

**AUTO ENCODER BASED ON TEMPORAL CODING IN A  
SPIKING NEURAL NETWORK**

Kaluhath Dhanushka Niroshan De Abrew

179313A

M.Sc. in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

April 2020

**AUTO ENCODER BASED ON TEMPORAL CODING IN A  
SPIKING NEURAL NETWORK**

Kaluhath Dhanushka Niroshan De Abrew

179313A

Thesis/Dissertation submitted in partial fulfillment of the requirements for the Degree of  
Master of Science in Computer Science and Engineering

Department of Computer Science and Engineering

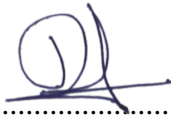
University of Moratuwa

Sri Lanka

April 2020

## DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works.



.....

K.D.N. De Abrew

29/05/2020

.....

Date

I certify that the declaration above by the candidate is true to the best of my knowledge and that this project report is acceptable for evaluation for the CS6997 MSc Research Project.

.....

Dr. Charith Chitraranjan

.....

Date

## **ABSTRACT**

Auto Encoders using Artificial Neural Networks have achieved a high level of regeneration accuracy whereas Auto Encoders using Spiking Neural Networks are still in their early stage and only a few SNN Auto Encoders have been introduced but with lesser accuracies compared to ANN Auto Encoders. Using SNNs for Auto Encoders is desired as SNNs are one step closer to understand the communication and processing in biological neural networks. Sparse discrete events known as Spikes make SNNs energy efficient especially when implemented using Neuro-morphic hardware and Temporal coding scheme with the mapping of ‘input value to time of the first generated spike’ makes it even more efficient in terms of power consumption and time to generate an output where power consumption and time to encode/decode are the key metrics.

However, the direct application of gradient descent methods is not possible for SNN as the activation functions are non-differentiable. Training an Auto Encoder requires a way to adjusting the network parameters so that the reconstruction loss is minimized. Due to the lack of such training models for SNNs especially with multiple hidden layers, it is a challenging task to implement an Auto Encoder using SNN. In this research, models enable such learning, were analyzed with the aim of selecting a promising model. Based on the selected model for adjust the synaptic weights of the network, an SNN Auto Encoder model is developed which allows the user to configure the network structure and number of neurons in each layer to achieve the desired compression ratio. Considering the demonstrated reconstruction accuracy and convergence rate of the SNN Auto Encoder, it can be concluded that the introduced model is one of the first models which enables to use multilayer Auto Encoder using Spiking Neural Networks.

## **ACKNOWLEDGEMENTS**

I wish to express my sincere gratitude to my supervisor, Dr. Charith Chitraranjan, for his guidance and invaluable assistance by providing me useful knowledge materials, advice, and supervision throughout this research work. His expertise and encouraging guidance enabled me to complete my work successfully.

I am indebted to my parents for their love and untired support throughout my life.

# TABLE OF CONTENT

DECLARATION .....	i
ACKNOWLEDGEMENTS .....	ii
ABSTRACT .....	iii
TABLE OF CONTENT .....	iv
LIST OF FIGURES .....	vii
LIST OF TABLES .....	viii
LIST OF ABBREVIATIONS .....	ix
CHAPTER 1 INTRODUCTION .....	1
1.1    Auto Encoder .....	2
1.2    Learning in Auto Encoder.....	3
1.3    Auto Encoders using Spiking neural networks .....	3
1.4    Neuron Model for SNN .....	4
1.4.1    Leaky Integrate and Fire .....	4
1.4.2    Integrate and Fire (IF) (Non - leaky) .....	6
1.4.3    Spike Response Model (SRM).....	7
1.5    Neural coding.....	8
1.5.1    Rate Coding .....	8
1.5.2    Temporal Coding .....	8
1.6    Motivation.....	9
1.7    Objective.....	10
2    CHAPTER 2 LITERATURE REVIEW .....	11
2.1    Spike Time Dependent Plasticity (STDP) for learning.....	12
2.1.1    Spike Time Dependent Plasticity (STDP) .....	12

2.1.2	Learning in SNN using STDP.....	13
2.2	Learning in SNN using temporal coding .....	14
2.2.1	Spike Prop.....	15
2.2.2	ReSuMe (Remote Supervised Method) .....	17
2.2.3	BP-STDP .....	19
2.2.4	Using back propagation for learning in SNN using temporal coding.....	21
2.3	Comparison of SNN Models.....	25
3	CHAPTER 3 METHODOLOGY AND IMPLEMENTATION .....	27
3.1	Using a SNN Model for Auto Encoder.....	28
3.2	Overview.....	29
3.3	Convert input/output values to spike times.....	33
3.4	Structure of the network .....	34
3.5	Error function for Auto Encoder.....	34
3.6	Python Implementation.....	35
4	CHAPTER 4 EXPERIMENTAL ANALYSIS & MODEL EVALUATION .....	39
4.1	Input Dataset.....	40
4.2	Performance Comparison with ANN AutoEncoder developed using Keras .....	41
4.3	Performance of Auto Encoder vs network structure .....	48
4.3.1	Performance vs Number of Hidden layers.....	49
4.3.2	Performance vs compression ratio .....	51
4.4	Hyper parameter analysis.....	53
4.4.1	Mini Batch Size .....	54
4.4.2	Learning Rate.....	55
4.4.3	Frobenius norm threshold .....	56
4.5	Hidden Layer Spike Time Analysis.....	56

4.6	Hardware information.....	59
5	CHAPTER 5 CONCLUSION.....	60
5.1	Contribution.....	61
5.2	Limitations.....	61
5.3	Future work.....	62
	REFERENCES .....	64
	APPENDIX.....	67



## LIST OF FIGURES

Figure 1.1 : Auto Encoder.....	2
Figure 1.2 : LIF Neuron modeled as a RC circuit.....	5
Figure 1.3 : A: Potential increase in LIF and IF models with time for a constant input current, B: Firing rates against input current for LIF and IF models.....	6
Figure 1.4 : Kernel functions of SRM corresponding to each phase of membrane potential change when an output spike is generated.....	7
Figure 2.1: Synaptic weight change against time gap between pre/post synaptic neuron spikes	13
Figure 2.2: Single connection between two neurons consist of different delays and weights.....	16
Figure 2.3: Weight adjustment with Spike trains. $S_i(t)$ – Input, $S_o(t)$ – Actual Output, $S_d(t)$ – Desired Output.....	18
Figure 2.4: Membrane potential and summation of input synaptic current of a neuron changes as per the input spike times.....	24
Figure 3.1 : Task wise flow of Auto Encoder model.....	32
Figure 3.2: Calculating derivative of output spike time w.r.t. synaptic weight.....	36
Figure 4.1: Training error curve Vs. Number of layers.....	50
Figure 4.2 : Training Error curve vs. Compression ratio 1.....	52
Figure 4.3 : Training Error curve vs. Compression ratio 2.....	53
Figure 4.4 : Training Error curve with Mini Batch size.....	54
Figure 4.5 : Training Error curve with Learning Rate.....	55
Figure 4.6 : Training Error curve with Frobenius norm threshold.....	56
Figure 4.7: Reconstructions for sample 10 images:.....	57
Figure 4.8 : Heat map of Hidden layer spike time for different input images.....	58

## LIST OF TABLES

Table 2.1: Comparison of SNN Models.....	26
Table 3.1 : Hyper Parameters of the Auto Encoder Model.....	30
Table 3.2 Input value to spike time conversion .....	33
Table 4.1: [144 x 24 x 144] Reconstruction comparison .....	42
Table 4.2 : [144 x 48 x 24 x 48 x144] Reconstruction comparison.....	43
Table 4.3: [144 x 48 x 24 x 24 x 48 x144] Reconstruction comparison.....	44
Table 4.4 : [144 x 48 x 24 x 16 x 24 x 48 x144] Reconstruction comparison.....	46
Table 4.5 : Reconstruction Error comparison of SNN AutoEncoder vs. Standard AutoEncoder (Optimizer - 'AdaDelata').....	47
Table 4.6 : Reconstruction Error comparison of SNN AutoEncoder vs. Standard AutoEncoder (Optimizer -'SGD').....	48
Table 4.7 : Reconstruction Error with number of layers.....	49
Table 4.8 : Network Structure vs. Training Error .....	50
Table 4.9 : Compression ratio vs Training error .....	51
Table 4.10 Training error after 100 epochs for different mini batch sizes .....	54
Table 4.11 : Avg. Spike generation time for hidden layer neurons .....	57
Table 4.12 : Hardware information.....	59

## **LIST OF ABBREVIATIONS**

<b>SNN</b>	Spiking Neural Network
<b>ANN</b>	Artificial Neural Network
<b>STDP</b>	Spike Time Dependent Plasticity
<b>IF</b>	Integrate and Fire
<b>LIF</b>	Leaky Integrate and Fire
<b>SRM</b>	Spike Response Model
<b>PSP</b>	Postsynaptic Potential
<b>SGD</b>	Stochastic Gradient Decent
<b>MSE</b>	Mean Squared Error
<b>PSNR</b>	Peak Signal to Noise Ratio