VIII Simpósio de Iniciação Científica 2025

"Do conhecimento acadêmico à transformação sustentável: inovação com validação científica"

IMPLEMENTAÇÃO DE CIRCUITO NEUROMÓFICO BASEADO NA ARQUITETURA DE REDE ELM EM HARDWARE DIGITAL UTILIZANDO LINGUAGEM VERILOG

Felipe R. Malizia[1] (IC), Gabriel A. Fanelli de Souza (PQ)[1]
[1]*Universidade Federal de Itajubá.*

Palavras-chave: Neuromórfico. redes neurais. ELM. circuito digital. verilog.

Introdução

A Extreme Learning Machine, como descrito em [1,3], é uma arquitetura de rede neural derivada da Feedforward Neural Network (FNN) que tem se destacado no campo de redes neurais em hardware devido à sua simplicidade e eficiência. Nesta arquitetura, há apenas uma camada oculta de neurônios, os pesos de entrada são fixos e gerados de forma aleatória e os pesos de saída são obtidos através da realização do método dos mínimos quadrados.

Esta iniciação científica é uma continuação de um trabalho prévio realizado pelo grupo de microeletrônica da Universidade Federal de Itajubá, onde estava sendo pesquisada uma extreme learning machine (ELM) analógica para descrição de superfícies de controle não lineares [2]. Neste projeto de pesquisa, foi proposto o desenvolvimento de um modelo de neurônio digital como base para uma rede neural ELM para descrição de superfícies de controle não lineares, dessa forma, bastaria alterar os pesos e entradas deste único circuito digital para produzir o efeito de vários neurônios e formar uma rede, permitindo técnicas de multiplexação no tempo e versatilidade na variação de parâmetros da rede como número de neurônios e camadas ocultas, por exemplo. A mudança desta pesquisa para a área digital, ao invés da analógica, se baseia na flexibilidade obtida através do circuito digital, uma vez que só é preciso mudar os pesos na memória alocada para o circuito e rodar novamente para que se aprenda uma nova superfície, enquanto analogicamente, essa flexibilidade quanto a superfícies variadas não é tão simples.

Metodologia

O projeto foi dividido em cinco etapas, sendo elas o desenvolvimento teórico da rede, simulação da rede, descrição do circuito em hardware, simulação da rede em hardware e, por fim, modificação para ponto flutuante.

Em primeiro lugar, construiu-se o funcionamento do neurônio com base no funcionamento do neurônio da camada oculta da ELM, como o objetivo era a descrição de superfícies, decidiu-se por constituir um neurônio que recebe duas entradas, respectivamente as coordenadas do ponto da superfície a ser calculado no momento. Em seguida essas duas entradas seriam multiplicadas pelos pesos de entrada do neurônio, somadas e comparadas ao bias daquele neurônio, caso o valor da soma das entradas ultrapasse o valor do bias, a rede enviaria um sinal lógico alto para que o peso de saída fosse adicionado ao resultado daquele ponto. Este processo se repetiria pelo número de neurônios da rede, que seriam representados pelo tamanho do elemento de memória do circuito, uma vez que cada endereço da memória seria responsável por guardar os dois pesos de entrada, o bias e o peso de saída daquele neurônio em questão. Logo, ao final do processo, o ponto em questão da superfície seria formado pela soma de todos os pesos de saída dos neurônios que enviaram o sinal lógico alto.

Uma vez construída a base teórica para a rede, escolheu-se uma superfície teste que pudesse ser usada como objetivo durante todo o processo de desenvolvimento da rede. Esta superfície é formada pela função: $\sin(x/10) + \cos(y/10)$ e seu gráfico é demonstrado na figura (2). Após a escolha da superfície, realizou-se a primeira simulação da rede, sendo ela construída por um código em Octave. Neste código os pontos de (0,0) até (128,128) são transformados em uma matriz de pontos, os pesos de entrada são escolhidos de forma aleatória entre potências de 2 que variam de 2^0 a 2^7, essa decisão foi tomada visando a facilidade da implementação do circuito em hardware e a possibilidade de representar os pesos como números de 8 bits. Para uma primeira abordagem, os bias foram escolhidos como números que variam de 0 a 199 na rede multiplicados por 128. Desta forma, os bias não passariam de um número que pode ser representado por 16 bits. Por fim, os pesos de saída foram adquiridos através da operação dos mínimos quadrados

na matriz contendo os pontos da superfície alvo e o circuito pode ser calculado através da função de ativação Heaviside deslocada pelo bias, tendo assim o mesmo efeito da ideia original do circuito. Além disso, após algumas simulações alterando o número de neurônios, foi decidido que as simulações ocorreriam com 200 neurônios na rede.

Após as simulações realizadas em Octave provarem o funcionamento da rede, como pode ser visto na sessão de resultados, iniciou-se a descrição em hardware da rede através da linguagem de descrição de hardware Verilog no software Quartus. Por uma questão de simplicidade do circuito, decidiu-se fazer uma quantização nos pesos de saída, isto é, alteraram-se os valores que anteriormente eram descritos como ponto flutuante de dupla precisão para uma versão multiplicada por 128 e transformada em inteiro (houve o descarte das casas decimais restantes). Dessa forma, os pesos de saída puderam ser representados como valores de 16 bits, assim como os bias anteriormente. Além disso, durante a produção do circuito, percebeu-se que a operação de multiplicação em hardware era muito custosa e houve a possibilidade de se alterar a descrição da multiplicação dos pesos de entrada pelas entradas no circuito por uma operação de deslocamento. Isto decorre do fato de que os pesos de entrada são potências de 2 e podem ser representados como valores de 8 bits com apenas um deles em estado lógico alto. Logo, cada deslocamento tem o efeito de uma multiplicação por 2 no valor original da entrada. Após os pesos de entrada deslocarem as entradas originais, o resultado da soma e comparação habilita ou desabilita o acumulador, que tem a função de manter o seu valor, caso não habilitado, ou de somar o peso de saída ao seu valor atual, caso esteja habilitado.

A figura (4) demonstra o resultado final do neurônio através da função RTL Viewer do software Quartus, enquanto que, por motivos de melhor visualização e clareza do funcionamento da rede, a figura (3) demonstra o fluxo de dados na rede de maneira simplificada.

Uma vez construída a rede em hardware, simulou-se através do software ModelSim-Altera e comparou-se o resultado ao resultado obtido anteriormente pela simulação realizada no Octave (alterado para trabalhar com inteiros da mesma forma).
Como última etapa do projeto, decidiu-se realizar mudanças no circuito para que o mesmo fosse capaz de trabalhar com pesos de saída em ponto flutuante de precisão simples, buscando uma melhor precisão com a rede alvo. Dessa forma, foram realizadas as devidas alterações para que o acumulador fosse capaz de trabalhar com o padrão de ponto flutuante IEE754. Para validar as mudanças realizadas, 50 simulações foram feitas variando os pesos de entrada aleatoriamente, exibindo os gráficos e valores do erro médio quadrático (EMQ) no caso em que ambas as versões da rede se saíram melhor.

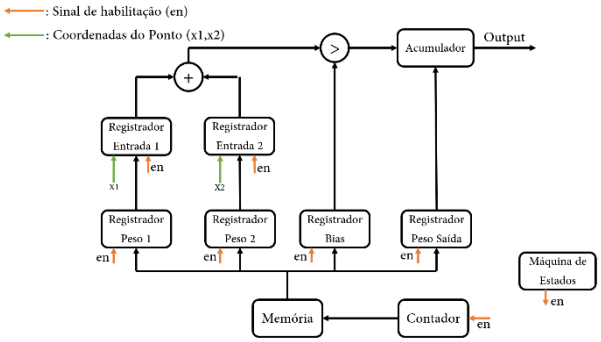

Figura 3 –Circuito do neurônio simplificado.

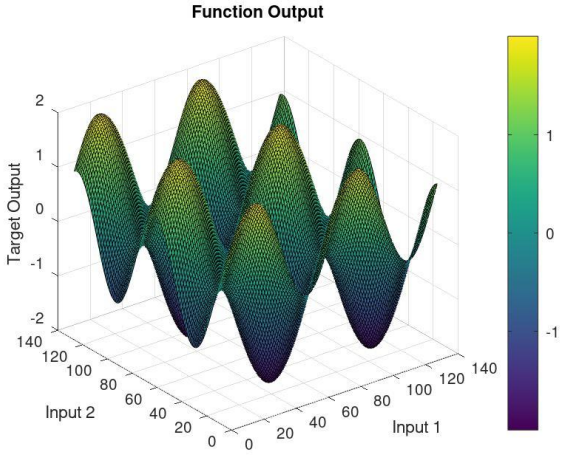

Figura 2 – sin(x/10) + cos(y/10), superfície alvo.

Resultados e discussão

Como indicado na sessão anterior, foram rodadas 50 simulações com os pesos de entrada e bias aleatórios, entre os limites estabelecidos, com o objetivo de encontrar o conjunto que mais se aproxima da superfície original. Para este conjunto, obteve-se a figura 5.

Além disso, para uma visualização gráfica, realizou-se a diferença entre a matriz da superfície original e da superfície construída com pesos de saída inteiros na figura 6.

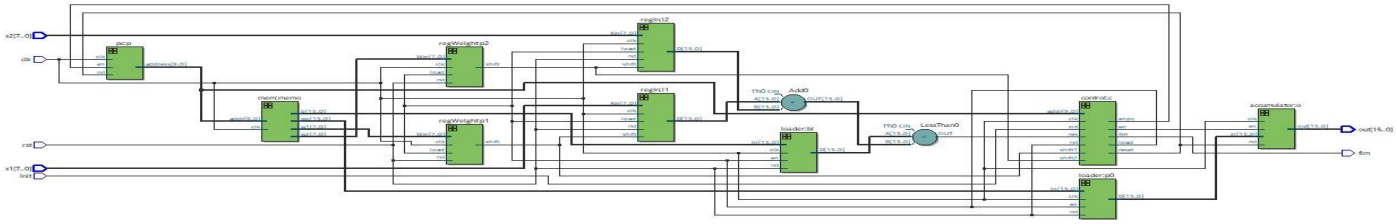

Figura 4 – RTL Viewer Quartus.

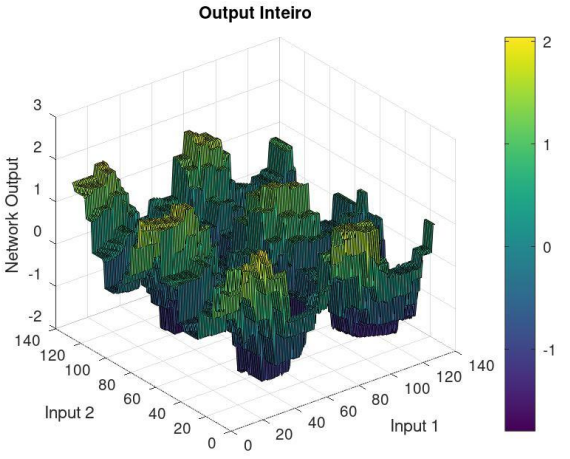

Figura 5 – Superfície construida utilizando pesos inteiros.

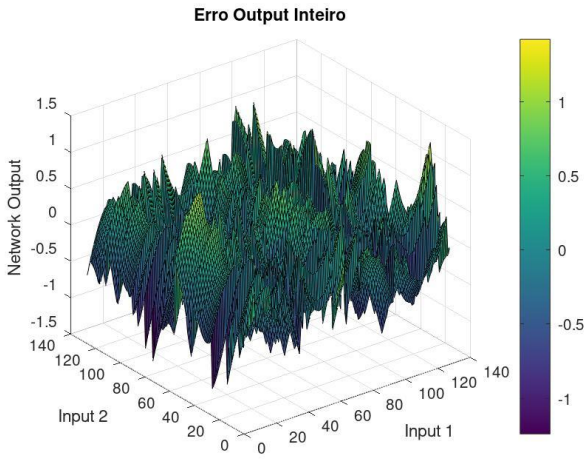

Figura 6 – Diferença entre superfície alvo e pesos inteiros.

Por fim, para se ter um parâmetro mais geral e concreto, mediu-se o erro médio quadrático (EMQ) entre a matriz alvo e a gerada pela rede, disponibilizado na tabela (1).

Ao se comparar graficamente as duas superfícies, percebe-se que a superfície projetada pela rede falha principalmente em momentos muito íngremes da função alvo, o que provavelmente se deve ao fato de se estar utilizando uma função degrau como função de ativação, o que forma um conjunto de plataformas ao longo do gráfico e que, com 200 neurônios, não consegue representar de maneira consistente a forma como a função alvo cresce e decresce. Entretanto, ao se analisar o EMQ, percebe-se que a superfície errou menos do que se aparenta pelo gráfico de erro, pois os picos e vales da superfície alvo estão corretamente representados na superfície construída.

Após uma análise do resultado do circuito trabalhando com pesos de saída inteiros, decidiu-se fazer as devidas atualizações nos módulos em Verilog e trabalhar com pesos de saída em ponto flutuante padrão IEEE 754. O resultado da superfície com os mesmos pesos de entrada e biases é demonstrado na figura 7.

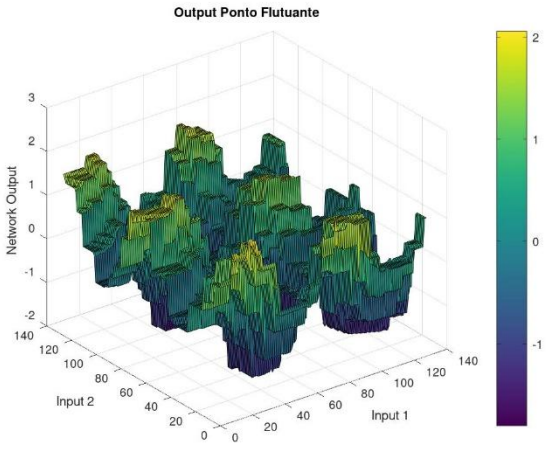

Figura 7 – Superfície construida utilizando pesos ponto flutuante.

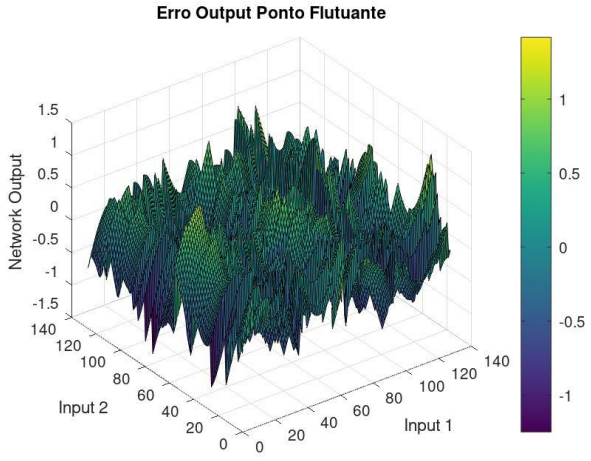

Figura 8 – Diferença entre superfície alvo e pesos ponto flutuante.

Assim como a superfície construída com pesos inteiros, analisou-se o erro em relação à superfície alvo, figura 8, e o EMQ, sendo que o resultado do EMQ está descrito na tabela (1). Além disso, uma comparação de elementos lógicos entre os circuitos descritos em Verilog no software Quartus foi feita com o objetivo de comparar quão válido seria a alteração do circuito digital inteiro para ponto flutuante.

Superfície	EMQ	Elementos Lógicos
Inteira	0.1130	121
Ponto Flutuante	0.1129	1284

Tabela 1 – Erro Médio Quadrático e Número de elementos lógicos

Como é possível observar, a rede teve um erro de aproximadamente 5,64% comparado à superfície alvo, o que pode ser ocasionado pelo número de neurônios ou pela forma como os bias estão dispostos pela rede. Além disso, pela diferença entre ambas as redes ser mínima no EMQ, mas multiplicada em 10 no número de elementos lógicos, não vale a pena realizar esta alteração.

Conclusões

O projeto de pesquisa comprovou que uma rede ELM digital é capaz de aprender uma superfície e pode ser utilizada para esta área do desenvolvimento. Além disso, o projeto foi capaz de comprovar a versatilidade do circuito digital neste meio, uma vez que há uma possibilidade de mudança no algoritmo de produção dos bias da rede, como também para alteração do número de neurônios, possibilitando uma melhor representação das curvas na rede. Após as simulações, também foi possível concluir, através do número de elementos lógicos e valor do erro quadrático médio, que não há uma melhora significativa na alteração da rede para se trabalhar com pontos flutuantes de precisão simples, sendo a melhor opção realizar a quantização dos valores calculados pelo método dos mínimos quadrados.

Agradecimentos

Agradeço a CNPq, a Universidade Federal de Itajubá, ao grupo de microeletrônica e seus colaboradores pelo financiamento. Sou grato ao Gabriel A. F. de Souza pela oportunidade, ajuda e pelo conhecimento adquirido. Por fim, agradeço aos meus amigos Felipe H. Puccinelli e Fábio P. S. Stolf por estarem ao meu lado durante essa jornada.

Referências

[1] DING, Shifei; XU, Xinzheng; NIE, Ru. Extreme learning machine and its applications. *Neural Computing and Applications*, London: Springer-Verlag, v. 25, p. 549–556, 2014.

[2] ALVES, Euller Lucas Mendes; SOUZA, Gabriel Antonio Fanelli de. Análise e implementação de circuito neuromórfico utilizando a arquitetura ELM (Extreme Learning Machine) em hardware. In: **SIMPÓSIO DE INICIAÇÃO CIENTÍFICA**, 7., 2024, Itajubá. *Anais...* Itajubá: Universidade Federal de Itajubá, 2024

[3] WANG, J.; LU, S.; WANG, S. H.; et al. A review on extreme learning machine. *Multimedia Tools and Applications*, v. 81, p. 41611–41660, Dec. 2022. Disponível em: https://doi.org/10.1007/s11042-021-11007-7. Acesso em: 21 ago. 2025.