A Spiking Neural Network Based Autonomous Reinforcement Learning Model and Its Application in Decision Making

Guixiang Wang[1], Yi Zeng[1,2], and Bo Xu[1,2]

[1] Institute of Automation, Chinese Academy of Science, Beijing, 100190, China
[2] Center for Excellence in Brain Science and Intelligence Technology Chinese Academy of Sciences, Shanghai, 200031, China
yi.zeng@ia.ac.cn, guixiang.wang@ia.ac.cn

Abstract. In this paper, we propose an autonomous spiking neural network model for decision making. The model is an expansion of the basal ganglia circuitry with automatic environment perception, which constructs environmental states automatically from image inputs. The work in this paper has the following contributions: (1) In our model, the simplified Hodgkin-Huxley computing model is developed to achieve calculation efficiency closed to the LIF model and is used to obtain and test the ionic level properties in cognition. (2) A spike based motion perception mechanism is proposed to extract key elements for learning process from raw pixels without large amount of training. We apply our model in the "flappy bird" game and it play well after dozens of trainings. The model gets similar learning performance with human at the start of training. Besides, our model simulates cognitive defects when blocking some of sodium or potassium ion channels in the Hodgkin-Huxley model and this is an exploration of cognition deep into ionic level.

Keywords: Spiking neural network, Hodgkin-Huxley, Basal Ganglia, motion perception

1 Introduction

Human brains can handle complex tasks well, from seeing the situation to giving an available plan. When playing a game on some devices, brain does not just play it; it analyzes what it perceived, gets key elements and gives the next movement according to feedback. The brain understands the situation and traditional artificial intelligence doesn't. Algorithms inspired by brain structure and information processing way may give a better solution to understand and solve problems.

Brain-inspired neural networks provide new opportunities to achieve general intelligence, and became popular in recent years.

In this paper, we build a spiking neural network that could play strategic games by analyzing figures presented every time. The agent knows nothing about the game at the beginning. After several times of detection, it can feels motion and available movements. Knowing movement limitations and the environment, the agent will learn how to move to get better game scores. The policy learning is a reinforcement training process. Our spiking neural network model consists of two modules, one is environment perception and another one is autonomous learning.

Our motivation is to build a brain-inspired cognitive model for reinforcement learning tasks, so that agents with this model will have autonomous learning ability without telling what the environment is. The agents can perceive the environment with their eyes and abstract key elements for task with little prior knowledge. To complete strategic tasks such as reinforcement learning, our autonomous cognitive model borrows ideas in the brain both from the structural and the functional perspectives. We hope that our work could give a possible way to approach human-like intelligence.

The rest of this paper is organized as follows. Section 2 gives a description of the basal ganglia model and the Hodgkin-Huxley model. Section 3 is our methods, including the simplified Hodgkin-Huxley computing equation for our autonomous learning model; the motion perception algorithm we propose; the whole autonomous learning model. Section 4 is experiments and application of our model. Finally, Section 5 concludes the paper and discusses our model's limitations.

2 Previous Work

2.1 The Basal Ganglia Model

Biological brain gives us many inspirations. The basal ganglia are one of the important brain structures. They are important in many cognitive behaviors, including decision making and reinforcement learning.

The basal ganglia is composed of striatum, the STN (subthalamic nucleus), the GPe (globus pallidus external), and two output nuclei, the SNr (substantia nigra pars reticulata) and the GPi (internal globus pallidus) [3, 5-8]. Other nuclei, including the SNc (substantia nigra pars compacta) and the VTA (ventral tegmental area), are also seen as part of the basal ganglia. Each nucleus has different function in the brain's learning process.

There are some important nuclei need to be introduced. Striatum is the input nucleus of the basal ganglia and it receives direct input from cortical areas and modulatory afferents (DA) from SNc. It has two types of DA receptors, they are D1 and D2. The StrD1 receptor enhances response of inputs, while the StrD2 has the contrary effect [5, 25]. The STN is receives direct input from cortical areas. The outputs from the STN are excitatory, compared to inhibitory outputs from other nuclei in the basal ganglia. The GPi is the output nuclei of basal ganglia. The SNc and VTA release dopamine (DA) as an important modulatory signal, which are critical for cognitive behaviors [4, 5, 7].

In cognition process, the basal ganglia need to cooperate with other related brain regions to form the basal ganglia circuitry [4, 5, 6, 8, 25]. The basal ganglia receive input from the prefrontal cortex (PFC) and send output to the thalamus. The thalamus sends the basal ganglia's output back to the PFC [25].

The cognitive research on the basal ganglia has produced a large number of models. Start from the "box and arrow" model of the basal ganglia [14, 15] based on anatomical data of the brain, Many computing models are put forward [5, 7-10, 13, 15, 16]. These models try to improve the original basal ganglia circuitry based on the structure and function of the basal ganglia as well as their associated brain regions. Most of the basal ganglia models are built using artificial neural network [5, 9, 15], which cannot give a good description on neural activities. Ionic level neuron model such as the Hodgkin-Huxley model is not used in these basal ganglia models. Besides, these models are used to do simple action-selecting experiments, rather than solving real and complex cognitive tasks.

We will build our autonomous learning model based on the basal ganglia circuitry. Before describing our model, we need to introduce the mathematical model of the basal ganglia that gives us the inspiration.

2.2 A Spike Coding Model of the Basal Ganglia

Our work is related to a mathematical basal ganglia model for action selection [10] and its expanded spiking coding model [1, 8].

We develop our autonomous learning model based on the spiking coding model because of two reasons. (1) The model has low computational complexity and keeps main biological characters about the basal ganglia. (2) The model can deal with wide range of inputs with hundreds of dimensions [1].

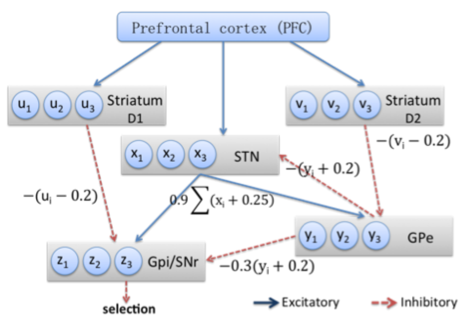


Fig. 1. Mathematical model of the basal ganglia [1]. The input vector of the basal ganglia is from the prefrontal cortex. x, y, z, u and v denote vector outputs from each nucleus respectively. The values in the equations, such as 0.25, are from the mathematic model of the basal ganglia [1].

The spike coding model's selection mechanism can be described as a linear equation [10]:

$$f(x_i) = \begin{cases} 0 & x_i < e_i \\ m(x_i - e_i) & e_i \leq x_i < 1/m + e_i \\ 1 & x_i > 1/m + e_i \end{cases} \tag{1}$$

where $f(x_i)$ is the output of a nucleus, and x_i is the input with $0 \leq x_i \leq 1$ as the value interval.

Figure 1 shows the connections among nuclei and the exact equations for each nucleus in the basal ganglia. The prefrontal cortex projects inputs vector (represent a set of actions) to the basal ganglia. The latter chooses one action according to the their values, and gives it's output through the GPi to the thalamus (not drawn). Note that the output of the basal ganglia is inhibitory, therefore, the selected action will have the smallest value that is closed to zero.

Each nucleus in the basal ganglia is represented by a group of biological neurons, as shown in Fig. 1, noted by circles. It is generally believed that biological neurons carry information by producing complex spike sequences [1, 11, 18]. These spikes encode input stimulus as firing rates [17, 18]. The firing frequency increases with the increment of the input stimulus.

The action potential of neuron i can be written as:

$$v_i(t) = G_i(J_i(x(t))) \tag{2}$$

where $v_i(t)$ is the action potential of a neuron and $G(x)$ is the neuron model. $J(x)$ is the current input, usually a linear function of $x(t)$. When the action potential reaches the threshold, a spike is released. The spike firing rate $r_i(x)$ during a time period can be obtained using the neuron model, therefore, the input x will be encoded as a firing rate.

After the encoding process, we need the decoding operation to regain the estimated input. The inputs are usually a function written as $f(x(t))$ and its estimated result can be expressed as:

$$f(\hat{x}) = \sum_i^N r_i(x) d_i \tag{3}$$

where d_i denotes the linear decoders and N denotes the number of neurons used to encode the input stimulus. To estimate the input function, we can use the least-squared-error method [1, 8, 12]. The deviation between the estimated value and the input can be written as:

$$E = \int [(f(x) - f(\hat{x}))]^2 dx \tag{4}$$

Through minimizing the deviation E, we obtain the least-squared version estimation of the decoder d_i [1,8]:

$$d = \Gamma^{-1} \Upsilon \qquad \Gamma_{ij} = \int r_i r_j dx \qquad \Upsilon_i = \int r_i f(x) dx \tag{5}$$

In our autonomous learning neural network, the Hodgkin-Huxley model is used to build the basal ganglia model using the method above. We will run the H-H equations to fit the relationship between firing rate and current input.

2.3 The Hodgkin-Huxley Model

The Hodgkin-Huxley model was found by Alan Lloyd Hodgkin and Andrew Fielding Huxley in 1952 [20]. They got it from the squid giant axon. Many parameters in the model are fitted using the experimental data. The H-H model is regarded as the best one to describe action potential's dynamic properties in the ionic level. The model is written as four equations [20]:

$$\begin{aligned} C\dot{V} &= I - \overbrace{\overline{g}_k n^4 (V - E_k)}^{I_k} - \overbrace{\overline{g}_{Na} m^3 h (V - E_{Na})}^{I_{Na}} - \overbrace{g_L (V - E_L)}^{I_L} \\ \dot{n} &= \alpha_n(V)(1-n) - \beta_n(V)n \\ \dot{m} &= \alpha_m(V)(1-m) - \beta_m(V)m \\ \dot{h} &= \alpha_h(V)(1-h) - \beta_h(V)h \end{aligned} \tag{6}$$

where V is the membrane potential, C is the membrane capacitance with a value $C = 1\mu F/cm^2$, and I is the externally current input.

The ionic current consists of three components, the sodium (Na+) current with three activation gates and one inactivation gate, the potassium (K+) with four activation gates, and the leak current carried primarily by chlorine (Cl-) [19, 21]. n, m and h in the equations represent the open probability of different ionic gates respectively [21], with n for potassium (K+), m and h for sodium (Na+).

The transition rates of the gate between open and closed state are denoted by $\alpha(V)$ and $\beta(V)$, which are voltage-dependent [19, 21] and named rate constant. The transition rate functions are fitted by the voltage clamp experiments [20, 21]. Please refer to [20, 21] for exact equations.

In this paper, we use the H-H model to build the basal ganglia model following the spike coding method described above. It can simulate more biological effects related to cognition. And we will apply the model to real cognitive task to test the model's ionic property and simulate it's influence on cognitions.

3 Methods

3.1 The Simplified Computing Hodgkin-Huxley Model

We use the Hodgkin-Huxley equations to build the spiking neural network model of reinforcement learning, based on the previous work about the basal ganglia in section 2. We use the H-H model here for two reasons. (1) The H-H model explains experimental results accurately and enables quantitatively voltage analysis on the nerve cell. (2) Sodium and potassium are associated to human cognition. The Changes on the ionic conductance in sodium or potassium channel may affect the decision making or learning process, thus gives us an opportunity to investigate on our intelligence deep into the ion level.

The basal ganglia model is usually built using the leaky integrate-and-fire (LIF) neuron. The reason is that LIF model has low computational complexity and is easy to control. While, LIF neuron has poor description on dynamic action potential and on ionic level activities. We cannot simulate cognitive defeats well just in LIF-neuron level. The H-H model does well deep into the ionic level, and this give us more information to analyze and predict biological properties.

Different from the LIF neuron, the H-H model is more elaborate. To reach the action potential threshold, the H-H model need to simulate hundreds of steps rather than one step as LIF. If LIF uses one step to climb up to fire a spike in 1 millisecond, the H-H model will spend hundreds of steps to reach the threshold in the same time period because of the ionic channels' dynamic properties. This phenomenon is determined by the H-H equations.

The H-H model releases spikes automatically in some frequency if the current input is larger than a certain value. If the LIF model runs in milliseconds, the H-H model runs in nanoseconds. This means when simulating using the H-H model, two loops are necessary, the external loop is simulation itself, the internal loop is ionic level simulation to accumulate the action potential. This needs more simulations and costs more computational complexity, therefore, it is necessary to present simplified computing equations for the H-H model.

To achieve calculation efficiency closed to the LIF model, several small changes are made in H-H equations in this paper. (1) The membrane capacitance C becomes smaller than the original value and is set to the value of $0.01-0.03\mu F/cm^2$. This will greatly increase the action potential in one step. And fewer steps are needed before reach the threshold. (2) To ensure a large enough firing rate , the current input should be larger than $6-7\mu A/cm^2$, and greater current inputs can ensure better accuracy. However, the current input should not be too large, otherwise, it will reduce the ionic properties of the H-H model. According to our experiments, a value of 30 is a better choice for the current input. (3) There is no internal loop for action potential accumulation in the simulation of H-H equations. The first two changes gain enough voltage increment in one step. And one step simulation gets the same accuracy than before. (4) The H-H equations fire spikes automatically in some frequency, therefore, to control H-H neuron's activity to encode different functions, intervention mechanisms are performed. We need to reset the voltage to zero after a spike and assign a refractory time period to the equations according to the function represented and the encoding process.

We rewrite the voltage equation of the H-H model in the following, with some improvement described above:

$$C\dot{V} = J(x) - \overbrace{\overline{g}_k n^4 (V - E_k)}^{I_k} - \overbrace{\overline{g}_{Na} m^3 h (V - E_{Na})}^{I_{Na}} - \overbrace{g_L (V - E_L)}^{I_L}$$

$$
\begin{aligned}
V &= 0, & if \quad V > 60mV \\
J(x) &= 30(ax+b), & -1 < a, b < 1 \\
C &= 0.02\mu F/cm^2
\end{aligned} \tag{7}
$$

where $J(x)$ is the current input, a linear function of the original input x, with a and b the random value for different neurons in the spike coding model. And 30 is a scale of the current input to guarantee enough firing rate of the H-H model. The peak voltage is about 80-100 mV, so we count a spike if the potential exceeds 60 mV and set the voltage to zero to let it restart right now or wait for a while before the end of the refractory.

Usually, the resting action potential is set as $V = V_{rest} = 0$ for easy calculation. Meanwhile, other potential and conductance parameters should be given the following values [19, 22, 24]:

$$
\begin{aligned}
&E_K = -12mV &&E_{Na} = 120mV &&E_L = 10.6mV \\
&\overline{g}_k = 36mS/cm^2 &&\overline{g}_{Na} = 120mS/cm^2 &&g_L = 0.3mS/cm^2
\end{aligned}
$$

When the input current is larger than a specific value, which is about $6 - 7\mu A/cm^2$ in our simulations, regular spiking activity is observed [22]. If the spike interval is T, the average firing rate can be expressed as $f = 1/T$, and it increases as the stimulus enhanced [22,23,24], as shown in Fig. 2.

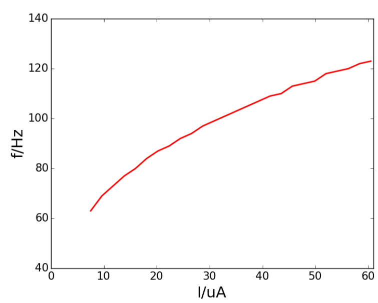


Fig. 2. Firing rate curve of the Hodgkin-Huxley model under various current inputs. In our experiments, the neuron starts to have stable regular spikes when the current input is larger than 7.5 $\mu A/cm^2$. The resting potential is set to 0. The time constant is 0.01ms. Simulation period is 1s.

3.2 Spike Based Motion Perception

Eyes can detect the moving objects in the environment and pass the results back to the brain as important references for cognition. In this paper, we build a neural network model as the eyes of agents. Agents with this model could percept and understand the environment to some extend. With the basal ganglia as the reinforcement learning model, agents will get the autonomous learning ability to do some complex cognitive task with image input rather than simple action selection.

The perception of an image happens in multilayer, including the input layer, the perceptive layer and the output layer. For an image input, Every pixel has a group of neurons to represent its value in the input layer. Population coding algorithm is used to change the pixel to spike sequences of a period of time. These spike sequences are the inputs of two special neurons in the perceptive layer. In this layer, two neurons represent a pixel. Neuron A works in the current time, and neuron B works in the past with a constant time delay τ_0. It means that neuron A receives the current inputs and B receives the last inputs. If there is movement in the two neurons' position, they will have different spike outputs. In the output layer, each pixel has one neuron. The neuron's inputs are neuron A and neuron B. Neuron A is excited, and B has the contrary effect. If there is no movement, the neuron has no spike, otherwise, it will release spikes.

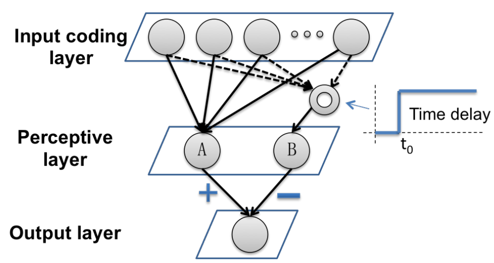


Fig. 3. Multilayer structure of the motion perception model. The time delay here is used to keep the spiking sequences of the last time, therefore, the output layer will detect movements over time.

The whole motion perception process is shown in Fig. 4. Each pixel in the image has its representing neurons from the input coding layer to the output layer. In the input coding layer, each pixel is represented by a group of neurons and converted into spiking sequences using the population coding [26]. These neurons have overlapping gaussian receptive fields as shown in Fig. 4. The number of neurons and the encoding interval can be variable to obtain good coding results. With good parameters, the population coding can distinguished two similar values by different spiking patterns. The experiment in the next section shows its good performance in pixel encoding.

In the population coding, suppose that $[I_{min}, I_{max}]$ is the encoding interval, and M is the number of neurons, the center of gaussian receptive field for each neuron can be calculated as [26]:

$$C_i = I_{min} + (i-1) \cdot \frac{I_{max} - I_{min}}{m-2} \tag{8}$$

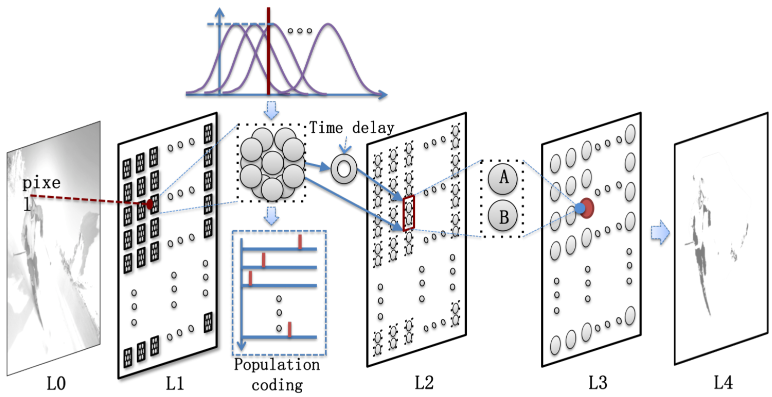


Fig. 4. The motion perception process for raw image inputs. L0 is the raw image, L1 is the input coding layer, L2 is the perception layer, L3 denotes the spike output layer, and L4 is the spike activity of the original image, showing the movement area. The population coding process is performed in L1. Every pixel is converted into a group of neurons' spiking sequence in L1.

and the variance equation of each neuron is written as [26]:

$$\sigma_i = \beta \cdot \frac{I_{max} - I_{min}}{m - 2} \tag{9}$$

where the common values of β are at $[0, 2]$. The coding process will calculate a value's intersection with each gaussian receptive field and obtain spiking times for these gaussian neurons.

The output of the motion perception is an image with movement areas highlighted by spikes, as shown in L4 in Fig. 4. In the environment with less background variation, it's easy to get a object's moving direction and separate the moving area from other backgrounds. This is also useful for moving obstacle avoiding, which will be described in the next section.

3.3 Spiking Neural Network Based Autonomous Reinforcement Learning Model

In this section, we will describe our whole autonomous reinforcement learning model. The model consists of two important spiking models, which are the motion perception and the basal ganglia model. Our autonomous model is designed to interact with the environment and perform reinforcement learning to avoid moving obstacles without giving environment states. All the important model in this paper are built using the Hodgkin-Huxley model.

Our autonomous learning model is shown in Fig. 5. The model forms a Q learning loop to learn to interact with the environment. The motion perception model is used to detect moving objects from the image inputs and obtain new

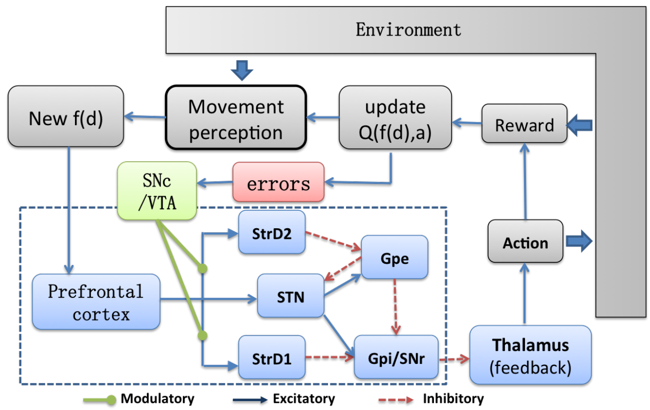


Fig. 5. The autonomous reinforcement learning model built using spiking neural network. The basal ganglia structure and their action selection function (in the rectangle) is referred from [1] and [2]. $f(d)$ is a function of distance d, named the state function. The Q-value is a function of state function and action a. The environment states is calculated by the state function and not created manually. And the agent knows nothing except several basic survival rules.

environment state function. The basal ganglia model takes the state function as input and performs action selection among all available actions every time step.

As introduced in section 2, the basal ganglia model is built using methods in [1,2]. Learning in the basal ganglia circuitry is achieved by the updating of the synaptic connections between the basal ganglia and prefrontal cortex. Learning will not finish until the basal ganglia getting good action selections and the agent can interact well with the environment.

If the agent wants to survive in the environment, it has to follow two rules: (1) There should be no obstacles in agent's forward direction. (2) The agent should be away from obstacles as far as possible if no other goals. The state function and rewards are calculated according to these two rules.

4 Experiments and Applications

In this section, we apply our model to the game called "flappy bird" to test our method's performance. As described above, the simplified computing Hodgkin-Huxley equations are used through our model and also need experiments. In the following, we will first make experiments for the H-H model. And next, we make experiments in the game of "flappy bird" using our model.

4.1 The simplified Hodgkin-Huxley Model

As discussed above, the Hodgkin-Huxley model has many ionic properties that the LIF model hasn't. We proposed our simplified H-H equations to reduce the computing complexity and keep their ionic properties at the same time. The experiments in Fig. 6 show the performance of our simplified H-H model. According to the experiments results, we can conclude that, the simplified H-H model has lower computing complexity and similar performance with the LIF model.

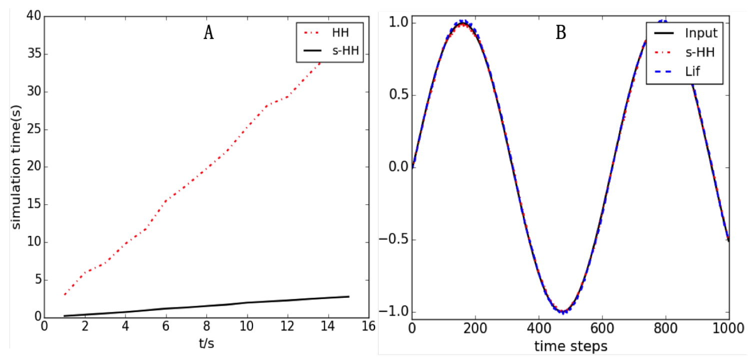


Fig. 6. The experiments for the simplified H-H equations. A: The simulation time comparison between the original H-H model and its simplified model. B: The function tracking experiment for the simplified H-H model and the LIF model. The result shows that our simplified H-H model has similar function recover performance with the LIF model.

4.2 Autonomous Reinforcement Learning Model in the Game

Before playing the "flappy bird" game, we will explain how to calculate the environment state functions and rewards.

In the "flappy bird" game, the pipes move toward the bird, and the bird move up and down to adjust its position to try to pass through the pipes' gap without collision. According to the two surviving rules proposed above, the bird should avoid colliding with the pipes along their moving directions, and try to stay away from each pipe's terminal as far as possible. Because there are pipes both in the above and below, the bird should try to approach the center of the pipes' gap.

The environment state is a key element that describes the environment. Without specified in advance, it can be a function of distance to the center of the pipe's gap, written as $f(d)$. The bird gets the center of the pipes' gap using the motion perception model when a pair of pipes coming. The environment state function $f(d)$ is calculated every time step according to the bird's position. For each value of state function, there are two actions the bird can take, which are up and down. The Q-value function of the Q learning process is represented by the function of $f(d)$ and action a, written as $Q(f(d), a)$. Every time step the Q-value updates, the error is used for the autonomous model's learning.

The reward is also important for the learning process. It can be calculated according to the motion perception results.If there are no pipes in the bird's forward direction (pipes? approaching direction), the rewards are positive, if not, the rewards should be negative. If considering the center of the pipe's gap, the reward can also be the function of the distance d and has different values as d varies.

A. The Motion Perception Instead of telling the model the key elements of the environment, our model deals with raw image inputs. The proposed motion perception algorithm will help detect the obstacle's movement and key informations such as state and reward could be calculated according to the spiking results. The motion perception results is shown in Fig. 7.

The spike results shows a good input coding performance of the population coding algorithm. The coding process converts a pixel into spike sequences in a certain time interval. There are dozens of coding steps in this time interval. From Fig. 7. B, we can see that, the motion perception process can detect the main moving areas even in single time step. The spiking results in Fig. 7. A is the result collection of all the coding time steps.

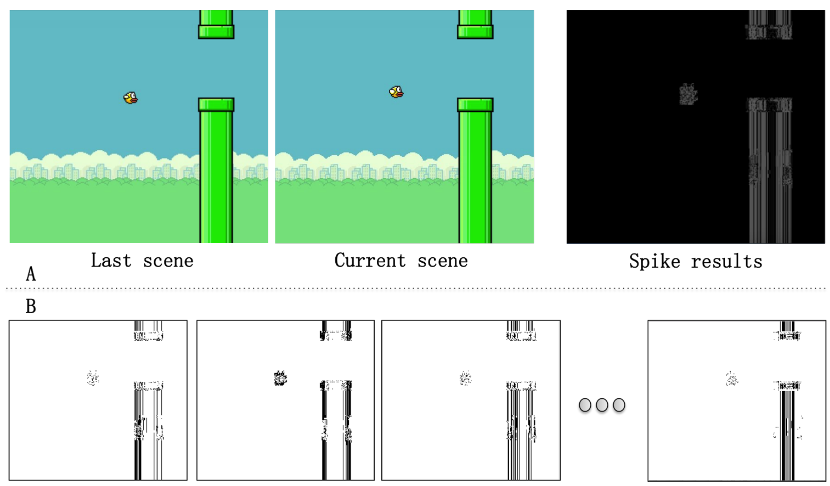


Fig. 7. The motion perception result for the "flappy bird" game. A: The motion perception model takes the game scene as input in the current time step and gives the spike results in the next time step. B: The spiking results of every population coding step.

B. Playing the Game We train our autonomous learning model in the "flappy bird" game. The experimental results are shown in Fig. 8. Under the Hodgkin-Huxley model, our model can play the game well after about 50 attempts. One can see from the Fig. 8. B that, at the beginning, the bird usually gets low scores before colliding pipes. After a couples of attempts, the bird has learned how to avoid the pipes and achieves more than 30 scores during its last attempts, and even larger scores if the game continues. This achievement can be compared with a good human player.

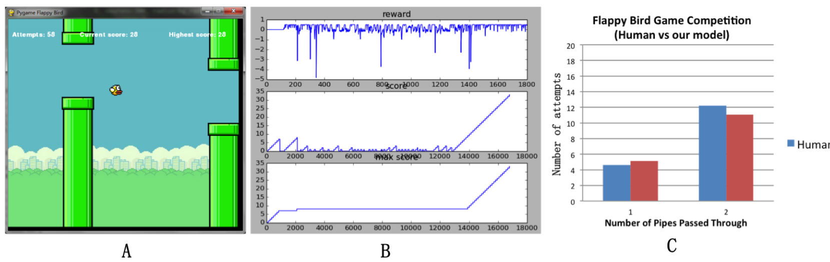


Fig. 8. The game playing results using our model. If the bird collides with the pipe, it costs one attempts. If the bird passes one pipe, its score plus 1. A: the game scene. B: the score recording during the learning process. C: the performance comparison between human and our model.

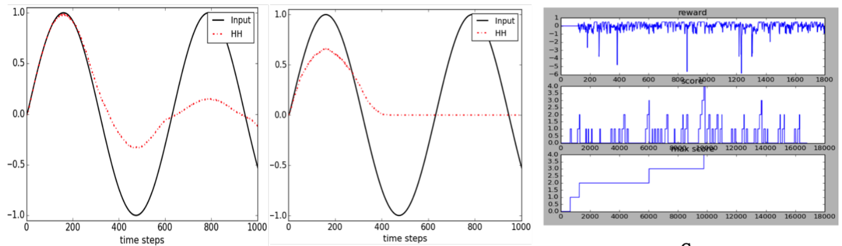
In the Fig. 8. C, we make an comparison between human player and our model. The results show that, at the beginning of learning, our model has similar performance with general human player. That means our autonomous learning model has the ability to simulate simple human decision making process and gets similar results. That's important to the application of models which are built using spiking neural network.

C. Ionic Property Test of the H-H Model The sodium and potassium ion in the H-H equations are very important to the cognition process in biological brains. Fig. 9 shows some simulation results of the insufficiency of these two ions in the H-H model. The experiments show that, lacks of sodium or potassium ion causes cognitive deficits. Fig. 9. C shows that, the bird can only pass through several pipes and can not get a higher score even in longer training.

5 Conclusion

This paper proposes an autonomous model for decision making. The model is built using spiking neural network and can do autonomous learning without telling a lot of the environment's information.

In this paper, we propose the simplified Hodgkin-Huxley computing model to achieve calculation efficiency closed to the LIF model and keep the ionic level properties in cognition. To detect the moving objects in the environment, a spike based motion perception mechanism is developed to extract key elements from



A B C

Fig. 9. The insufficiency experiments of sodium and potassium. A: lack of sodium ion. B: lack of potassium ion. C: playing the "flappy bird" game with the lack of sodium ion.

raw pixels without large amount of training. In the experimental section, we apply our model in the flappy bird game and it play well after less than 60 trainings. Besides, our model simulates cognitive defects when blocking some of sodium or potassium ion channels in the Hodgkin-Huxley model and this is an exploration of cognition deep into ionic level.

However, our model is not perfect, it has several limitations. It can not detect the moving object correctly when the background is changing over time. And its basal ganglia model is not real enough comparing to the brain. These limitations are something we are heading towards. And we will improve our model to get better cognitive performance.

Acknowledgements

This study was funded by the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB02060007), and Beijing Municipal Commission of Science and Technology (Z151100000915070, Z161100000216124).

References

1. Stewart, T. C., Bekolay, T. and Eliasmith, C.: Learning to select actions with spiking neurons in the basal ganglia. Frontiers in Neuroscience. vol. 6, no. 2, 1-14 (2012)
2. Bekolay, T. and Eliasmith, C.: A general error-modulated STDP learning rule applied to reinforcement learning in the basal ganglia. Computational and Systems Neuroscience conference, Salt Lake City, Utah. 24-27 (2011)
3. Eliasmith, C.: How to build a brain. Reprint edition. New York: Oxford. 121-171 (2013)
4. Chakravarthy, V. S., Joseph, D. and Bapi, R. S.: What do the basal ganglia do? A modeling perspective. Biological Cybernetic, vol. 103, no. 3, 237-253 (2010)
5. Frank, M. J.: Dynamic dopamine modulation in the Basal Ganglia: a neuro computational account of cognitive deficits in medicated and nonmedicated parkinsonism. Journal of Cognitive Neuroscience, vol. 17, no. 1, 51-72 (2005)

6. Utter, A. A. and Basso, M. A.: The basal ganglia: an overview of circuits and function. Neuroscience and Biobehavioral Reviews, vol. 32, no. 3, 333-342 (2008)
7. Redgrave, P., Rodriguez, M., Smith, Y., Rodriguez-Oroz, M. C., et al: Goal-directed and habitual control in the basal ganglia: implications for Parkinson's disease. Nature Reviews Neuroscience, vol. 11, 760-772 (2011)
8. Stewart, T. C., Choo, X. and Eliasmith, C.: Dynamic behavior of a spiking model of action selection in the basal ganglia. Proceedings of the 10th International Conference on Cognitive Modeling, 5-8 (2010)
9. Frank, M. J.: Hold your horses: a dynamic computational role for the subthalamic nucleus in decision making. Neural Networks, vol. 19, no. 8, 1120-1136 (2006)
10. Gurney, K., Prescott, T. J. and Redgrave, P.: A computational model of action selection in the basal ganglia. Biological Cybernetics, vol. 84, no. 6, 401-410 (2001)
11. Stewart, T. C. and Eliasmith, C.: Large-scale synthesis of functional spiking neural circuits. Proceedings of IEEE, vol. 102, no. 5, 881-898 (2014)
12. MacNeil, D. and Eliasmith, C.: Fine-tuning and the stability of recurrent neural networks. Public Library of Science (PLoS One), vol. 6, no. 9, 1-16 (2011)
13. Gurney, K., Prescott, T. J., Wickens, J. R. and Redgrave, P.: Computational models of the basal ganglia: from robots to membranes. Trends in Neuroscience, vol. 27, no. 8, 453-459 (2004)
14. Albin, R. L., Young, A. B. and Penney, J. B.: The functional anatomy of basal ganglia disorders. Trends in Neuroscience, vol. 12, no. 10, 366-375 (1989)
15. Bar-Gad, I. and Bergman, H.: Stepping out of the box: information processing in the neural networks of the basal ganglia. Current Opinion in Neurobiology, vol. 11, no. 6, 689-695 (2011)
16. Iqarashi, J., Shouno, O., Fukai, T. and Tsujino, H.: Real-time simulation of a spiking neural network model of the basal ganglia circuitry using general purpose computing on graphics processing units. Neural Networks, vol. 24, no. 9, 950-960 (2011)
17. Cessac, B., Paugam-Moisy, H. and Viéville, T.: Overview of facts and issues about neural coding by spikes. Journal of Physiology-Paris, vol. 104, no. 1, 5-18 (2010)
18. Dayan, P. and Abbott, L. F.: Computational and mathematical modeling of neural systems: Model Neurons I: Neuroelectronic. Cambridge: MIT Press, (2003)
19. Izhikevich, E. M.: Dynamical systems in neuroscience: the geometry of excitability and bursting. Cambridge, MA: MIT Press, (2004)
20. Hodgkin, A. L. and Huxley, A. F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. The Journal of Physiology, vol. 117, no. 4, 500-544 (1952)
21. Nelson, M. E.: Electrophysiological Models In: databasing the brain: from Data to Knowledge. Wiley, New York, (2004)
22. Gerstner, W. and Kistler, W. M.: Spiking neuron models. Single Neurons, Populations, Plasticity. Cambridge: Cambridge University Press, (2002)
23. Wells, R. B.: Introduction to Biological signal processing and computational Neuroscience. Moscow, ID, USA, (2010)
24. Long, L. N. and Fang, G. L.: A review of biologically plausible neuron models for spiking neural networks. AIAA InfoTech Aerospace Conference, Atlanta, GA, 20-22 (2010)
25. Weber, C., Elshaw, M., Wermter, S., Triesch, J. and Willmot, C.: Reinforcement Learning: Theory and Applications: Reinforcement learning embedded in brains and robots. Vienna, Austria, (2008)

26. Bohte, S. M., Poutre, H. L. and Kok, J. N.: Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks. IEEE Transactions on Neural Networks. vol. 13, no. 2, 426-435 (2002)