

Research Results

Designing of Future State Estimation of Finite State Machine By Deep Learning Algorithm

Akash Kaushal¹, Prof. Nishi Pandey², Prof. Abhishek Agwekar³

¹M.Tech. Scholar, Truba Institute of Engineering & Information Technology, Bhopal, (M.P.), INDIA

²Assistant Professor, Truba Institute of Engineering & Information Technology, Bhopal, (M.P.), INDIA

³HOD, EC, Truba Institute of Engineering & Information Technology, Bhopal, (M.P.), INDIA

ABSTRACT

Digital circuits lie at the heart of almost all modern circuits working on digital technology. The design and testing of digital circuits and systems has undergone a paradigm shift in recent times with the advent of artificial intelligence. It was customary to design digital circuits using the knowledge of the internal circuitry or the input-output mapping or truth table of the circuit. Some modern applications though need interactive behaviour from systems which finally boils down to the interactive design of digital circuits. Some examples can be the internal circuitry of interactive gaming consoles, vending machines, sequence detectors, human machine interfaces (HMI) etc. The aim is to design an Artificial Intelligence based system which would get trained using the limited input-output mapping of the circuit and be able to predict the future behaviour of the circuit. This would make digital circuits interactive. In the present work, Artificial Neural Networks have been used for predicting the behaviour of digital circuits. In the present work, different digital circuits have been simulated such as the AND Gate, Half Adder, JK Flip Flop and Sequence Detecting Finite State Machines using AI based systems.

KEYWORDS

Artificial Intelligence, Artificial Neural Networks, Finite State Machine, Levenberg Marquardt Algorithm, Mean Square Error

1. INTRODUCTION

DIGITAL CIRCUIT DESIGN

Digital circuits can be thought of as circuits which accept only two levels of voltage and give back the output in the same two levels of voltage. These two levels of voltage are customarily termed as 0 and 1. The basic building blocks of digital circuits are Gates. Gates can be thought of as entities which take logic 0 or 1 as input combinations and yield an output depending on the logical operation on the inputs. Needless to say, the output is also in the form of 0 and 1. Logic gates serve as the basic building block for digital circuits. Digital circuits can be broadly classified as:

- 1) Combinational Circuits.
- 2) Sequential Circuits.

Combinational Circuits: Combinational circuits are those circuits in which the present output depends only on the present inputs and not on past inputs. Such circuits are said to possess No Memory.

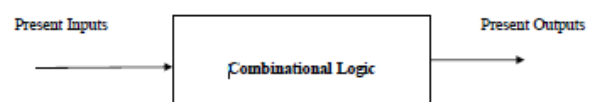


Figure 1.1: Combinational Circuit

Sequential Circuits:

Sequential circuit works on a clock cycle which may be synchronous or asynchronous. The figure shows a basic diagram of sequential circuits. Sequential circuits use current inputs and previous inputs by storing the information and putting back into the circuit on the next clock cycle.

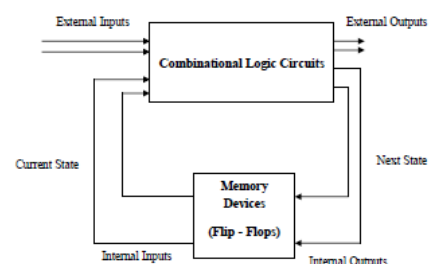


Figure 1.2: Sequential Circuit

2. LITERATURE REVIEW

Li et al in this paper “Towards Acceleration of Deep Convolutional Neural Networks using Stochastic Computing” the authors proposed a Convolutional Neural Network Mechanism for the predictive modelling of finite state machines used for stochastic computing. The transfer function used for the analysis of the data was the convolution integral and the output of each neuron was computed based on the Convolutional integral. The simulations were run for several finite state machine from input size of 8 to that of 64. It has been shown that the proposed system achieves an error of 0.9 for a finite state machine of 64 inputs.

Song et.al. in the paper “Design of logic gates using spiking neural P systems with homogeneous neurons and astrocytes-like control” proposed an approach to investigate the spiking neural P systems, i.e. SN P systems, with astrocyte-like control which were proven to have “Turing completeness” as computing models. In biological nervous systems, the operation of interacting neurons depends largely on the regulation from astrocytes. In this research, it is proposed and established SN P systems with astrocyte-like control to emulate logic AND, OR, NOT, NOR, XOR and NAND gates. The obtained SN P systems are simple and homogeneous, which infers that each neuron in the systems has the same unique spiking rule. The systems proposed in this work provide theoretical models to construct neural-like logic gates with universal information processing units. It will be beneficial if parallel hardware, such as GPU, can be utilized and harnessed to realize the neural-like digital logic gates and logic circuits.

Eisner et al. in the paper “Weighting Finite-State Transductions with Neural Context” proposed a new approach to deep learning to tasks such as morphological re inflection, which stochastically edits one string to get another. The traditional architecture is kept as such, and a stack of bidirectional LSTMs reads the input string from left-to-right and right-to-left, in order to summarize the input context in which a transducer arc is applied. These learned characteristics and traits are intertwined with the transducer to define a probability distribution over aligned output strings, in the form of a weighted finite-state automaton. This lessens hand-engineering of features, allows learned features to examine unbounded context in the input string, and still permits exact inference through dynamic programming.

Kayri et al. in the paper “Predictive Abilities of Bayesian Regularization and Levenberg- Marquardt Algorithms in Artificial Neural Networks: A Comparative Empirical Study on Social Data” demonstrated the objective of the study which was to compare the predictive ability of Bayesian regularization with Levenberg-Marquardt Artificial Neural Networks. Since complex models are penalized in accordance with the Bayesian approach, this approach explores complex architecture smoothly. All in all, the model with two-neurons is the best architecture because of the highest importance level of predictors.

Gross et al. in the paper “VLSI Implementation of Deep Neural Network Using Integral Stochastic Computing” suggested that an integer form of stochastic computation

and introduce some elementary circuits. An efficient implementation of a DNN based on integral stochastic computing is put forward. An efficient stochastic implementation of a deep belief network is proposed using integral SC. Going by the simulation and implementation results, it shows that the proposed design reduces the area occupation by 66% and the latency by 84% with respect to the state of the art. It also indicates that the proposed design consumes 21% less energy than its binary radix counterpart. Moreover, the proposed architectures can save up to 33% energy consumption w.r.t. the binary radix implementation by using quasi-synchronous implementation without compromising on performance and keeping the efficiency intact.

3. PROBLEM FORMULATION

Conventionally, all digital circuits are designed by the following two techniques:

1) Prior knowledge of the internal circuitry. For example consider the circuit diagram of a half adder circuit

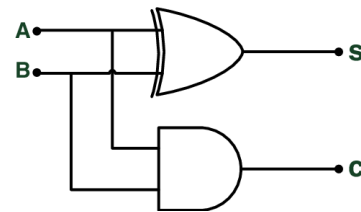


Figure 3.1 Internal Circuitry of a Half Adder

2) Input - Output mapping of the circuit also called the Truth- Table.

Truth Table			
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 3.1 Truth Table of a Half Adder

In both the cases, either the internal circuit or the input-output mapping of the circuit was known which meant circuit design would be possible following standard design steps. The major challenge arrives when neither the internal circuit nor the truth table of the circuit is known but only certain input output mapping pairs are known which tend to change. In such cases, it becomes extremely complex to predict the behaviour of the circuit. Such applications are needed in Human Machine Interfaces (HMI) and interactive systems.

The design of such systems requires the need of Artificial Intelligence. There are several AI based algorithms that can be utilized for the design and prediction of digital circuits. We mainly focus on systems with feedback or systems with **Back Propagation**.

The evaluation of the performance of the designed AI based system generally is computed on the basis of the following performance metrics:

1) Mean Square Error (MSE): It gives an idea about the difference in predicted output and actual output (ideal output or target)

It's mathematically defined as:

$$MSE = \frac{1}{n} \sum_{i=1}^n e_i^2$$

Here e=predicted output- actual output

2) The execution time: The lesser the execution time, the faster the system.

3) Regression: It is a parameter which gives an idea about the coherence in the results. It has an ideal value of 1. The closer the value is to 1, the more coherent is the result.

4) Iterations: It the system attains a tolerable error margin in lesser iterations; it points to the fact that the system exhibits lesser time and system complexity.

4.PROPOSED METHODOLOGY

ARTIFICIAL NEURAL NETWORKS (ANN)

Artificial Neural Networks (ANN) are computing systems or technique that are inspired by the learning architecture of human brain to discover the relations between the input and target variables of a system. Human brain consists of a large set of structural constituents, known as neurons, which form a well-connected network to respond to an input signal to perform all its computations / calculations in a certain complex task such as image and voice recognition task and they do this with incredible speed and accuracy. Neurons are simple processing units, which has the ability to store experimental data and which work as parallel distributed processor. The speed of human brain is several thousand time faster than traditional computer because in brain unlike traditional computer as whole information is not passed from neuron to neuron they are rather encoded in the neuron network. This is reason why neural network is also named as connectionism.

A biological model of neuron is basically comprised of dendrites, a cell body or soma, and an axon. The cell body, also called the soma, holds the nucleus of neurons. The dendrites are the branches that are linked to the cell body and stretched in space around the cell body to receive signals from neighbouring neurons. The axon works as a transmitter of the neuron. It sends signals to neighbouring neurons. The synapse or synaptic terminal are the connection between the axon of one neuron and the dendrites of neighbouring neuron, which is the communication link in between the two neurons.

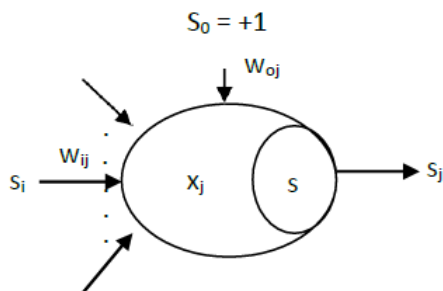


Figure 4.1 Mathematical equivalent of Neuron.

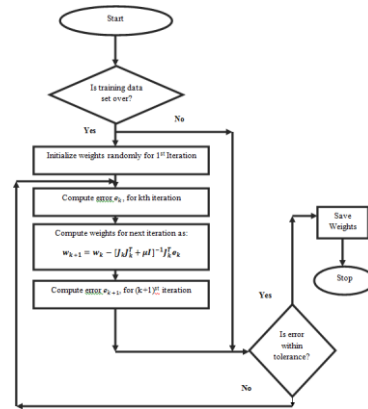


Figure 4.2: Block diagram for training using Levenberg-Marquardt algorithm.

5. RESULTS AND DISCUSSIONS

Before proceeding towards complex FSMs, we start our discussion with the design and implementation of logic gates. Subsequently, we move on to the design of combinational circuits, sequential circuits and finally sequence detectors.

In all the above examples, the customary rule of taking 70% of the data for training and remaining 30% of the data for testing and validation has been followed. The circuits simulated are:

- 2) HALF ADDER using LM Algorithm
- 3) JK FLIP FLOP using LM Algorithm
- 4) A Jordan network implementation of the Half Adder
- 5) Sequence detector design using LM, BR and SCG algorithms. The performance indices considered here are:

- 1) Mean Square Error (MSE)
- 2) Epochs or execution time
- 3) Iterations
- 4) Error Histograms for Graphical Error Representation
- 5) Regression

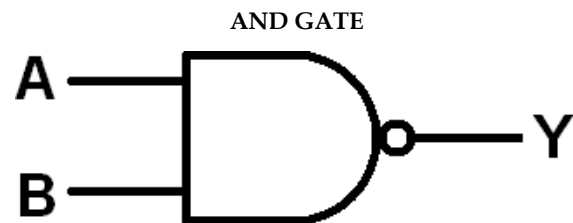


Figure 5.1: Logic Diagram of AND Gate

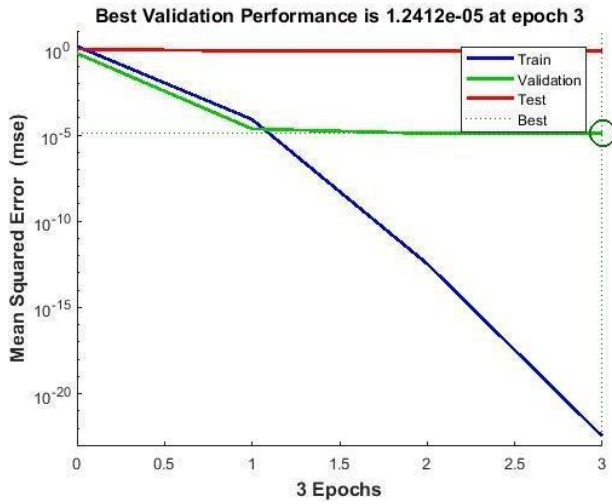


Figure 5.2: MSE analysis for AND Gate

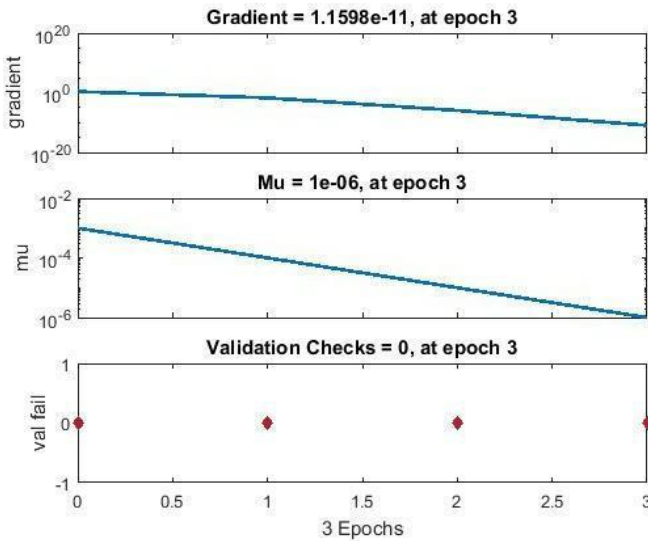


Figure 5.3: Training States for AND Gate



Figure 5.4: Neural Network Design for AND Gate

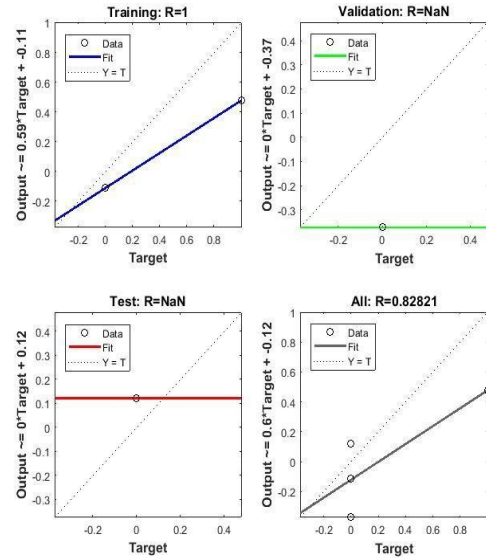


Figure 5.5: Regression Analysis for AND Gate

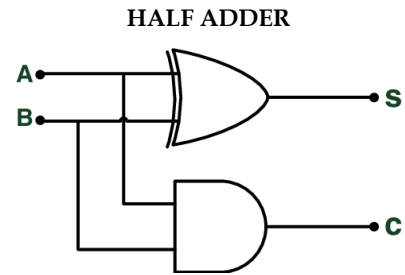


Figure 5.6: Logic Diagram of Half Adder

Truth Table			
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Figure 5.7: Truth Table for Half Adder

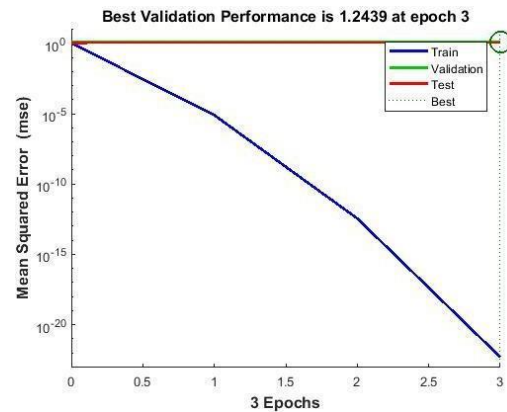


Figure 5.8: MSE analysis for Half Adder

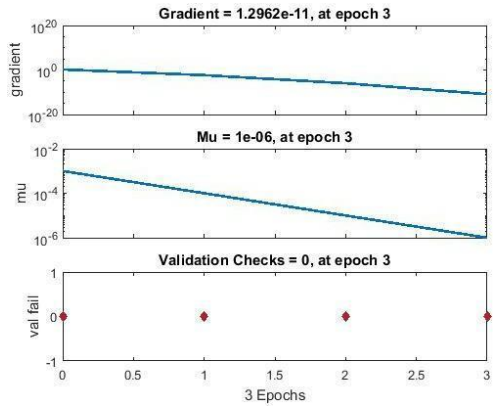


Figure 5.9: Training States for Half Adder

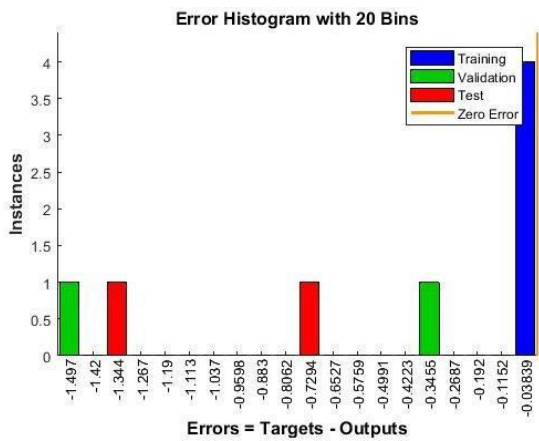


Figure 5.10: Error Histogram for Half Adder



Figure 5.11: Neural Network Design for Half Adder

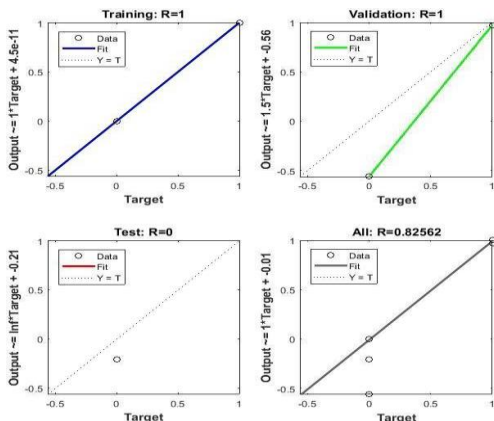


Figure 5.12: Regression Analysis for Half Adder

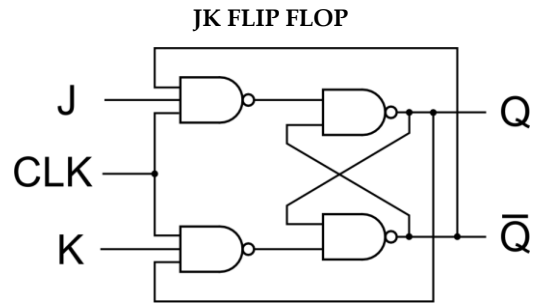


Figure 5.13: Logic Diagram of JK Flip Flop

Q	J	K	Q(t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Figure 5.14: Truth Table for JK Flip Flop

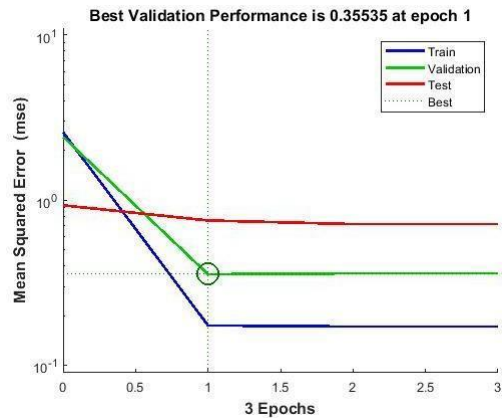


Figure 5.15: MSE analysis for JK Flip Flop

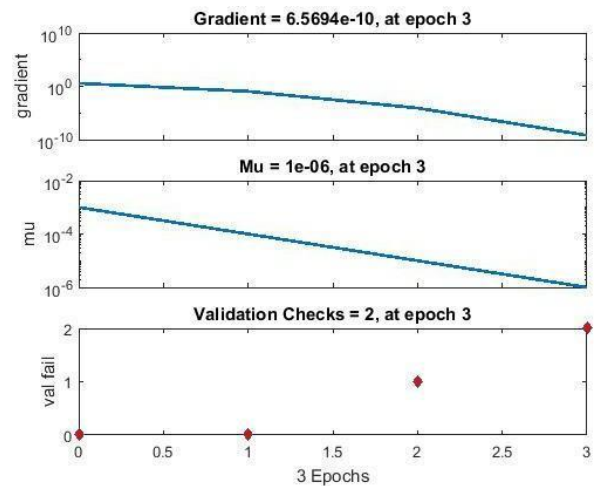


Figure 5.16: Training States for JK Flip Flop

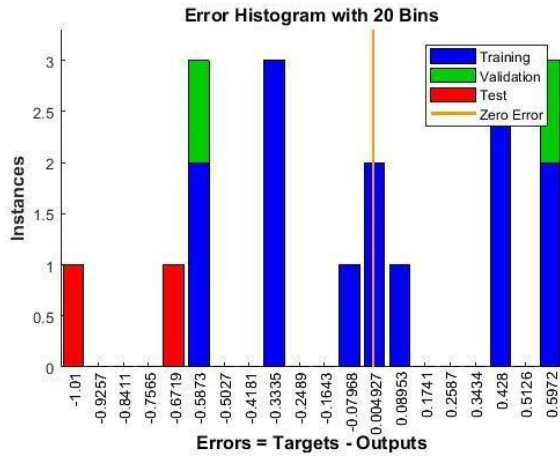


Figure 5.17: Error Histogram for JK Flip Flop



Figure 5.18: Neural Network Design for JK Flip Flop

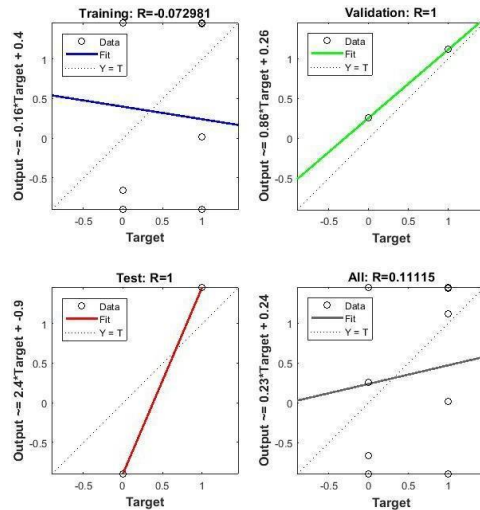


Figure 5.19: Regression Analysis for JK Flip Flop

5. CONCLUSION AND FUTURE SCOPE

It can be concluded from the previous discussions that the design and prediction of complex digital circuits is possible using Artificial Neural Networks (ANN). The benefit of such a system lies in the fact that there is no need of knowing the internal circuitry or the complete input output mapping of the digital system to predict its performance. The behaviour of the circuit can be predicted by the ANN based system once the system has been trained. This approach finds extensive application in interactive gaming, Human Machine Interfaces (HMI), hardware level cryptography. In the present work, different digital circuits

have been simulated such as the AND Gate, Half Adder, JK Flip Flop and Sequence Detector using AI based systems. As a standard convention, 70% of the total data has been employed for training, and the rest of the 30% has been employed for testing and validation. It has been found that the behaviour of these circuits can be predicted with ease using Artificial Neural Networks. A comparative error analysis clearly shows that the proposed system outperforms the previous system. It can be seen that as the number of inputs increases, the errors also increases due to the complexity of the system. The proposed system attains a maximum error of 0.087 is is relatively 10 time lesser than the error performance of the base paper [1]

6. FUTURE SCOPE

The future scope can be thought of as in improving and optimizing the present algorithms to attain even better performance indices such as lesser number of iterations, better regression and lesser MSE. Also the design of cascaded algorithmic structures can be designed to while retaining the merits of each of the algorithms. Also, more complex circuits can be tested in real time applications.

REFERENCES

- [1] Ji Li1, Ao Ren2, Zhe Li, Caiwen Ding, Bo Yuan, Qinru Qiu2 and Yanzhi Wang "Finite State Machine SoC Design Using Deep Neural Networks for Future State Estimation," IEEE 2022.
- [2] Design of logic gates using spiking neural P systems with homogeneous neurons and astrocytes-like control Tao Song, Pan Zhen, M.L. Dennis Wong, Xun Wang, Elsevier 2021
- [3] Weighting Finite-State Transduction swith Neural Context Pushpendre Rastogi and Ryan Cotterell and Jason Eisner, NAACL-HLT Proceedings 2020
- [4] Predictive Abilities of Bayesian Regularization and Levenberg-Marquardt Algorithms in Artificial Neural Networks: A Comparative Empirical Study on Social Data Murat Kayri, MDPI, 2019
- [5] VLSI Implementation of Deep Neural Network Using Integral Stochastic Computing ArashArdakani, Student Member, IEEE, François Leduc-Primeau, NaoyaOnizawa, Member, IEEE, Takahiro Hanyu, Senior Member, IEEE and Warren J. Gross, Senior Member, IEEE 2018
- [6] State assignment for area minimization of sequential circuits based on cuckoo search optimization q Aiman H. El-Maleh, Sadiq M. Sait, AbubakarBala Elsevier 2017
- [7] Jordan Neural Network for Modelling and Predictive Control of Dynamic Systems AntoniWysocki and MaciejŁawry nczuk, IEEE 2016
- [8] On adaptive experiments for nondeterministic finite state machines Natalia Kushik · Khaled El-Fakih · Nina Yevtushenko · Ana R. Cavalli, Springer 2015
- [9] Deep Learning in Neural Networks: An Overview Technical Report IDSIA-03-14 / arXiv:1404.7828 v4 [cs.NE] (88 pages, 888 references) JürgenSchmidhuber The Swiss AI Lab IDSIA IstitutoDalleMolle di Studisull'Intelligenza Artificiale, IEEE 2014
- [10] A Synthesis Flow for Sequential Reversible Circuits Mathias Soeken Robert Wille Christian Otterstedt Rolf Drechsler, IEEE 2014
- [11] Back propagation and Levenberg-Marquardt Algorithm for Training Finite Element Neural Network Arnold Reynaldi, Samuel Lukas, Helena Margaretha, IEEE 2012.

-
- [12] RH Byrd, GM Chin, W Neveitt, J Nocedal, "On the use of stochastic hessian information in optimization methods for machine learning", SIAM Journal of Optimization 2011
 - [13] J Rhinelander, XP Liu, "Stochastic subset selection for learning with kernel machines", IEEE 2011.
 - [14] J Bae, LS Giraldo, P Chhatbar, J Francis, "Stochastic kernel temporal difference for reinforcement learning", IEEE 2011
 - [15] F Orabona, L Jie, B Caputo, "Online-batch strongly convex multi kernel learning", IEEE 2010
 - [16] L Bottou, "Large-scale machine learning with stochastic gradient descent", Springer 2010.
 - [17] S Yi, D Wierstra, T Schaul, J Schmidhuber, "Stochastic search using the natural gradient", ACM 2009
 - [18] P Vamplew, R Dazeley, E Barker, A Kelarev, "Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks", Springer 2009
 - [19] J Ye, JH Chow, J Chen, Z Zheng, "Stochastic gradient boosted distributed decision trees", ACM 2009
 - [20] Design and Implementation of Parallel Hierarchical Finite State Machines Valery Sklyarov, IouliiaSkliarova, IEEE Explore 2008