Biomedical & Pharmacology Journal, March 2025. Vol. **18**(Spl.), p. 229-243

Special Issue – AI-Driven and Multimodal Innovations in Biomedical Imaging and Sensing

Comparative Study of Hardware Footprints of Neural Network Activation Functions for EMG Based Diabetic Sensorimotor Polyneuropathy Severity Classifier

Sandeep Kumar Pandey and Geetika Srivastava*

Department of Physics and Electronics, Dr. Rammanohar Lohia Avadh University, Ayodhya, India.
*Corresponding Author Email: geetika_gkp@rediffmail.com

https://dx.doi.org/10.13005/bpj/3084

(Received: 16 December 2024; accepted: 12 February 2025)

Diabetic Sensorimotor Polyneuropathy (DSPN) is a common complication of diabetes, significantly increasing the risk of diabetic foot ulcers and potential amputations. Timely monitoring and early detection of DSPN severity are crucial for prevention. Recent advancements in Machine Learning (ML) have led to highly accurate diagnostic models in the medical field and with the advancements in emerging edge devices these models offer patient-friendly solutions for continuous monitoring. Developing edge-compatible devices necessitates less complex ML models to optimize power consumption, resources, and processing speed. The choice of activation function is critical, as it directly impacts model complexity and performance. While complex data requires sophisticated functions to maintain accuracy, resource-constrained edge platforms demand a balance between complexity and effectiveness. This study presents a performance comparison of hardware-implemented Neural Network (NN) classifiers utilizing various linear and non-linear activation functions for Electromyography (EMG)-based DSPN classification, tested on the ZCU102 FPGA board. Results indicate that the NN employing the ReLU activation function achieved 78% accuracy with only 4.33 W power dissipation, a time delay of 20.1 mS, and resource utilization of 70,134 Look Up Tables (LUT) and 123 Block RAM (BRAM). These findings demonstrate that ReLU-based NNs offer better power efficiency, resource utilization, and speed compared to other activation functions for EMG-based DSPN classification. These insights serve as a valuable reference for researchers developing hardware-friendly NN models for edge-based ML applications in biomedical devices.

Keywords: Activation Function, DSPN, FPGA, ML, NN.

Diabetes Mellitus (DM) is increasing globally at a fast rate with expected 578 million people suffering with DM by the year 2030 worldwide. The annual health expenditure on DM is estimated to 825 billion USD by 2030.[1] DSPN is a common and debilitating problem in DM, primarily associated with high and poorly controlled blood sugar levels over an extended period of time.[2–6] It is a primary factor contributing to complications in diabetic foot, giving rise to issues like ulceration and Charcot neuroarthropathy (a condition that affects the joints, particularly in the feet, of individuals with peripheral neuropathy, commonly due to diabetes), which becomes a cause for lower-limb amputation. Some Other problems of DSPN included distressing pain and an increased risk of falls. Most disturbing, DSPNs, foot ulceration and lower- limb amputation are independently associated with an increased risk of mortality.[7] Early detection and prevention can reduce the risk of DSPN.

Existing diagnostic methods for DSPN, such as Nerve Conduction Studies (NCS),[8] Vibration Perception Threshold (VPT),[9] Michigan Neuropathy Screening Instrument,[10] Neuropathy Disability Score (NDS),[11] Achilles Tendon Reflexes,[12] pinprick and temperature sensation,[13] are time consuming, expensive, require expertise, and in some cases, cause discomfort.

ML based edge devices can help in early detection of DSPN as the combination of machine learning and edge computing is transforming the healthcare industry by allowing real time analysis of EMG data and personalized treatment.[14] ML algorithms analyse patient's EMG data, to aid in accurate diagnostics and predictive analytics. At the time edge computing ensures that this analysis takes place on devices reducing delays and enhancing security. Together these technologies enable healthcare systems to offer sophisticated patient monitoring, personalized medicine and efficient integration, with devices. Software based ML models in biomedical diagnosis are relatively easy to develop and can provide high accuracy reaching upto 100% as given in literature.[5,15–17] The development of edge devices using these ML algorithms on resource limited hardware is not an easy task as complex ML models when implemented at hardware utilises more resources, consume more power and offer slow response time. The implementation on hardware need some compromise between performance of ML model and hardware conditions such as resource, power and delay.

While extensive research has been conducted on the development and optimization of neural network architectures and training algorithms, the focus on hardware implementation with diverse activation functions is relatively limited. Y. Xie et al.[18] presented a FPGA implementation of NN for different activation function with two fold LUT (t-LUT) concept which requires only half of the total number of bits compared to the original LUT implementation. Z. Hajduk et al.[19] presented implementation of hyperbolic tangent function in FPGA by using direct polynomial or Chebyshev approximation which provided very high accuracy, short calculation time and moderate resource utilization. L. D. Medus et al.[20] presented FPGA implementation of Feed Forward Neural Network (FFNN) with an Activation Function Block (AFB) which contains ReLU, Logistic Sigmoid and hyperbolic tangent activation function and provide hardware utilization report for Iris and MNIST datasets. Z. Pan et al.[21] presented implementation of sigmoid function using three modules namely piecewise linear approximation, Taylor series approximation and Newton-Raphson method based approximation provided balance between accuracy, performance and resource utilization, but there is very few articles are available in literature which provided performance evaluation of different activation functions for hardware implementation.

Different activation functions have unique computational characteristics, edge devices often have specific requirements and constraints that differ from traditional computing systems. Research on hardware implementations of neural networks with different activation functions for biomedical signals can lead to tailored solutions that maximize performance and efficiency in edge computing scenarios. As per my study this is the first article which presents the performance and hardware footprint of different activation functions in FPGA implementation of NN for the development of EMG based edge device to detect DSPN. The software model of designed NN is hardware coded and then tested for evaluating its hardware footprint using Xilinx ZCU102 FPGA board. The targeted FPGA offers great amount of configuration flexibility with high computation power to support complex models for signal processing as compared to general purpose CPU.[22]

Presented study analyses the hardware encoding of activation functions for hardware implementation as there are three important components of any neuron; the weight, the bias and the activation function. Activation function fetches the attention because in a NN model it adds non linearity and without it the NN can simply do a linear transformation. Learning of a complex task by NN is just impossible without adding nonlinearity.[23,24]

The primary selection of the activation function is based on the nature of input, type of output, computation efficiency and range of output.[25] Smooth shape of the activation function help NN converge faster and the complex shape functions are capable of learning complex patterns however difficult to optimize.[26] For non-complex data patterns the linear or piecewise linear activation functions are computationally efficient than the complex shaped functions, and speed up the learning. However the complex patterns in the input data can only be learnt by non-linearity of activation function.[27] Gradients of weights are updated on the error value which is difference between actual and predicted value and for small gradients network will learn slowly and face difficulty in convergence. Activation functions with smaller values of slope face vanishing gradient problem whereas the activation functions with larger slope suffers from exploding gradient problems where the large error gradients accumulate and update the weights with very large values resulting in instability.[28,29] Choosing the right activation function for NN is a typical task as too ambitious accuracy objectives with complex mathematical computations may lead to consumption of large amount of hardware resources and will result in slower or power hungry performance.[30] There is no such result reported so far, regarding the hardware footprint of ML using different types of activation function. Such comparative study on hardware footprint is the need of the hour as edge computing is gaining much popularity due to advantages of latency and security. For edge computing the proper selection of activation function is even crucial as the resource constrained NN easily loses accuracy.

This paper compares the performance of software based NN algorithm with different activation functions in hardware implemented version using FPGA for DSPN severity classification using EMG signal. The comparison is based on resource utilization, timing performance and power consumption for approximately 78% accuracy. Rest of the paper is arranged as follows Preliminaries about activation functions, research methodology, experimental results, discussion and conclusion.

Preliminaries

Figure 1 shows a NN in which inputs represent various features or variables that are used as the initial information for the network to process. Weights are the parameters that the neural network learns during training. These weights determine the strength of the connections between the inputs and the neurons in the network. The net input is the weighted sum of the inputs. It indicates the total input to a neuron, taking into account both the inputs and their corresponding weights.

Activation functions are a crucial component of artificial neural networks and are used to introduce non-linearity into the output of a neuron. This non-linearity allows neural networks to model complex relationships in data, making them capable of learning and performing a wide variety of tasks including classification, regression, and more.

Activation functions in neural networks have evolved over time to address various challenges and improve the performance of the networks.

The earliest neural networks, such as the perceptron introduced by Frank Rosenblatt in the late 1950s, used a step function as the activation function. The step function outputted a 1 if the input was greater than or equal to 0, and otherwise. However, the step function’s lack of differentiability limited its application in training neural networks using gradient-based optimization methods.

In the 1980s and 1990s, the sigmoid function, also known as the logistic function, became popular as an activation function in neural networks. The sigmoid function maps any input value to a range between 0 and 1, and it addresses the differentiability issue of the step function. It was widely used in the early days of neural networks, particularly in binary classification tasks. However, it suffers from the vanishing gradient problem, which can slow down or even stall the training of deep neural networks.

The hyperbolic tangent function (Tanh) was introduced as an alternative activation function to address the vanishing gradient problem associated with the sigmoid function. Its development coincides with the early 1990s as part of efforts to improve the training of deeper neural networks.

The Rectified Linear Unit (ReLU) emerged as a significant development in the early 2010s, providing an effective solution to the vanishing gradient problem and becoming the default activation function for many types of neural networks. The concept of the ReLU activation function was introduced by Hahnloser et al. in a research paper titled "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit", published in 2000.[31] The paper outlined the functioning of the rectified linear neuron, which is the basis for the ReLU activation function.

The ReLU activation function is a piecewise linear function that outputs the input directly if it is positive, otherwise, its output is zero. The ReLU activation function is a combination of linear units with a derivative value allowing backpropagation.[32,33] It has an advantage that not all the neurons are active at the same time. The linear non saturating property accelerates the convergence of gradient descent towards the global minimum of loss function. During backpropagation the dying ReLU is the problem with derivative function where the negative input values become zero and the weight and biases are not updated, hence ability of NN to fit data decreases. The problem of dying ReLU is solved by Leaky ReLU in which there is a small positive slope in negative area.[34] The dead neurons for negative inputs are not encountered during backpropagation. The learning time in Leaky ReLU is more. Parametric ReLU is another form of ReLU in which the most appropriate value of the slope of the negative part is learnt during backpropagation.[35] The drawback with parametric ReLU is that it performs differently of same input for different value of slope. Exponential Linear Unit or ELU is another variation of ReLU in which the log curve represents the negative part of the function.[36] ELU smooths slowly and avoids dying ReLU problem. ELU increases computational time and suffers exploding gradient problem.

Swish is another self-gated activation function that outperforms ReLU for deep network. Swish is a smooth function which is bounded below but unbounded above. Small negative values are not zeroed in swish unlike ReLU. Swish enhances the expression of data and weight for learning.[37]

Gaussian Error Linear Unit or GELU is the activation function that combines the properties of dropout, zoneout and ReLU. The input is multiplied either by zero or one which is stochastically determined and is dependent on the input. GELU's non linearity is better than ReLU and ELU and shows better performance in many applications.[38]

Scaled Exponential Linear Unit or SELU is a self-normalizing network and each layer preserves the mean and variance from previous layer. SELU can output both negative and positive value so the mean can be shifted both positive and negative in value.[39] The network converges faster. This study uses non-linear activation functions ReLU, Leaky ReLU, ELU, GeLU, SeLU and Swish for comparative study of their hardware footprint. Table 1 show the mathematical equations of various

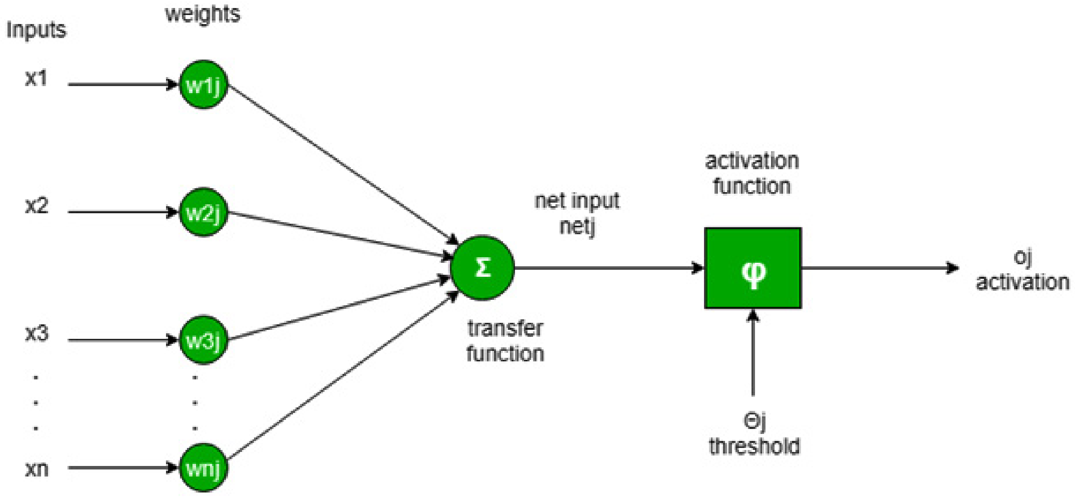


Fig. 1. General NN structure

activation functions and figure 2 shows the plot of these activation functions.

MATERIALS AND METHODS

Dataset Description

The dataset comprised 142 samples without DSPN and 72 samples with severe DSPN. As lower limb muscles are affected by DSPN therefore the data contains EMG signals from three mostly affected lower limb muscles named as Vastus Lateralis (VL), Gastrocnemius Lateralis (GL) and Tibialis Anterior (TA) as shown in figure (3).[40] Data was divided properly for training, validation and testing as given in table 2.

Table 1. Mathematical Equations of Considered non-linear Activation Functions

Activation Function	Mathematical Equation
ReLU	$f(x) = \max(0, x)$
Leaky ReLU	$f(x) = \max(0.1\, x, x)$
ELU	$f(x) = \begin{cases} x, x \geq 0 \\ \alpha(e^{x} - 1), x < 0 \end{cases}$
GeLU	$f(\alpha, x) = \lambda \begin{cases} \alpha(e^{x} - 1) 0\ for\ x < 0 \\ x\ for\ x \geq 0 \end{cases}$
SeLU	$f(\alpha, x) = \lambda \begin{cases} \alpha(e^{x} - 1) 0\ for\ x < 0 \\ x\ for\ x \geq 0 \end{cases}$
Swish	$f(x) = x * sigmoid(x)$

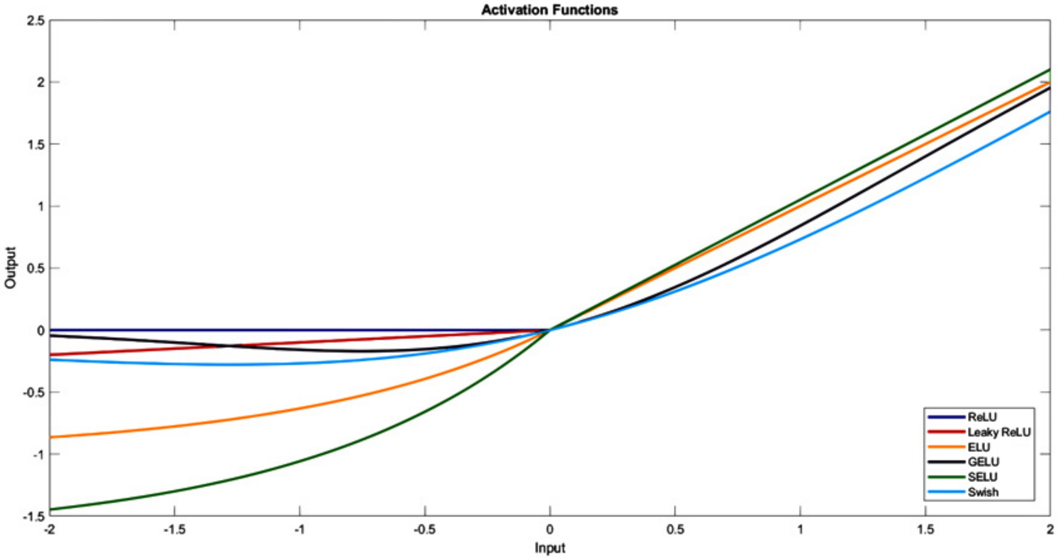


Fig. 2. Plot of Various activation functions

Figure 4 shows the Non DSPN EMG signal and DSPN EMG signal collected from these affected lower limb muscles. Pre-processed imbalanced dataset was balanced using Synthetic Minority Oversampling Technique (SMOTE) data augmentation method in python.[41] Feature selection was done using Relieff technique[42] in Matlab for all GL, VL and TA muscles EMG data. Figure 5(a), 5(b) and 5 (c) shows six important features selected from each data.

Neural Network Design

The study was conducted on MATLAB R2021a and Vivado design suite (v 2022.2) by Xilinx and further for hardware implementation Xilinx's ZCU102 FPGA board is used.

To evaluate the performance of activation functions, designed a 3 inputs, 3 layers fully connected neural network as shown in block diagram of figure 6 in MATLAB and then the model was freezed and ported to hardware (ZCU102) using Vivado design suite. Details of designed NN models was given in table 3.

The inputs of the NNs are taken from the selected features of the lower limb muscles data named as GL-Enhanced mean Absolute Value, VL-Sigma (signal entropy) and TA-Final/min, as they are the first important feature in respective dataset. Using these features, multiple NNs are designed using selected activation functions and then the models are exported to Vivado for hardware implementation. Figure 7 shows the step by step procedure for implementing a NN in hardware and figure 8 shows the hardware setup.

Implementation on ZCU102 FPGA board is completed using Look Up Tables (LUT) implementation of activation function. The LUT is implemented by using Block memory generator in Vivado. Since each activation function receives an input that corresponds to a specific value of outputs, the LUT is a fast way of generating this input-output correspondence. The LUT need a special memory initialization file called *.coe which was generated with MATLAB. Since each neuron in one layer uses the same LUT, this means they all use the same *.coe file for one type of network. This file populated the Block RAM memory. The memory addresses of the Block RAM serve as the input, while its addressed contents as the output. This file renamed with the *.coe extension. In Vivado, a BRAM unit is constructed using the IP Catalog for the Block Memory Generator then it is loaded with the memory initialization file. One "*.coe" file used at the output of all the neurons in that particular single layer. This process repeated

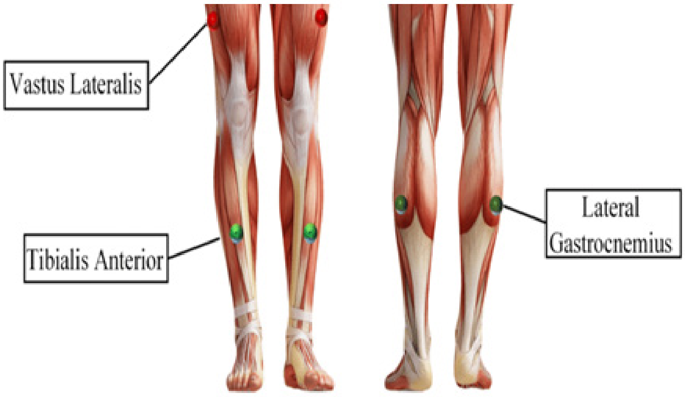


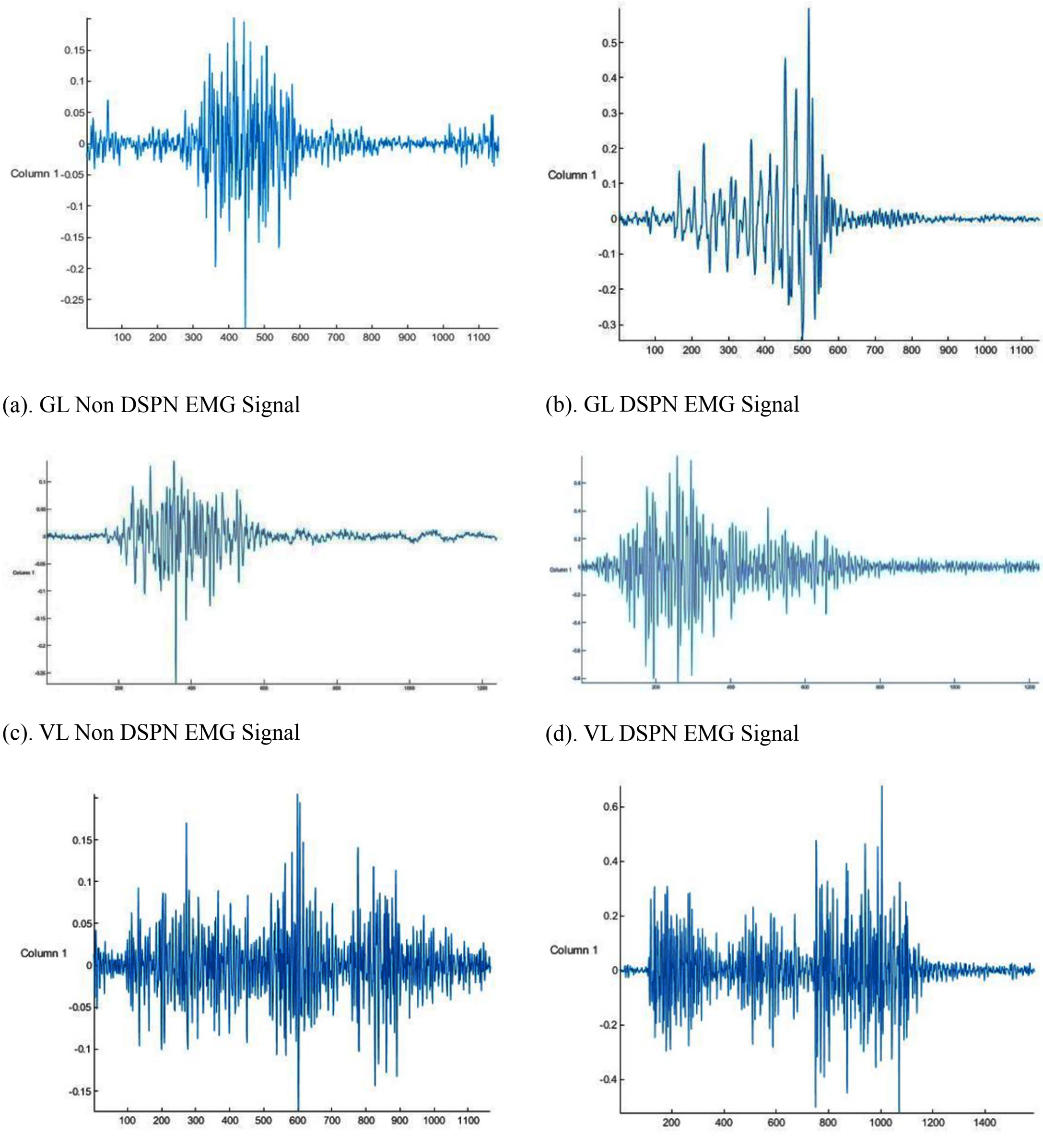
Fig. 3. Mostly Affected Lower Limb Muscles

Table 2. Details of dataset

Class	Training (70%)	Validation (15%)	Testing (15%)	Total
Non DSPN	100	21	21	142
DSPN	50	11	11	72
Total	150	32	32	214

for each of all other layers in the design, but each layer's activation LUT must be considered and designed separately

The number of bits used for representation of activation function affects the performance of design and also impacts the resource utilization in hardware design.[43] Therefore Bit value is chosen to accurately represent the output with an acceptable loss of information. In present study for justified comparison between all the chosen activation function bit value is set to 18 for all three layers and memory location to 131072 and then the hardware simulation for NNs is performed. Table 4 shows the evaluation metrics for proposed study.



(a). GL Non DSPN EMG Signal

(b). GL DSPN EMG Signal

(c). VL Non DSPN EMG Signal

(d). VL DSPN EMG Signal

(e). TA Non DSPN EMG Signal

(f). TA DSPN EMG Signal

Fig. 4. Non DSPN and DSPN EMG Signals from GL,VL and TA Muscles

RESULTS

Performance comparison of chosen activation functions is based on power dissipation, resource utilization and delay. Table 5 shows the accuracy, power dissipation and delay of designed NN implementation with all considered activation function. It shows that the device with ReLU activation function dissipate less power and takes less time than others i.e. use of ReLU activation function saved power and works faster comparison to others.

Table 6 shows the resource utilization of implemented design with various activation functions. It shows that device with ReLU activation function uses fewer resources than others, it means that the device with ReLU activation function is compact than others.

DISCUSSION

DSPN is a growing complication for individuals with DM. Despite the introduction of various diagnostic tools, such as symptom scores, quantitative sensory testing, and electrophysiology, researchers continue to face challenges in effectively screening and stratifying DSPN. The current methods for manual grading and decision-making by healthcare professionals are subjective, relying heavily on individual expertise. Addressing the need for a more accurate and non-invasive approach to detect DSPN, a portable device is essential that should intelligently identify DSPN, offering higher accuracy and robustness, and also acceptable by clinicians. Early detection of DSPN can significantly enhance the management of painful diabetic neuropathy, reduce the risk of foot

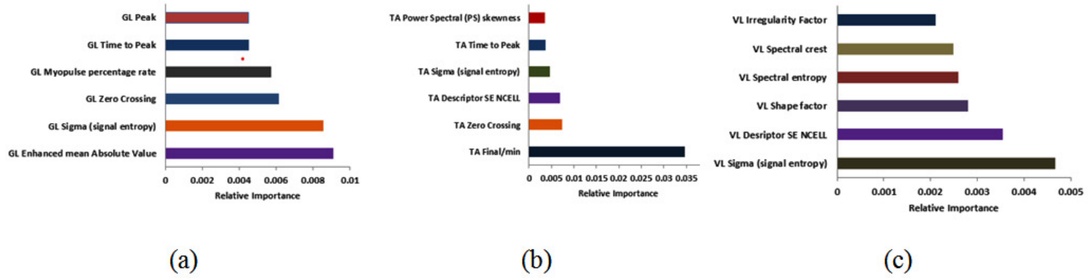


Fig. 5. (a). Six important features of GL data; (b). Six Important features of TA data; (c). Six important features of VL data

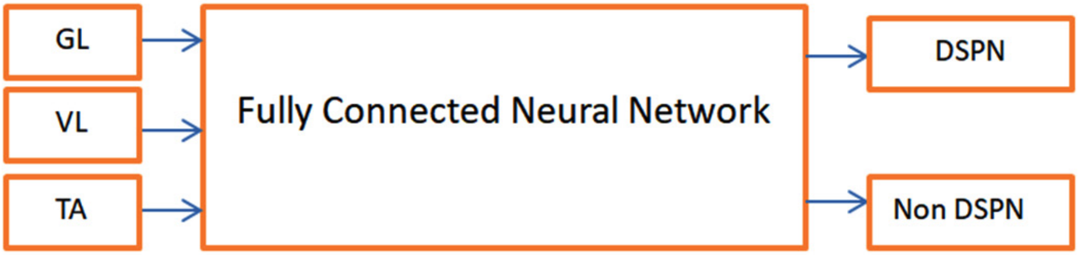


Fig. 6. Block Diagram of Designed Neural Network

Table 3. Details of designed Neural Network

Input Layers (No. of Neurons)	Activation Functions	Total Learning Parameters	Output Layers (No. of Neurons)
03	Various Activation functions used in the study	49	01

ulceration, amputation, and other complications associated with diabetes.

ML based edge devices plays crucial role in point of care diagnosis of diseases in healthcare. Development of edge devices needs adjustment between software model and hardware conditions. As activation functions are important parameters for a NN design which decided the complexity of model and also affect the hardware parameter such as power consumptions, resource utilization and latency of device. The discussion provided the performance of activation functions for FPGA implementation and suggested that which one is better for hardware implementation.

Figure 9(a) shows the utilization of CLB registers in percentage. It shows that ReLU consumes 15% and all others consumes more than ReLU. Fewer CLB registers utilization means design is resource efficient, fast, less power consuming and design complexity is also less.

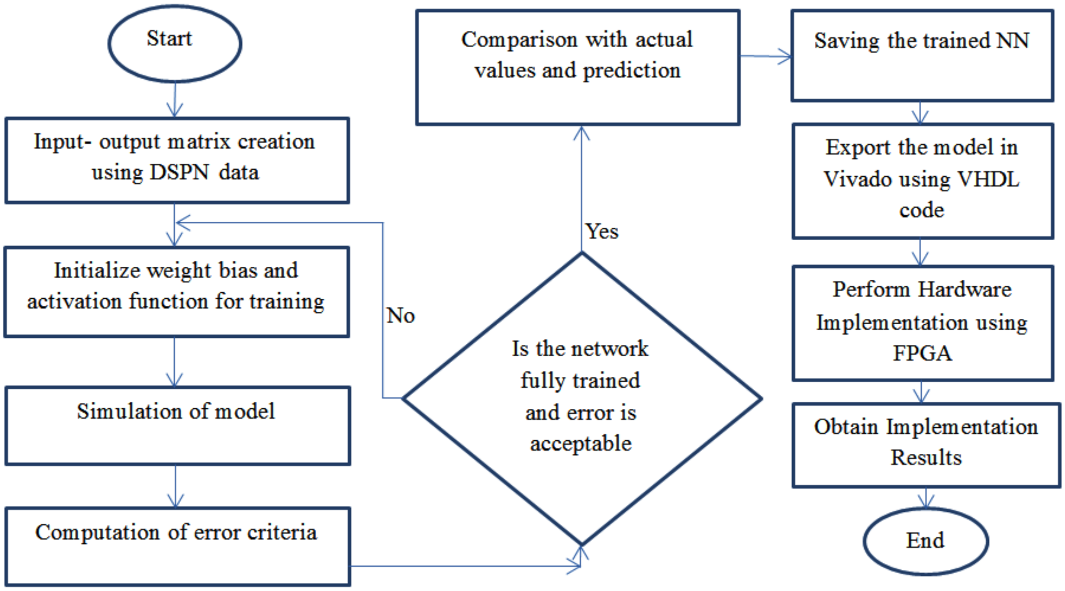

Start
Input- output matrix creation using DSPN data
Initialize weight bias and activation function for training
Simulation of model
Computation of error criteria
Is the network fully trained and error is acceptable
No
Yes
Comparison with actual values and prediction
Saving the trained NN
Export the model in Vivado using VHDL code
Perform Hardware Implementation using FPGA
Obtain Implementation Results
End

Fig. 7. Steps of Hardware implementation of NNs

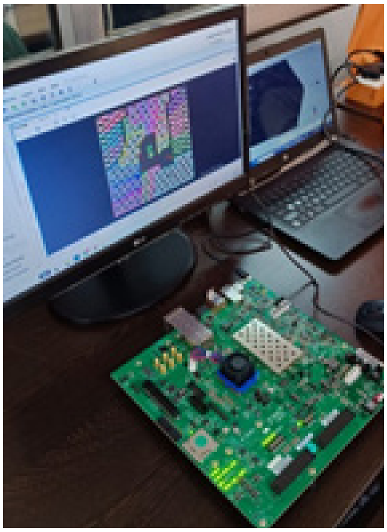


Fig. 8. Hardware Setup

Figure 9(b) shows the percentage utilization of LUTs for various activation functions. It shows that NN with ReLU, Leaky ReLU and GeLU consumes 15% and others utilised more LUTs. Figure 9(c) shows the BRAM percentage utilization for NNs with different activation functions. It shows that NN with ReLU activation function utilized only 11% BRAM and all others utilized more than ReLU. From Figure 9(a), 9(b) and 9(c) we found that although for LUT utilization for ReLU, GeLU and Leaky ReLU are same but CLB register and BRAM utilization report shows that ReLU perform better than others. Figure 9(d) shows the ReLU's time delay percentage is approximately 5% less than all others i.e. NNs with ReLU are fast operating.

Table 4. Evaluation Metrics

S.No.	Name	Description
1.	Accuracy	Accuracy in a classification system is an important parameter and underpins the reliability and effectiveness of the entire system. High accuracy ensures that the system makes reliable predictions.
2.	Power dissipation	Edge devices are often deployed resource-constrained environments where access to continuous power sources may be limited. Efficient power dissipation is crucial for maximizing the operational lifespan of these devices while minimizing energy consumption.
3.	Time delay	Time delay (latency), plays a crucial role in the performance and functionality of edge devices. Edge devices are frequently utilized in applications that demand real-time data processing and decision-making. A device with low latency is good for real time operation.
	Resource Utilization	
1.	LUT	Look-Up Table (LUT) in FPGAs is a fundamental building block that provides the ability to implement user-defined logic functions. Higher LUT utilization implies increased complexity, whereas lower LUT utilization may suggest simplicity, available resources for future expansion, and ease of design.
2.	BRAM Tiles	BRAM tiles in FPGAs are specialized blocks of memory that provide on-chip storage for data and program variables. It offer fast and low-latency memory storage directly on the FPGA chip. Designs with lower BRAM tile utilization may involve less complex data storage requirements, resulting in simpler design implementations and reduced memory management overhead, leave room for scaling or accommodating future enhancements that may require additional memory resources within the FPGA design.
3.	CLB Register	CLB Registers are specialized elements located within the Configurable Logic Blocks (CLBs) that provide storage for data in the form of flip-flops. These registers play a crucial role in holding intermediate and final results during the operation of the FPGA design. Designs with higher CLB register utilization may indicate a more complex and performance-critical application.
4.	CLB.	CLBs are fundamental building blocks within the FPGA fabric that provide the ability to implement a wide range of logic functions and interconnections. Lower CLB utilization suggests the implementation of simpler or less complex logic functions within the FPGA, possibly indicating a less resource-intensive design.

Table 5. Accuracy, Power Dissipation and delay Summary for various activation function

Activation Function	Accuracy (%)	Power Dissipation (W)	Delay (mS)
ReLU	78	4.33	20.1
Leaky ReLU	77	5.34	28.1
ELU	75	6.52	30.6
GeLU	77	5.82	27.8
SeLU	76	6.15	28.4
Swish	77	6.35	30.0

Table 6. Resource Utilization Summary

	CLB Register (548160)	CLB (34260)	LUT (274080)	BRAM Tiles (912)	DSP (2520)	Bonded IOB (328)
GeLU	1305	10759	72515	203.5	58	37
SeLU	1243	11428	77525	158.5	58	32
Swish	1237	12923	87459	240.5	58	37
ELU	1232	13042	87336	269.5	58	37
Leaky ReLU	1204	10825	71877	133	58	37
ReLU	1108	10232	70134	123	58	34

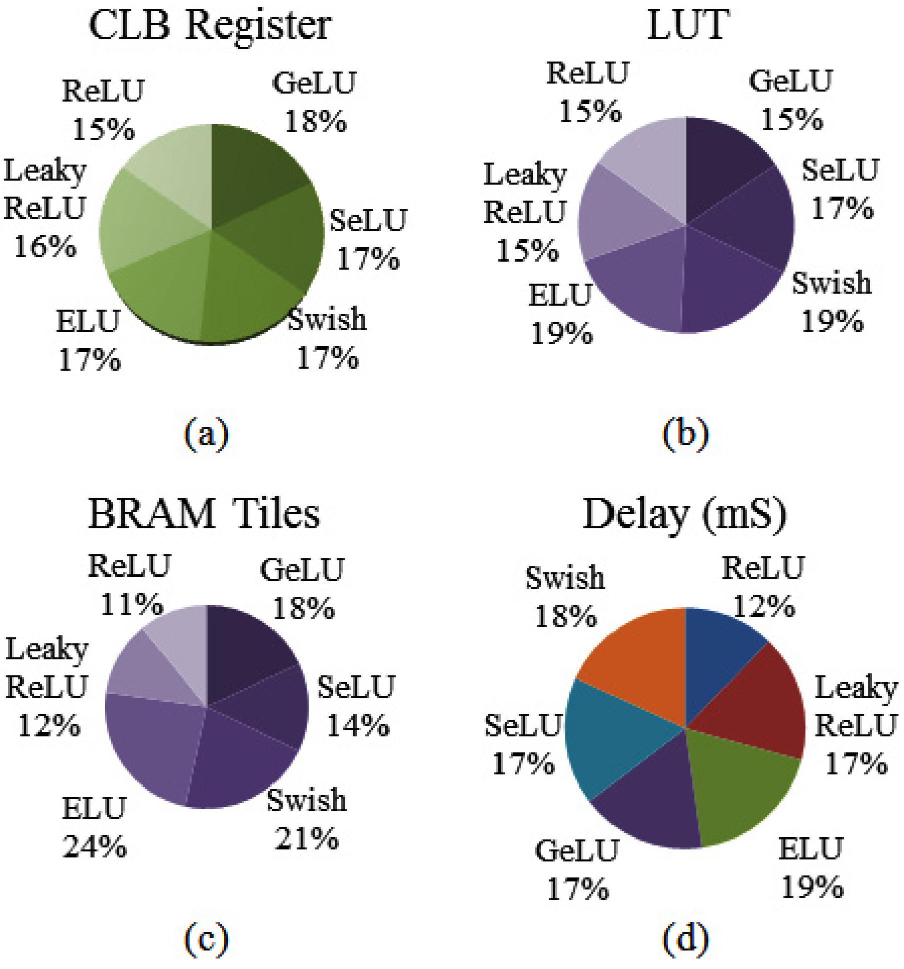


Fig. 9. (a). Percentage Utilization of CLB Registers; (b). Percentage Utilization of LUTs; (c). Percentage Utilization of BRAM Tiles; (d). Percentage Delay Report

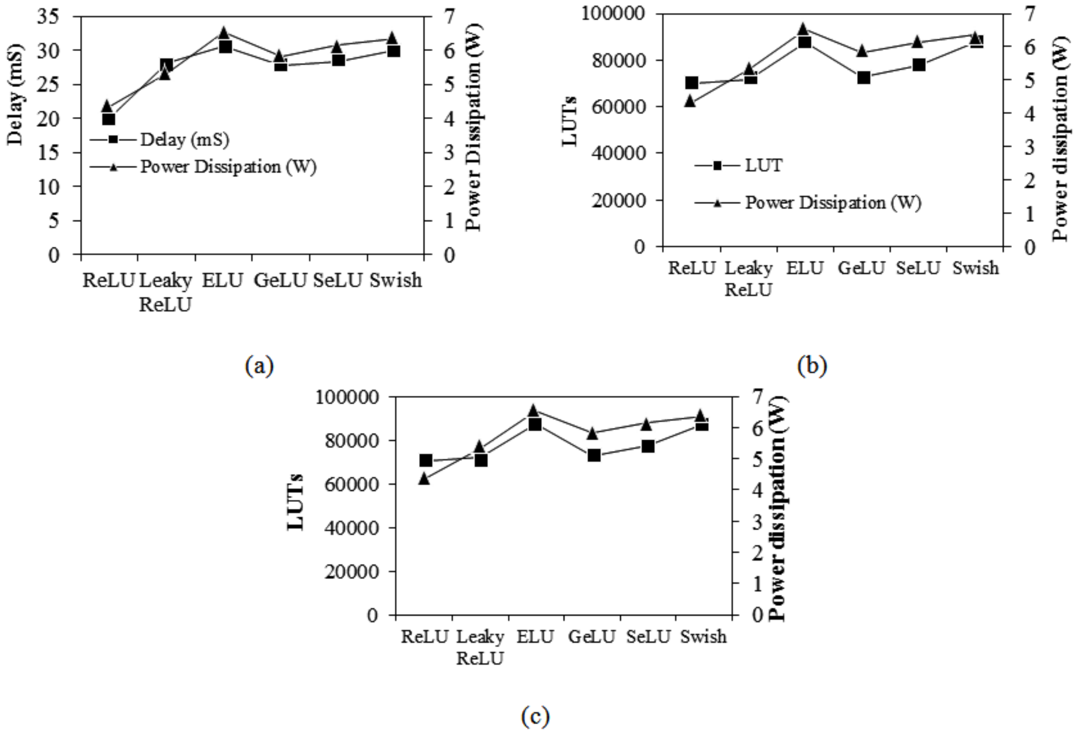


Fig. 10. (a). Time delay and power dissipation curve (b). LUT Utilization and Power Dissipation Curve (c). Delay and LUT utilization curve

Figure 10 shows the power dissipation, LUT Utilization and time delay for different activation functions. Figure 10(a): As the ReLU activation function is computationally simple, the delay and power dissipation through the ReLU activation layer is lower as compared to layers using more complex activation functions. Figure 10(b) shows the LUT utilization and power dissipation in NN using different activation functions. Higher LUT utilization causes higher power dissipation as shown in graph. When more LUTs are used in design, it means that more logic resources are active, which typically consumes more power. However, the relationship is not strictly linear as shown in figure, and the power consumption may not only varies proportionally with LUT utilization, due to other factors like operating frequency and especially if the FPGA has to activate or deactivate additional routing resources or interconnects. The graph shows that NN with ReLU activation function utilized less number of LUTs and hence consume less power. Figure 10(c) shows the LUT utilization and delay respectively in FPGA design as the relationship between delay and Look-Up Table (LUT) utilization in Field-Programmable Gate Arrays (FPGAs) is an important consideration when designing FPGA circuits. As shown in figure, delay and LUT utilization for ReLU activation function is less than others. The reason is that the higher LUT utilization requires the FPGA to run at a higher clock frequency to meet timing constraints. Increasing the clock frequency lead to shorter clock periods, this in turn, reduced the amount of time available for logic to complete its computations, hence increasing the overall delay. As NN with ReLU activation function utilized less number of LUTs therefore the delay for NN with ReLU activation function is less than others.

These parameters indicate that ReLU activation function exhibits lower delay and power dissipation as compared to other activation functions in the neural network (NN) design and utilize lower LUTs than others due to the inherent simplicity and piecewise linearity of the ReLU function, which can be implemented in FPGA hardware efficiently.

Although ReLU works well in feed forward neural networks, selected here in this study, but it may not be the best choice for other NNs such as for Recurrent Neural Networks (RNNs) due to issues like vanishing gradient problem. In such cases variants of ReLU like the Leaky ReLU or Parametric ReLU are often used in RNNs to address these problems.[44]

CONCLUSION

Selection of proper activation function is crucial for hardware implementation of NN. In software design and testing one can get better accuracy with varying the activation function but whether or not it can implement in resource constrained hardware; is always a challenging task. If selected NN or the chosen activation function is more complex then it will consume more power, resource and perform slow. This study summarizes hardware requirements for hardware implementation of NN using FPGA ZCU102 board to detect DSPN, the NN with ReLU activation function performed better than others. ReLU in general utilises less resources in FPGA so it consumes less power and reduces delay in operation. ReLU activation function is a better choice for many NNs but it may not good for all tasks. In those cases variants of ReLU and different activation function can perform better than ReLU. This study is an effort towards efficient hardware implementation of NN Algorithms for edge computing. For this study the hardware implementation of NN for DSPN detection in binary class classification is taken as a target problem. This work summarizes the hardware foot print of NN with varying activation functions and will help researchers in proper selection of activation function for edge based NN.

ACKNOWLEDGEMENTS

The authors would like to express their sincere gratitude to ASEAN-India Collaborative Research Project, Department of Science and Technology, Science and Engineering Research Board (DST-SERB), Govt. of India, ref. no. CRD/2020/000220; Research and Development Scheme, Department of Higher Education, Government of Uttar Pradesh, India, ref. no. 80/2021/1543(1)/70-4-2021-4(28)/2021 and ICTP, Italy for supporting and facilitating this research work. Special thanks to Dr. Fahmida Haque (Affiliated to: Artificial Intelligence Resource, National Cancer Institute, National Institutes of Health, Bethesda, MD, 20814, USA) and Prof. MBI Reaz (Affiliated to: UKM, Bangi, Selangor & IUB, Bangladesh) for their supports.

Funding Source

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Conflict of Interest

The authors do not have any conflict of interest.

Data Availability Statement

The manuscript incorporates the datasets produced or examined throughout this research study.

Ethics Statement

This research did not involve human participants, animal subjects, or any material that requires ethical approval.

Informed Consent Statement

This study did not involve human participants, and therefore, informed consent was not required.

Clinical Trial Registration

This research does not involve any clinical trials.

Author's Contribution

Sandeep Kumar Pandey: Conceptualization, Methodology, Analysis and Writing – Original Draft; Geetika Srivastava: Visualization and Supervision; All authors made a significant and equal contribution to this work.

REFERENCES

1. Saeedi P, Petersohn I, Salpea P, et al. Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas, 9th edition. *Diabetes Res Clin Pract*. 2019;157:107843. doi:10.1016/j.diabres.2019.107843.
2. Pop-Busui R, Boulton AJ, Feldman EL, et al. Diabetic Neuropathy: A Position Statement by the American Diabetes Association. *Diabetes Care*. 2017;40(1):136-154. doi:10.2337/dc16-2042.
3. Haque F, Reaz MBI, Ali SHM, Arsad N,

Chowdhury MEH. Performance analysis of noninvasive electrophysiological methods for the assessment of diabetic sensorimotor polyneuropathy in clinical research: a systematic review and meta-analysis with trial sequential analysis. *Sci Rep.* 2020;10(1):21770. Published 2020 Dec 10. doi:10.1038/s41598-020-78787-0

4. Zinman LH, Ngo M, Ng ET, Nwe KT, Gogov S, Bril V. Low-intensity laser therapy for painful symptoms of diabetic sensorimotor polyneuropathy: a controlled trial. *Diabetes Care*. 2004;27(4):921-924.
5. Haque F, Reaz MBI, Chowdhury MEH, Hashim FH, Arsad N, Ali SHM. Diabetic sensorimotor polyneuropathy severity classification using adaptive neuro fuzzy inference system. *IEEE Access*. 2021;9:7618-7631.
6. Haque F, Reaz MBI, Chowdhury MEH, et al. A Machine Learning-Based Severity Prediction Tool for the Michigan Neuropathy Screening Instrument. *Diagnostics (Basel)*. 2023;13(2):264. Published 2023 Jan 11. doi:10.3390/diagnostics13020264.
7. Sloan G, Selvarajah D, Tesfaye S. Pathogenesis, diagnosis and clinical management of diabetic sensorimotor peripheral neuropathy. *Nat Rev Endocrinol.* 2021;17(7):400-420. doi:10.1038/s41574-021-00496-z.
8. Mallik A, Weir AI. Nerve conduction studies: essentials and pitfalls in practice. *J Neurol Neurosurg Psychiatry.* 2005;76 Suppl 2(Suppl 2):ii23-ii31. doi:10.1136/jnnp.2005.069138.
9. Wiles PG, Pearce SM, Rice PJ, Mitchell JM. Vibration perception threshold: influence of age, height, sex, and smoking, and calculation of accurate centile values. *Diabet Med.* 1991;8(2):157-161. doi:10.1111/j.1464-5491.1991.tb01563.x.
10. Moghtaderi A, Bakhshipour A, Rashidi H. Validation of Michigan neuropathy screening instrument for diabetic peripheral neuropathy. *Clin Neurol Neurosurg.* 2006;108(5):477-481. doi:10.1016/j.clineuro.2005.08.003.
11. Weintrob N, Amitay I, Lilos P, Shalitin S, Lazar L, Josefsberg Z. Bedside neuropathy disability score compared to quantitative sensory testing for measurement of diabetic neuropathy in children, adolescents, and young adults with type 1 diabetes. *J Diabetes Complications.* 2007;21(1):13-19. doi:10.1016/j.jdiacomp.2005.11.002.
12. Rosenbaum D, Hennig EM. The influence of stretching and warm-up exercises on Achilles tendon reflex activity. *J Sports Sci.* 1995;13(6):481-490. doi:10.1080/02640419508732265.
13. Papachristou S, Pafili K, Trypsianis G, Papazoglou D, Vadikolias K, Papanas N. Skin Advanced Glycation End Products among Subjects with Type 2 Diabetes Mellitus with or without Distal Sensorimotor Polyneuropathy. *J Diabetes Res.* 2021;2021:6045677. Published 2021 Nov 28. doi:10.1155/2021/6045677.
14. Hayyolalam V, Aloqaily M, Özkasap Ö, Guizani M. Edge intelligence for empowering IoT-based healthcare systems. *IEEE Wirel Commun*. 2021;28(3):6-14.
15. Haque F, Bin Ibne Reaz M, Chowdhury MEH, et al. Performance Analysis of Conventional Machine Learning Algorithms for Diabetic Sensorimotor Polyneuropathy Severity Classification. *Diagnostics (Basel).* 2021;11(5):801. Published 2021 Apr 28. doi:10.3390/diagnostics11050801.
16. Haque F, Reaz MBI, Chowdhury MEH, et al. Performance Analysis of Conventional Machine Learning Algorithms for Diabetic Sensorimotor Polyneuropathy Severity Classification Using Nerve Conduction Studies. *Comput Intell Neurosci.* 2022;2022:9690940. Published 2022 Apr 25. doi:10.1155/2022/9690940.
17. Khandakar A, Chowdhury ME, Reaz MB, et al. DSPNet: A self-ONN model for robust DSPN diagnosis from temperature maps. *IEEE Sens J.* 2023;23(5):5370-5381.
18. Xie Y, Raj ANJ, Hu Z, Huang S, Fan Z, Joler M. A twofold lookup table architecture for efficient approximation of activation functions. *IEEE Trans Very Large Scale Integr Syst*. 2020;28(12):2540-2550.
19. Hajduk Z, Dec GR. Very high accuracy hyperbolic tangent function implementation in fpgas. *IEEE Access*. 2023;11:23701-23713.
20. Medus LD, Iakymchuk T, Frances-Villora JV, Bataller-Mompeán M, Rosado-Munoz A. A novel systolic parallel hardware architecture for the FPGA acceleration of feedforward neural networks. *IEEE Access*. 2019;7:76084-76103.
21. Pan Z, Gu Z, Jiang X, Zhu G, Ma D. A modular approximation methodology for efficient fixed-point hardware implementation of the sigmoid function. *IEEE Trans Ind Electron*. 2022;69(10):10694-10703.
22. Zhang S, Cao J, Zhang Q, Zhang Q, Zhang Y, Wang Y. An fpga-based reconfigurable cnn accelerator for yolo. In: *2020 IEEE 3rd International Conference on Electronics Technology (ICET)*. IEEE; 2020:74-78.
23. Singaravel S, Suykens J, Geyer P. Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction. *Adv Eng Informatics*. 2018;38:81-90.
24. Sharma S, Sharma S, Athaiya A. Activation functions in neural networks. *Towar Data Sci.*

2017;6(12):310-316.

25. Nwankpa C, Ijomah W, Gachagan A, Marshall S. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*. Published 2018 Nov 8.

26. Yu Y, Adu K, Tashi N, Anokye P, Wang X, Ayidzoe MA. Rmaf: Relu-memristor-like activation function for deep learning. *IEEE Access*. 2020;8:72727-72741.

27. Goodfellow I, Yoshua B, Aaron C. Deep learning. *MIT press*. Published 2016.

28. Pascanu R. On the difficulty of training recurrent neural networks. *arXiv Prepr arXiv12115063*. Published online 2013.

29. Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int J Uncertainty, Fuzziness Knowledge-Based Syst*. 1998;6(02):107-116.

30. Wuraola A, Patel N. Resource efficient activation functions for neural network accelerators. *Neurocomputing*. 2022;482:163-185.

31. Hahnloser RHR, Sarpeshkar R, Mahowald MA, Douglas RJ, Seung HS. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*. 2000;405(6789):947-951.

32. Movva P. Survey on activation functions: A comparative study between state-of-the-art activation functions and oscillatory activation functions. *https://engrxiv.org/preprint/view/2250* Published 2022 April 3. doi: https://doi.org/10.31224/2250.

33. Banerjee C, Mukherjee T, Pasiliao Jr E. An empirical study on generalizations of the ReLU activation function. In: *Proceedings of the 2019 ACM Southeast Conference*. ; 2019:164-167.

34. Dubey AK, Jain V. Comparative study of convolution neural network's relu and leaky-relu activation functions. In: *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018*. Springer; 2019:873-880.

35. Lau MM, Lim KH. Review of adaptive activation function in deep neural network. In: *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*. IEEE; 2018:686-690.

36. Pedamonti D. Comparison of non-linear activation functions for deep neural networks on MNIST classification task. *arXiv Prepr arXiv180402763*. Published online 2018.

37. Jinsakul N, Tsai C-F, Tsai C-E, Wu P. Enhancement of deep learning in image classification performance using xception with the swish activation function for colorectal polyp preliminary screening. *Mathematics*. 2019;7(12):1170.

38. Lee M. Gelu activation function in deep learning: a comprehensive mathematical analysis and performance. *arXiv Prepr arXiv230512073*. Published online 2023.

39. Kýlýçarslan S, Adem K, Çelik M. An overview of the activation functions used in deep learning algorithms. *J New Results Sci*. 2021;10(3):75-88.

40. Sawacha Z, Spolaor F, Guarneri G, et al. Abnormal muscle activation during gait in diabetes patients with and without neuropathy. *Gait Posture*. 2012;35(1):101-105. doi:10.1016/j.gaitpost.2011.08.016.

41. Blagus R, Lusa L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*. 2013;14:1-16.

42. Urbanowicz RJ, Meeker M, La Cava W, Olson RS, Moore JH. Relief-based feature selection: Introduction and review. *J Biomed Inform*. 2018;85:189-203.

43. Libano F, Rech P, Neuman B, Leavitt J, Wirthlin M, Brunhaver J. How reduced data precision and degree of parallelism impact the reliability of convolutional neural networks on FPGAs. *IEEE Trans Nucl Sci*. 2021;68(5):865-872.

44. Bikku T. Multi-layered deep learning perceptron approach for health risk prediction. *J Big Data*. 2020;7(1):50.