FPGA Based Accelerators of Deep Learning Networks for Learning and Classification

** Subhabaha Pal*
*** Ravikanth Paturi*

Abstract

A recent trend is to execute computationally intensive algorithms (or work flows) for business analytics using cloud environments which provide machine learning hardware support in the form of GPUs and TPUs. Businesses obtain their data at the sensor level and then perform algorithmic operations on the data via these cloud services. As a result, there can be high input/output data latency which tends to slow down productivity. This thesis work will explore the topic of executing computationally complex algorithms, such as the Convolutional Neural Network (CNN), Recurrent Neural Network(RNN) and Spike Neural Network (SNN), at the sensor level through the use of FPGAs (Field Programmable Gate Arrays) as an alternative to cloud-bound GPU and TPU services.

Keywords : **Deep learning networks, FPGA**

NOMENCLATURE

FGPA Field Programmable Gate Array
MNIST Modified National Institute of Standards and Technology Database
GPU Graphics Processing Unit
TPU Tensor Processing Unit
ELM Educational Leadership and Management
ASM American Society for Microbiology

I. INTRODUCTION

Neural networks are a group of training networks that resemble human brain and can be used in most of prediction tasks. Psychologists and neurobiologists are the first to do research in the area of neural networks. Neural networks are a group of connected layers with each layer consisting of many neurons. Each connection has weights associated with it. The weights are adjusted during training phase and the class labels are matched as near as possible by changing weights. The presence of connections between different layers has also earned the name as connectionist learning. In general, the training by adjusting weights of neural network takes a long time and must be worked out only in the applications that it matches well. Parameters such as network topology or structure must be experimentally determined. Also, the main setback for a neural network is its poor interpretability. We can never understand the relation of data to the structure or number of units and weights. All these drawbacks initially made neural networks to be the last choice for researchers and came to use in case of non-linear networks. On the other hand, they have the benefit in terms of classifying patterns on which they have not been trained and can work even in case of noisy data. Ideally, we use in case of continuous variables but can be applied in case of categorical variables. The applications of neural networks range from handwritten digits recognition to biological and laboratory medicine. The major breakthrough is in the application of reading English text. We run neural networks in parallel that helps to speed up computation process. Among neural networks, back propagation has gained reputed name in the analysis and applications

Manuscript Received : August 6, 2020 ; Revised : August 20, 2020 ; Accepted : August 22, 2020. Date of Publication : September 5, 2020.
* S. Pal is Associate Professor with TAPMI School of Business, Manipal University Jaipur, Dehmi Kalan, Jaipur-Ajmer Express Highway, Jaipur, Rajasthan - 303 007, India. (email : subhabaha.pal@jaipur.manipal.edu)
** R. Paturi is Researcher with Manipal Academy of Data Science, MAHE-Bangalore, Karnatak, India (email : ravikanth@instadatahelp.com)
DOI : 10.17010/ijcs/2020/v5/i4-5/154787

A. Statement of the Problem

Because of late advances in computerized innovations, and accessibility of believable information, a territory of man-made reasoning, profound learning has risen and has exhibited its capacity and adequacy in taking care of complex learning issues which were earlier impractical earlier. Specifically, Convolutional Neural Systems (CNNs) have exhibited their adequacy in picture discovery and acknowledgment applications. Notwithstanding, they require concentrated CPU tasks and memory transfer speed that causes general CPUs to neglect accomplishing the ideal execution levels. Thus, equipment quickening agents that utilize application-explicit incorporated circuits, field-programmable entryway clusters (FPGAs), and realistic preparing units have been utilized to improve the throughput of CNNs. All the more, of late FPGAs have embraced quickening the execution of profound learning systems because of their capacity to boost parallelism and their vitality effectiveness. In this paper, we survey the ongoing existing procedures for quickening profound learning systems on FPGAs. We feature the key highlights utilized by different procedures for quickening execution. We give suggestions to upgrading the usage of FPGAs for CNNs increasing speed. The procedures examined in this paper describe the ongoing patterns in the FPGA-based quickening agents of profound learning systems. Subsequently, this paper is relied upon to coordinate the future advances on productive equipment quickening agents and to be valuable for profound learning specialists. Identification and machine translation to a known language would be helpful.

B. Multilayer Feed-Forward Neural Network

Back propagation algorithm works with feed forward neural network. It trains with all the weights in the network connections. It consists of an input layer and an output layer. Input layer consists of all the neuron units each of which represents the attributes of input. The input to the network consists of many tuples defined by attributes. The input layer then works with hidden layer with further more units connected by weights. The output of this next layer can again be input to another layer and these proceed till the output layer which is nothing but the predictions layer. Input layer is called input nodes and the layer from second to the last are hidden layers and last layer is the output layer. Neurons of each unit are often referred to as neurodes. In case of binary outputs if we have one layer after input layer, we term this as a two layer neural network. Similarly, if two layers are hidden, then we term this as a three layer neural network. Feed forward obtains its name from the fact that the weights are multiplied forward and are not fed back. Each layer in the neuron units takes inputs from previous units and outputs based on the weighted sum of inputs.

After the weighted sum of inputs is calculated, we apply activation function to this output that is non-linear. So, ultimately since this activation function can be non-linear, we are in fact performing non-linear combination of inputs and then predict the label. So, this classification or regression task using neural networks can be non-linear. As stated before, the researcher has to assume the initial topology of neural network that is, the number of hidden layers, number of neuron units, and type of activation function to be used. In general, to improve the response time, we normalize the inputs and perform the classification using neural network. So the values fall between 0 and 1 in the input layer. As in case of other algorithms, even in this case we dummy down the categorical variables. So, we will have separate units in the input layer for each of the categorical variables. We can perform regression for continuous valued variables or classification for discrete valued variables in case of neural networks. In case of binary classification, we can use a single unit to represent either 0 or 1 and in case of classification for more than one class, we need one unit per class in the output layer.

We do not have any thumb rule for number of hidden layers and need to work on trial or error process based on the accuracy achieved. The initial weights also affect the convergence achieved and accuracy. So, we have many factors that affect accuracy in designing the topology. We check the accuracy in the process and then repeat the procedure of changing assumed weights and structure of network, that is, the number of hidden layers till good accuracy is obtained. In order to validate the model of neural network among many options we use cross validation techniques.

C. Back Propagation

In this technique, we iteratively start with each row of input data and then adjust weight after observing the difference between the output from data and predictions. This applies to both in case of classification in case of categorical variables and regression in case of continuous

valued variables. We use mean squared error as a means of estimate to adjust weights in each case and repeat the process in order to minimize the error. So the adjustment of weights is done in backward direction, hence the name. In case of feed forward network we do not adjust weights in each tuple case of the data. We target to achieve the convergence and stop the learning process at some point. We randomize the weights with real numbers for initialization and then start adjusting the weights till the mean square error between outputs and predictions are reduced and target accuracy is achieved. At the start, we give the row of data to the input layer and direct output to the next layer. In each layer for each unit, the output is obtained by calculating the linear combination of inputs and then bias is added. This bias is used as threshold and is used to vary the activity of the unit. We may use many different activation functions, such as logistic or sigmoid functions. This function is applied after bias is added to the input combination with weights. Sigmoid function maps from larger values input to output in the range of 0 to 1. Once the iteration is finished we calculate the error in the output layer and this error is used to find the difference in weights needed to adjust the weights. We then multiply the error by learning rate to find the difference in weights required. Learning rate in this context is similar in nature to gradient descent method for optimization problem. Learning rate cannot be too small or too large. Setting learning rate to too small value leads to very slow convergence, taking longer time than usual and setting learning rate too large value. It leads to oscillation between wrong solutions which will never converge to global minimum value. We have two types of updates on weights and bias. One is case updating and other is epoch updating. In case updating, we update bias and weights on iteration, whereas in case of epoch updating we update the weights and bias after data set is presented at one go. We can set conditions for terminating algorithm.

One of the terminating conditions can be the minimum of weight difference achieved for convergence, or if the error is below pre-specified value. Also if the number of iterations reached a pre-specified value as we cannot run the algorithm indefinitely.

Interpretation is one of the drawbacks of neural network as we cannot understand what the weights do and initial biases depend on and also number of hidden layers that is basically the network topology. We use sensitivity analysis to interpret the internal understanding of the structure.

D. Metrics on Model Selection and Evaluation

As we are not certain in case of neural networks on initial weights and biases, we also need to validate models based on metrics. Some measures of accuracy are more appropriate than others. When we declare the accuracy of the model it is certainly not a good choice to decide the accuracy based on each input tuple as is done in case of back propagation explained in the earlier section. This is because the data is already known to the models on labels, during training we may get good accuracy whereas, in case of testing dataset, the accuracy may drop drastically leading to wrong estimation of accuracy. So we need to test the data and then decide actual accuracy so that we will test with the labels that are not introduced during training phase. Let us in the first instance assume that all the training examples that lead to label of 'yes' can be positive, others as negative to understand different metrics. We may have four different categories of outcomes. In case of true positives, the outcome is positive as the actual is positive. These examples are from the category that is correctly classified. True negatives are the examples that are correctly classified as negative that is the actual and the predicted outcomes are negative. False positive have outcome of prediction to be positive and the actual to be negative. So, these examples are falsely predicted as positive although the input is negative. False negatives have outcome of prediction to be positive and the actual to be negative. So, these examples are falsely predicted as negative although the input is positive. It certainly depends on the field of application to decide which measure is very important and which measure if poor can be disastrous.

Accuracy or the recognition rate is the ratio of true positives and true negatives to the total outcomes consisting of all the above categories. Recognition rate can be checked for both actual positives case and actual negative. On this we understand the classifier performance on positive and negative examples separately. Data balance is also important for training examples as it plays a major role in accuracy. We in certain cases instead of accuracy deal with error rate. This is obtained by subtracting accuracy from 1 or it is the ratio of false positive and false negatives to total number of examples. This is also termed in research as re-substitution error. Accuracy as a measure may give

better value but may not always lead to correct interpretation as the classifier may have poor values of other metrics.

We have data imbalance as in case if the input data is not properly balanced in terms of labels. As an example, consider the example of classification in case of cancer and non-cancer prediction. If we have few training examples of training set that are cancerous then we may end up with good accuracy. If the accuracy is 98% in this case, it does not always lead to correct interpretation. So, we always encourage checking separately for classifier performance on positive examples and negative examples. In this context, we have sensitivity and specificity to deal and compare classifier performance for positive and negative examples separately. Sensitivity is the ratio of true positive to actual positive, also called as true positive rate or recall. Specificity is the ratio of true negatives to actual negatives that is also called as the true negative rate. We also perform a plot between true positive rate and false positive rate that is (1- sensitivity) to analyze the area under curve in most of the classification problems for the measure of metric. Precision as a measure of other metric is the ratio of true positive to the entire predicted positive. So, precision is the exactness that is how many of predicted positives are actually positive whereas specificity measures completeness,that is how many of the actual positive examples are predicted as positive. Higher precision indicates how good among predicted are correctly classified, whereas it does not convey the information on how many are misclassified. Higher rate on sensitivity indicates how good the classifier classified from the given class, but does not convey how good it performed for other labels misclassified as belonging to this class. Alternatively, we may also combine precision and recall into single measure to obtain all the information. That is, we want the detail as to how many of predicted class are correctly predicted and also from given class how many are correctly labelled. In the process we do not lose the information on the number of misclassified labels from other classes and also from given class how many are miss-classified. So, technically we combine these details into a measure called f1 score. It is a harmonic mean of precision and recall.

In case of most of the practical examples, we observe that the data belongs to more than one class. Suppose that the features of a training set may correspond to two classes simultaneously. This is multiclass classification problem. We cannot be accurate in this case. In this case the output of neural network may represent probability of claszs distribution. In addition to these metrics, speed of computation is an important factor for validating the model. Also, the ability of classifier to perform in case of noisy data or missing data or in case of increasing degree of noise, this is where neural network comes to our rescue. Another important factor is scalability, that is even with increasing dataset size the model performance must not be affected. In summary, accuracy is a good measure of performance only in case of data what is balanced and evenly distributed, other metrics helps to analyze in case of individual classes.

In actual process we do hold out method where we segregate the data into training and test data into roughly 70% into training and remaining 30% into test data. This will lead to pessimistic result as we are dealing with fraction of data for evaluation. To overcome this, we do cross validation where initial data is randomly partitioned into k parts that are mutually exclusive, that is, each set is independent of the other set. Now we start with the first set and reserve this for testing the model and training with the combination of the remaining k-1 sets. This is repeated with second, third, till k^{th} set. The overall accuracy is the average of all the accuracies obtained. The accuracy is the overall correctly classified examples from all the k models divided by the total number of examples in the initial data. In this procedure, sometimes we may also face data imbalance issues and we deal with this by initially stratifying data so that all the k data sets have equal number of classes so that data in all the classes is balanced. On the other hand, bootstrap method samples the given training examples uniformly with replacement. In order to solve this problem, we can use the technique of bootstrap, that is sampling with replacement. In this procedure, the data is randomly sampled and used for training the model. In this process, there can be a situation where the data input for training is repeated, but this leads to uniform distribution of data into training and testing. In case of large dataset we observe that probability of training data set is 63.2% whereas there is 36.8% chance that the data is not selected for training and is taken for test. We repeat the procedure of bootstrap k times like in case of cross validation and in iteration we do bootstrap for dividing into train and test sets. We have observed that bootstrap will lead to higher value of accuracy, which can lead to over estimation of accuracy. In general, bootstrap can be adopted in case of small datasets.

When using cross validation on different models we may observe one model to be performing better than the other model. In some cases there can be similar accuracies within the *k* models while performing cross validation but the overall accuracies of two models may show up to be different. In this case we may have to use inferential statistics to test if there is significant difference in model accuracies. In order to accomplish this task, we can perform t-test and check if there is significant difference in the mean value of error rates in both the models.

E. Convolution Neural Network

It is possible to pack the data in a get-together into a singular model which is significantly less difficult to send as exhibited via Carauna and his associates and we can develop this technique for others by using an elective weight methodology. Some dumbfounding results on MNIST have been cultivated by us and by characterizing the data that RUPA models into a single model. Third stage authority models can be arranged rapidly and in equivalent as paired to a mix of pros.

This helpful numerical procedure requires extraordinary equivalent displays to be applied to a gigantic Cops And issues and to extend its usage despite the fact that it has the ability to give organize ace game plans and increasingly accurate results in the standard CFD - DEF couplings. Abusing the overhauled multi quick nature of a twofold Matrix coupling to gain flexibility in the space parasailing while simultaneously keeping a low between process correspondence cost are the plants proposed by the given parallelization system. Execution grasping complex passing system is a record of a steam line free stanza feast correspondence that keeps up in vital good ways from between process correspondence among CFD and DEM programming.

The incapacitated multi-scale coupling has better equivalent execution and has all its ordinary focal points over a mono scale coupling. To study the handling and execution of the method, three benchmark cases are presented. The proposed grants keeping an incredible equivalent execution when work in excess of 1,000 procedures is exhibited by this methodology. Genuine low Mach numbers streams with the thickness-based solvers, this article supervisors acoustic computation. There is incredible significance for a prepared arrangement to oversee fix low Mach number streams so as to ensure a not too bad objective of the low Match number base stream. The acoustic estimations are attempted with the as of late proposed low Match number fixes. As demonstrated by numerical results they are not exact for acoustic counts. To be, accurate for reliable Lo Mac numbers groups and to be an expansion for acoustic figuring's, issues are raised with acoustic estimations with low Mach number fixes and another arrangement is likewise made. There is improvement of the proposed plot concerning the state of craftsmanship through different numerical tests. Different difficulties and plan affirmations and AI have been won by the profound phony neural frameworks (tallying irregular ones). Thus, we can summarize significant work moderate partner and that too a great deal of it from the previous years through this recorded review.

Through the significance of the credit task, ways are the chains of learnable, causal associations among exercises and effects variable to perceive shallow and profound students. Controlled learning (also repeating the chronicled backdrop of back engendering) solo learning, stronghold learning, and transformative calculation and indirect journey for momentary ventures in coding significant and tremendous frameworks are significant by I review. Starting late unprecedented learning machine ELM has increment venturing into the ASM from various investigation fields. The current status of the theoretical lapse and practical advancement this issue are to be relied upon to report in this overview. This effort is inducing the examination of non-parametric quantifiable methodology, request trees, and neural framework headway for credit scoring applications.

The firm-focus virtual FGPA was acclimatized over the Xilinx Spartan physical FGPAs and besides 18 standard circuits were executed over the virtual FGPA by us. The final product uncovers a six times turn down in execution and multiple times greater hardware resource business as the virtual FGPA approach contrasts with planning the circuits allowably to the physical FGPA surface. Aside from the referenced realities, this evaluation is going to help to normalize the up and coming virtual FGPA approaches that may decrease the space above it in various manners. Without a moment to spare (JIT) gradual addition for Field-Programmable Gate Arrays (FGPAs) entitles the improvement of a standard gear twofold. Standard gear comparable to, will give robustness and permitting the producers in structuring an extraordinary number of FGPAs with fundamental models utilizing lone hardware net list. Over the span of hoisting the gear structure for the particular FGPA, a FGPA

supporting JIT get together would have the option to do free game plan from the standard hardware twofold to the FGPA's programmable thinking. In programming plans, association gives extraordinary favorable circumstances. Social affair includes downloading an item equal setup onto a chip, and passably and legitimately re-requesting that twofold to the chief arrangement of the specific processor.

II. LITERATURE REVIEW

A. Relevant Works

[1] is an important work in this direction. Because of ongoing advances in computerized innovations, and accessibility of valid information, a territory of artificial intelligence, deep learning, has risen and has exhibited its capacity and viability in taking care of complex learning issues impractical previously. Specifically, convolution neural systems (CNNs) have exhibited their adequacy in the image detection and acknowledgment applications. Notwithstanding, they require serious CPU tasks and memory transmission capacity that cause general CPUs to neglect to accomplish the ideal execution levels. Therefore, equipment quickening agents that utilization application-explicit coordinated circuits, field-programmable entryway exhibits (FPGAs), and realistic handling units have been utilized to improve the throughput of CNNs. All the more accurately, FPGAs have been as of late embraced for quickening the execution of profound learning systems because of their capacity to boost parallelism and their vitality productivity. In this paper, we survey the ongoing existing methods for quickening profound learning systems on FPGAs. We feature the key highlights utilized by the different methods for speeding up execution. Likewise, we give proposals for improving the usage of FPGAs for CNNs speeding up. The procedures explored in this paper describe the ongoing patterns in the FPGA-based quickening agents of profound learning systems. Therefore, this paper is required to coordinate future advances on productive equipment quickening agents and to be helpful for profound learning analysts.

[2] is another paper on this topic. It majorly discusses Stochastic Gradient Descent (SGD). Stochastic Gradient Descent (SGD) is a regularly utilized calculation for preparing direct AI models. In view of vector variable based math, it profits by the characteristic parallelism accessible in a FPGA. In this paper, we first present a solitary exactness skimming point SGD execution on a FPGA that gives comparative exhibition as a 10-center CPU. We at that point adjust the structure to make it equipped for handling low-exactness information. The low-exactness information is got from a novel pressure conspire called stochastic quantization, explicitly intended for AI applications. We test both full-exactness and low-accuracy structures on different relapse and grouping informational collections. We accomplish up to a significant degree preparing speed-up when utilizing low-accuracy information contrasted with a full-exactness SGD on the equivalent FPGA and a cutting edge multi-center arrangement, while keeping up the nature of preparing. We open source the structures introduced in this paper

B. Approaches

The CNN algorithm is used to classify the image classification using MATLAB and we have used FPGA accelerator to improve the speed since in implementing these complex algorithms it is highly difficult to use the general processors to implement. FPGA is designed based on the algorithms generated by Matlab. While the paper quickly addresses the increasing speed strategies for profound taking in calculations and CNNs from both programming and equipment point of view, the centre of this article has been the survey of late strategies utilized in the increasing speed of CNNs on FPGAs. careful up-to-date survey is given that outlines the work of different prospects and strategies, for example, misuse of parallelism using circle tiling and circle unrolling, viable utilization of inner memory to amplify information reuse, activity pipelining, and viable utilization of information sizes to limit memory impression, and to upgrade FPGA asset usage. The paper likewise introduced the utilization of instruments for creating Register Move Level (RTL) contents that not just assist in computerizing the plan procedure, yet additionally help in investigating the configuration space and recommending effective equipment. The paper talks about the utilization of investigation, for example, (i) remaining burden examination in deciding the calculations that can be parallelized, (ii) ideal circle unrolling factors (iii) deciding access examples to improve information area, and so forth. Also, a brief survey of the utilization of non-deterministic heuristics in un-ravelling NP-hard combinatorial advancement issues in the structure what's

more, usage of CNNs has been introduced. At last, the paper sums up the key highlights utilized by the different FPGA-based CNN quickening procedures and gave suggestions for improving the viability of using FPGAs in CNNs increasing speed.

III. CONCLUSION

We have worked on using FPGA architecture to execute complex algorithms like CNN for image classification tasks. The objective of this thesis work is to explore the topic of executing computationally complex algorithms at the sensor level through the use of FPGAs (field programmable gate arrays) as an alternative to GPUs and TPUs. An area of data analytics concerned with the implementation of algorithms and data processing at the sensor level of production/operation. A branch of artificial intelligence dealing with the creation of artificial neural networks (ANNs) which mimic the human brain with the aid of neuromorphic chips, which are analog data processors inspired by the brain. Neuromorphic computation takes inspiration from biology, physics, mathematics, computer science, and electronic engineering to design systems, such as vision systems, head-eye systems, auditory processors, and autonomous robots, whose physical architecture, and design are based on biological nervous systems.

REFERENCES

[1] A.Shawahna, S. M. Sait, and A. El-Maleh, "FPGA-based accelerators of deep learning networks for learning and classification: A review," *IEEE Access, vol. 7,* pp. 7823–7859, 2019. doi: 10.1109/ACCESS.2018.2890150

[2] K. Kara, D. Alistarh, G. Alonso, O. Mutlu, and C. Zhang, "FPGA-accelerated dense linear machine learning: A precision-convergence trade-off," In *2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Napa, CA,* pp. 160–167, 2017. doi: 10.1109/FCCM.2017.39

[3] K. Abdelouahab, M. Pelcat, J. Serot, and F. Berry, *"Accelerating CNN Inference on FPGA: A Survey,"* [Online]. Available: https://arxiv.org/abs/1806.01683

[4] N. M. Nawi,A. Khan,M.Z.Rehman,H. Chiroma, and T. Herawan, *"Weight optimization in Recurrent Neural Networks with Hybrid Metaheuristic Cuckoo Search Techniques for data classification,"* Hindawi., Doi: https://doi.org/10.1155/2015/868375

[5] A. X. M. Chang, B. Martini, and E. Culurciello, *"Recurrent neural networks hardware implementation of FPGA,"* [Online]. Available: https://arxiv.org/abs/1511.05552

[6] J. C. Ferreira and J. Fonseca, "An FPGA implementation of a long short-term memory neural network," In *International Conference on ReConFigurable Computing and FPGAs (ReConFig), Cancun, 2016,* pp. 1–8. doi: 10.1109/ReConFig.2016.7857151

[7] D. Honegger, H. Oleynikova and M. Pollefeys, "Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU," In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL,* pp. 4930 – 4935, 2014. doi: 10.1109/IROS.2014.6943263.

[8] A. N. Ide and J. H. Saito, "FPGA Implementation of Neocognitrons," In A. R. Omondi, J. C. Rajapakse (eds) FPGA Implementation of Neural Networks, *Springer, Boston, MA.* pp-197–224, 2006. Doi: https://doi.org/10.1007/0-387-28487-7_7

[9] K. Kara, D. Alistarh, G. Alonso, O. Mutlu and C. Zhang, "FPGA-accelerated dense linear machine learning: a precision-convergence trade-off," In *IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), Napa, CA,* pp. 160–167,2017. IEEE. doi: 10.1109/FCCM.2017.39

[10] R.Keim, *"What is an FPGA? An introduction to programmable logic," 2018.* [Online]. Available: https://www.allaboutcircuits.com/technical-articles/what-is-an-fpga-introduction-to-programmable-logic-fpga-vs-microcontroller/

[11] M. Nazemi, S. Nazarian and M. Pedram, "High-performance FPGA implementation of equivariant adaptive separation via independence algorithm for Independent Component Analysis," In *2017 IEEE 28th International Conference on Application-specific Systems, Architectures and Processors (ASAP), Seattle,* WA, pp. 25–28, 2017. doi: 10.1109/ASAP.2017.7995255

[12] K. L. Rice, M. A. Bhuiyan, T. M. Taha, C. N. Vutsinas, and M. C. Smith, "FPGA Implementation of Izhikevich spiking neural networks for character recognition," In

2009 International Conference on Reconfigurable Computing and FPGAs, Quintana Roo, pp. 451 – 456, doi: 10.1109/ReConFig.2009.77

[13] J. P. Singh, "Designing an FPGA sythesizable computer vision algorithm to detect the greening of potatoes,"*International Journal of Engineering Trends and Technology, vol. 8,* no. 8, pp. 438 – 442, 2014. doi: https://arxiv.org/ct?url=https%3A%2F%2Fdx.doi.org%2F10.14445%2F22315381%2FIJETT-V8P275&v=9f891483

[14] R. Solovyev, A. Kustov, D. Telpukhov, V. Rukhlov and A. Kalinin, "Fixed-point Convolutional Neural Network for real-time video processing in FPGA," In *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Saint Petersburg and Moscow,* Russia, pp. 1605 – 1611, 2019. doi: 10.1109/EIConRus.2019.8656778

[15] D. Soni, *"Spiking neural networks, the next generation of machine learning,"* 2018. [Online]. Available: https://towardsdatascience.com/spiking-neural-networks-the-next-generation-of-machine-learning-84e167f4eb2b

[16]A. Zuppicich and S. Soltic, "FPGA implementation of an evolving spiking neural network,"In *Koppen, M., Kasabov, N., Coghill, G. (eds) Advances in Neuro –Information Processing, International Conference on Neural Iinformation Processing,* pp. 1129 – 1136, 2008.

About the Authors

Dr. Subhabaha Pal is a seasoned Data Scientist and Academician with over 16 years of experience working in varied fields of Information Science and Analytics. He had been nominated as the Top 20 Data Science and Machine Learning Academicians in India in 2018 by Analytics India Magazine. He is Ph.D. from the University of Calcutta. He has taught Data Science at well-renowned institutions like Manipal University, T. A. Pai Management Institute, International Institute of Digital Technologies among others. He had worked in senior software-related roles in organizations like Kuwait Petroleum Corporation and Manipal Global. He has around 40 research papers in the field of Data Science and Analytics to his credit and published 3 books in renowned publications to his credit. He is a hardcore Data Scientist delivering many data science projects to different SMEs and has worked in various domains.

Ravikanth Paturi is a Researcher with Manipal Academy of Data Science, MAHE-Bangalore, Karnataka.