“NeuronGlow”; Perceptron Learning Tool

Course Title : CO542 Neural Networks and Fuzzy Systems

Group Name : FuzzBusters

Team Members :

- S.A.R.R. Suraweeraarachchi (E/19/393)
- M.H.P. Thiwanka (E/19/407)
- L.M.L.M. Senevirathne (E/19/365)
- R.A.R.S Jayasekara (E/19/159)
- J.K. Wanasinghe (E/20/420)

Project Domain and Research Proposal

1. Goal

The goal of this project is to develop a hardware-based learning tool that helps students, researchers, and learners of all ages understand how a perceptron works. This tool will use an Arduino/FPGA-based system with LEDs and potentiometers to provide a tangible, real-time visualization of the perceptron’s operations. By allowing users to interact with the system, they will be able to observe how the perceptron updates its weights, applies activation functions, and makes classification decisions based on the input data.

This hands-on approach aims to bridge the gap between theoretical learning and practical application. Traditional methods of learning neural networks often rely on abstract software simulations that can be difficult to understand, particularly for beginners and younger learners. With this project, learners will be able to see and manipulate the perceptron’s components in a physical form, reinforcing key neural network concepts like weighted summation, activation functions, and the training process.

Moreover, the simplicity of the perceptron, combined with the visual feedback provided by LEDs and resistors, makes this tool an accessible educational resource for a wide range of audiences. It will allow even children or those with limited prior knowledge of machine learning to engage with and learn about neural networks in an intuitive and interactive way, helping to foster interest in the field of artificial intelligence from an early age.

2. Problem Definition

Understanding neural networks can be challenging, especially for beginners, as the learning process often relies on mathematical formulas and abstract software simulations. Many existing educational resources focus on programming-based implementations, which may not provide a clear and intuitive way to grasp how a perceptron updates its weights during training and makes classification decisions. Without hands-on experience, learners may struggle to comprehend key concepts such as weighted summation, activation functions, and learning iterations.

This project addresses these challenges by <u>creating a tangible, hardware-based perceptron visualization tool that makes the learning process more interactive and accessible</u>. Instead of relying solely on software simulations, this system allows users to physically see and manipulate how a perceptron processes two inputs, adjusts weights, and classifies data in real-time. The use of LED indicators and adjustable resistors provides instant visual feedback, making abstract machine learning concepts more engaging and easier to understand.

Additionally, this tool is designed to be simple and user-friendly, making it suitable for a wide range of learners, including young students with no prior knowledge of neural networks. The visual nature of the system ensures that even small children can grasp the basic functioning of a perceptron by observing how input changes affect outputs. This hands-on approach encourages curiosity and fosters a deeper interest in artificial intelligence and machine learning, making neural networks more approachable for students of all ages.

By providing a physical, interactive experience, this project bridges the gap between theory and practical application, improving learning outcomes and making neural networks more accessible to a broader audience.

3. Scope

The NeuronGlow: Perceptron Learning Tool is designed as an interactive educational tool for students and researchers interested in neural networks. Unlike traditional software-based simulations, this project aims to provide a tangible and hands-on approach to understanding perceptron learning. By implementing a single-layer perceptron with two numerical inputs, the tool focuses on demonstrating how a perceptron updates its weights, processes input data, and makes classification decisions.

- Real-Time Representation of Learning: The project utilizes an Arduino/FPGA-based system to display weight adjustments in real-time. As the perceptron learns from its inputs, the hardware components will visually represent changes in the learning process, making it easier for users to follow the iterative training steps.
- Visualization Using LEDs and Variable Resistors: The perceptron's decision-making and learning process will be illustrated using LED indicators and variable resistors. LEDs will display activation outputs, while variable resistors will allow users to manually adjust input weights, providing a dynamic and interactive experience.
- Portable Demonstration Setup: The tool is designed to be compact and mobile, making it suitable for lecture halls, workshops, and hands-on training sessions. Unlike software simulations that require computers, this physical model can be easily transported and set up, allowing educators and students to engage with perceptron learning in various environments.
- Enhancing Conceptual Understanding: By allowing users to experiment with real-time learning scenarios, the tool reinforces theoretical concepts such as weighted summation, activation functions, and training iterations. The interactive nature of the model ensures a deeper understanding compared to static mathematical explanations.

4. Justification for Using Perceptrons

Perceptrons are the simplest type of artificial neural networks, making them an excellent starting point for learning machine learning concepts. Since perceptrons are designed for binary classification tasks, they provide a clear and straightforward method for understanding how neural networks make decisions.

- Fundamental to Neural Networks: The perceptron model is the building block of more complex artificial neural networks (ANNs). Understanding how a perceptron learns and adjusts its weights is essential for grasping deeper AI concepts, including multi-layer perceptrons and deep learning models.
- Structured Learning Process: The perceptron follows an iterative weight adjustment process, where it learns by minimizing errors. This structured approach makes it suitable for real-time visualization, as users can observe changes in weights as the perceptron improves its classification accuracy.
- Engaging Hands-On Experience: Unlike abstract mathematical descriptions, this hardware-based learning tool provides a tangible way to see how a perceptron learns. This approach bridges the gap between theoretical knowledge and practical understanding, making it especially useful for beginners.
- Real-Time Decision-Making Demonstration: The project enables users to manipulate inputs and observe immediate changes in outputs, which enhances engagement and retention of learning concepts. Instead of relying on graphs or software-based outputs, users can physically see how the perceptron reacts to new inputs and adjusts accordingly.

By leveraging interactive hardware elements, the NeuronGlow: Perceptron Learning Tool ensures that learners can actively engage with perceptron training and decision-making, making it an effective, intuitive, and innovative educational tool for neural network fundamentals.

5. Literature Review

The perceptron algorithm, introduced by Rosenblatt in 1958, is one of the earliest machine learning models used for classification tasks. Since its introduction, various implementations and advancements have been made in both software and hardware to facilitate its understanding and application. Traditional perceptron implementations are software-based, but hardware-based educational tools have been explored in recent years. Some key literature related to the perceptron algorithm, hardware-based implementations, and educational tools for neural networks are stated below.

1. Perceptron Algorithm and Its Evolution

Rosenblatt (1958) introduced the perceptron as a supervised learning algorithm for binary classifiers. The algorithm relies on an adjustable weight vector and a simple threshold function for decision-making. Subsequent research refined the perceptron model, highlighting its limitations in handling linearly inseparable data.[2] These limitations led to the development of multi-layer perceptrons (MLPs) and backpropagation algorithms, extending neural networks' capabilities beyond binary classification.

2. Traditional Software-Based Implementations

Software-based implementations of perceptrons are widely used in machine learning courses and applications. Popular frameworks such as TensorFlow and PyTorch provide built-in perceptron models for classification tasks.[3] These tools help users understand perceptron learning through code-based simulations but lack interactive, tangible representations.

3. Hardware-Based Perceptron Implementations

Recent studies have explored microcontroller-based perceptron models that provide real-time, physical visualization of neural computations. Arduino microcontrollers are commonly used due to their simplicity and affordability.[5]

4. Hardware-Based Neural Network Implementations

4.1 Arduino-Based Neural Networks

Several research studies and open-source projects have demonstrated the feasibility of implementing perceptrons using microcontrollers like Arduino. These projects often use LEDs, voltage-sensitive components, and interactive visualizations to illustrate neural network concepts.

- Arduino-Based Perceptron Simulations: Numerous projects have explored the implementation of perceptrons using Arduino microcontrollers. These systems process inputs and activate LEDs based on learned weights, helping students visualize weight adjustments.[6]

- NeuroArduino: This open-source project provides a framework for implementing various neural network architectures on Arduino boards. It offers libraries and examples for training and simulating neural networks, including perceptrons.

4.2 Analog Circuit Implementations

- Analog Circuit Perceptrons: Some studies explore resistor networks to model weight adjustments in learning.[7]

- Op-Amp-Based Perceptrons: Op-amps can be used to implement the weighted sum and activation functions of a perceptron. By adjusting the resistances in the circuit, the weights of the network can be modified.

4.3 FPGA-Based Implementations

Field-Programmable Gate Arrays (FPGAs) have been widely explored for implementing perceptrons due to their parallel processing capabilities, energy efficiency, and real-time adaptability. Unlike microcontroller-based implementations, FPGA-based perceptrons enable faster computations and support large-scale neural network applications. Several studies have demonstrated the use of FPGA architectures to accelerate perceptron learning and inference.

- FPGA-Accelerated Perceptrons: Researchers have developed FPGA-based perceptron models to achieve high-speed classification tasks while minimizing latency and power consumption. These implementations leverage hardware description languages (HDLs) such as VHDL and Verilog for efficient circuit design.[8]

- Low-Power FPGA Implementations: Power-efficient FPGA-based perceptrons are designed for embedded and IoT applications where energy constraints are critical. These implementations optimize resource utilization by reducing redundant computations and using approximate arithmetic techniques.

- Scalable Perceptron Architectures on FPGA: Some studies propose scalable FPGA architectures that support multi-layer perceptrons (MLPs) by integrating memory-efficient weight storage and dynamic reconfiguration. These architectures enhance adaptability for real-time AI applications.

By leveraging the parallel computing capabilities of FPGAs, perceptron models can achieve significant improvements in computational speed and scalability. These implementations provide a promising direction for real-time, hardware-accelerated neural networks.

4.4 Educational AI Tools

- Educational AI Kits: Several educational kits have been developed to teach AI concepts, including neural networks. These kits often include hardware components like microcontrollers, sensors, and actuators, along with software tools for programming and simulation.

- AI Education Platforms: Online platforms like TensorFlow Playground and Google Colaboratory provide interactive tools for learning about neural networks. While these platforms are software-based, they can be used in conjunction with hardware implementations to enhance the learning experience.

6. High-Level Design of the Proposed Model

The perceptron model for this project follows a single-layer architecture with two main inputs. The architecture consists of:

- Input Layer: Two Numerical inputs (e.g., temperature, humidity, pressure) | Potentiometer
- Label: Binary label (e.g., Good / Bad weather)
- Weights: Represented by intensities of the weight representing LEDs
- Bias: Represented by intensities of the bias representing LED
- Summation Function: Computes weighted sum of inputs
- Activation Function: Step function (ON/OFF) or sigmoid function (to be tested on both)
- Output Layer: LED or display indicating classification

6.1 Diagram of the High Level Design

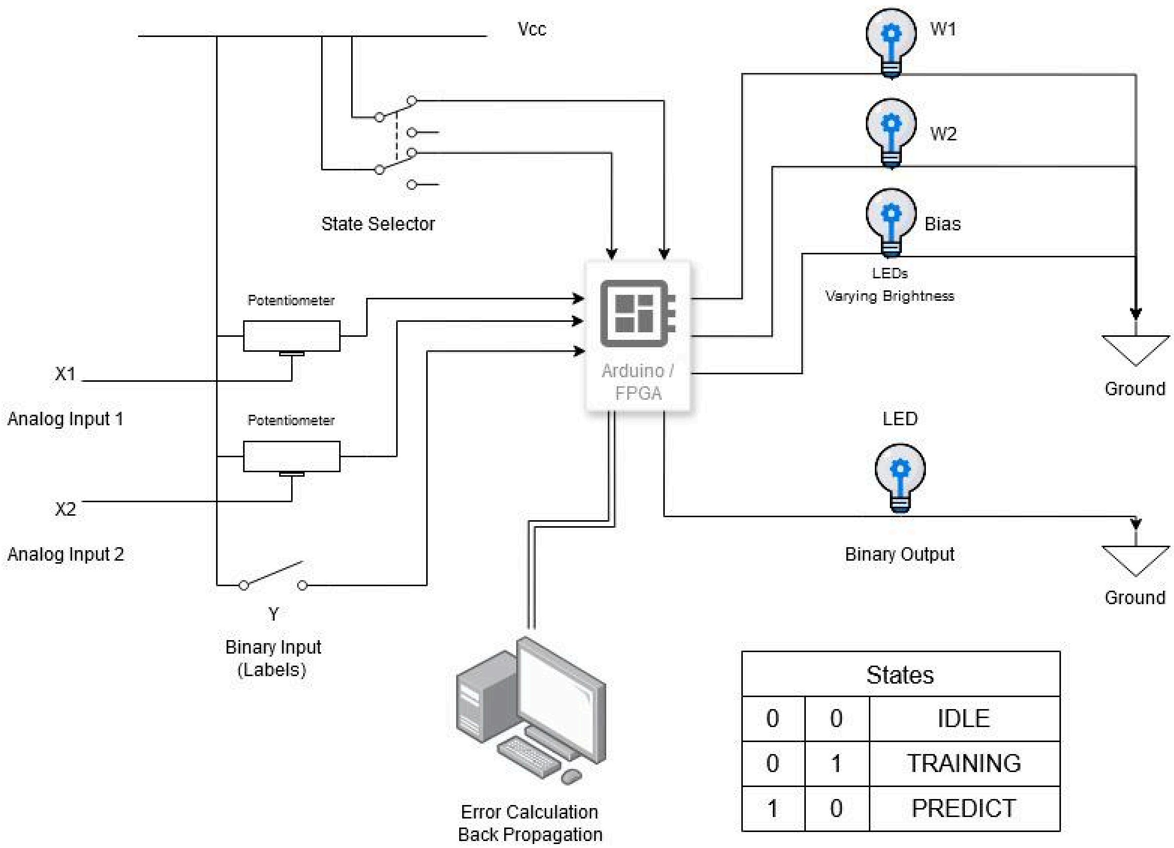

Figure 01: High Level Design Diagram

6.2 Implementation process

Overview :

- The setup has 3 states
  - Idle State
  - Training State
  - Prediction State
- The input values are scaled to the potentiometer range
- Binary input for the labels is set by a switch (Applicable for binary classification)
- Error calculation and Back propagation algorithms are performed by a connected computer
- Initial bias and weights will be set, and while being trained they would be adjusted and will be visible by the respective LED intensities
- Binary output LED will be used when a prediction is being performed

6.3 Explaining the process based on the states

1) Idle state
   Inputs and Labels will be adjusted at this stage. Neither the predictions nor training will occur during this state. This state is for feature and label adjustments

2) Training state
   After setting up the required features and labels, the system is transitioned to this state, where the voltage levels which are mapped according to their values at the features and label will be taken into the system.

   Each labeled data is set at idle state, and the training state is switched so that the system will take the voltages at that point to the network. After weights are adjusted (observed based on the intensity) the system is taken back to idle state, data is set and training state will be switched. New data set will be then taken to the system and weights will be adjusted. This can be performed until sufficient labeled data is given

3) Prediction state
   When the system is trained sufficiently, this state is switched, and based on the non labeled data given by the two feature inputs, the binary output will be predicted by the output LED.

7. Conclusion

This hardware-based perceptron visualization tool will serve as an educational aid for understanding fundamental neural network concepts. The implementation will include Arduino/FPGA integration, LED-based weight representation, and interactive real-time training feedback.

8. Reference

[1] F. Rosenblatt, "**The perceptron: A probabilistic model for information storage and organization in the brain,**" **Psychological Review**, vol. 65, no. 6, pp. 386–408, 1958.

[2] M. Minsky and S. Papert, **Perceptrons: An Introduction to Computational Geometry**. Cambridge, MA: MIT Press, 1969.

[3] M. Abadi et al., **"TensorFlow: A system for large-scale machine learning,"** in Proc. 12th USENIX Symp. Oper. Syst. Des. Implement.*, 2016, pp. 265–283.

[4] A. Paszke et al., **"PyTorch: An imperative style, high-performance deep learning library**," in **Adv. Neural Inf. Process. Syst.**, vol. 32, 2019.

[5] A. Suresh et al., **"Hardware-based neural network simulation using Arduino,"** **Int. J. Artif. Intell. Appl.**, vol. 11, no. 4, pp. 32–45, 2020.

[6] W. Jiang et al., "**An interactive perceptron learning tool with Arduino**," **Educ. Robot. J**., vol. 9, no. 3, pp. 18–30, 2021.

[7] A. Gupta and S. Verma, "**Analog implementations of perceptron-based classification**," J. Electron. Learn. Technol., vol. 6, no. 1, pp. 55–68, 2019.

[8] S. Rothhaupt, W. Meyerle, and B. Kormann, "**FPGA-based hardware acceleration of artificial neural network inference**," in **Proc. 6th AccML Workshop**, 2024, pp. 1–6.