Spatial Localization using SNNs

Bachelor Thesis Project (Stage-2) by

Preetam Pinnada 170070042

Supervisor:

Prof. Udayan Ganguly

Department of Electrical Engineering

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

2021

Contents

Abstract

Neurons present in the Entorhinal Cortex (EC) of a mammalian brain produce interesting spatial firing patterns. Studies suggest that these spatial representations play a crucial role in understanding and navigating through the environment. These neurons are classified based on their firing patterns, viz. Grid cells, Place cells, Border cells, Band cells, Head direction cells, etc. It is believed that these representations encode the environment and perform path integration. Recent work in machine learning community supports this hypothesis. RNNs were trained to perform spatial localization and firing patterns in the hidden layers were observed to be strikingly similar to that of neurons in EC. As a part of BTP-1, a spatial localization system using RNNs was built to perform path integration for artificial agents. Further, in BTP-2, the attempt is to build a spatial localisation system using SNNs which will enable energy-efficient neuromorphic path integration system.

Keywords: Entorhinal Cortex, Path Integration, Spiking Neural Network, LIF Neuron

Chapter 1

Introduction

Path integration is the ability of an organism to calculate its updated position using idiothetic cues such as vision, motor neurons, directional cues, etc. Previously, in BTP-1 (refer section 8.1.1) an RNN was built to perform path integration and the representations learnt by the model were observed to appreciate the spatial firing patterns. This model can be used with any artificial agent to perform spatial localisation for navigation.

Now in BTP-2, the attempt is to perform such spatial localisation using spiking neural networks (SNNs). This will provide a neuromorphic path integration system which can be integrated with the C.Elegans work [10] to serve the bigger goal of developing a biologically plausible, energy efficient, navigation system for artificial robots. So the main question here is, how to build an SNN that performs spatial localisation?

Because of discrete activity of spiking neural networks, training becomes challenging. So in this work the approach is to make use of already trained artificial neural network, translate it to a spiking neural network and execute energy efficient forward-pass through it. This work explores transfer learning from RNNs to spiking domain and addresses the challenges encountered.

In chapter 2, the reviewed literature is discussed. In chapter 3, the details of model implementation and investigation is reported. Chapter 4 summarises the work and outlines potential directions for future work. Chapter 5 lists all the references, and chapter 6 links to the associated blogposts and code repositories.

Chapter 2

Literature Review

A couple of methods exist in literature that attempt to train and build spiking neural networks, most notable of them being – spike-timing dependent plasticity (STDP), tandem learning, and transfer learning.

Wu et al. 2019b [2]; Taherkhani et al. 2019 [3] use algorithms specific to spiking neural networks which are mainly based on spike-timing dependent plasticity. But until date, these methods have reported to effective only for shallow networks and simple datasets.

Wu et al. 2019a [4] discusses tandem learning. In this method, forward passes are made on both artificial neural network and the twin spiking neural network. The backward pass of errors and gradients from SNN are used to update the weights of twin ANN. These updated weights are then transferred to the to the twin SNN for the forward pass. But this requires both ANN and SNN while training and is costly.

Hence, taking inspiration from Z.Yan et al. 2021 [5], transfer learning is explored. In this method, artificial neural network is trained and the weights are appropriately transferred to spiking neural network. And then energy and cost effective forward passes are made through SNNs.

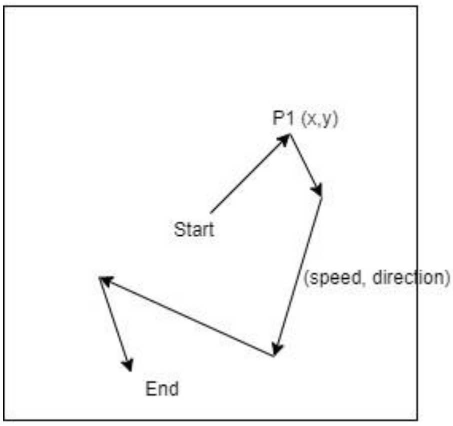
However, it is important to note that all the above literature discusses mainly about implementing CNNs in spiking domain. Whereas, the requirement in this work is to implement RNN in spiking domain. There's very little to no concrete research on recurrent spiking neural networks. This work will ideate and take the first steps towards building R-SNNs

Chapter 3

Approach

3.1 The ANN Model

The model consists of input layer of 2 neurons that accepts speed and absolute direction. Followed by a simple recurrent layer of 100 neurons to represent the hidden state, a gaussian noise layer, and a fully connected dense layer that maps the hidden state to an output (x, y) as shown in Fig 1(b). At every time step, speed and direction is input to the model and updated position at the next time step is estimated by the model.



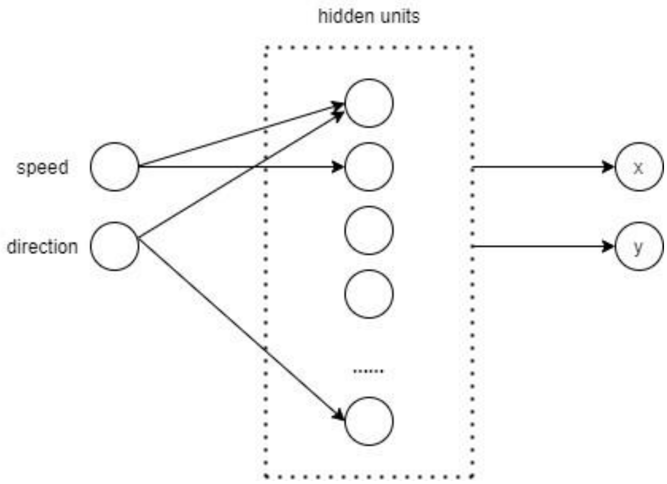


Fig 1. (a) Agent moving around in a rectangular environment; (b) Network architecture

State Update

$$\tau \frac{dx_i(t)}{dt} = -x_i(t) + \sum_{j=1}^{N} W_{ij}^{rec} u_j(t) + \sum_{k=1}^{N_{in}} W_{ik}^{in} I_k(t) + b_i + \xi_i(t)$$

$u_i(t) = tanh(x_i(t))$
$x_i(t)$: state of the i^{th} unit
$u_i(t)$: firing rate of the i^{th} unit
$\xi_i(t)$: noise

Output

$$y_j(t) = \sum_{i=1}^{N} W_{ji}^{out} u_i(t)$$

Cost function

$$E = \frac{1}{MTN_{out}} \sum_{m,t,j=1}^{M,T,N_{out}} (y_j(t,m) - y_j^{target}(t,m))^2$$

Regularization

1. $R_{L2} = \frac{1}{NN_{in}} \sum_{i,j=1}^{N,N_{in}} (W_{ij}^{in})^2 + \sum_{i,j=1}^{N,N_{out}} (W_{ij}^{out})^2$
2. $R_{FR} = \frac{1}{NTM} \sum_{i,t,m=1}^{N,T,M} u_i(t,m)^2$

3.2 Data

Two datasets are generated, one in a rectangular arena and other in a circular arena, to understand if there is any dependence of firing patterns on the shape of arena. Each dataset consists of 100000 paths with 10-20 steps in every path starting from the origin (0,0). Step size varies randomly between 0-10 units in a rectangular arena of size 100x100 units (or in a circular arena of radius 50 units). Every path generated is a random walk in 2D space within the boundary. An example path is shown in Fig 2.

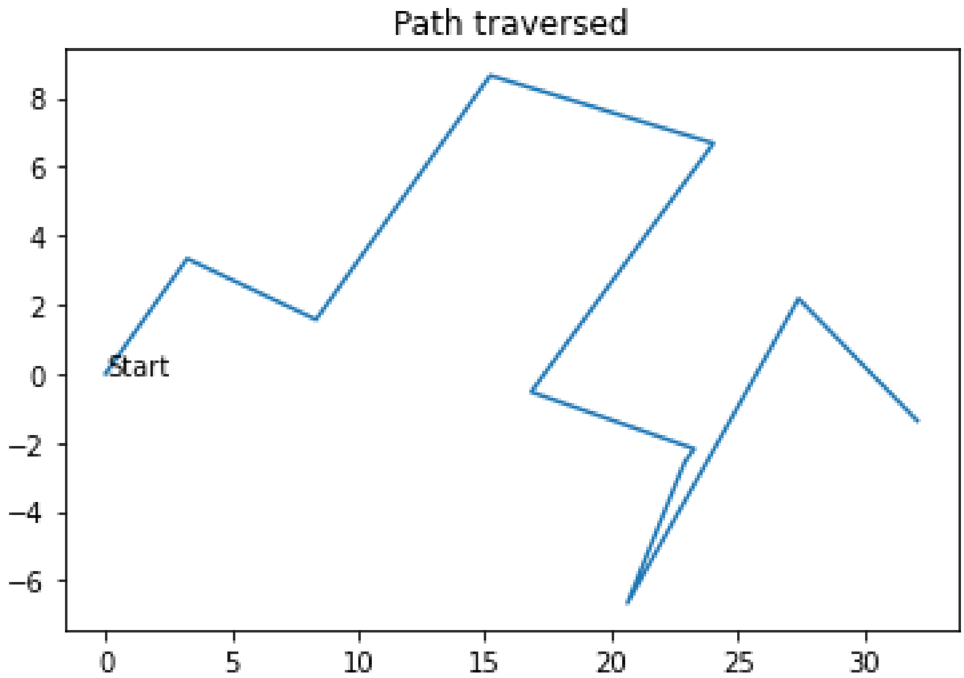


Fig 2. An example path from the recatngular arena dataset

This dataset is rate-coded using input layer LIF neurons of the SNN model. In rate-coding, the analog input value is manifested as firing rate (number of spikes in unit time) of corresponding LIF neuron.

3.3 LIF Neuron

The neuron model used to build spiking neural network is Leaky-Integrate and Fire (LIF) neuron. These LIF neurons replace the nodes in model graph of RNN in Fig 1(b). The following equation describe the time evolution of state of LIF neurons –

$$\frac{dV}{dt} = \frac{1}{C}[-g(V - E_l) + I]$$

where,

V : membrane potential

I : current into the neuron

E_l : resting potential

g, C : conductance and capacitance of the neuron

τ_{ref} : refractory period

$$rate(I) = \frac{1}{\tau_{ref} - \tau_{RC} ln(1 - \frac{g(V_{th} - E_l)}{I})}$$

for $I > g(V - th - E_l)$

3.4 Synapse

Synapses form the connection between neurons. Synapses replace the edges in the model graph of RNN in Fig 1(b). Following equation describes the current through synapse –

$$I(t) = wI_0(e^{-\frac{t-t_s}{\tau_m}} - e^{-\frac{t-t_s}{\tau_s}})$$

$$\tau_m > \tau_s$$

3.5 The SNN Model

Now the SNN model is built using the LIF neurons and synapses described above. Rate-coded input is passed to network of LIF neurons and synapses arranged as per model graph in Fig 1(b).

Since

$$x_i(t+1) = x_i(t) + \tau \frac{dx_i(t)}{dt}$$

State update equation can be re-written as

$$x_i(t+1) = \sum_{j=1}^{N} W_{ij}^{rec} u_j(t) + \sum_{k=1}^{N_{in}} W_{ik}^{in} I_k(t) + b_i + \xi_i(t)$$

1. Weight multiplication is implemented as synaptic weights and addition of signals is executed by combining neurons while input to next layer.
2. Rate coding of analog inputs (v, θ) to spikes is done by applying proportional current to input layer LIF neurons.
3. The analog outputs (x, y) readout from spikes is calculated from firing rates of output layer LIF neurons.

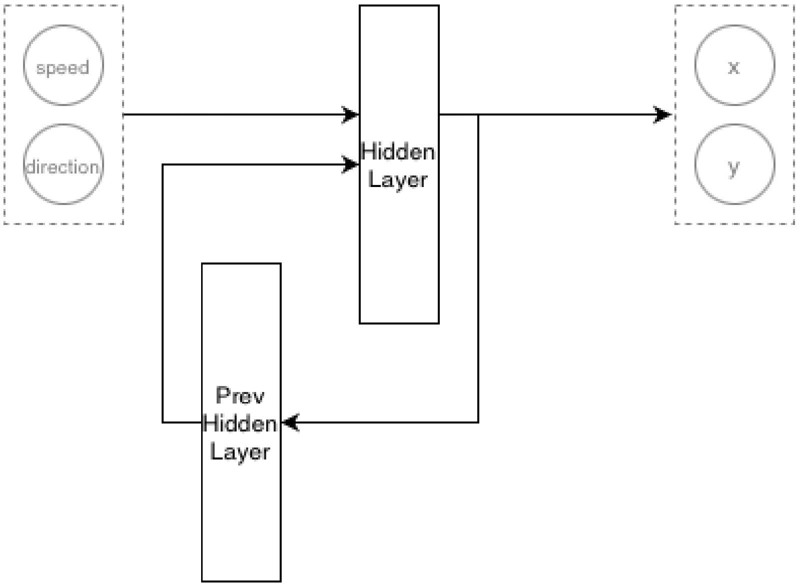


Fig 3. SNN architecture

3.6 Recurrent connections

To implement recurrent connections an extra set of hidden layer (100 LIF neurons) is maintained. These neurons operate with a time delay T, where T is the simulation window for one step. These 'delayed' hidden layer LIF neurons along with input layer LIF neurons together drive the hidden layer LIF neurons.

3.7 Tanh activation

Since the entire implementation needs to be end-to-end spiking, the activation function must be incorporated in the model of neuron chosen. The parameters of the neurons should be set such that the firing rate vs input response resembles tanh activation.

To replicate the positive tanh activation, set the refractory period (T_{ref}) for the neuron. Then the number of spikes in a given time window (T) will grow linearly from zero input and saturate to T/T_{ref} for larger positive inputs.

But more work needs to be done in this front to understand how to model the negative tanh activation.

Chapter 4

Summary & Future Work

Described a theoretical construct of spiking neural network that will perform spatial localisation. Methods to implement recurrent connections and tanh activation function in spiking neural network have been proposed. Implementation of these methods is yet to be completed. This would require more enquiry into normalising inputs and dealing with negative tanh activation. This will complete the construction of spatial localisation system for the environment described in section 3.2.

The next step will be to implement the SNN on Loihi. After that, training of SNN in spiking domain would be taken up. This SNN will then be generalised to larger and more diverse environments. Finally, this work shall integrate with C.Elegans work [10] to complete a navigation system and perform some basic navigation tasks.

Chapter 5

References

1. Moser EI, Kropff E, Moser MB. Place cells, grid cells, and the brain's spatial representation system. Annu Rev Neurosci. 2008;31:69-89. doi: 10.1146/annurev.neuro.31.061307.090723. PMID: 18284371.
2. Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Xie, Y.; and Shi, L. 2019b. Direct training for spiking neural networks: Faster, larger, better. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 1311–1318.
3. Taherkhani, A.; Belatreche, A.; Li, Y.; Cosma, G.; Maguire, L.; and Mcginnity, T. 2019. A review of learning in biolog- ically plausible spiking neural networks. Neural Networks 122. doi:10.1016/j.neunet.2019.09.03
4. Wu, J.; Chua, Y.; Zhang, M.; Li, G.; Li, H.; and Tan, K. C. 2019a. A Tandem Learning Rule for Effective Training and Rapid Inference of Deep Spiking Neural Networks.
5. Z. Yan, J. Zhou, and W.-F. Wong, "Near Lossless Transfer Learning for Spiking Neural Networks", AAAI, vol. 35, no. 12, pp. 10577 10584, May 2021.
6. Christopher J. Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. ICLR 2018
7. Banino, A., Barry, C., Uria, B. et al. Vector-based navigation using grid-like representations in artificial agents. Nature 557, 429–433 (2018). https://doi.org/10.1038/s41586-018-0102-6
8. Hawkins Jeff, Lewis Marcus, Klukas Mirko, Purdy Scott, Ahmad Subutai. A Framework for Intelligence and Cortical Function Based on Grid Cells in the Neocortex. Frontiers in Neural Circuits 2019. DOI=10.3389/fncir.2018.00121
9. Tale L Bjerknes, Edvard I Moser, and May-Britt Moser. Representation of geometric borders in the developing rat. Neuron, 82(1):71–78, 2014.
10. Shukla S., Pathak R., Saraswat V., Ganguly U. (2020) Adaptive Chemotaxis for Improved Contour Tracking Using Spiking Neural Networks. In: Farkaš I., Masulli P., Wermter S. (eds) Artificial Neural Networks and Machine Learning – ICANN 2020. ICANN 2020. Lecture Notes in Computer Science, vol 12397. Springer, Cham. https://doi.org/10.1007/978-3-03061616-8_55

Chapter 6

Other Resources

8.1 Repository

1. BTP-1

   https://drive.google.com/drive/folders/11glL_OFT1oWZ6gHkDTL8omV1aIp108LX

2. Code

   https://colab.research.google.com/drive/11mAVNSjsQKbRqneGliAR9mr3nqjo8bVN

8.2 Blogposts

1. Spatial Localisation using SNNs

   http://iitbnf.iitb.ac.in/udayanresearch/index.php/2021/06/30/spatial-localisation-using-snns/

Chapter 7

Acknowledgment

I sincerely appreciate the efforts put in by Prof. Udayan Ganguly, Vivek Saraswat, MeLoDe research group, and the Department of Electrical Engineering for making stage-2 Bachelor Thesis Project (BTP) possible by working remotely during Covid-19 pandemic.

This short stint as an undergraduate researcher was a steep learning curve.

I thank my parents for their unconditional love and emotional support that helped me fight my severe covid infection in May 2021. I also thank my mentor and professor for understanding my situation. I would like to dedicate this work to the close family members I have lost in the second wave of covid pandemic in India.