Real time human motion recognition via spiking neural network

Yang Jing¹, Wu Qingyuan², Huang Maiqi³ and Luo Ting⁴
Beijing Normal University, Zhuhai campus, Zhuhai, Guangdong, China
¹ yangjing@bnuz.edu.cn; ² wuqingyuan@bnuz.edu.cn; ³ huangmaiqi@outlook.com; ⁴ Gloria_LuoTing@163.com

Abstract. Real time human action recognition is to recognize the human motion type based on skeleton movement in real time and is always a challenging task. In this paper, a novel method is proposed to accomplish the classification by using Spiking neural network (SNN) which is biology oriented neural network dealing with precise timing spikes. First, a new temporal encoding scheme is used to encode the real time motion capture data into a series of spikes and the according type of the motion is represented by a spike time. Second, a two-layered spiking neural network is initiated and trained through a gradient descent learning algorithm. The experimental results show that this method achieves a good learning precision and generalization.

1. Introduction
With the rapid development of motion capture techniques and systems, more and more motion data are available for human action recognition in the research area of computer vision. Especially depth cameras can provide 3D depth data [1] of joint positions of the human skeleton which are showed in figure 1. That may be more helpful in recognition of the motion type than just using 2D data. However the depth map also increases the amount of the data and makes the recognition process more complex. As we know the human body is an articulated system of rigid segments connected by joints. The shape information carried by the contour can be extended to the 3D case for motion recognition [2]. The 3D depth data not only facilitates a rather powerful human motion capturing technique, but also makes it possible to efficiently model human-object interactions and intra-class variations [3].

![Figure 1. Skeleton of human body.](image-url)
SNN is the third generation neural network model and its inputs and outputs are both multiple spikes, i.e., spike trains. It’s widely accepted that artificial SNNs can stimulate any sigmoidal neural networks and even capable of exploiting time as a resource for coding and computation in a much more sophisticated manner than typical neural computational models [4]. As SNN deals with spike times the encoding of information is an important step in its applications. There are two ways of encoding, rate encoding and temporal encoding. Supervised learning algorithm is another hottest research area of SNN. For the learning of spiking neuron (SN), ReSuMe (remote supervised method) [3] is based on the rules of STDP and anti-STDP and Xu and el. [6] proposed another method based on gradient descent. For the learning of SNN, SpikeProp [7] is also a gradient-descent-based learning algorithm and there some improved algorithm based on SpikeProp. However only single spike is allowed to fire in SpikeProp and that narrows the application.

In this paper, we propose a novel real time motion data recognition approach. This method encodes the motion data into several spike trains and trains the motion samples with a two-layered SNN. The main contribution of this paper lies in the new encoding scheme of the motion data. The arrangement of the paper is as follows. The SNN model is briefly introduced in Section 2. Section 3 gives the encoding scheme of motion data and discusses the learning process of SNN. In Section 4 some experiments are conducted to demonstrate the efficiency of the classification method. Finally a conclusion is drawn in Section 5.

2. SNN model
The multilayer SNN is a fully connected feedforward network. Each neuron in the network is a spiking neuron (SN) and there are three main SN models, Leaky Integrate-and-Fire model (LIF), Hodgkin-Huxley model (HH) and Spike Response model (SRM). Because the gradient descent learning algorithms require computation of partial derivatives we take the SRM, whose internal state can be expressed intuitively, as SN model in this paper. As the figure 2 shows, the neuron receives all the weighted spikes from the presynaptic neurons via several synapses. When the neuron’s internal state exceeds the threshold it fires a spike and then drops immediately to zero.

Figure 2. Spiking neuron.

Figure 3 is the architecture of SNN. Between two neurons in adjacent layers there are several synapses with different transmit delays and weights, as shown in Fig 3 with dotted line. For the neuron \( j \) in layer \( l \), its internal state at time \( t \) is expressed as,

\[
    u_j(t) = \sum_{i=1}^{N_{l+1}} \sum_{k=1}^{K} w_{jk} g(t - t^{(f)}_{ij} - d^k) + \eta(t - t^{(o)}_{i})
\]

where \( N_{l+1} \) is the number of neurons in layer \( l+1 \), \( K \) is the number of synapses between neuron \( l \) and its presynaptic neuron, \( F_i = \{t^{(f)}_{ij}, t^{(2)}_{ij}, \ldots, t^{(F_i)}_{ij}\} \) is the spike train fired by neuron \( i \) in layer \( l+1 \) and \( t^{(o)}_{i} \) is the time of the most recent output spike for neuron \( j \) prior to the current time \( t \), \( d^k \) is the
$k$ th synapse’s delay, $w_{ij}^k$ is the weight of the $k$ th synapse between presynaptic neuron $i$ and postsynaptic neuron $j$, $R_a$ is the length of the absolute refractory period which means the input spikes futile in the period of $(t^{(p)}_i, t^{(p)}_j + R_a)$. $e(t)$ is the spike response function expressed as

$$e(t) = \begin{cases} t e^{-\frac{t}{\tau}} & \text{if } t > 0 \\ 0 & \text{if } t \leq 0 \end{cases} \quad (2)$$

where $\tau$ is the time decay constant. $\eta(t)$ expressed in formula 3 is refractoriness function and it can make internal state drop immediately after firing to prevent firing in succession. $\theta$ is the neuron threshold and $\tau^k$ is the time decay constant.

$$\eta(t) = \begin{cases} -2.9e^{-\frac{t}{\tau^k}} & \text{if } t > 0 \\ 0 & \text{if } t \leq 0 \end{cases} \quad (3)$$

3. Motion data recognition

3.1. Motion data encoding
As the inputs of the spiking neural network is a series of spikes it’s an important step to encode the captured motion data into spike times. There are different ways to encode the inputs. For example, population encoding scheme is to encode each input data separately by $M (>2)$ identically shaped overlapping Gaussian functions centered at $M$ different locations. And the input data is finally encoded as $M$ spike times between 0 and 10 ms. Apparently, this kind of encoding method produces artificial spike times which imply no real time sequences. As the human motion data is composed of all the joints’ position information at each frame it has a close connection with time. The spiking time in SNN is a kind of relative time interval so the frame in the motion capture can just be considered as spiking time interval.

Based on these explanations we introduced a new coding method based on the timely human joint position sequences in this paper. First, we consider joint $i$ in the human body. It has 3 coordinates information, that is, $p_i(t) = (x_i(t), y_i(t), z_i(t))$ at a frame $t$. For $x$-coordinate, the following figure shows how the position of one joint changes with time.

![Figure 4. X-coordinate position change of on joint](image)

From the above figure, we can see that the joint’s position rises or drops from time to time. Each local lowest point means the joint’s position will rise at the next frame and each local highest point means it will drop at the next frame. We name these key frames because they indicate the change in
direction of joint movement. We divide the key frames into two inputs, that is, input for local highest point and input for local lowest point and set an input spike at each local extreme point. So the information in Figure 4 can be encoded into two spike trains which are showed in the following figure.

![Spike train 1](image1)

![Spike train 2](image2)

**Figure 5.** Parts of encoded spike train of one joint.

It can be seen in Fig 5 that the spike train 1 have six input spikes which means there are six movement shifts and each one represents the change of movement direction. And the spike train 2 means the adverse direction movement shift. In this encoding scheme the movement range and velocity information is abandoned. We think the movement range plays less important role in the motion recognition as a huge, fast wave and a small, slow wave should be put into the same category. But there is another import information should not be omitted, the initial position. This information shows where the joint begins to move so the initial position of each joint should also be encoded. Moreover, to avoid the inconsistency of position data caused by the human movement all the joints’ positions are normalized based on the benchmark of joint 7 showed in Figure 1. The initial position of each joint can be encoded by population encoding scheme as an input to SNN.

### 3.2. Learning

Supervised learning algorithm based on temporal encoding of spikes is adopted in the training of SNN in this paper. The goal of SNN learning is to make the output neuron fire spikes at desired time that is, minimize the error function defined as the following formula,

$$ E = \frac{1}{2} \sum_{j=1}^{N} \sum_{f=1}^{F} (t_{j}^{(f)} - s_{j}^{(f)})^2 $$

(4)

where \( t_{j}^{(f)} \) denotes the desired output and \( s_{j}^{(f)} \) denotes the actual output. The weight updating rule based on gradient descent is expressed as

$$ \Delta w_{ij} = -\alpha \nabla E_{ij} $$

(5)

The detailed weight adjustment steps have been described in reference 6. The number of synapse between two neurons should be designed carefully because few synapses results in reduced convergence rates and classification accuracy whereas more synapses reduces the learning efficiency with no improvement in convergence or classification accuracy. Actually setting synapses aims to add the number of adjustable weights to improve the convergence of learning. If a motion data has 20 joints and each one has three coordinates there would be enough inputs after encoding. So adding synapses between input layer and hidden layer can be skipped. The neuron threshold should also be considered. For this motion recognition example, there are many input spikes after encoding and the running time is relatively long so too small threshold like 1 would lead to frequently firing. In this case the threshold can be increased one by one until the frequency of firing is appropriate.

### 3.3. Readout

The readout part aims to extract information about the stimulus from responses of the learned SNN. There are different ways to represent a certain class of patterns. For example, the number of output neuron can be set equal to the number of the motion type and each output neuron represents one motion type or we can use several SNNs with each net to learn one class or we can use just one output neuron indicating the class by different output spike train. Note that the set of desired output spike
train may have a significant influence on the experimental results. So a possible solution is to compute the output of the SNN before learning and observe the possible output distribution. Making the SNN’s output neuron fire spikes at the exact desired time is not easy. A frequently used method is to set a tolerance \( t \) for the desired time \( t_j^{(f)} \). The output spike between \([t_j^{(f)} - t, t_j^{(f)} + t]\) can be considered as a correct output.

4. Experiments

4.1. Dataset and experiment setup

The motion data used in our experiments comes from MSR-Action3D dataset available at http://www.uow.edu.au/~wanqing/#MSRAction3DDatasets. This dataset is skeleton data in real world coordinates contains twenty motion types. The 3D joint positions are extracted from the depth sequence. Every motion has twenty joints information. The SNN in this experiment has one hidden layer with 10 hidden neurons and one output layer with just one output neuron. There is no synapse between input neuron and hidden neuron but 5 synapses between hidden neuron and output neuron and the time delay of the synapse is 1, 3, 5, 9 and 13. The parameters \( \gamma \) and \( M \) in the population encoding are 1.5 and 4 individually. The neuron parameters used here are as follows: \( \tau = 11 \) ms, \( \tau_a = 80 \) ms, \( \theta = 4 \) and \( R_a = 1 \) ms. A SNN with one output neuron is used to learn the motions and indicate the motion type and the output spike train consists of two spikes. 60% samples are used to train the SNN and the rest samples are used to test the recognition accuracy.

4.2. Results

The following table 1 shows learning accuracy and testing accuracy to different learning epoch. As the large amount of motion data would lead to the increase of learning time we divide the dataset to three different action sets. Each action set has three types of motion data and all the data is encoded into several spike trains. The results show that after learning the SNN can recognize more than 80% training samples while the generalization is about 75%.

| Data set       | epoch | Accuracy |           |           |
|---------------|-------|----------|-----------|-----------|
|               |       | Training set | Test Set |           |
| Action set 1  | 200   | 0.94      | 0.81      |           |
|               | 150   | 0.91      | 0.7       |           |
| Action set 2  | 200   | 0.85      | 0.76      |           |
|               | 150   | 0.82      | 0.70      |           |
| Action set 3  | 200   | 0.90      | 0.82      |           |
|               | 150   | 0.86      | 0.75      |           |

5. Conclusion

This paper focuses on the application of SNN to do the motion recognition. The 3D coordinates motion data is encoded to several spike trains. The recognition takes advantage of the time character of the spike and the precise time of each joint’s moving direction shift is kept as the key input information. Through SNN learning the network can remember and recognize different motion type. Our future work will focuses on further improving the recognition accuracy and recognizing more types of similar motions.

Acknowledgements

This research was supported by the National Science Foundation of China under grants (No.61503031), financially supported by the grant from the Soft Science Research Project of
Guangdong Province(Grant No.2014A030304013), a grant from the Guangdong province Science & Technology Plan(No.2016A040403029) and Guangdong province Philosophy Social Science "12th five" plan Interdisciplinary project（No.GD15XGL02）

References
[1] S Zhang 2010 Real-time 3D shape measurement using digital fringe projection techniques Optics and lasers in engineering 48(2) 149-58
[2] W Li, Z Zhang and Z Liu 2010 Action recognition based on a bag of 3D points Computer vision & pattern recognition workshops pp 9-14
[3] J Wang, Z Liu, Y Wu and J Yuan 2014 Learning Actionlet Ensemble for 3D Human Action Recognition Springer International Publishing 36 (5) 1290-97
[4] Maass W 1997 Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons Advances in Neural Information Processing System 9 211-7
[5] Ponulak F, Kasinski A 2010 Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting Neural Computation 22 (22) 467-510
[6] Xu Y, Zeng X, Han L and Yang J 2013 A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks Neural Networks 43(4) 99-113
[7] S M Bohte, J N Kok and J A La Poutre 2002 Error-backpropagation in temporally encoded networks of spiking neurons Neurocomputing 48(1) 13-37