaluated [4–12,15]. However, it is worth noting that simplicity is another important factor to evaluate any model, while greater complexity in modeling causes a large computational error during the simulation.

By using avascular models, simulating, analyzing and finding an optimal treatment plan are possible; however, the effect of the vascular network to make a heterogeneous environment for tumor growth is neglected [4,17–19]. In comparison with avascular models, vascular modeling is an attempt to model cancer dynamics by considering them realistically [4–10].

The vascular network is an important factor to simulate tumor evolution. It supplies the nutrient factors for tumor growth [4–12,28–30]. Depending on the structure of the vascular network and distribution of oxygen and glucose, the process of tumor evolution will vary heterogeneously. On the other hand, by releasing Vascular Endothelial Growth Factor (VEGF) by hypoxic cells, the structure of the vascular network will be changed proportionally. Finally, tumor growth should be investigated as a multiscale dynamical system (i.e., macroscopic, mesoscopic and microscopic scale) [4–10]. A complete discussion about multiscale cancer modeling and drug effect on proliferating cells is presented in [5–9].

Although the multiscale model has many advantages, complexity of the model, a large number of parameters and its long lasting numerical algorithm of simulation are some of disadvantages of this model [6,7]. These limitations causes the number of cancer cells in the model to be strongly less than the typical number of cancer cells in a real tumor (around 10^3 cells compared with 10^6 cells and more) [6,7,17,18]. In addition, the effect of decreasing the stress on normal tissue [4,16] especially when a part of tumor removed by any drug is not addressed by the model. Because of the elasticity of normal tissue, after removing a part of the tumor, tissue gradually occupies the freed-up space.

1.2. NHSOM for simulating multiscale cancer model

In order to reduce the computational complexity of the multiscale model, artificial neural networks are appropriate choices. Neural networks are defined as powerful tools which can solve many problems in different areas such as Image Processing [31–34], Pattern Recognition [31,35,36] and Nonlinear Dynamical Systems and Control [31]. Under the notion of learning, neural networks may be distinguished two different types of learning: supervised and unsupervised. By using unsupervised neural networks, the task of clustering by grouping similar input patterns, could be done [31,35,37].

In [33], an enhanced version of the Self Organizing Map (SOM) with a vigilance parameter is introduced. This version of SOM is a combination of best features of famous unsupervised neural networks: Adaptive Resonance Theory (ART1) and SOM. The enhanced version of SOM can assign a new cluster, for some patterns, if necessary. This method is not dependent on the input presentation order, because during the training steps, it computes an average value for the patterns which all belong to one cluster. The computational complexity decreases in comparison with SOM, while the quality of the clustering increases. However, determining appropriate values for the vigilance parameter and cluster number plays an important role in the line of network performance, especially when applying it hierarchically.

Nested HSOM is the hierarchical version of the enhanced SOM, in which the vigilance parameter and number of neurons varies in the layers of the network. It has a bottom-up/top-down architecture. The overall bottom-up design prevents the under-dimension problem in which the number of the neurons in the output layer must be greater than the real number of output clusters, to avoid saturation of the network by high frequency input patterns [33,38]. In addition, the embedded top-down architecture reduces the computational complexity [39,40]. The hierarchical relations between the input data are reflected in a straightforward manner like other versions of Hierarchical SOM (HSOM). Additionally, by applying an adaptive version of the vigilance parameter, the new method requires a lower number of training steps in comparison with other ones. The vigilance parameter is also a guarantee to preserve the quality of results, even with a low number of training samples.

Because of the considerable quality and low run time of NHSOM, it is a suitable candidate for simulating the multiscale model. By executing the numerical algorithm of the multiscale model [5–9], the output of each iteration is achieved which is a combination of macroscopic, mesoscopic and microscopic states. NHSOM will be trained using a combination of these states. In the training phase, after determining what is the best cluster at lowest (bottom) layer for a training sample, the next state of the given sample (state) will be stored. In simulation phase, after finding the closest cluster at the bottom layer, this stored information will be retrieved.

By using NHSOM, the simulation process is done more than 650 times faster than the numerical method. As expected, feature selection has an important role in the line of network performance. In addition, NHSOM has an extra-emergent advantage in the way of simulating the effect of decreasing stress on the normal tissue.

1.3. Gompertz model to chemotherapy of multiscale model

The state space of multiscale model has a discrete nature. Although by applying NHSOM, the process of simulation is done faster than its numerical algorithm, it is not possible to check a simple binary tree of treatment plans with fixed treatment gaps and only two choices for drug dosage. In order to build a search space, in which each treatment plan stands for a state who lasts 84 days with a weekly drug injection and two specified choice of drug dosage, 2^{12} states are required. The time needed to build and check such a tree by the numerical algorithm is minimally 2465 days (using a MATLAB program on a 2

GHz processor). Under the same conditions, by applying NHSOM instead of the numerical method, it takes at least 91 h to do the same test. As a result, to find the best treatment plan with the floating nature of drug dosage and treatment gap, methods such as dynamic programming are not applicable. In order to solve the aforementioned problem, the best treatment plans of a simpler cancer model (Gompertz) will be used [18].

Additionally, by considering the fact that Doxorubicin (which is used for chemotherapy in this paper) attaches to "cytochrome p450 reductase" which has a lower size than red blood cells [41], it is assumed that the distribution of Doxorubicin in vessels has a linear relation with blood flow rate. The technical advantage of this assumption is that by using a simple linear relation, a given diffusion of Doxorubicin can be scaled to other dosages. Therefore, with a limited number of training samples, various states with various drug dosages can be obtained.

The next sections are arranged as follow: In Section 2, NHSOM and its advantages will be discussed. The chemotherapy of the multiscale model will be explained in Section 3 and Simulation results will be given in Section 4. Finally, some conclusions are made in Section 5.

2. Nested hierarchical self organizing map

In order to simulate multiscale cancer model faster than its numerical algorithm, neural networks may be suitable choices. To avoid error propagation during simulation, the quality and speed are some of the necessary features of the neural network applied. As a result, a new version of SOM is introduced in this paper (Nested Hierarchical Self Organizing Map or NHSOM). The quality and speed of the new method is considerably better than traditional SOM and another hierarchical version of SOM [32]. This examination is done for recognizing handwritten digits of [42][1] and for hierarchical image segmentation.

A complete discussion about advantages and disadvantages of ART1, SOM and hierarchical versions of SOM are presented in [31–40,43–45]. In Section 2.1 the process, advantages and disadvantages of enhanced SOM will be reviewed. In Section 2.2 an adaptive version of enhanced SOM in a bottom-up architecture will be considered. The embedded top-down design of NHSOM will be the discussion of Section 2.3.

2.1. Enhanced SOM with vigilance parameter

The under-dimension problem is one of the disadvantages of SOM, in which the number of the neurons in the output layer must be greater than the real number of output clusters to avoid saturation of the network by high frequency input patterns [32,33,38]. Referring to [33] one may find that, the under-dimension and high computational complexity of SOM, stems from the fact that in each training step, a weight which has the maximum similarity measure with the input pattern, is updated. However, is it enough to choose the maximum similarity? On the other hand, the maximum value of all activation functions must be greater than a minimum threshold.

In the case of Enhanced SOM, the new cluster will be added to the network when the maximum value of all activation functions does not pass the vigilance test (as in ART1) [31–36,43]. Updating the winner cluster, same as SOM [31–33,35–40, 44,45], prevents dependency of the network on the input presentation order [32,33]:

$$\text{If } \langle X, W_i^t \rangle > \rho, \quad W_i^{(t+1)} = W_i^{(t)} + \lambda^{(t)}(x_k - W_i^{(t)}). \tag{1}$$

In (1), W_i is the weight of the winner cluster, $\lambda^{(t)}$ is the learning parameter, decreasing in time, x_k is the normalized input pattern and ρ is the vigilance parameter. The input patterns and weights of the neurons are normalized. $\langle X, W_i^t \rangle$ presents the inner product of the input pattern and the weight of the winner neuron (i.e., the similarity measure among them [33]).

As a result, when an input pattern has a low frequency compared with others, a specified class will be assigned to it. By using a vigilance parameter, high frequency patterns cannot change it during the training process. Since the vigilance parameter is a guarantee for existing minimum similarity between the pattern and its cluster center, selecting a finer learning parameter (λ) will be sufficient. In the meantime, a large number of training is not needed.

The main advantages of enhanced SOM compared with ART1 and SOM are as follows: there is no need to know the number of final clusters (neurons), a lower number of training steps (approximately 20 times) and qualitative results [32,33]. However, determining appropriate value for vigilance parameter plays an important role in the line of network performance.

2.2. Bottom-up adaptive enhanced som

As mentioned before, enhanced SOM with a vigilance parameter, reduces the number of training steps significantly. The success of the results depends on the vigilance parameter, which should be predetermined. In [32,39,40], the advantages of HSOMs to reduce computational complexity are explained. In order to combine best features of enhanced SOM and HSOMs, an adaptive version of vigilance parameter is required. Contemplation about bottom-up process in hierarchical clustering methods gives a good idea for defining the adaptive vigilance parameter. In the first (bottom) layer of the hierarchy, patterns

[1] For digit recognition task, after a training phase, the label of each cluster will be determined using maximal accepted patterns.

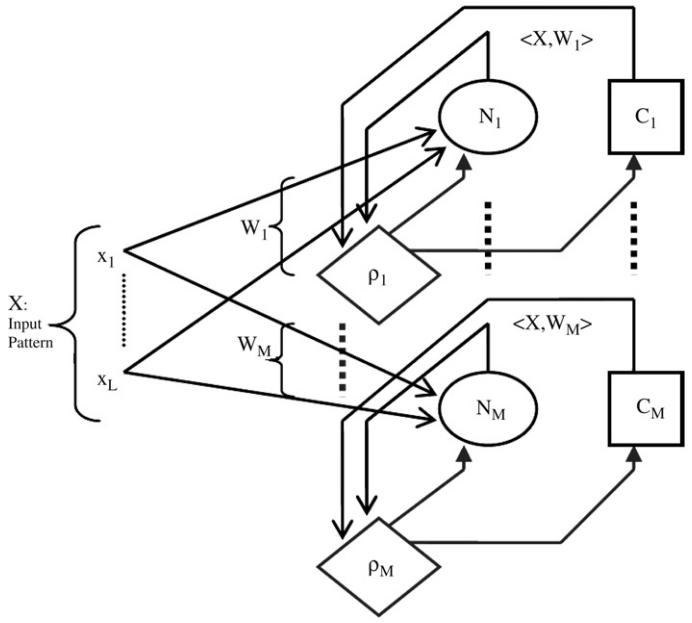

Fig. 1. The architecture of the neurons for a layer of bottom-up adaptive enhanced SOM: W is the weight of the neuron, C counts the number of the accepted patterns and ρ is the vigilance parameter. The weight and the counter are updated, if the vigilance test is passed. The value of the vigilance parameter depends on the similarity measure between the accepted patterns and the weight of the neuron, counter and the corresponding layer number.

of the given data set will be presented. In the next layers, clusters which are formed by the previous layer, will be considered as input. Therefore, the similarity measure of the accepted patterns by a cluster in the first layers is greater than the next ones. On the other hand, for high frequency similar patterns presented to a specified layer, it is better to have a greater vigilance parameter, which results in a better quality of clustering these patterns.

Fig. 1 presents the proposed structure for neurons with the adaptive vigilance parameter for a layer (Adaptive Enhanced SOM). Each neuron has a counter, by which the number of accepted patterns in each training step is counted. After identifying the winner neuron for a given pattern, if the similarity measure between them passes the vigilance test, the counter of the winner neuron will be increased using (2)(e) and the weight of the neuron will be updated finely based on (2)(a) and (2)(b). The value of the vigilance parameter is affected by the measured similarity, number of the accepted patterns and corresponding layer number (see (2)(c) and (2)(d)).

$$\text{If } \langle X, W_i^t\rangle > \rho \quad \text{then} \begin{cases} \Delta W_i^t = X - W_i^t \cong 1 - \langle X, W_i^t\rangle \text{ [31]}, & \text{(a)} \\ W_i^{t+1} = W_i^t + \beta^t \Delta W_i^t, & \text{(b)} \\ \Delta\rho_i^t = \left(1 - \dfrac{1}{\text{counter}(i)}\right)^{\text{LayerNo}} (\langle X, W_i^t\rangle - \rho_i^t), & \text{(c)} \\ \rho_i^{t+1} = \rho_i^t + \lambda^t \Delta\rho_i^t, & \text{(d)} \\ \text{counter}(i) = \text{counter}(i) + 1. & \text{(e)} \end{cases}$$

Else add a new cluster. (2)

In (2)(a)–(2)(e), X is an input pattern, i presents the winner neuron, t corresponds to the step of the training, W_i^t is the weight of the winner neuron, *LayerNo* is the current layer number, $\langle X, W_i^t\rangle$ presents the inner product of the input pattern and the weight of the winner neuron (i.e., the similarity measure between them [31]), "counter(i)" counts the accepted patterns by ith neuron, ρ_i^t is the corresponding vigilance parameter for ith neuron and β^t and λ^t are the learning parameters, both decreasing in time based on (3) and (4). As explained in Section 2.1, the initial value of β^t must be selected finely.

$$\lambda^{(t)} = \frac{\lambda^0}{t}. \tag{3}$$

$$\beta^{(t)} = \frac{\beta^0}{t}. \tag{4}$$

In (2)(c), the expression $(1-\frac{1}{\text{counter}(i)})^{\text{LayerNo}}$ is always in the range $[0, 1)$. Lower values of the layer number (bottom layers) or greater number of accepted patterns, consequences higher values of the above expression and it causes more effect of the similarity measure on vigilance parameter.

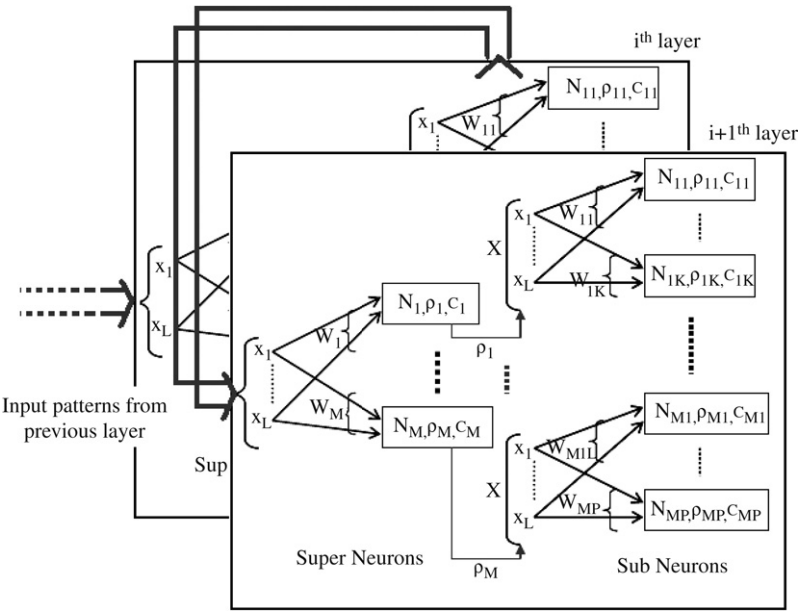

Fig. 2. Nested hierarchical SOM consists of top-down adaptive enhanced SOM, which are organized hierarchically in a bottom-up architecture. The weights of the sub neurons of each layer are the input patterns of the next one. In the top-down adaptive enhanced SOM, each super neuron refers to a set of sub neurons.

In bottom-up adaptive enhanced SOM there is no need to have pre-knowledge about proper number of clusters in each layer. By using only a neuron in the first stage of learning in each layer, the number of neurons will be determined automatically by the evolutionary process of learning. Referring to the explanation of Section 2.1, a low number of training steps is sufficient to form the network well.

2.3. Nested hierarchical SOM

Without considering the high number of training steps in traditional top-down HSOMs, these versions of neural networks have, potentially, a good ability to reduce the number of comparisons [32,39,40]. The low number of neurons in the first layers of HSOM causes such effect; however, it also causes the under-dimension problem, in which the main characteristics of the low frequency patterns may not be reflected in these layers and it may be possible that these patterns would be forgotten in the next layers [32,33,38].

If in *each layer* of bottom-up adaptive enhanced SOM, the number of neurons decreases; it will causes lower number of comparisons in that layer. In order to reflect the features of all patterns in the network, the number of neurons should not decrease as in traditional HSOMs [39,40]. It must be great enough to prevent the under dimension effect. To achieve this goal, each layer of bottom-up adaptive enhanced SOM is reconfigured by a top-down design (Fig. 2). In the new architecture the input patterns will be presented to the super neurons, which are guidances to a set of sub neurons.

Since more patterns are accepted by the super neurons, the sensitivity of their vigilance parameter must be less than the sub neurons. Relation (5) reduces the sensitivity of vigilance parameter for a super neuron.

$$\Delta\rho_i^t = \left(1 - \frac{1}{\text{counter}(i)}\right)^{\text{LayerNo}^* \text{counter}(i)} (\langle X, W_i^t\rangle - \rho_i^t). \tag{5}$$

One of the disadvantages of top-down approaches is k-competition to achieve better results [39]. The k-competition can recursively continued in more than two layers of the hierarchy. Therefore, the greater depth of the recursion causes greater computational complexity. In case of each layer of NHSOM (top-down adaptive enhanced SOM), the depth of nested top-down hierarchy is limited to two layers. Therefore, its effect on the overall computational complexity in the test phase is negligible. Additionally, it is shown that in the training phase, there is no need to use a k-competition approach, while the vigilance parameter guarantees the quality of the network performance.

Fig. 3 explains schematically what happens in the training phase. By presenting an input pattern to a layer, the best super neuron will be selected. It might be possible that the best sub neuron which belongs to a further super neuron be closer to the pattern. In Fig. 3, the super neuron 1 (Box A) is closer to the given pattern (black circle) in comparison with the super neuron 2; however, the sub neuron 4 which belongs to 2, is closer to the input compared with neuron 3 which belongs to 1. In the *test phase* by using a k-competition approach, the best case (i.e., the closest sub neuron, say neuron 4) will be selected. If in the *training phase* the best sub neuron does not pass the vigilance test, a new neuron will be added to the network;

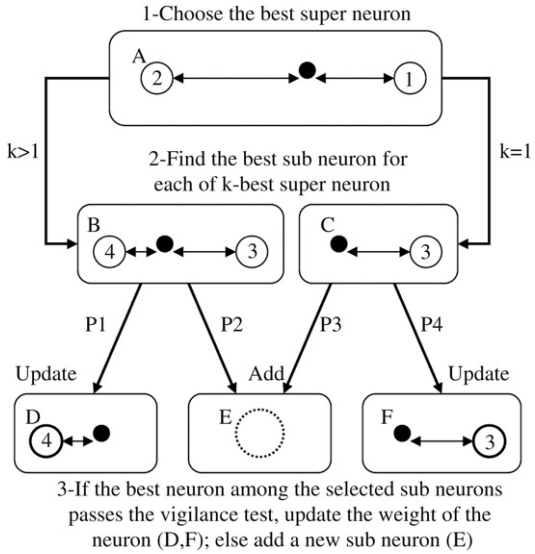

Fig. 3. After choosing the closest super neuron (no. 1 in box A), it may be possible that the best sub neuron (no. 4) which belongs to a further super neuron (no. 2), be closer to the input pattern. By choosing neuron no. 4 the probability of adding a new sub neuron (P2) will be less than the case in which sub neuron no. 3 is chosen (P3).

otherwise the best sub neuron will be updated. In case of k-competition approach ($k > 1$) the probability of adding a new sub neuron (P2) is less than the case of one-competition (P3); because neuron no. 4 has a greater similarity with the input pattern in comparison to neuron no. 3.

Nested Hierarchical SOM (NHSOM) consists of top-down adaptive enhanced SOM, which are organized hierarchically in a bottom-up architecture (Fig. 2. The weights of the sub neurons (in a layer) are the input pattern of the next layer. After the training phase, the *index* of weights from previous layer will be stored in the corresponding accepted sub neuron of the next layer. In this way, the top down relations between the layers will be saved to apply for the top-down test phase.

The important features of NHSOM which are mainly the advantages of applying *adaptive vigilance parameter*, are summarized bellow [32]:

1. Low number of training steps.
2. Qualitative results, even with a low number of training samples.
3. Adjusting the proper number of neurons in each layer of the network automatically in an evolutionary process.
4. The algorithm presents the best results in a one-competition approach in the training phase, which causes less computational complexity for learning, compared with other hierarchical networks.
5. The overall bottom-up architecture of NHSOM and the vigilance parameter prevents the under-dimension problem in which the main characteristics of the low frequency patterns are not reflected in the network.
6. The embedded top-down design of NHSOM reduces runtime of the algorithm.
7. A proper determination of number of layers to have a balance between quality of the results and runtime of the algorithm is required.

3. Chemotherapy of multiscale model

Feature selection is an important phase before applying a pattern recognition method [35]. In Section 3.1, the process of feature selection before applying NHSOM will be presented. By using a simple linear relation, the suggested drug dosages of Gompertz model will be scaled which will be the discussion of Section 3.2.

3.1. Feature selection for NHSOM

By simulating the multiscale model using its numerical algorithm, an array of cellular automata, in which each cell contains a state vector will be achieved [5–7,9]. Diffusion of chemicals around a cell and the next state of a proliferating cell depend on the state of the neighbors around that cell [6–9]. Therefore, it is not enough to consider the current states of an element, to predict the next state of it, while there is an indirect relation between the next state vector and the current states of its neighbors.

As a result, an n^*n array of automaton (Fig. 4) has a better ability to reflect both direct and indirect effects on the next states. The size of the selected arrays of cells should be large enough to reflect the effect of neighbors, and should be small enough to avoid extra large feature vectors. After determining a suitable size for array of automaton, each feature vector will

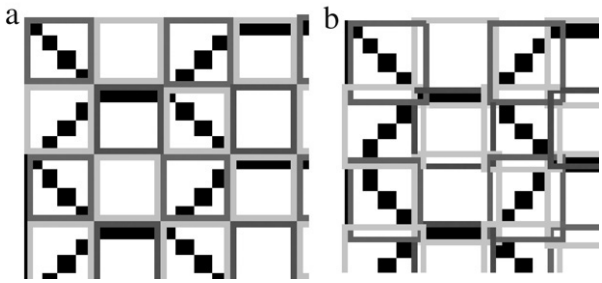

Fig. 4. The selected features are n^*n arrays of automaton. (a) 4*4 arrays; some feature vectors do not contain any automaton corresponding to vessels. (b) 5*5 arrays; all of feature vectors contain the automaton corresponding to vessels.

Table 1
The average runtime of NHSOM, with 2 and 4 layers for different feature selection methods

Feature code	Selected futures						No. layers	Mean runtime of train (s)	Mean runtime of test (s)
	Applying the second weights	Scaling some elements	Shifting some elements	Repeating some elements	Size of the selected array	Length of feature vector			
1	×	×	×	×	4*4	209	2	62.84	4.28
							4	84.1	2.93
2	✓	×	×	×	4*4	209	2	68	7
							4	95.21	3.75
3	✓	✓	×	×	4*4	209	2	68	7
							4	95.21	3.75
4	✓	✓	✓	×	4*4	209	2	68	7
							4	95.21	3.75
5	✓	✓	✓	✓	4*4	241	2	73.98	8.45
							4	98.49	4.3
6	✓	✓	✓	✓	5*5	376	2	104.44	13.34
							4	146.77	7.6

be created, using a sequence of state vectors of corresponding elements of array. In order to presave the length of feature vector before normalization, an extra element which equals to one, will be added to it.

It is worth remembering that for simulating a multiscale model, by presenting a current feature, the next state of the feature is expected; however, NHSOM is a clustering method by which the closest neuron (cluster) to a given input will be recognized. A review on the structure of filing systems gives a useful idea to change NHSOM, in order to retrieve the next feature after presenting any current one. In these systems to have a fast access to data (or fetch), a hierarchical structure of indices is used to address the page of memory containing the corresponding "data" [46]. The hierarchical arrangement of NHSOM acts as these indices, and the next features can be stored in the bottom layer of NHSOM (say stored features) such as "data" in a filing system. A similar (however, not the same) idea is applied in ART1 to store original weight of a presented input (i.e., the "v" weighs) [31,43].

The stored features are saved in the bottom layer of NHSOM during the training phase. The number of neurons will be determined during the evolutionary process of learning. As a result, each layer begins the training phase with one neuron. By adding each new sub neuron in the bottom layer of NHSOM, the related next feature of the presented pattern will be stored.

To select the best feature and architecture of NHSOM, two types of the network: with two layers (four nested layers) and four layers (eight nested layers), are tested, using various feature selection methods. In order to train NHSOM, using 26 steps of simulation of the main model, which contains 61*61 cells of the automaton, 5850 feature vectors are extracted. The number of training steps in both networks is 20. After the training phase, the training samples (current features) will be again presented to the network, to determine the closest cluster. The average of closeness between these samples and corresponding winner clusters (elementary weight of the winner cluster) and also, between the next state of these samples and corresponding stored features, are used to evaluate various feature selection methods (Fig. 6). In addition, the runtime of training and testing these samples are given in Table 1.

The runtime of the *test phase* of NHSOM with four layers is less than the versions with two layers (Table 1); however, because of error propagation in the four layer network, the average of closeness for the two layer version is better than the case with four layers (Fig. 5). Additionally, it is obvious in Fig. 5 that by increasing the feature code, the performance of NHSOM increases.

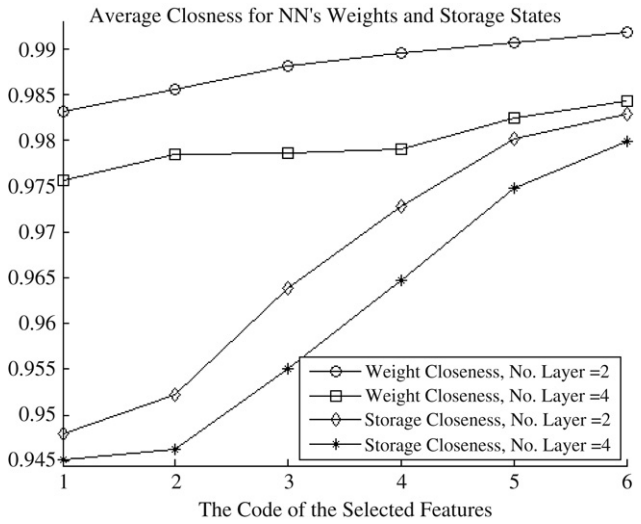

Fig. 5. The average closeness between training samples (features) and corresponding elementary weight of best neuron (cluster) and also between the next feature of the given samples and corresponding stored features, for NHSOMs with two and four layers.

The specifications of the feature selection methods are given bellow:

Feature Code 1: At first step of feature selection, 4*4 arrays of automaton are selected (Fig. 4a). In each extracted feature, the 13 length state vectors of automaton are arranged continuously. 13 elements of the state vectors correspond to: type of the element (cancer/normal cell, vessel or empty space), no. of divisions which cell has done in the time step before, concentration of oxygen, VEGF and drug around the cell, mass of the cell, death of the cell by toxic drug effect and concentration of chemicals in the cell (Cdh1, Cyc-CDK, p27, p53, RBNP and inter-cellular VEGF). For more details about chemicals one may see [5–9]. The next corresponded features are stored in the bottom layer of NHSOM.

Feature Code 2: In this step, another weight, which is the normalized-next feature of the elementary weight of a neuron, will be added to it. By using the second weight for each neuron, the winner will be updated if the vigilance test passes for both elementary and second weight of it. Adding a new test for second weights, increases the runtime of the training phase in comparison with feature code 1. In addition, there are some similar current features with different next states. By applying feature code 2, these features will be distinguished by more. Increment in the number of neurons causes increment in runtime of the training and test phase. The performance of feature code 2 is improved compared with feature code 1 (Fig. 5). It is obvious that the second weights are used only during the training phase.

Feature Code 3: In addition to the characteristics of feature code 2, some important elements of each state vector are scaled in feature code 3. The selected elements are type and the mass of the cell, number of divisions which has cell undergone during the last step and death of the cell by a chemotherapic drug. According to the fact that there is no change in the length of the new feature vector, the runtime of the algorithm is approximately equal to feature code 2.

Feature Code 4: The difference between this feature code and feature code 3 is that the element corresponding to the drug concentration around a cell, is shifted. At the time which a cell attempts to divide, the division will fail and cells will die, if the drug concentration around it exceeds a specified threshold. As a result, *in simulation phase,* it occurs that a given sample which has a close similarity with the elementary weight of a cluster; however, with a small difference in drug concentration (one less and the other more than the drug threshold), assigns to a false cluster with a different next feature. Therefore, for each drug concentration the values which are more than (or equal to) the drug threshold, should shift up to, and have a greater difference with the lower values. Although the performance of NHSOM increases with the new change, its runtime is approximately the same as feature codes 3 and 2.

Feature Code 5: In order to obtain a better performance of NHSOM, some important elements of the state vectors are repeated. These important elements are type of the cell and drug concentration. By increasing the length of each state vector from 13 to 15, the length of feature vectors increases to 241 from 209. Therefore, the runtime of the algorithm will increase in comparison with previous feature selections.

Feature Code 6: Vessels adjacency to a cell, have important role in forming the dynamics of that cell. By applying 4*4 arrays of automaton, it is possible that some feature vectors do not contain any automaton of vessels (Fig. 4a). In Fig. 4b, the 5*5 arrays of automaton are presented so that all of them contains at least an automaton of vessels. These 5*5 arrays increase the length of feature vector to 376 and additionally, the performance of the network.

The quality of the feature code 6 for NHSOM with two layers is the best in comparison with the other features, and also compared with NHSOM with four layers. Therefore, the simulation of multiscale model in Sections 4.1 and 4.2 is done using feature code 6 and NHSOM with two layers. The two layered NHSOM is trained by results of 26 steps of numerical simulation

of 61*61 array of automaton (it is equal to 390 h of tumor growth [9] in a 2 mm^2 of square tissue [13]) with a maximum drug dosage of 10.

3.2. Scaling the treatment plans

In this paper, an extra assumption is considered for drug diffusion. In order to transmit drugs in a vascular network, some blood proteins are used. The chemotherapy drug in this paper is Doxorubicin, which is carried by cytochrome p450 reductase [41]. Since each cell consists of various proteins, the size of proteins is generally smaller than cells. As a result, the distribution of the drug in the vascular network is assumed to be proportional to the blood flow rate, based on (6). An important technical advantage of this assumption is that a linear relation can be found between two different drug diffusions with the same automaton structure.

$$\theta_{\text{vess\_}i} = \frac{\dot{Q}_i}{\dot{Q}_{\text{input}}}\theta_{\text{total}}. \tag{6}$$

In (6), θ_{total} is the total input drug for vascular simulated network, $\dot{Q}_i$ and $\theta_{\text{vess\_}i}$ are blood flow rate and the drug of the ith vessel and $\dot{Q}_{\text{input}}$ is the total input blood flow rate. Relation (7) shows the dynamic of the injected drug dosage [17,18].

$$\dot{\theta}_{\text{total}} = -\gamma\theta_{\text{total}}. \tag{7}$$

In the above relation, γ depends on the half life of the injected drug (HLD) (relation (8)) [17,18]. The value of HLD for Doxorubicin is taken from [13] equal to 26 (hour).

$$\gamma = \frac{\ln(2)}{HLD}. \tag{8}$$

In order to solve the diffusion of chemicals, after expansion of the appropriate PDEs [5–7,9], an equation in the form of $\underline{\underline{A}}\,\underline{x} = \underline{b}$ should be solved. For a drug distribution, $\underline{x}$ (say $\underline{\theta}$ for the case of drug) is the vector of θ_is in which i is the index of each automaton. $\underline{\underline{A}}$ is a sparse matrix which contains the coefficients D_θ (diffusion constant) and λ_θ (constant rate to consume drug). $\underline{b}$ is a vector contains zero elements and drug concentration inside the vessels ($\theta_{\text{vess\_}i}$), which is multiplied in h_θ (the rate at which chemicals cross the vessel walls). Therefore, $\underline{b}$ is rewritten as follows:

$$\underline{b} = h_\theta\underline{\theta}_{\text{vess}}. \tag{9}$$

The drug concentration for cells ($\underline{\theta}$) will be solvable using (10).

$$\underline{\theta} = \underline{\underline{A}}^{-1}h_\theta\underline{\theta}_{\text{vess}}. \tag{10}$$

The combination of (6) and (10), results in (11).

$$\underline{\theta} = \underline{\underline{A}}^{-1}h_\theta\frac{\theta_{\text{total}}}{\dot{Q}_{\text{input}}}\underline{\dot{Q}}. \tag{11}$$

In (11), $\underline{\dot{Q}}$ is a vector, the same as $\underline{\theta}_{\text{vess}}$, consists of zero elements and $\dot{Q}_i$. Therefore, if the drug diffusion for a specified incoming drug (say $\theta_{\text{total\_1}}$) has been calculated before (say $\underline{\theta}_1$) and without any changes in the statues of the cells and vascular structure, only total incoming drug is changed to $\theta_{\text{total\_2}}$, the new drug diffusion $\underline{\theta}_2$ will be equal to (12).

$$\underline{\theta}_2 = \underline{\underline{A}}^{-1}h_\theta\frac{\theta_{\text{total\_2}}}{\dot{Q}_{\text{input}}}\underline{\dot{Q}}. \tag{12}$$

The expression $\underline{\underline{A}}^{-1}h_\theta\frac{1}{\dot{Q}_{\text{input}}}\underline{\dot{Q}}$ may be computed by replacing $\theta_{\text{total\_1}}$ and $\underline{\theta}_1$ in (11):

$$\underline{\underline{A}}^{-1}h_\theta\frac{1}{\dot{Q}_{\text{input}}}\underline{\dot{Q}} = \frac{1}{\theta_{\text{total\_1}}}\underline{\theta}_1. \tag{13}$$

Finally, (12) will be rewritten as follows:

$$\underline{\theta}_2 = \frac{\theta_{\text{total\_2}}}{\theta_{\text{total\_1}}}\underline{\theta}_1. \tag{14}$$

Relation (14) shows that there is a linear relation between two different drug diffusions with the same automaton structure. Therefore, it is possible to train NHSOM, using limited samples with a maximum dosage of 10 and to scale the incoming drug dosage to any other one, unlimitedly.

4. Simulation results

In this section, the multiscale model will be simulated, using NHSOM and the scaled treatment plans of Gompertz model with various cost functions (i.e., the discussion of Section 4.1). In Section 4.2, after simulating the multiscale model, the effect of stress reduction on normal tissue will be observed emergently.

4.1. Simulating the chemotherapy suggestions of Gompertz model

It has been mentioned that simulation will be done, using the suggested treatment plans of Gompertz model [18] and NHSOM. In addition, drug diffusion can be scaled to any desired one using relation (14). The simulation time which is needed for 84 days of treatment plan by a numerical algorithm is more than 67 h. By applying NHSOM, it takes 81 s; however, even by using NHSOM, it is not possible to check a simple binary tree of treatment plans with fixed treatment gaps and only two choices for drug dosage. More details about the required times are given in Section 1.3.

As a result, the best treatment plans of a simpler model (Gompertz) are used. Chemotherapy in Gompertz model is suggested, using an optimized estimation of drug dosage and swarm intelligence [18]. The main advantage of the proposed method, in comparison with some classic versions [17,19] is the discrete-floating nature of treatment gaps, and also, the floating nature of drug dosages. Therefore, these treatment plans have a realistic view for the process of treatment [18].

The maximum drug dosage of Gompertz model is 50 [17,18]; however, NHSOM has been trained by a maximum dosage of 10 (Section 3.1). Therefore, treatment plans of Gompertz model will be used in the scales of 1/5, 2/5 and 3/5 (i.e., with maximum dosage of 10, 20 and 30). Relation (7) supplies the dynamic of the incoming drug in each step of simulation. In addition, after retrieving the next state of the automata using NHSOM and also by applying relation (14), the incoming drug dosage of the automata changes simply to which relation (7) determines. The new automata with the desired incoming drug will be presented again to NHSOM.

Relation (15) presents the general form of the applied cost function of Gompertz model, in which C_N and C_D are positive constants, related to the special conditions of the patient, $N(T)$ is the final number of cancer cells and $\sum_{i=1}^{M} \sigma_i$ is total injected drug dosage [18].

$$J = C_N \log_{10}(N(T)) + C_D \sum_{i=1}^{M} \sigma_i. \tag{15}$$

Candidate cost functions in this paper are as follows:

1. J_1 with $C_N = 1$ and $C_D = 0$: The goal is minimizing the final number of cancer cells at the end of the treatment plan.
2. J_2 with $C_N = 0$ and $C_D = 1$: The goal is minimization of the drug dosage during the treatment plan.
3. J_3 with $C_N = 500$ and $C_D = 1$: The goal is to minimize the number of final cancer cells with the minimum drug dosage. According to the logarithmic form of the number of cells in relation (15), C_N is taken 500 times greater than C_D.

For each cost function, a set containing 30 best results of Gompertz model is used to simulate the multiscale model. It takes more than 113 days to simulate multiscale model, using numerical algorithm and, approximately, 40 min by NHSOM.

4.1.1. Minimizing number of cancer cells

For the cost function J_1, 30 treatment plans of Gompertz model are scaled with scaling factors of 1/5, 2/5 and 3/5. The multiscale model simulates using these plans. After determining the best result of the multiscale model, number of cancer cells is plotted in time (Fig. 6). The best treatment plan of Gompertz model is the best candidate for scaled versions 1/5 and 2/5 (Fig. 7a) of multiscale model; however, with a small difference in the last time of drug injection, the best result of scale 3/5 is equal to the best of Gompertz (Fig. 7b).

4.1.2. Minimizing number of cancer cells

In Fig. 8 the number of cancer cells is plotted in response of the scaled-best suggestion of Gompertz model. Although the number of cells in Gompertz model has a decreasing behavior, they saturate in the scaled versions of the multiscale model. According to the definition J_2, the best result of Gompertz model and all scaled versions of the multiscale model are the same (Fig. 9).

4.1.3. Minimizing number of cancer cells

For the cost function J_3 (Fig. 10), the best result of Gompertz model (however, with a small difference in drug scheduling) is approximately equal to scales 1/5, 2/5 and 3/5 (Fig. 11).

It is worth reminding that because of the logarithmic form of final number of cells in relation (15), C_N is taken 500 times greater than C_D for cost function J_3. However, in case of the multiscale model there is no need to use logarithmic form, while number of initial cells is 1057. Therefore, the result of the multiscale model with cost function J_3 has some similarity with its result in case of cost function J_1. As a result, after some statistical tests, a mapping between cost functions of Gompertz and the multiscale models should be found.

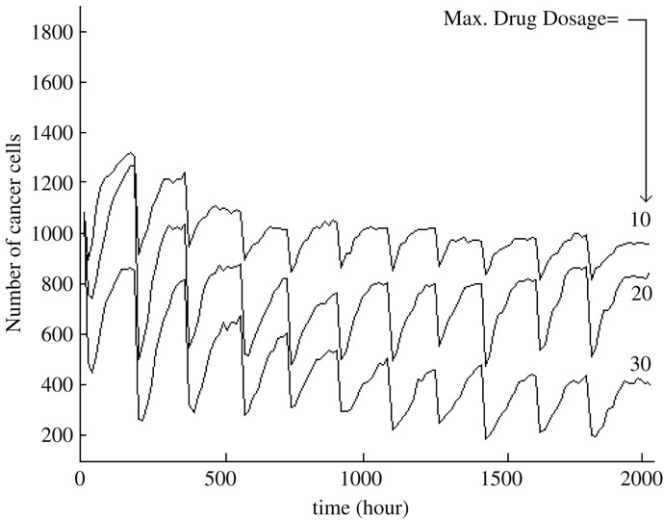

Fig. 6. Number of cancer cells for different scales of cost function J_1.

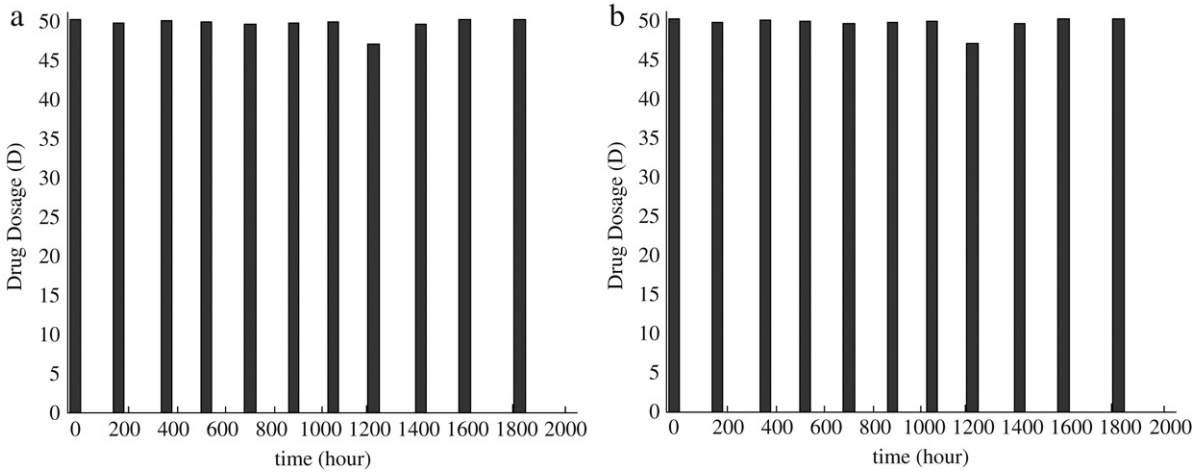

Fig. 7. (a) The best treatment plan of Gompertz model and scales 1/5 and 2/5 of the multiscale model. (b) The best treatment plan of scale 3/5 of the multiscale model.

4.2. Simulating the stress reduction emergently

Referring to Section 3.1, 61*61 array of automaton is used to simulate 2 mm^2 square of tissue [13]. It contains 1057 cancer cells. In Fig. 12 series of images in the case of cost function J_1, for the scale of 3/5, are given. Each image is taken after injecting the drug dosage, which is given in Section 4.1.

Simulation using NHSOM reveals some expectations about variations of cancer cells during chemotherapy. Cancer cells have a greater sensitivity to Doxorubicin's effects, and the main distortion of these cells is around the vessels, especially vessels which are near the inlet or outlet of the vascular network. Additionally, by eliminating the cancer cells, especially for the scales 2/5 and 3/5, normal tissue can gradually occupy the freed-up space. Such a phenomenon which occurs without considering direct issue in the system, is called *Emergent* [47–52].

"Luc Steels writes [48]: *A component has a particular functionality but this is not recognizable as a subfunction of the global functionality. Instead a component implements a behaviour whose side effect contributes to the global functionality [...]. Each behaviour has a side effect and the sum of the side effects gives the desired functionality*. In other words, the global or macroscopic functionality of a system with *emergent functionality* is the sum of all *side effects*, of all emergent properties and functionalities" [52].

The reason why NHSOM simulates the effect of decreasing stress on normal tissue is justifiable by a small of error during simulation using NHSOM. One of the necessary cautions in applying neural networks is error in retrieving information and diffusion of it [31,35]. Although NHSOM has greater accuracy in comparison with other NNs [32], it is not an error-free clustering method. However, a small of error during the simulation of the multiscale model sounds useful! It causes the

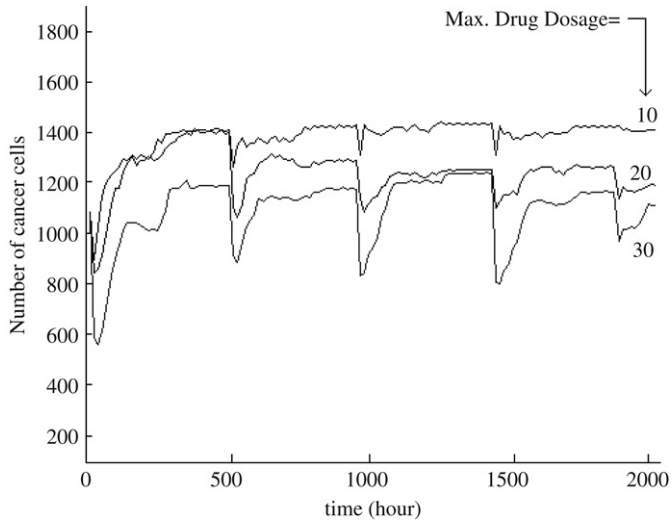

Fig. 8. Number of cancer cells for different scales of cost function J_2.

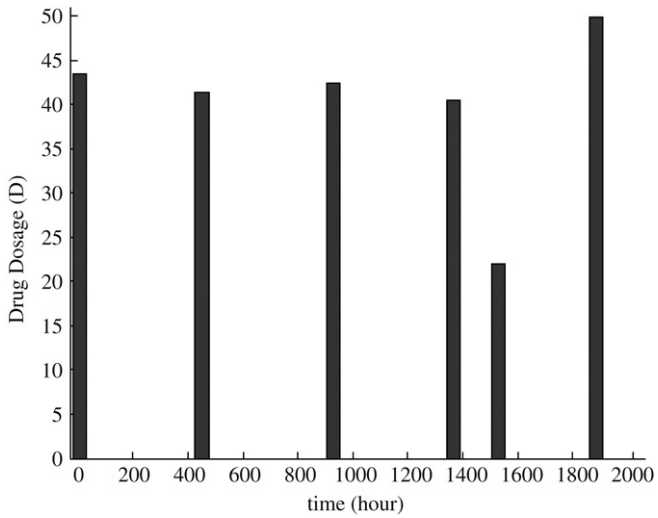

Fig. 9. The best treatment plan of Gompertz model and scales 1/5, 2/5 and 3/5 of the multiscale model.

next feature of a presented pattern, contains some normal cells instead of empty spaces after removing cancer cells. As a result, it seems that the compressed tissue around the tumor can gradually capture the freed-up space after injecting the chemotherapy drug.

In Figs. 13 and 14, series of images for cost functions J_2 and J_3 for scales 1/5 and 2/5, are given. For treatment plans with lower drug injections (cost function J_2), the cells have a saturating manner. It is worth remembering that the multiscale model is sensitive to adjustment of its parameters [5–9]. As a result, changes in them forces NHSOM to train with a new sample set, and to change the final results accordingly.

5. Conclusion

In this paper, an artificial neural network called NHSOM (Nested Hierarchical Self Organizing Map) was presented to simulate the multiscale cancer model faster than the numerical algorithm. Because of the advantages of NHSOM, the proposed neural network has, potentially, a good ability to apply for data sets which do not have a large set of training data, or in cases where there is no predetermined statistical information about some features of the network, such as the number of neurons. Therefore, NHSOM is a suitable candidate for simulating the multiscale model.

Before applying NHSOM, a feature selection phase is passed. For the best performance of NHSOM, each feature vector is chosen as a sequence of state vectors of a 5*5 array of an automaton. Some important elements of the vector are scaled,

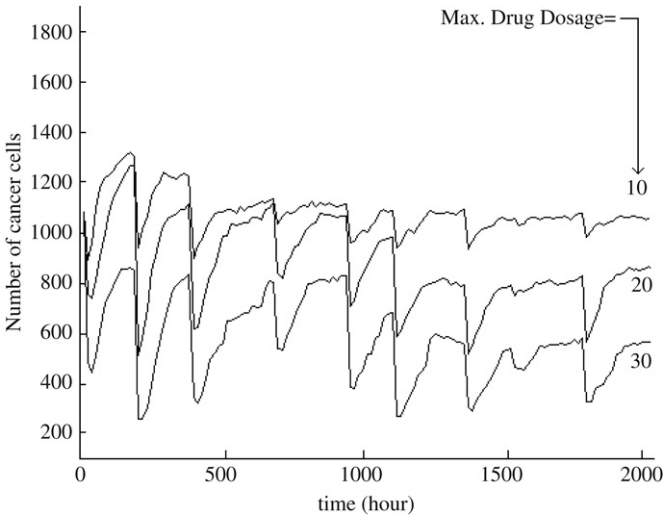

Fig. 10. Number of cancer cells for different scales of cost function J_3.

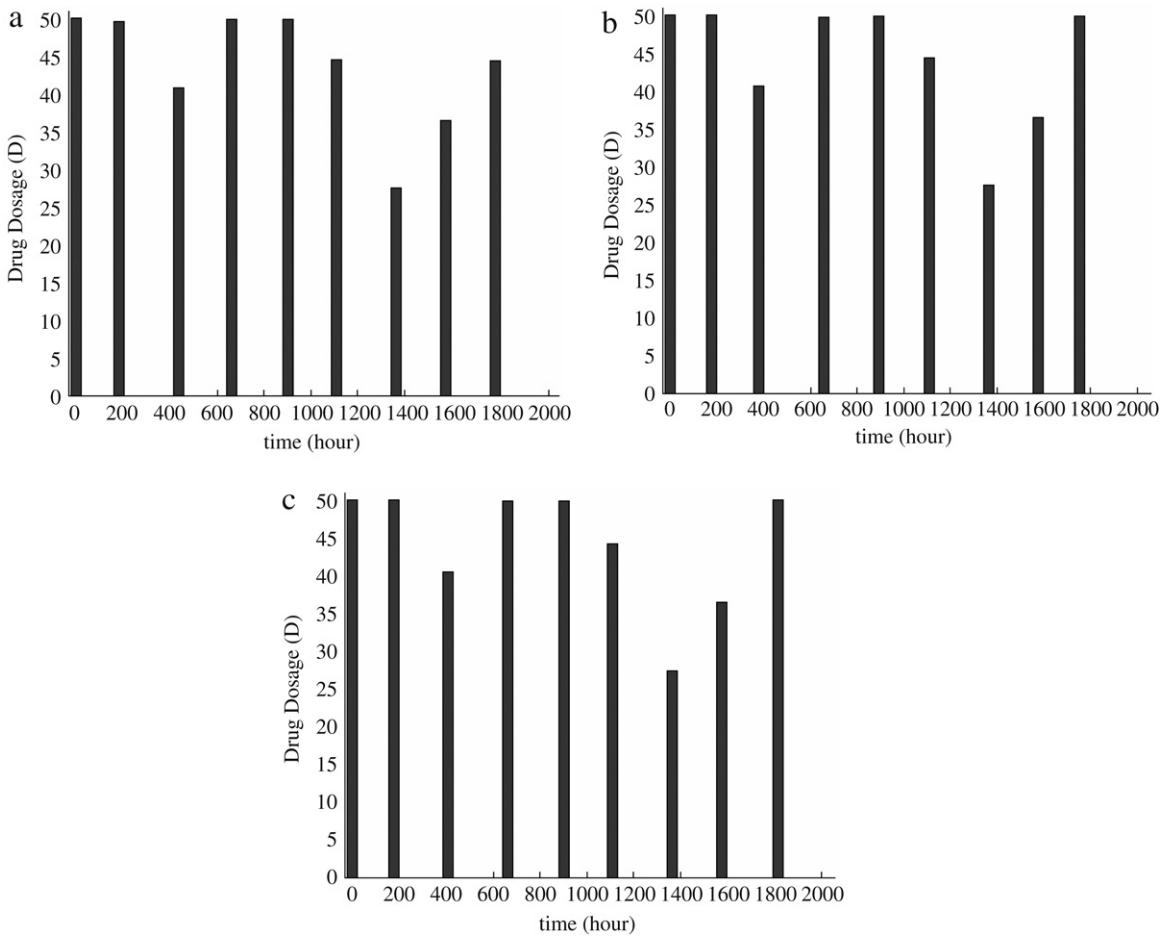

Fig. 11. (a) The best treatment plan of Gompertz model. (b) The best treatment plan of scale 1/5 of the multiscale model. (c) The best treatment plan of scales 2/5 and 3/5 of the multiscale model.

shifted and repeated. In the training phase, the next features of the presented samples are saved at the bottom layer of NHSOM. In addition, a second weight is added to each neuron, which is the normalized-next feature of the elementary

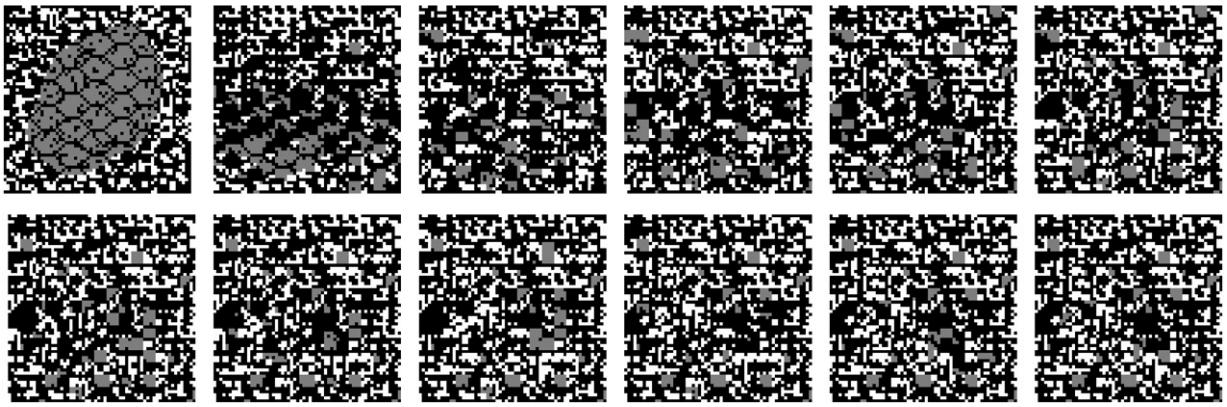

Fig. 12. The series of images (*from left to right and top to down*) are taken after injecting drug dosage (Fig. 7b) for cost function J_1 in the scale of 3/5.

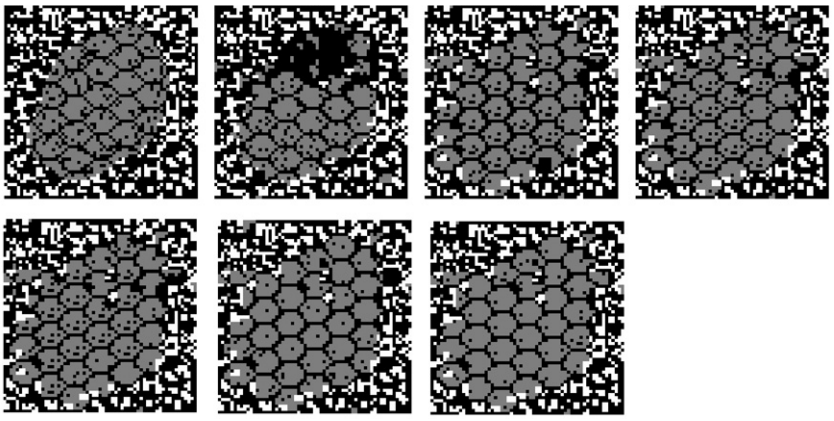

Fig. 13. The series of images (*from left to right and top to down*) are taken after injecting drug dosage (Fig. 9) for cost function J_2 in the scale of 1/5.

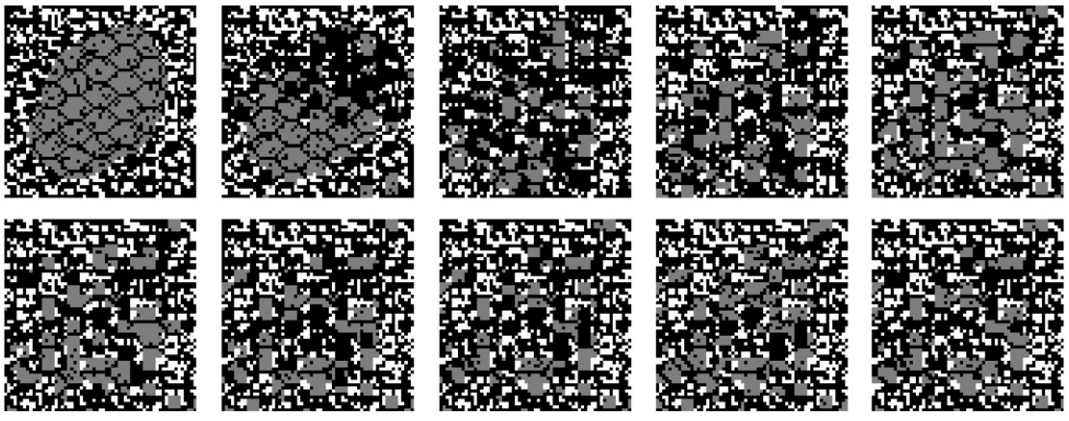

Fig. 14. The series of images (*from left to right and top to down*) are taken after injecting drug dosage (Fig. 11c) for cost function J_3 in the scale of 2/5.

weight of that neuron. By using the second weight for each neuron, the winner will be updated if the vigilance test passes both elementary and secondary weights.

Although the performance of NHSOM is better than those of other neural networks, even with a low number of training samples, it is not an error-free network. A small error during the simulation phase causes some normal cells to gradually occupy the freed-up space, when retrieving the stored features after drug injection. As a result, the effect of decreasing stress on the normal tissue is simulated emergently.

Because of long lasting process of checking a simple state space of the multiscale model, even by applying NHSOM, the best results of Gompertz model are applied to chemotherapy of the multiscale model. However, more statistical tests are needed to determine the correct mapping between these models.

Another advantage introduced in this paper, is that by assuming that the distribution of drug proportional to blood flow rate, a linear relation between two different drug diffusions with the same automaton structure is found. The main advantage of the mentioned relation, is that it is possible to train NHSOM using limited samples, and to scale the incoming drug dosage to any other one, unlimitedly.

However, there are many drawbacks to test the validity of the proposed methods. It has been mentioned that some statistical tests need to find the correct mapping between two models. If such a mapping is found, there will be no need to test various treatment plans in the multiscale model. A combination of two models may be used to apply multi treatment plans, such as a combination of chemotherapy and radiotherapy. In this case, by applying the best treatment plan of Gompertz model, the multiscale model will present the variations of the tumor via images. Therefore, the border of the tumor after applying any chemotherapy treatment will be determined, which is needed for radiotherapy.

NHSOM is an appropriate tool in order to extend the size of a cellular automaton array. In this case, it might be better to build a coarse structure of a vascular network which is divided to some finer vascular structures applied in the proposed setting of this paper and others [5–7,9]. A tumor with normal size consists of some arrays of automata with finer vascular structure.

In the proposed technique, the feature selection phase is done step by step by a human designer. This process may be done systematically, using some artificial methods and in combination with some classical methods such as Principle Component Analysis (PCA) [35].

References

[1] E. Haga, Computer Techniques In Biomedicine And Medicine: Simulation And Modeling, Health Care, And Image Processing, Auerbach Publishers, 1973.
[2] F.C. Hoppensteadt, C.S. Peskin, Modeling and Simulation in Medicine and the Life Sciences, Springer, 2002.
[3] S. Cotin, D.N. Metaxas, Medical Simulation: International Symposium, ISMS 2004, Cambridge, MA, USA, June 2004, Proceedings, Springer, 2004.
[4] L. Preziosi, Cancer Modelling and Simulation, in: Mathematical and Computational Biology, vol. 3, CRC Press, 2003.
[5] T. Alarcon, H.M. Byrne, P.K. Maini, A cellular automaton model for tumour growth in inhomogeneous environment, Journal of Theoretical Biology 225 (2003) 257–274.
[6] T. Alarcon, H.M. Byrne, P.K. Maini, A multiple scalel model for tumor growth, Multiscale Modeling Simulation 3 (2) (2005) 440–475.
[7] H.M. Byrne, M.R. Owen, Modelling the response of vascular tumors to chemotherapy: A multiscale approach, Mathematical Models and Methods in Applied Sciences 16 (7S) (2006) 1219–1241.
[8] T. Alarcon, H.M. Byrne, P.K. Maini, A mathematical model of the effects of hypoxia on the cell-cycle of normal and cancer cells, Journal of Theoretical Biology 229 (2004) 395–411.
[9] T. Alarcon, H.M. Byrne, P.K. Maini, Towards whole-organ modelling of tumour growth, Progress in Biophysics & Molecular Biology 85 (2004) 451–472.
[10] H.M. Byrne, T. Alarcon, M.R. Owen, S.D. Webb, P.K. Maini, Modelling aspects of cancer dynamics: A review, Philosophical Transactions of the Royal Society of London, Series A 364 (2006) 1563–1578.
[11] A. Marchiniak-Cozchra, M. Kimmel, Modelling of early lung cancer progression: Influence of growth factor production and cooperation between partially transformed cells, Mathematical Models & Methods in Applied Sciences 17 (2007) 1693–1720.
[12] Y. Kim, M.A. Stolarska, H.G. Othmer, A hybrid model for tumor spheroid growth in vitro I: Theoretical development and early results, Mathematical Models & Methods in Applied Sciences 17 (2007) 1773–1798.
[13] B. Ribba, K. Marron, Z. Agur, T. Alarcon, P.K. Maini, Amathematical model of Doxorubicin treatment efficacy for non-Hodgkin's lymphoma: Investigation of the current protocol through theoretical modeling results, Bulletin of Mathematical Biology 67 (2005) 79–99.
[14] R.S. Day, Challenges of biological realism and validation in simulation-based medical education, Artificial Intelligence in Medicine 38 (2006) 47–66.
[15] N. Bellomo, N.K. Li, P.K. Maini, On the foundations of cancer modelling: Selected topics, speculations, and perspectives, Mathematical Models & Methods in Applied Sciences 18 (2008) 593–646.
[16] F. Mollica, L. Preziosi, K.R. Rajagopal (Eds.), Modelling Biological Tissues, Birkäuser, Boston, 2006.
[17] R.B. Martin, Optimal control drug scheduling of cancer chemotherapy, Automatica 6 (1992) 1113–1123.
[18] E. Bavafa, M.J. Yazdanpanah, B. Kalaghchi, Chemotherapy using linear analysis and swarm intelligence, in: International Federation of Automatic Control (IFAC), 2008, pp. 5233–5238.
[19] A. Swiernlak, U. Ledzewicz, H. Schättler, Optimal control for a class of compartmental models in cancer chemotherapy, International Journal on Applied Mathematics and Computer Science 13 (3) (2003) 357–368.
[20] M. Kimmel, A. Swierniak, Using control theory to make cancer chemotherapy beneficial from phase dependence and resistant to drug resistance, Archives of Control Science 14 (2) (2004) 105–145.
[21] N.V. Mantzaris, S. Webb, H.G. Othmer, Mathematical modeling of tumor-induced angiogenesis, Journal of Mathematical Biology (2004) 233–257.
[22] H.A. Levine, S. Pamuk, B. Sleeman, M. Nilsen-Hamilton, Mathematical modeling of capillary formation and development in tumor angiogenesis: Penetration into the stroma, Bulletin of Mathematical Biology 63 (2001) 801–863.
[23] B.D. Sleeman, I.P. Wallis, Tumour induced angiogenesis as a reinforced random walk: Modelling capillary network formation without endothelial cell proliferation, Mathematical and Computer Modelling 36 (2002) 339–358.
[24] T. Hillen, H. Othmer, The diffusion limit of transport equations derived from velocity jump processes, SIAM Journal on Applied Mathematics 61 (2000) 751–775.
[25] Y. Dolak, C. Schmeiser, Kinetic models for chemotaxis: Hydrodynamic limits and spatio temporal mechanisms, Journal of Mathematical Biology 51 (2005) 595–615.
[26] F.A. Chalub, Y. Dolak-Struss, P. Markowich, D. Oeltz, C. Schmeiser, A. Soref, Model hierarchies for cell aggregation by chemotaxis, Mathematical Models & Methods in Applied Sciences 16 (2006) 1173–1198.
[27] N. Bellomo, A. Bellouquid, J. Nieto, J. Soler, Multicellular growing systems: Hyperbolic limits towards macroscopic description, Mathematical Models & Methods in Applied Sciences 17 (2007) 1675–1693.
[28] V.T. DeVita, S. Hellman, S. Rosenberg, in: Lippincott Williams (Ed.), Cancer: Principles and Practice of Oncology, 7th ed., 2005.
[29] J. Folkman, Angiogenesis and apoptosis, Seminars in Cancer Biology 13 (2003) 159–167.
[30] L.M. Franks, N.M. Teich, Introduction to the Cellular and Molecular Biology of Cancer, third ed., OXFORD, 1997.
[31] J. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, 1992.

[32] E. Bavafaye-Haghighi (Bavafa), Applying Intelligent Methods for Chemotherapy of Multiscale Cancer Model, M.Sc. Thesis, School of Electrical and Computer Engineering, University of Tehran, 2008.
[33] E. Bavafa, M.J. Yazdanpanah, Image compression using an enhanced self organizing map algorithm with vigilance parameter, in: International Joint Conference on Neural Networks, 2006, pp. 1923–1928.
[34] L. Cinque, G.L. Foresti, A. Gumina, S. Levialdi, A modified fuzzy art for image segmentation, in: 11th International Conference on Image Analysis and Processing, ICIAP'01, 2001, pp. 102–17.
[35] S. Theodoridis, K. Koutroumbas, Pattern Recognition, second ed., Elsevier Academic Press, 2003.
[36] Sh. Abe, Pattern Classification, Neuro-Fuzzy Methods, and their Comparison, Springer, 2001.
[37] J.J. Aleshunas, D.C.St. Clair, W.E. Bond, Classification characteristics of SOM and ART2, in: Proceedings of the ACM Symposium on Applied Computing, 1994, pp. 297–302.
[38] R. Tagliaferri, N. Capuano, G. Gargiulo, Automated labeling for unsupervised neural networks: A hierarchical approach, IEEE Transactions on Neural Networks 10 (1999) 199–203.
[39] J. Ontrup, H. Ritter, Large-scale data exploration with the hierarchically growing hyperbolic SOM, Neural Networks 19 (2006) 751–761.
[40] M. Ditenbach, A. Rauber, D. Merkel, Uncovering hierarchical structure in data using the growing hierarchical self-organizing map, Neurocomputing 48 (2002) 199–216.
[41] A.G. Gillman, L. Goodman, The Pharmacological Basis of Therapeutics, 10th ed., McGraw Hill, 2001.
[42] http://yann.lecun.com/exdb/mnist/.
[43] G.A. Carpenter, S. Grossberg, The art of adaptive pattern recognizing by a self organizing neural network, Computer 21 (3) (1988) 77–88.
[44] T. Kohonen, Things you haven't heard about the self-organizing map, in: IEEE International Conference on Neural Networks, 1993, pp. 1147–1156.
[45] T. Kohonen, Comparisons of som point densities based on different criteri, Neural Computation 11 (1999) 2081–2095.
[46] Gio wiederhold, File Organization for Database Design, McGraw-Hill, 1987.
[47] M.J. Mataric, Behavior-based robotics as a tool for synthesis of artificial behavior and analysis of natural behavior, Trends in Cognitive Sciences 2 (3) (1998) 82–87.
[48] Luc Steels, Towards a theory of emergent functionality, in: From Animals to Animats (Proceedings of the First International Conference on Simulation of Adaptive Behaviour), Bradford Books, MIT Press, 1990, pp. 451–461.
[49] L. Steels, Emergent functionality in robotic agents through on-line evolution, in: Proceedings of Artificial life IV, MIT Press, Cambridge, 1994, pp. 8–14.
[50] O. Shehory, S. Kraus, O. Yadgar, Emergent cooperative goal-satisfaction in large scale automated-agent systems, Artificial Intelligence 110 (1) (1999) 1–55.
[51] M.A. Bedau, Weak Emergence, in: James Tomberlin (Ed.), Philosophical Perspectives: Mind, Causation, and World, vol. 11, Blackwell Publishers, 1997, pp. 375–399.
[52] http://en.wikipedia.org/wiki/Emergence.