Neural Dynamic Programming for Optimal Control of Large Genetic Regulatory Networks

Ahmad T. Abdulsadda
Communication Department
Al-Najaf Technical College
Technical Educational Foundation of Iraq

Abstract:

The modeling and control of genetic regulatory networks carries tremendous potential for gaining a deep understanding of biological processes, and for developing effective therapeutic intervention in diseases such as cancer. A dynamical programming control has been proposed for determining an optimal intervention policy to shift the steady-state distribution of the network. The dynamic programming solution is, however, computationally prohibitive for large gene regulatory networks, as its complexity increases exponentially with the number of genes. Since the number of genes considered is directly related to the accuracy of the model, it is imperative to be able to design optimal intervention policies that can be reasonably implemented for large gene regulatory networks. To this endeavor, we will design a neural dynamic programming controller to optimize the same dynamic programming performance measure, while requiring only a polynomial time complexity. The proposed neural dynamic programming structure includes two networks: action and critic. The critic network is trained toward optimizing a total reward to objective, namely to balance the Bellman equation. The action network, constrained by the critic network, generates the optimal control strategy. Both the control strategy and the critic output are updated according to an error function that changes from one step to another. General theory of non-homogeneous Markov chain will be used to find the optimal strategies of non uniform policy method.

Keywords:

Genomic Signal Processing (GSP) , Boolean Network structure, Neural Dynamic Programming networks (NDP), Bellman Optimal Equation.

برمجة العصبية الحيوي لفي التحكم الأمثل لشبكات كبيرة التنظيمية الوراثية

د. احمد طه عبد الساده جبار

الخلاصة:

النمذجة والتحكم في الشبكات التنظيمية الوراثية يحمل إمكانات هائلة لاكتساب فهم عميق من العمليات البيولوجية، والتدخل العلاجي لتطوير فعالية في أمراض مثل السرطان. وقد تم اقتراح برمجة التحكم الديناميكي لتحديد سياسة التدخل الأمثل

لتحويل توزيع ثابتة للدولة للشبكة. الحل البرمجة الديناميكية، مع ذلك، باهظة حسابيا لشبكات الجينات التنظيمية الكبيرة، وتعقيدها يزيد أضعافا مضاعفة مع عدد من الجينات. منذ يرتبط ارتباطا مباشرا على عدد من الجينات التي تعتبر دقة النموذج، لا بد أن تكون قادرة على وضع سياسات التدخل الأمثل التي يمكن تنفيذها بشكل معقول لشبكات الجينات التنظيمية الكبيرة. في هذا المسعى، ونحن تصميم العصبية البرمجة الديناميكية تحكم لتحسين أداء نفس الإجراء البرمجة الديناميكية، في حين لا تتطلب سوى تعقيد الوقت متعدد الحدود. هيكل المقترح العصبية البرمجة الديناميكية تضم اثنين من الشبكات: العمل والناقد. يتم تدريب نحو تحسين شبكة الناقد مكافأة لإجمالي الهدف، ألا وهو تحقيق التوازن في المعادلة المنادي. شبكة العمل، مقيدة شبكة الناقد، يولد استراتيجية التحكم الأمثل. يتم تحديث كل من استراتيجية السيطرة وإخراج الناقد وفقا لوظيفة من الخطأ أن التغييرات خطوة واحدة إلى أخرى. وسوف تستخدم النظرية العامة للسلسلة ماركوف غير متجانسة للعثور على استراتيجيات الأمثل للأسلوب غير سياسة موحدة.

INTRODUCTION

Genomic signal processing (GSP) is the engineering discipline that studies the processing of genomic signals. Genomic signals must be processed to characterize their regulatory effect and their relationship to changes at both genotype and phenotype levels. GSP contains different methodology involving detection, prediction, classification, control, statistical, and dynamical modeling of gene networks. Let's give the reader some basic idea about the natural operation for human cells, and how they are replicating by cell division, and irreparably damaged cells remove themselves by process called apoptosis. Each cell contains biological instruction called DNA and must be replicated and handed down unchanged to the cells progeny when it divides. Each cell could make copy from DNA, which is called RNA and the final stage generates another protein type called the messenger MRNA. What a designer needs to figure out is that if the gene express (it is on) there is protein, otherwise the cell does not express (the gene is off) and there is no protein generated. These processes occur in the natural condition with the healthy person. Control process inside the cells depends on complex interaction between the products of the cell and those environments. As might be expected from a highly complex, efficient and survivable system, control is highly distributed and redundant. Any uncontrolled process inside the cells causes uncontrolled divide in them.

From a translational perspective, the ultimate objective of genetic regulatory network modeling is to use the network to design different approaches for affecting network dynamics in such a way as to avoid undesired phenotypes, such as, cancer [1]. Recently, in the hardware side there are new equipments, they did convert the information code from the genotype to binary information, one of those strategies is the micro-array strategy used to obtain the binary and ternary gene expression in discrete case (quantization process) is generically called Probabilistic Boolean Networks (PBNs), [2-4]. The basic PBNs structure introduced by Kauffman [5-7] to allow the incorporation of uncertainty into the inter-gene relationships. Any given PBN should have a state transition matrix or transition probability to move from one state to another. The process control for this dynamic state matrix can be studied in the context of homogeneous Markov chains with finite state space. Basically, the major goal of functional genomics is to screen for genes that determine specific cellular phenotype (disease) and model their activity. Engineering therapeutic tools involves nonlinear dynamical networks, to characterize gene regulation, and developing intervention strategies to modify dynamical behavior.

Intervention studies have used three different approaches to deal with problem: (i) resetting the state of the PBN to a more desirable initial state [3] ; (ii) changing the steady-state (long run)

behavior of the network by minimally altering its predicator function[4]; (iii) manipulating external (control) variables that alter the transition probabilities of the network[8]. In [8]they proposed control method, dynamic programming is employed to finite a finite horizon control sequence intervention. In [9], they solved the Bellman dynamic programming using infinite horizon control (there was not needed to know the terminal state information). Basically, in the two methods of [8] and [9] is well known that the direct application of optimal control methods is limited by the size of the state space-the curse of dimensionality. For larger biological models involving interactions among many genes, a stochastic control method has been polynomial time complexity. Recently, there is a group in university of Texas [13], they formulated the problem of controlling a context sensitive PBN as a Markov chain with reward. The refinement learning method (Q learning) was used to find the optimal strategies. Although the Q learning method deals with a larger genetic problem, it is still suffering from lengthy calculation time. It took two days to find the optimal strategies for a problem had 15 genes coded in binary form. Nidhal and at el [14-17], have been proposed an optimal perturbation control scheme to solve the dynamic equation, we thought this scheme still consuming a lot of time to do the calculation.

In this paper, we proposed a novel method using neural dynamic programming controller to optimize the same dynamic programming performance measure, while requiring only a polynomial time complexity. The proposed neural dynamic programming structure includes two networks: action and critic. The critic network is trained toward optimizing a total reward to objective, namely to balance the Bellman equation. The action network, constrained by the critic network, generates the optimal control strategy. Both the control strategy and the critic output are updated according to an error function that changes from one step to another. General theory of non-homogeneous Markov chain will be used to find the optimal strategies of non uniform policy method. Simulation has been conducted to examine the effectiveness of the proposed scheme.

The remainder of the paper is organized as follows. Problem formulation: the definition of Boolean networks, general control process strategies, and solution using dynamic programming in section 2. Dynamic Neural Programming (DNP) structure was described in section 3. Section 4, contains the results of the control strategies process. Finally, section 5 Conclusion.

PROBLEM FORMULATION

Boolean Network Structure

The Context-sensitive Probabilistic Boolean networks PBN consists of a set $V = \{x_1,\ldots, x_n\}$, of n nodes ,where $x_i \in \{0,1,\ldots,d-1\}$, and a set $\{f_1,f_2,\ldots,f_k\}$ of vector-valued functions, called predictor functions. In the framework of gene regulation, each x_i, for i= 1… n, represents the expression value of a gene. It is common to mix terminology by referring to x_i as the i^{th} gene. Each vector valued function f_1 which has the form of $f_l = (f_{l1},\ldots,f_{1n})$, determines a constituent network of the context-sensitive PBN. The context sensitive PBN with control can be modeled as a stationary discrete time dynamic system:

$$x(k+1) = f(x_k, u_k, w_k) \quad (1)$$

Where for all k, the state x_k is an element of a space *S*, the control input u_k is an element of space C, the w_k is disturbance in the space D. finally, $f : S \times C \times D \rightarrow S$.

In the particular case of context sensitive PBNs of n genes composed of N Boolean networks with perturbation probability p and network transition probability q, S =[0,1,2,…,2^n-1], where n numbers of gene. The control signal u_k should be in space C =[0,1,2,…,2^m-1],where m is the number of control inputs. Another equivalent way to represent the dynamical system in (1) is as a finite state Markov Chain described by the control dependent one step transition probability $p_{ij}(u_k)$, where for any k =0,1,2,…,N ; i,j ε S and u ε C:

$$p_{ij}(u_k) = P(x_{k+1} = j | x_k = i, u_k = u) \ . \qquad (2)$$

The one-step evolution of the probability distribution in the case of a PBN containing 2^n states with control inputs can be described with following equation [8, 9]:

$$pd_{k+1} = pd_k A(u_k) \ . \qquad (3)$$

Where pd_k is the 2^n dimensional state probability distribution vector at time *k*, and $A(u_k)$ is the $2^n \times 2^n$ matrix of control dependent transition probabilities ,i.e., $A(u_k)$ is the matrix whose ijth element is $P_{i-1,j-1}(u_k)$. Equation (3) represented the main point in our work, because if the system starts with any initial state probability vector, it could end with the desired one depends on the probability transition matrix and the input control.

Solution Using Dynamic Programming

The optimal control problem can now be stated as follows: Given an initial state *x(0)*, find a control law $\pi = \{u_0, u_1, \ldots, u_{M-1}\}$ that minimizes the cost functional

$$J_\pi(x(0)) = E\left[\sum_{k=0}^{M-1} C_k(x_k, u_k(x_k)) + C_M(x(M))\right] \ ,$$

Subject to the constraint

$$\Pr\{x_{k+1} = j | x_k = i\} = a_{ij}(u_k) . \qquad (4)$$

Where $a_{ij}(u_k)$ is the i^{th} row, j^{th} column entry of the matrix $A(u_k)$. Optimal control problems of the described by equation (4) can be solved using the technique of dynamic programming. The dynamic programming solution to eq.(4) was derived in [10, 11]:

$$J_M(x(M)) = C_M(x(M)) ,$$

$$J_k(x_k) = \min_{u_k \in \{1,2,\ldots,2^m\}} \left\{ C_k(x_k, u_k) + \sum_{j=1}^{2^n} a_{ij}(u_k) . J_{k+1} \right\}, k = 0, 1, 2, ..., M-1; i = 1, 2, ..., 2^n . \qquad (5)$$

Where $C_M(x(M))$ is the terminal cost at terminal state $x(M)$, M is the finite number of steps. Note that the expectation on the right hand side of equation (5) for each x_k and k:

$$E\left[J_{k+1}(x_{k+1}|x_k,u_k)\right]=\sum_{j=1}^{2^n}a_{ij}(u_k)J_{k+1}. \tag{6}$$

Thus, the final solution of the dynamic programming system (equation (4)), which is known as Bellman equation is given as [10, 11]

$$\begin{aligned} &J_M(x(M))=C_M(x(M)),\\ &J_k(x_k)=\min_{u_k\in\{1,2,\ldots,2^m\}}\left\{C_k(x_k,u_k)+\sum_{j=1}^{2^n}a_{ij}(u_k).J_{k+1}\right\},k=0,1,2,\ldots,M-1;i=1,2,\ldots,2^n. \end{aligned} \tag{7}$$

Neural Dynamic Programming

The objective of a dynamic neural programming controller is to optimize a desired performance measure by learning to create appropriate control action through interaction with the environment the controller is designed to learn to perform better over time using only sampled measurement and with no prior knowledge about the system. Figure 1 shown as a schematic diagram of Neural Dynamic Programming (NDP) online learning control scheme, which has two main neural networks, the Action and the Critic networks.

To be more quantitative, consider the critic network shown in Figure 2, the output of the critic element is the objective function J, which represented approximates the discounted total reward-to-go. Specifically, it approximates R_k at time k. It can be calculated as:

$$R_k=r_{k+1}+\gamma\, r_{k+2}+\ldots \tag{8}$$

Where R_k is the future accumulative reward-to-go value at time k, γ is a discount factor for infinite horizon problem ($0<\gamma<1$), since the exact value for the discount factor is given by:

$$\gamma=\frac{1}{r+1} \tag{9}$$

Where r is the external reinforcement value at time k, r_{k+1} is the external reinforcement value at time $k+1$. Before go deeply in critic and action networks details, let's defined the Ultimate function U and reinforcement r_k.

The ultimate function is the only source of information the Adaptive Process (ADP) has about the task for which it is designing the controller. When the statement is made that dynamic programming designs an optimal controller, optimality is defined strictly in terms of the ultimate

function. It is important to recognize that a different *U* function will (typically) yield a different controller. So that, in our case we defined it as:

$$U = \begin{cases} -1 & \text{if} \quad \text{u} = 1 \\ 0 & \text{if} \quad \text{u} = 0 \end{cases} \tag{10}$$

As we mentioned before, the reinforcement signal *r* is used to find the critic error which referred the adaptive error the adjusted the weights of critic network depend upon the back propagation principle. If deal with state probability vector we can define the following reward per stage function

$$r_{k} = -\sum_{k}^{M} \frac{pd_{k} - pd^{*}{}_{k}}{pd_{\max}} \ , \tag{11}$$

Where $pd^{*}{}_{k}$ is a desired probability distribution vector, pd_{k} current distribution vector. In practice, the reward values will have to mathematically capture the benefits and costs of intervention and the relative preference of probability state, and have to set by physicians in accordance with their clinical judgment, [9].

Action-Critic Neural Networks Structure

Generally, the neural dynamic programming provides a suitable structure to solve the dynamic programming equation exactly like the Bellman equation. So that NDP aims to find the optimal objective function:

$$J^{*}{}_{k} = r_{k} + \gamma J^{*}{}_{k+1} \tag{12}$$

Based on Figure 2, the predication error of the critic network is calculated as:

$$e_{c}(k) = \gamma J_{k} - (J_{k+1} - r_{k}) \ , \tag{13}$$

$$E_{c}(k) = \frac{1}{2} e_{c}(k) \ , \tag{14}$$

where E_c is the mean square error for critic network. This error provides the desired objective function for critic network to minimize by tuning critic weights. Principles used in the weight update for the critic can be derived through gradient decent as below:

$$J = b^{2}{}_{c}(k) + \sum_{i=1}^{H} w^{2}{}_{ci}\, p_{i}, \tag{15}$$

where

$$p_i = \frac{1}{1+e^{-q_i}}; \quad \mathrm{i} = 1,2,...,\mathrm{H}, \tag{16}$$

and

$$q_i = b_c^1 + \sum w_c^1[ij]x[j]; \quad \mathrm{i} = 1,2,.....,\mathrm{H}. \tag{17}$$

The weight and bias update in critic networks are given as:

$$\begin{aligned} b_c(k+1) &= b_c(k) + \Delta b_c(k), \\ w_c(k+1) &= w_c(k) + \Delta w_c(k). \end{aligned} \tag{18}$$

Depending on back propagation algorithm, we have driven the exact values of the following biases and weights:

- Hidden to output bias and weights

$$\begin{aligned} b_c^2[j]_{k+1} &= b_c^2[j]_k - l_c \gamma e_c(k), \\ w_c^2[j]_{k+1} &= w_c^2[j]_k - l_c \gamma e_c(k) p[j]_k; \qquad \mathrm{j} = 1,2,...,\mathrm{H}, \end{aligned} \tag{19}$$

- Input to hidden bias and weights

$$\begin{aligned} b_c^1[j]_{k+1} &= b_c^1[j]_k - l_c \gamma e_c(k) w_c^2[\mathrm{i}]_k \left(\mathrm{p}[\mathrm{i}]_k (1 - p[i]_k)\right), \\ w_c^1[j]_{k+1} &= w_c^1[j]_k - l_c \gamma e_c(k) w_c^2[\mathrm{i}]_k \left(\mathrm{p}[\mathrm{i}]_k (1 - p[i]_k)\right) \mathrm{x[j]}; \ \mathrm{i} = 1,2,...,\mathrm{n}+1, \mathrm{j} = 1,2,...,\mathrm{H}. \end{aligned} \tag{20}$$

Now, we investigate the adaptation in the action network shown in Figure 3.

In action neural network the input layer has *n* input (probability distributed state vector) and one output node which represent the control u_k. The associated equations for the action network are:

$$e_a(k) = J_k - U, \tag{21}$$

$$E_a(k) = \frac{1}{2} e_a, \tag{22}$$

$$u[i]_k = \frac{1}{1+e^{-v[i]_k}}; \qquad \mathrm{i} = 1,2,...,\mathrm{m}, \tag{23}$$

$$\mathrm{v[i]_k} = b_a^2[i] + \sum_{j=1}^{H} w_a^2[ij] g[j]_k; \ \mathrm{i} = 1,2,...,\mathrm{m}, \tag{24}$$

(25)

$$g[j]_k = \frac{1}{1+e^{-h[i]_k}}; \qquad \mathrm{j}=1,2,...,\mathrm{H},$$
$$\mathrm{h[i]_k} = b_a^1[i] + \sum_{i=1}^{n} w_a^1[ji]x[i]_k; \quad \mathrm{j}=1,2,...,\mathrm{H} \tag{26}$$

The weight and bias update in action networks are given as:

$$b_a(k+1) = b_a(k) + \Delta b_a(k),$$
$$w_a(k+1) = w_a(k) + \Delta w_a(k). \tag{27}$$

The update rule for the nonlinear MLP action network also contains two sets of equations:

- Hidden to the output nodes in output layer bias and weights

$$b_{aj}^2(k+1) = b_{aj}^2(t) - l_a e_a(k)\,[u_j(k)(1-u_j(k))] \sum_{i=1}^{H} w_{ci}^2(k)[p_i(k)(1-p_i(k))w_{i,n+1}^1(k); j=1,2,\ldots,m$$

$$w_{aij}^2(k+1) = w_{aij}^2(k) - l_a e_a(k)\,[u_j(k)(1-u_j(k))]\, g_i(k) \sum_{i=1}^{H} w_{ci}^2(k)[p_i(k)(1-p_i(k))w_{i,n+1}^1(k); j=1,2,\ldots,m \tag{28}$$

- Input to hidden layer

$$b_{aj}^1(k+1) = b_{aj}^1(k) - l_a e_a(k)[u_j(k)(1-u_j(k))w_i^2[g_i(k)(1-g_i(k)) \sum_{i=1}^{H} w_{ci}^2(k)[p_i(k)(1-p_i(k))]w_{i,n+1}^1(k);$$
$$j=1,2,\ldots,m$$

$$w_{aij}^1(k+1) = w_{aij}^1(k) - l_a e_a(k)[u_j(k)(1-u_j(k))]w_i^2[g_i(k)(1-g_i(k))]x_j(k)$$
$$\sum_{i=1}^{H} w_{ci}^2(k)[p_i(k)(1-p_i(k))]w_{i,n+1}^1(k); j=1,2,\ldots,m \tag{29}$$

SIMULATION RESULTS

We apply the proposed neural dynamic programming control to a probabilistic Boolean network derived from gene expression data collected in a study of metastatic melanoma [8]. The abundance of *mRNA* for the gene *WNT5A* was found to be highly discriminating between cells with properties typically associated with high versus low metastatic competence. Furthermore, it was found that an intervention that blocked the *Wnt5a* protein from activating its receptor, the use of an antibody that binds the Wnt5a protein, could substantially reduce *Wnt5A*'s ability to induce a metastatic phenotype [8]. This suggests a control strategy that reduces the WNT5A genes action in affecting biological regulation. A seven-gene probabilistic Boolean network (PBN) model of the melanoma network containing the genes*WNT5A*, *pirin*, *S100P*, *RET1*, *MART1*, *HADHB*, and *STC2* was derived in [8-10]. Figure 4, derived in [8-10], illustrate the relationship between genes in the Human melanoma regulatory network. Note that the Human

melanoma Boolean network consists of $2^7 = 128$ states ranging from 00 … 0 to 11 …1, where the states are ordered as *WNT5A*, *pirin*, *S100P*, *RET1*, *MART1*, *HADHB*, and *STC2*, with WNT5A and STC2 denoted by the most significant bit (MSB) and least significant bit (LSB), respectively. We observer that the states from 0 to 63 have *WNT5A* down regulated (which means 0) and hence are desirable states, as compared to states 64 to 127 have *WNT5A* up regulated(which means 1) and hence undesirable. The steady state distribution of Human melanoma network of the original and controlled networks is shown in Figure 5. We can observe that the probability distributed state vector shifted from unwanted states (65-127) to the wanted states (0-64) that was our goal. The mean square error for both action and critic network is shown in Figure 5 (c), which is shown obviously how the action and critic weights are convergence.

Figure 5: Simulation results : (a) 2D-steady-state distribution results; (b) Steady-state distribution of gene-activity profile after intervention with optimal control policy using NDP method; (c) NDP mean square error.

DISCUSSION

We have formulated the NDP strategy to find an approximate stochastic control policy for a context sensitive PBN. NDP not only lowers computational complexity in comparison to the optimal stochastic control, but performs virtually the same as the optimal stochastic control when the learning duration is long enough. As shown in the melanoma case, applying suboptimal policy has the same effect in reducing the likelihood of visiting undesirable states, the ones with high chance of metastasis in the long run. The time complexity of the approximate control method is polynomial, whereas the time complexity of the optimal control algorithm is exponential in the number of genes.

ACKNOWLEDGMENTS

The author would like to thank Dr. Ranadip Pal from Texas Tech. University for providing the Human melanoma gene regulatory network dataset. Also, the author would like to thank Dr. Nidal Bouaynaya and Dr. Kameran Iqbal from university of Arkansas at little Rock for helping.

[1] Datta, A., Pal, R., and Dougherty, E.R.: 'Intervention in probabilistic gene regulatory networks', Current Bioinformatics, 2006, 1, (2), pp. 167-184.

[2] Shmulevich, I., Dougherty, E.R., and Zhang, W.: 'Gene perturbation and intervention in probabilistic Boolean networks', Bioinformatics, 2002, 18, (10), pp. 1319-1331.

[3] Shmulevich, I., Dougherty, E.R., and Zhang, W.: 'Control of stationary behavior in probabilistic Boolean networks by means of structural intervention', Biol. Syst., 2002, 10, (4), pp. 431-446.

[4] Shmulevich, I., Dougherty, E.R., Kim, S., and Zhang, W.: 'Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks', Bioinformatics, 2002, 18, (2), pp. 261-274.

[5] Kauffman, S.A.: 'Metabolic stability and epigenesis in randomly constructed genetic nets', J. Theor. Biol., 1969, 22, (3), pp. 437-467.

[6] Kauffman, S.A.: 'The origins of order: self-organization and selection in evolution' (Oxford University Press, 1993, 1st edn.).

[7] Kauffman, S.A., and Levin, S.: 'Towards a general theory of adaptive walks on rugged landscapes', J. Theor. Biol., 1987, 128, (1), pp. 11-45.

[8] Datta, A., Choudhary, A., Bittner, M., and Dougherty, E.R.: 'External control in Markovian genetic regulatory networks', Mach. Learning, 2003, 52, (1/2), pp. 169-191.

[9] Pal, R., Datta, A., and Dougherty, E.R.: 'Optimal infinite-horizon control for probabilistic Boolean networks', IEEE Trans. Signal Process., 2006, 54, (6), pp. 2375-2387.

[10] Bertsekas DP,' Dynamic Programming and Optimal Control' Athena Scientific 1995; 1 and 2.

[11] Bertsekas, D.P.: 'Dynamic programming and optimal control' (Athena Scientific, 1995, 2005, 3rd edn.)

[12] Bertsekas, D.P., and Tsitsiklis, J.N.: 'Neuro-dynamic programming' (Athena Scientific, 1996, 1st edn.).

[13] Faryabi B., Datta A. ,and E. R. Dougherty:' On approximate stochastic control in genetic regulatory networks', IET Syst.,2007 ,1,(6),pp. 361-368.

[14] X. Qian, Datta, A., Pal, R., and Dougherty, E.R.: 'Intervention in probabilistic gene regulatory networks via greedy control policies based on long run behavior', BMC System Biology, 2009, pp. 147-154.

[15] Nidal B., and Roman S.: 'Optimal Perturbation Control of General Topology Molecular Networks', IEEE Transaction on signal Processing, 2011, pp.1-15.

[16] Nidal B., and Roman S.: 'Inverse perturbation for optimal intervention in gene regulatory networks', Bioinformatics, 2011, pp.103-1011.

[17] Nidal B., M. Rasheed, and Roman S.: 'Intervention in general topology gene regulatory networks', IEEE Proc. Genomic Signal Processing, 2011.

[18] Nidal B., and Roman S.: 'Method for optimal Intervention in Gene Regularity Networks', IEEE Signal Processing Magazine, 2012, pp.1053-1088.

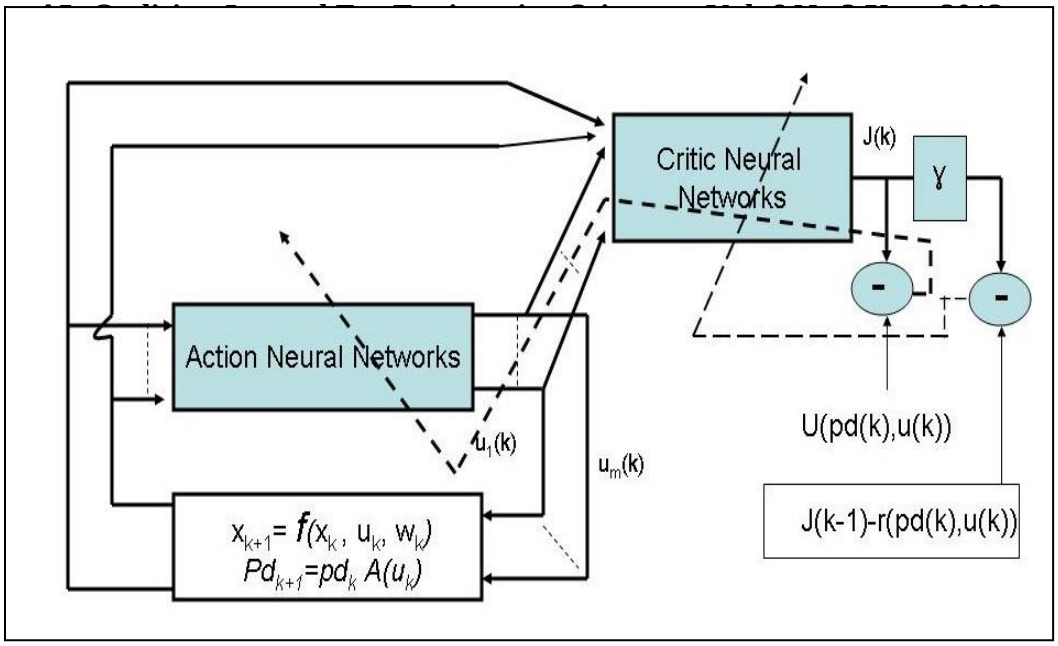


Figure 1: Schematic diagram for implementation of NDP, lines without sold represent the forward path, sold dash lines represented the back update weight path.

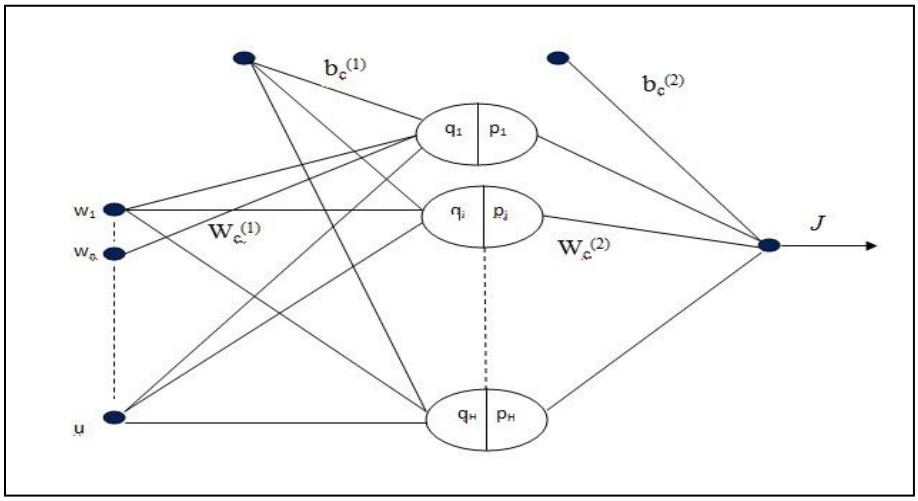


Figure 2: Nonlinear (sigmoid activation function) critic neural networks structure: *n+1* input nodes in input layer, *H* nodes in a one hidden layer, one node in the output layer.

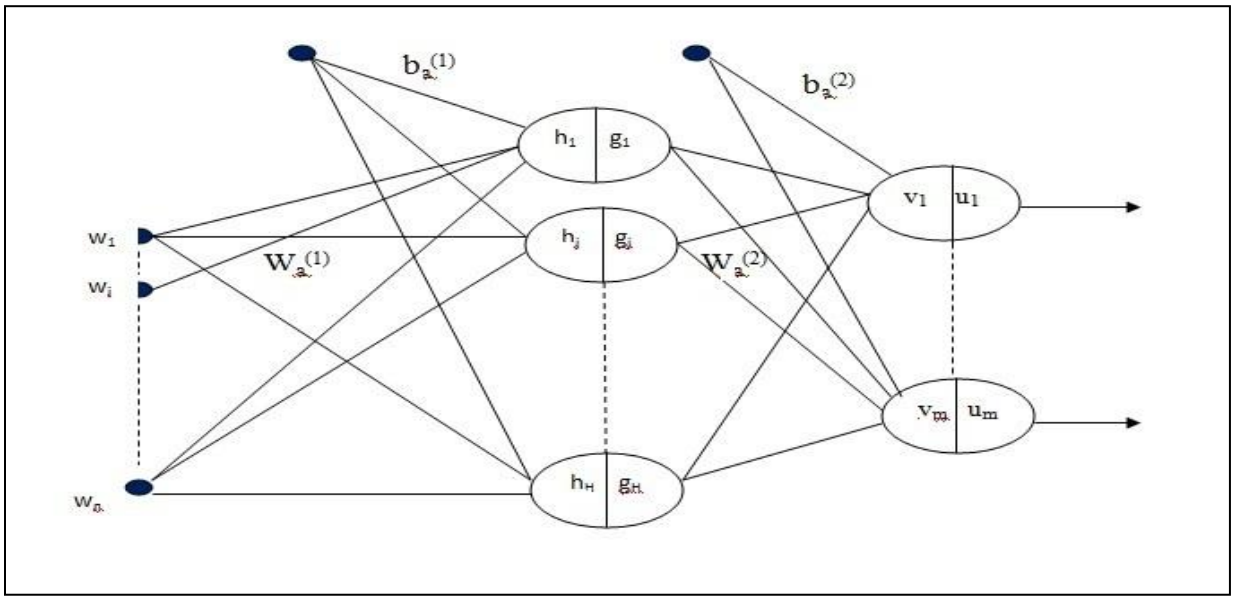


Figure 3: Nonlinear action neural networks structure: n input nodes in input layer, *H* nodes in a one hidden layer, m nodes in the output layer.

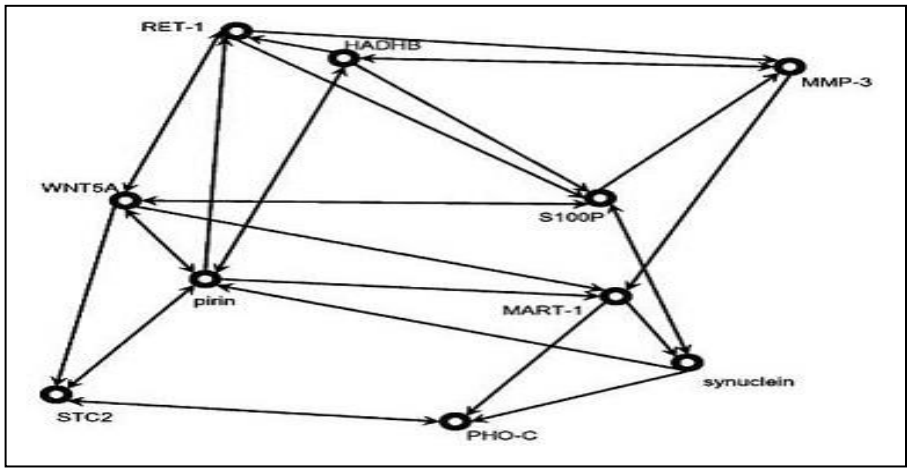
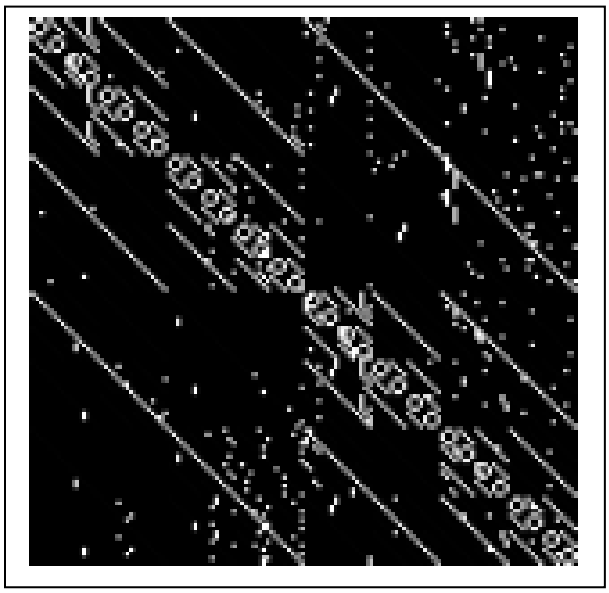
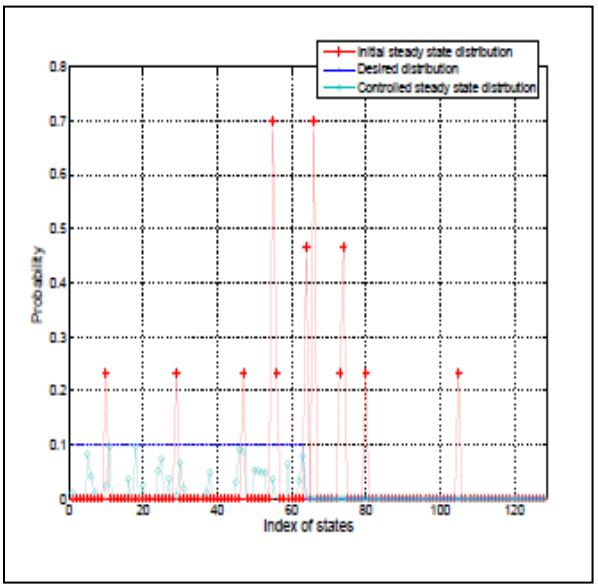


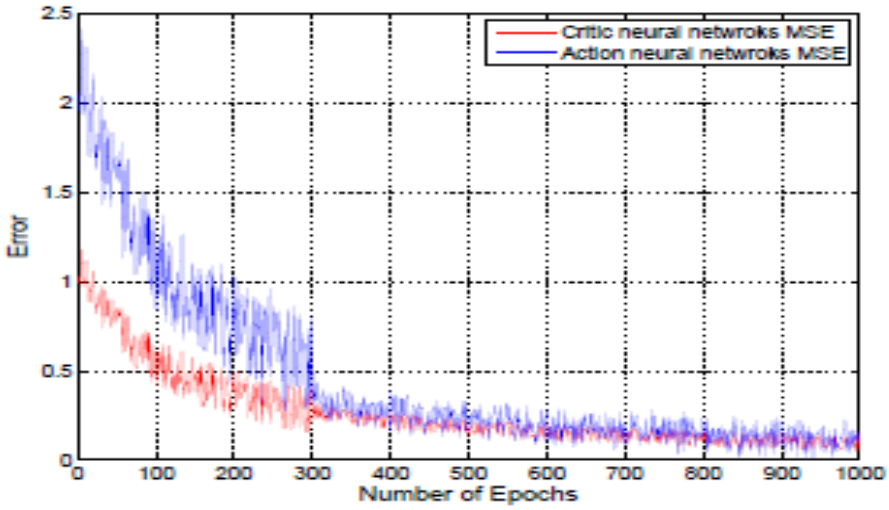
Figure 4: The probabilistic Boolean networks of the seven genes, [8].



(b)


Initial steady state distribution
Desired distribution
Controlled steady state distribution
Probability
Index of states
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
20
40
60
80
100
120

(a)


Critic neural netwroks MSE
Action neural netwroks MSE
Error
Number of Epochs
0
0.5
1
1.5
2
2.5
0
100
200
300
400
500
600
700
800
900
1000

(c)