Evolving Spiking Neural Networks for Control of Artificial Creatures

Arash Ahmadi
Electronic System Design Group, Electronics & Computer Science, University of Southampton
Tel: (+44) (0)23 8059 3545, Fax: (+44) (0)23 8059 2901, Southampton SO17 1BJ,
United Kingdom, http://users.ecs.soton.ac.uk/aa5/
Email: aa5@ecs.soton.ac.uk

Abstract:

To understand and analysis behavior of complicated and intelligent organisms, scientists apply bio-inspired concepts including evolution and learning to mathematical models and analyses. Researchers utilize these perceptions in different applications, searching for improved methods and approaches for modern computational systems. This paper presents a genetic algorithm based evolution framework in which Spiking Neural Network (SNN) of artificial creatures are evolved for higher chance of survival in a virtual environment. The artificial creatures are composed of randomly connected Izhikevich spiking reservoir neural networks using population activity rate coding. Inspired by biological neurons, the neuronal connections are considered with different axonal conduction delays. Simulations results prove that the evolutionary algorithm has the capability to find or synthesis artificial creatures which can survive in the environment successfully.

Keywords: Spiking Neural Networks (SNN), Izhikevich Model, Genetic Algorithm (GA), artificial creature.

1. Introduction

Artificial neural networks are inspired by natural neural systems such as brain. Human brain, for instance, consists of about 10^{11} neurons with over 6×10^{13} interconnections. Brain has an unapproachable potential in a wide variety of parallel information processing such as pattern recognition, speech processing, sensory system, locomotion, learning, etc., which are impossible by conventional computational devices. Therefore, scientists have been trying to understand different aspects about brain functionality. Following this path, different features of neurons, as the building blocks of the neural systems, have been investigated in terms of mathematical/circuit models, dynamic behavior and time-spatial and topological network properties and so on. As a result, a new generation of neural network is introduced: Spiking Neural Network (SNN). SNNs are the third generation of artificial neural network in which spikes train are the basis of the information exchange between individual neurons. In comparison, the first generation of artificial neural networks consists of McCulloch-Pitts neurons which can only present digital outputs, whereas in the second generation neurons communicate with continuous activation function [1-4].

In recent decades, the spiking neural network is progressively receiving attention because this approach to the neural networks expands the level of realism in simulation and dynamic analysis is considered as a powerful tool for examination of brain process such as information coding, plasticity and learning. In this regard, several mathematical models have been presented to describe the behavior of the spiking neurons [5]. Some models attempt to simulate the neuron very precisely. This group of models is usually computationally expensive and is not suitable for large scale neural network realization. There are models, on the other hand, which do not characterize the neuron very precisely but can be used to create large scale neural networks.

One important method in order to comprehend how the brain turns sensory input into actions is behavioral study of the artificial creatures. From this point of view, SNNs have the potential to be used in neural network controller of the artificial creatures. Although, compared to mammalians brain, far less number of neurons can be realized in the nervous system of artificial creatures, study of their behaviors is vitally beneficial for understanding complex nervous systems.

Different learning methods have been developed for both types of artificial and biological neural networks [6]. These learning mechanisms are usually based on adjustment of the synaptic strengths (or weights). Although this classic learning approach can be adequate in the first and second generation of neural networks, nature of SNN presents higher degree of complexity which needs more complicated learning approaches such as Spike Timing-Dependent Plasticity (STDP) [7]. There are a wide range of characteristics in a SNN which are essential in behavioral changes which must be taken into account in a learning algorithm such as network topology, neurons' type, synaptic weight and synaptic delays [8].

There are a many variations of Evolutionary Algorithms (EA) [9] to search within a huge number of possibilities for solutions. Evolutionary search can be applied to find the most successful network for specific applications and improve the learning capabilities of neural networks. Three basic types of these techniques are: Genetic Algorithms (GA), Evolutionary Programming (EP) and Evolutionary Strategies (ES). Among these evolutionary algorithms, GA [10] is the most popular and widely used computational model of evolutionary processes for solving practical problems and has been used as a systematic model of evolution in artificial-life systems. This algorithm emulates some of the processes observed in nature such as selection, mutation and population dynamics.

One of the active fields of research, in which evolutionary algorithms are utilized, is evolutionary robotic [11]. Evolutionary robotics was developed in 1990s as a field of research to build robot controllers without detailed hand design. The basic assumption of evolutionary robotics is to evolve a population of robots as autonomous artificial organism by evolutionary algorithms to improve them by generations. One main research application in evolutionary robotics is the evolution of neuro-controllers which use the neural networks for control of robots. In 1992 Dave Cliff, Philip Husbands and Inman Harvey [12] indicated that neural network control architecture can be evolved to make visually guided robots. Also, spiking neuro-controllers have been evolved by some researchers successfully, such as evolution of spiking neural controllers for autonomous vision-based robots by Dario Floreano and Claudio Mattiussi [13]. They used Spike Response Model neuron and showed that the evolution can find relatively quickly functional spiking controllers capable of navigating in irregularly textured environments without hitting obstacles. Evolving SNNs by adaptive Genetic Algorithm (GA) have also been used to control a Khepera robot [14]. The robot which is controlled by neural networks can perform interesting behaviors. The applications of neural robotics are increasing and have attracted significant attentions in research of artificial intelligence.

The main objective of this study is to discover optimal structures for an artificial creature with survival capability in virtual environment. Each creature starts with a limited level of energy and has the opportunity to find food objects to gain required energy for their survival. GA has been utilized to find the better creature structures, because the nature of biological neural networks are far more complex compared with traditional artificial neural networks; therefore, it is impossible to formulate and find the optimal structure using precise analytical approaches. In this paper a set of simulation-based artificial creatures using spiking neural network and Genetic algorithm are presented.

The paper is organized as follows: at first, background is described in Section 2. In Section 3, neural network structure of artificial creatures is explained; this section consists of two parts: part A, neuronal network model and part B, synaptic delay model. Section 4 presents the artificial creatures and environment. In Section 5 the evolutionary model which has been used is described. In Section 6 simulations are shown and results and conclusions are subsequently explained in Section 7.

2. Background

Despite recent advancements, scientists are still attempting to figure out wide range of unsolved problems about the most powerful organism of human body: the brain. One of the challenges in recent years has been recognizing the behavior of this complicated organism, in order to achieve more efficiency in neuromorphic computational and control systems. In this path, scientists discover electrochemical process of biological neurons, connectivity topologies, learning rules and etc. These concepts are used by computational neuroscientists to construct artificial models, such as artificial life and robots. A set of studies about artificial life and robotic have been developed recently. For instance; in [15] a C.elegans artificial creature has been presented. In this case since the C.elegans has a preliminary known specific neural structure there is no need for further investigation of the neural structure, whereas the approach which is presented in our work is to discover the most compatible structure among a large space of possible solutions. This is a step towards achieving a better understanding of real behavior in nature. For this reason, a reservoir neural network is used. In continue reservoir neural network and neuron model are discussed.

RESERVOIR NEURAL NETWORK

SNN is capable of producing a variety dynamic and behavior. They can be organized for many applications such as multilayer feed forward or recurrent networks, like traditional neural networks; however, SNN's operation is different from traditional networks. According to biological observations, a biological network topology can be composed of randomly connected neurons in which only active neurons process the information. Therefore, it does not seem necessary that a biological neural network has a regular structure as in traditional ANNs. Normally, in a natural system, neurons are distributed in a randomly connected form within a large scale consistent structure. With this explanation, a new form of artificial neural network was introduced: Reservoir Computing (RC). The main goal of the reservoir network is to enrich inner dynamics of the network in response to the input signals, as to attain the required flexibility, dynamic and adaptivity [16]. Initially, concept of reservoir computing structure, was extended by Maass [16], Jaeger [17].This type of network is suitable especially for temporal input and output pattern processing, where high efficiency can typically be achieved without too much concern about specific settings of the reservoir parameters. Principal specifications of reservoir computing model are [18-19]:

- Layer of K input neurons are connected to the reservoir.
- A main part consist of M randomly connected neurons.
- A layer of readout neurons with trained connections from reservoir network.

The main motivation of reservoir computing is a large solution space creation to discover the best and most compatible networks for special purposes. To enhance the strength of this network one choice is to applying Spike Time Dependent Plasticity (STDP) to neuron connections rules inside the reservoir design.

NEURON MODEL

As mentioned before, a variety of models have been developed for biological neurons with different degrees of precision and complexity. The choice of the model strongly depends on the application. This selection is a trade of between precision and computational cost, which is closely related to the complexity of these models. From a precision viewpoint, the Hodgkin Huxley (HH) [20] model is a ground breaking model with the highest accuracy in terms of biochemical modeling of the neurons but it has a high computational cost. Clearly, large scale simulation of neural network with this model is impractically expensive. On the other hand, Integrate and Fire (IF) is recognized as one of the simplest models with highest computational efficiency [21]. This model is a low cost but in accurate model which is incapable of producing complex dynamic of neurons such as bursting. IZ model, introduced by Izhikevich at al. 2003 [22], presents a compromise between accuracy and computability which has made it a widely accepted popular model. It is claimed to be as realistic as Hodgkin-Huxley model. This model is a 2 dimensional model that is described by two differential equations:

$$\begin{cases} \frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I \\ \frac{du}{dt} = a(bv - u) \end{cases} \qquad (1)$$

with the auxiliary after-spike resting equations:

$$v \geq V_{th} \text{ Then } \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \qquad (2)$$

where v is the membrane potential and u is the recovery variable. When a small pulse of current is applied, the membrane potential raise, if the current is adequate strong, membrane voltage crosses its threshold value (30mV); after which the membrane potential and the recovery variable reset according to the auxiliary reset equations. Here *a, b, c,* and *d* are dimensionless variables and can be described by:

a: Time scale of the recovery variable *u*. The lower a value the lower the recovery

b: Sensitivity of the recovery variable *u* to the sub threshold fluctuations of the membrane potential *v*

c: After spike reset value of the membrane potential *v* caused for the fast high-threshold K^+ conductance.

d: After-spike reset value of the membrane potential *v* caused for the slow high-threshold Na^+ and K^+ conductance.

The range of parameters in Izhikevich model can be seen in the Table [23].Since this model is computationally feasible and biophysically accurate it is used in this study. Its low computational cost makes it more suitable for large scale network simulations. Please note that the neural networks used in this paper are composed of Izhikevich neurons with a constant range of *a*, *b*, *c* and *d* parameters.

Table 1. Boundaris of parameters in Izhikevich model

	a	b	c	d
Min	0.002	0.1	-65	0.05
Max	0.1	0.3	-55	8

3. Neural network structure of artificial creatures

This section presents details of the spiking neural networks used in the artificial creatures.

Neuronal network model

A reservoir network has been used in this paper. This structure has been shown in Figure 1. As is observable this network has two input and output layers. The neuronal network that have been used is composed of *N = 150* randomly connected Izhikevich spiking neurons and different axonal conduction delays between each two neurons. Information is transferred between neurons of the networks through the links between every two neurons representing synapses. Each neuron is connected to *M = 15* random neurons, so that the probability of connection is *M / N = 0.15*. It is noticeable that in this network, not only the connection between two neurons is random, but also the neurons type selection is random, too.

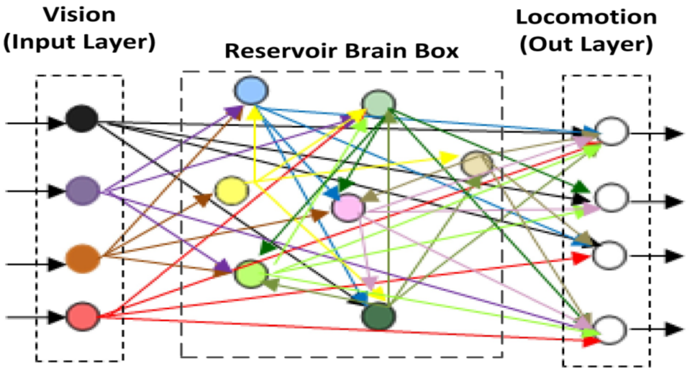


Figure 1. Typical neural networks

Synaptic delay model

Neurons transfer data as a spike trains through synapses. Because of the conduction time that is required for spike transformation between two neurons it is expected a synaptic delay. Delay has a key role in processes. In reality, in the mammalian nervous system, the axonal conduction delays are different depending on the type and spatial positioning of the neurons. In this study the value of each synaptic connection is assumed to have a fixed conduction delay between 1 ms and *D* = 20 ms [24]. For instance, in Figure 2, synaptic connection delays between neuron *A0* and *A1* is equal to 2 ms, so spike from *A0* arrives to *A1* after 2 ms.

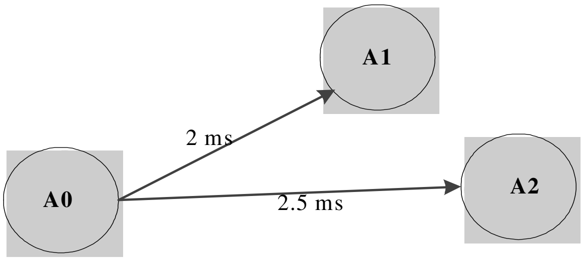
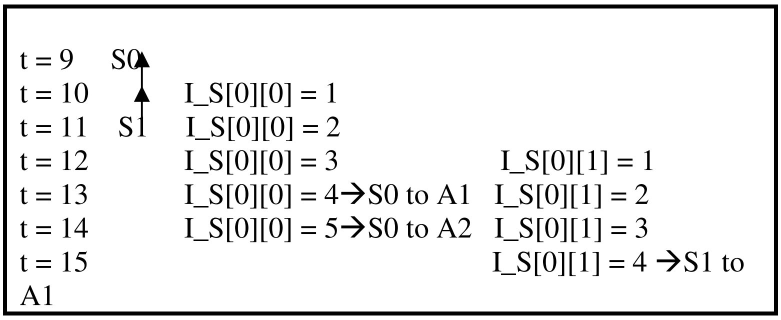


Figure 2. Example of different axonal conduction delay between presynaptic neuron A0 and postsynaptic neuron A1 and A2.

Different axonal conduction delays between every two neurons are applied as follows:

For instance, consider presynaptic neuron *A0* and postsynaptic neurons *A1* and *A2* in Figure 2. If neuron *A0* fires spike *S0* in *t = 9* and *S1* in *t = 11* and with a time step equal to 0.5 ms, *S0* arrives to *A1* in *t = 13* and to *A2* in *t = 14* where *S1* arrives to *A1* in *t = 15* and to *A2* in *t = 16*. This is implemented using counters assigned to each spike which increases with each time step in the time window. If counters arrive to axonal delay values, the effect of spikes from presynaptic neuron applies to the post synaptic neurons.

For neural network of each creature two arrays, *S* and *I_S* are defined. *S* is a one dimensional array for the number of spikes in each neuron and *I_S* is a two dimensional array, that one dimension is for neuron's numbers and another dimension is for spike's number and *I_S* shows the time counter of the spikes. Considering Figure 3, in *t = 12*, *S[0]* is equal to 2 which means *A0* has fired two spikes and *I_S[0][0] = 3* means three time steps (1.5 ms) passed from occurring *S0* in neuron *A0* and *I_S[0][1] = 1* means one time step (0.5 ms) passed from occurring *S1* in neuron *A0.* In each time step, axonal conduction delays between presynaptic neurons and postsynaptic neurons are examined whether they are equal to the elements of array *I_S*, to apply the respective spikes. The pseudo code for applying different axonal conduction delay between presynaptic neurons and postsynaptic neurons are shown in Figure 4.



t = 9	S0		
t = 10		I_S[0][0] = 1	
t = 11	S1	I_S[0][0] = 2	
t = 12		I_S[0][0] = 3	I_S[0][1] = 1
t = 13		I_S[0][0] = 4→S0 to A1	I_S[0][1] = 2
t = 14		I_S[0][0] = 5→S0 to A2	I_S[0][1] = 3
t = 15			I_S[0][1] = 4 →S1 to A1

```
t = 16                                    I_S[0][1] = 5 →S1 to
A2
t = 17               .
```

Figure 3. Example for applying different axonal conduction delay between neurons.

```
   for t = 0 : 400
      for e = 0 : N                                          // for all neuron
        v = v + 0.5 * ((0.04 * v + 5) * v + 140 – u + I);          // for numerical
        v = v + 0.5 * ((0.04 * v + 5) * v + 140 – u + I);          // stability time
        u = u + a * (0.2 * v - u);                                 // step is 0.5 ms
            if(v >= 30) S[e]++;             // increase the number of spike for neuron e
            if (S[e] > 0)
            {
           for f = 0 : S[e] do
              I_S[e][f]++;
              for c = 0 : m                  //for all persynaptic connection
                  if(I_S[e][f] == delay_synaptic[e][r])
                      // delay_synaptic[][] is delay between neuron e and presynaptic
                   // connections
                     {
                 // apply effect of spike to respective neuron
                     // ...
                     }
                 end;
           end;
            }
      end;
 end;
```

Figure 4: Pseudo code for applying different axonal conduction delay between presynaptic neurons and postsynaptic neurons.

4. Artificial creatures and environment

The artificial creatures are designed to find food objects in the environment and each of them starts with a limited level of energy. The energy level is decreased by every movement down to a certain level. If an artificial creature finds the food source, the energy level increases up to by a specific level. Accordingly, each creature can survive in the external environment until the energy level is not zero.

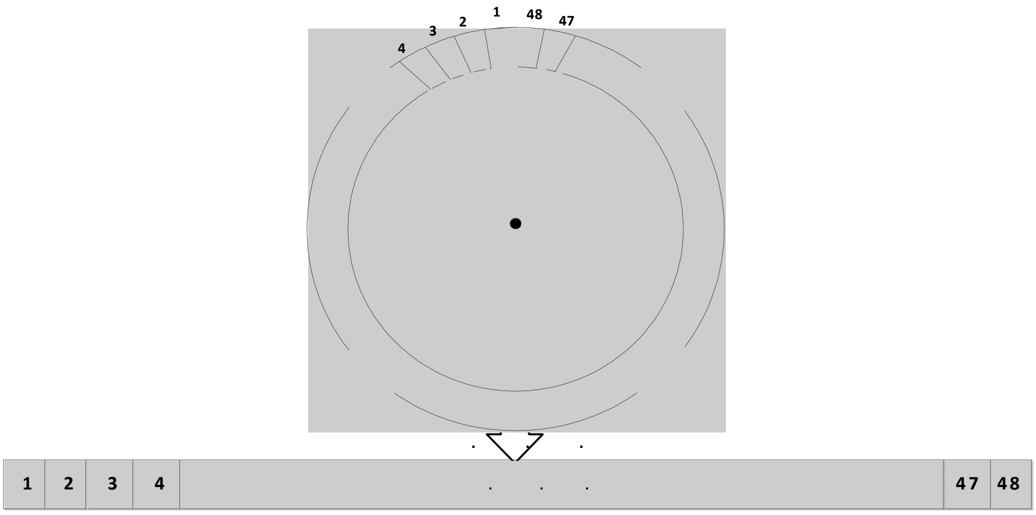


Figure 5. The periphery of circular area around of artificial creature in each position

It is assumed in the simulation that the artificial creature in each location stands in the center of a circular area with a radius of its vision range, i.e. 2 meters. If the creature turns around itself; it only can see objects in its visual range. When a food object appears in the periphery of this circular area, a creature should find this food. Moving around, another food appears in a random place on periphery of the circular area, and this procedure repeat for 5 iterations. As shown in Figure 5 the periphery of the circular area around the artificial creature in each position is divided to 48 slots, which can be considered as a rectangular. In the simulation program it is implemented as an array of 48 × 3.

The utilized neural networks of the artificial creature consist of three essential parts:

INPUT LAYER AS VISION:

Visual angle of the artificial creature has been assumed 150 degrees. This is similar to the visual angle of the human eye which is approximately 120 - 180 degrees. Since periphery of the circular area around the artificial creature is divided into 48 slots, from creature's viewpoint each slot is equaled to 7.5 degrees as:

$$\frac{360^{\circ}}{48} = 7.5^{\circ} \qquad (3)$$

Therefore, considering that the visual angle of the artificial creature has been assumed 150 degrees, an image in sight of the artificial creature is composed of 20 sub-divisions because:

$$\frac{150^{\circ}}{7.5^{\circ}} = 20 \qquad (4)$$

Figure 6 shows an image in sight of the creature, where the food in this image is the dark part of the vision. If the food object places in the vision edge of the artificial creature (2 meters), two squares in the image in sight of the creature becomes black and by getting the artificial creature closer to the food, more squares of image in sight become block and if all parts of the image in sight of the creature become black, the artificial creature has been successful in finding the food object. In Figure 6 each part of the image divided to 3 subparts because 3 neurons per part of image in the input layer of the artificial creature network (vision) have been considered. So vision of the artificial creature composed of 60 neurons due to:

$$\text{Number of parts} \times 3 = 20 \times 3 = 60 \qquad (5)$$

For each black part of image in sight of the artificial creature, three signals as the input signals is applied to the three respective neurons.

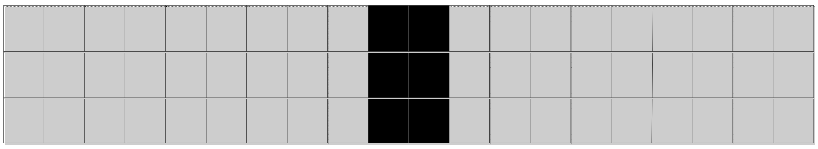


Figure 6, One image in sight of artificial creature.

OUTPUT LAYER AS LOCOMOTION PART:

Three movements for each creature have been defined: 15 degree right rotation, 15 degree left rotation and forward movement. In forward motion, the artificial creature 0.5 meter move to forward direction, so 4 forward motions is necessary for finding the food object in utmost of the creature location (4 meter). Depend on the food position in the vision, respective locomotion part of the artificial creature must be activated, for example if position of the food in sight of the artificial creature be right, the right part of locomotion must be activated. Figure 7a, Figure 7b, Figure 7c and Figure 7d, shows effect of different type of movement on the image in the vision of artificial creature if food be on vision boundaries. As mentioned each part of the image equal 7.5 degree. Therefore 15 degree left or right rotation locomotion equivalent two parts shift toward left or right. For motion to forward direction, size of the image has been reduplicated so that each part has been become to the two similar parts. Then half of new image in right side and left side has been deleted in order to create new close image in vision. Accordingly, if food be on vision boundaries, the number of black parts of the image for the food object in ultimate location is 2, by one movement to forward direction the number of these parts become to 4, by one movement to forward direction the number of these parts become to 8 and so on. After four movements to forward all part of the image is black and the creature is succeed find the food object.

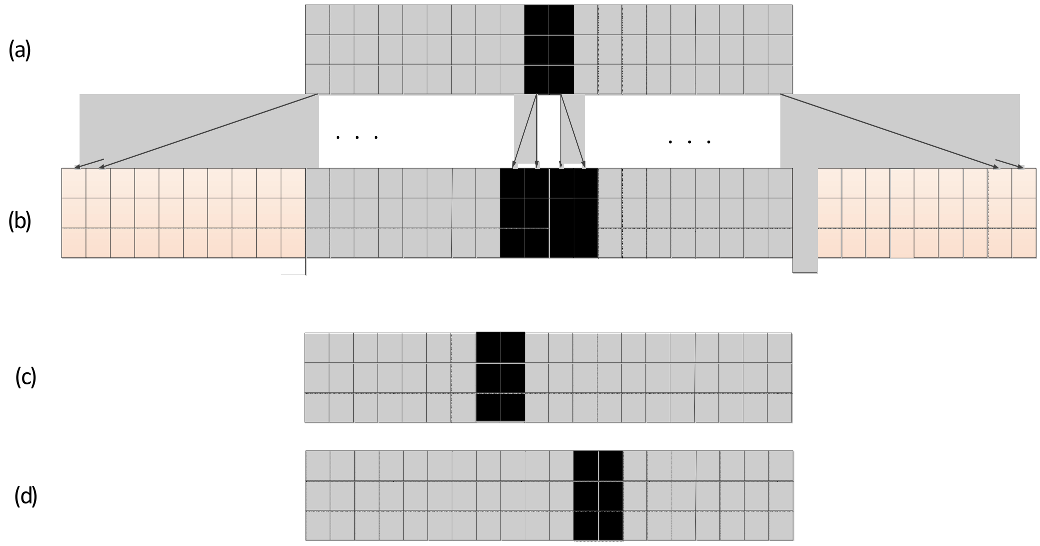


Figure 7. Effect of different type of movement on the image in the vision of artificial creature. a) Initial state, b) Forward, c) Right d) Left locomotion

Considering real nervous systems, in which many neurons react to the similar stimulus, in this paper three neurons are responsible for each locomotion part of the creature in response to the stimulus. Therefore, the output layer of the artificial creature neural network consisted of 9 neurons. To distinguish the direction of the movement, the artificial life form uses the population activity rate coding. This coding has been successfully applied in experiments on sensory or motor systems. In population activity, average number of firing in a population of the neurons is calculated in a fixed time window as:

$$A(t) = \frac{1}{\Delta t}\frac{n_{act}(t;t+\Delta t)}{N} \qquad (6)$$

where N is the population size, $n_{act}(t;t+\Delta t)$ is the number of spikes (summed over all neurons in the population) which are fired during the time window. Window time has a fixed length between t and $t+\Delta t$ and Δt is an appropriate time interval [5]. So in each time window the total number of spikes in each three neurons is compared with other three neurons and artificial creature moves toward direction that the respective neurons fired maximum number of spikes. These fixed time windows consist of 600 time-steps. Each time step is 0.5 ms. Flowchart in Figure 8 shows details.

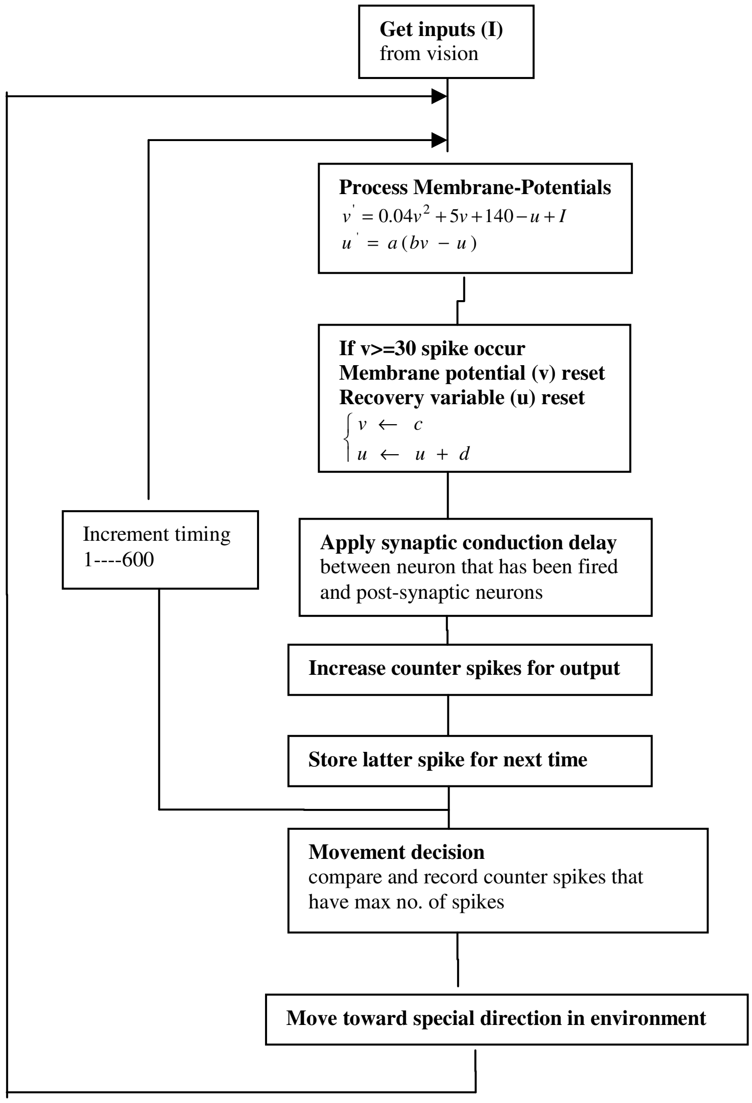

Get inputs (I)
from vision
Process Membrane-Potentials
$v' = 0.04v^2 + 5v + 140 - u + I$
$u' = a(bv - u)$
If v>=30 spike occur
Membrane potential (v) reset
Recovery variable (u) reset
$\begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases}$
Increment timing
1----600
Apply synaptic conduction delay
between neuron that has been fired
and post-synaptic neurons
Increase counter spikes for output
Store latter spike for next time
Movement decision
compare and record counter spikes that
have max no. of spikes
Move toward special direction in environment

Figure 8. Flowchart for movements of creature

MAIN BODY STRUCTURE:

As mentioned the reservoir neural network has been chosen for neural network of the artificial creature as main body structure. Figure 9 illustrates a typical artificial life form and circular area around of it. Maximum seeing of the artificial life form is periphery of the circular area and can't see places that have beyond of periphery of the circular area. The artificial creature by each movement is the center of a circular area so if the creature move to forward direct can see new places and some places are voided of seeing.

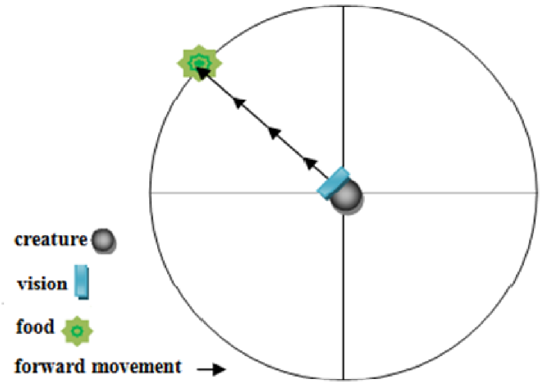


Figure 9. A typical artificial life form and circular area around of it

5. Evolutionary model

Although there is no methodical investigation has been carried out to find which evolutionary techniques work better than others, Genetic Algorithm (GA) has been selected in this study, because GA has been inspired from biological evolution and it is based on adaptation processes observed in nature, such as natural selection and genetic diversity. GA can find the best solution among a possible response space, efficiently [25-27].

First, an initial random population of creatures is generated where the neural networks of the creatures are coded as chromosomes, as shown in Figure 10a and Figure 10b. Each chromosome consists of four parts: *A1, A2, A3* and *A4*. Each part consists of *N* segments for *N* neurons of a typical neural network structure. The first part, *A1*, denotes *a, b, c* and *d* parameters of neurons Izhikevich model (discussed in (1) and (2)). Each segment of *A2* shows postsynaptic weights and connections for corresponding neuron and each segment of *A3* indicates postsynaptic delays of the connections. Segment *A4* shows postsynaptic neurons that are connected to corresponding neuron, as shown in Figure 10b. Each artificial creature (chromosome) is placed in an environment for searching and catching the randomly distributed foods in the peripheries. Then fitness value is calculated for each of them. Fitness function is the energy level. The ideal creatures are those who find food objects in 5 iteration that food sources appeared in random locations by minimum movements so that have maximum energy and they can survive. They should show some adaption abilities under different conditions.

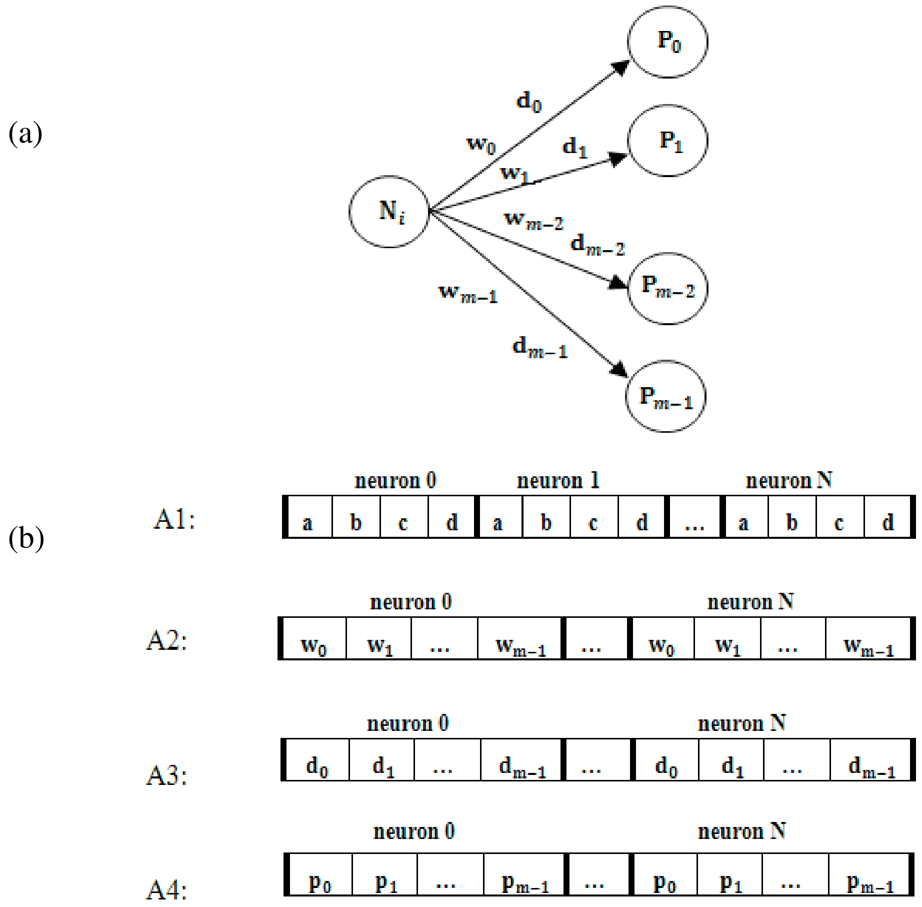

(a)
P_0
d_0
w_0
d_1
P_1
w_1
N_i
w_{m-2}
d_{m-2}
P_{m-2}
w_{m-1}
d_{m-1}
P_{m-1}
(b)
A1:
neuron 0 | a | b | c | d
neuron 1 | a | b | c | d
...
neuron N | a | b | c | d
A2:
neuron 0 | w_0 | w_1 | ... | w_{m-1}
...
neuron N | w_0 | w_1 | ... | w_{m-1}
A3:
neuron 0 | d_0 | d_1 | ... | d_{m-1}
...
neuron N | d_0 | d_1 | ... | d_{m-1}
A4:
neuron 0 | p_0 | p_1 | ... | p_{m-1}
...
neuron N | p_0 | p_1 | ... | p_{m-1}

Figure 10: a) A typical neuron and postsynaptic connections with weights and delays b) A typical chromosome A1: Parameters of the Izhikevich model, A2: Synaptic weights, A3: Conduction delays, A4: Post-synaptic neurons number

Then next generation are produced by combination of the elites (15%), crossover (55%) and mutation (30%) of the initial population. Elites are the best chromosomes which are directly transferred to the next generation. Because of long chromosome length, for crossover, five points are randomly chosen in each parent as cut points. Figure 11 shows a typical crossover with two cut points and Figure 12 illustrates a flowchart for the proposed evolutionary model. Selections are based on Roulette Wheel selection, more detailed information can be found in [28].

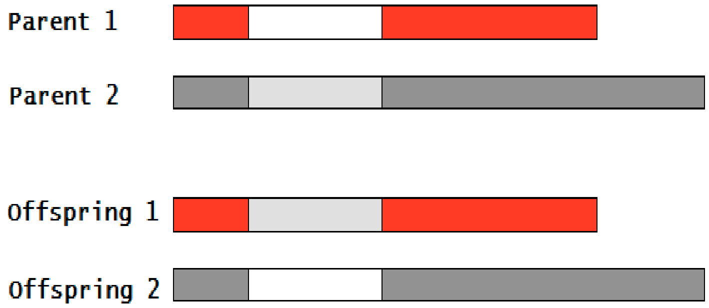


Figure 11. Crossover with two cut points [28]

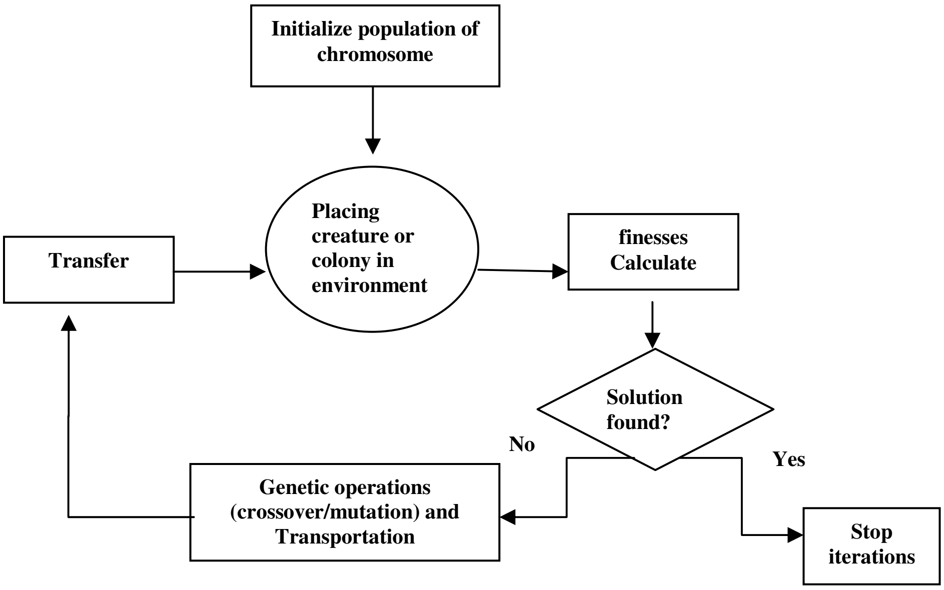

Initialize population of chromosome
Placing creature or colony in environment
Transfer
finesses Calculate
Solution found?
No
Yes
Genetic operations (crossover/mutation) and Transportation
Stop iterations

Figure 12. Genetic Algorithm flowchart

6. Simulation results

Artificial creatures and the virtual environment are designed and implemented in a C++ platform in which simulations are performed. In the GA algorithm, at First step a population of 100 complex artificial creatures that each of them had 150 neurons were tested and evaluated by GA; each neuron connected to 15 post-synaptic neurons with different axonal conduction delays between every two neurons. In every generation fitness function has been calculated for all population. The initial energy level for each creature is considered as 50.

Figure 13 compares the average fitness at generations in one typical program running. As can be seen from this figure, the average of fitness grows as the number of generations increases. Also as can be understood from Figure 14 the number of survived chromosome increases with generation progressing. For a better visual understanding of the creatures' movement, Figure 15 (a, b, c, d, e, f, g, h) illustrates movements of a typical successful creature for finding one food object.

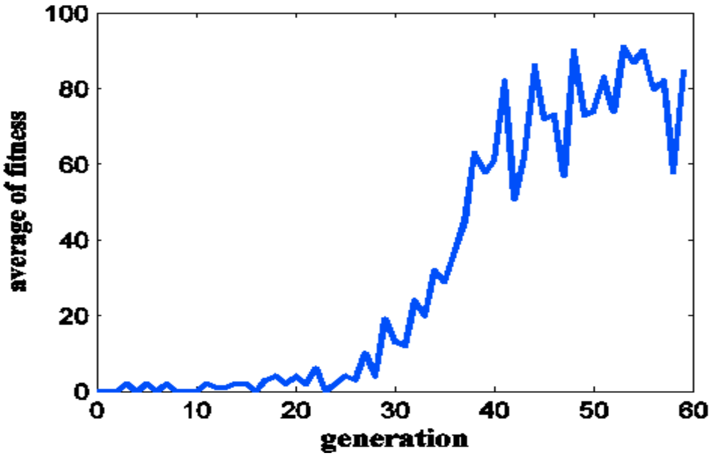


Figure 13. Average fitness at generation progressing

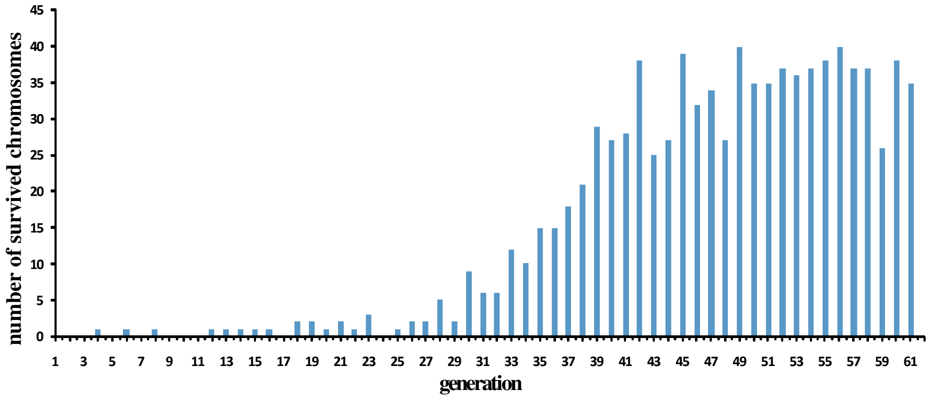


Figure 14. Number of artificial creatures that could survive in the external environment

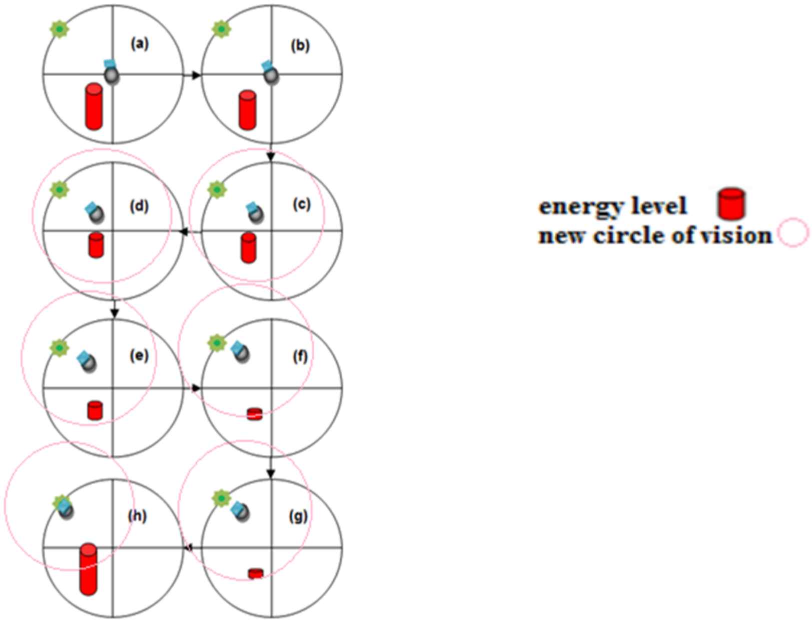


Figure 15. Movements a typical successful creature to find food.(a) Initial state, (b) Left, (c) Forward, (d) Left, (e) Forward, (f) Forward, (g) Left , (h)Forward and catching food

We believe that this study can be a step forward in understanding the morphology of artificial creatures. Also this paper suggests more complex artificial life examination by adding different part to these networks similar to different segments of the brain such as: vision, locomotion, hippocampus and communication in a future work. Because of the different axonal conduction delay between every two neurons in the neural network of artificial creatures in this paper, our next study is to enhance the artificial lives by STDP learning.

7. Conclusions

This work presented a Genetic Algorithm for evolving artificial life forms colonies. These artificial creatures composed of biological neurons which have been connected randomly. GA could be able discover the most compatible structures in the solution space. The results of simulations show that the Genetic Algorithm is efficient for this purpose. Winner Artificial creatures of GA algorithm have been located in the virtual environment in which they search to find the food objects in the environment.

References

[1] Dreyfus G. Neural Networks: Methodology and Applications. Springer, 2005.

[2] Mehra P, Wah BW. Artificial Neural Networks: Concepts and Theory. IEEE Comput Society Press, 1992.

[3] Paugam-Moisy H, Bohte S. Computing with spiking neuron networks. In: Rozenberg G, Bäck T, Kok J, editors. Handbook of Natural Computing. Springer Verlag, 2010.

[4] Vreeken J. Spiking neural networks: an introduction. Artificial Intelligence Laboratory, Intelligent Systems Group, Institute for Information and Computing Sciences, Utrecht University, 2003.

[5] Gerstner W, Kistler W. Spiking Neuron Models. Cambridge University Press, Cambridge, 2002.

[6] Judd JS. Neural Network Design and the Complexity of Learning. MIT Press, 1990.

[7] Roberts PD, Bell CC. Spike timing dependent synaptic plasticity in biological systems. Biol Cybern, 2002; 87: 392–403.

[8] Nolfi S, Floreano D. Learning and Evolution. Autonomous Robots, 1999; 1: 89-113.

[9] Bäck T, Schwefel HP. An overview of evolutionary algorithms for parameter optimization, Evolutionary Computation. 1993; 1: 1-23.

[10] Holland JH. Emergence. Oxford University Press, Oxford, 1998.

[11] Floreano D, Husbands P, Nolfi S. Evolutionary robotics. In: Siciliano B, Khatib O, editors. Handbook of Robotics. Springer, 2008. pp. 1423-1451.

[12] Cliff D, Harvey I, Husbands P. Issues in evolutionary robotics. Sussex University, 1992.

[13] Floreano D, Mattiussi C. Evolution of spiking neural controllers for autonomous vision-based robots. Swiss Federal Institute of Technology. EPFL, 2001.

[14] Hagras H, Pounds-Cornish A, Colley M, Callaghan V, Clarke G. Evolving spiking neural network controllers for autonomous robots. In: IEEE 2004 Robotics and Automation Conference; April 2004; New Orleans, USA: IEEE. pp. 4620- 4626.

[15] Bailey JA, Wilcock R, Wilson PR, Chad JE. Behavioral simulation and synthesis of biological neuron systems using synthesizable VHDL. Elsevier, Neurocomputing, 2011; 74: 2392-2406.

[16] Maass, W, Natschlàger T, Markram H. Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Computation, 2002; 14:2531-2560.

[17] Jaeger H. The echo state approach to analyzing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[18] Paugam-Moisy H, Martinez R, Bengio S. Delay learning and polychronization for reservoir computing. Neurocomputing, 2008; 71: 1143–1158.

[19] Schrauwen B, Verstraeten D, Campenhout JV. An overview of reservoir computing: theory, applications and implementations. In ESANN'07 Advances in Computational Intelligence and Learning Conference; 2007; pp. 471–482.

[20] Hodgkin AL, Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. Physiol J, Aug. 1952; 117: 500–544.

[21] Brunel N, Van Rossum MCW. Lapicque's 1907 paper: from frogs to integrate-and-fire. Biological Cybernetics, December 2007; 97: 337–339.

[22] Izhikevich E. Simple model of spiking neurons. IEEE Transactions on Neural Networks, 2003; 14: 1569–1572.

[23] Soares G, Borges H, Gomes R, Oliveira G. Emergence of neuronal groups on a self-organized spiking neurons network based on genetic algorithm. In: Neural networks, 2010 eleventh Brazilian symposium on; 2010; Brazil. pp. 43–48.

[24] Izhikevich E. Polychronization: computation with Spikes. Neural Computation. 2005; 18: 245-282.

[25] Davis L. Handbook of Genetic Algorithms. Van Nostrand Reinhold, 1991.

[26] Mitchell M. An Introduction to Genetic Algorithms. MIT Press, 1998.

[27] Goldberg DE. Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesly, 1989.

[28] Poli R, Langdon WB, McPhee NF. A Field Guide to Genetic Programming. Stanford University, USA, 2008.

[29] Dan Su. Neural networks for control of artificial life form. MSc, York University, 2010.

[30] Oros N, Steuber V, Davey N, Canamero L, Adams R. Evolution of bilateral symmetry in agents controlled by spiking neural networks. IEEE Symp, Artificial Life Conference; April 2009; Nashville, TN. pp. 116–123.

[31] Kasabov N. To spike or not to spike: a probabilistic spiking neuron model. Neural Networks, 2010; 23: 16-19.

[32] Brébisson AD (ahd11). Evolving spiking neural network controllers for virtual animates that explore and forage in unknown environments. MSc, Imperial College London, 2012.

[33] Moutarde F. A robot behavior-learning experiment using particle swarm optimization for training a neural-based animate. Mines Paris Tech, 2008.

[34] Urbanczik R, Senn W. Reinforcement learning in populations of spiking neurons. Nature Neuroscience. March 2009; 12: 250-2.

[35] Jin X. Parallel simulation of neural networks on spinnaker universal neuromorphic hardware. PhD, Computer Science, Manchester University, 2010.